

LAPORAN AKHIR

MAGANG & STUDI INDEPENDEN BERSERTIFIKAT

**PENGEMBANG MACHINE LEARNING DAN FRONT-
END WEB**

di PT Presentologics

Diajukan untuk memenuhi persyaratan kelulusan

Program MSIB MBKM

oleh :

Farel Arden/ 001201900068



Information Technology

President University

2021

Lembar Pengesahan
Information Technology
President University

STUDI INDEPENDEN PENGEMBANG MACHINE LEARNING DAN
FRONT-END WEB
di PT Presentologics

oleh :

Farel Arden/ 001201900068

disetujui dan disahkan sebagai

Laporan Studi Independen Bersertifikat Kampus Merdeka

Cikarang, 13/1/2022

Pembimbing Studi Independen

Information Technology

President University



Nur Hadisukmana, M.Sc

NIP: 0423076302

Lembar Pengesahan

**STUDI INDEPENDEN PENGEMBANG MACHINE LEARNING DAN
FRONT-END WEB**

di PT Presentologics

oleh :

Farel Arden/ 001201900068

disetujui dan disahkan sebagai

Laporan Studi Independen Bersertifikat Kampus Merdeka

Bandung, 13/1/2022

Senior Education Program Manager




Adrianus Yoza Aprilio

ID. 01032015004

Abstraksi

Studi Independen Bersertifikat Pengembang *Machine Learning* dan *Front-End Web* merupakan salah satu program yang ditawarkan oleh PT. Presentologics yang ditujukan untuk menghasilkan talenta berstandar tinggi yang sesuai dengan standar Industri di bidang pengembangan machine learning dan Front-End Web.

PT Presentologics merupakan perusahaan yang hadir di Dicoding Space Jl. Batik Kumeli no 50, Bandung, Indonesia. Dicoding menjadi platform edukasi teknologi terdepan yang mendorong akses literasi digital yang lebih luas untuk semua.

Pelaksanaan dari proyek ini dilaksanakan dalam dua sesi, yaitu sesi sinkronasi yang diselenggarakan oleh mentor secara langsung dalam sesi Bernama ILT (Instructor Led Training) dan sesi asinkronasi di mana pelajar diminta untuk mengerjakan modul – modul yang sudah disediakan oleh pihak Dicoding sebagai bahan pembelajaran dan penilaian selama empat bulan. Tidak hanya pengetahuan di bidang *Machine Learning* dan *Front-End*, tDicoding juga mempersiapkan kelas softskill yang bertujuan untuk memudahkan pelajar Studi Independen untuk masuk ke dunia kerja.

Terdapat 14 modul yang harus diselesaikan oleh peserta Studi Independen Bersertifikat Pengembang *Machine Learning* dan *Front-End Web* dalam proses pembelajaran yang bersifat asinkronasi. Dicoding pula mempersiapkan 8 ILT untuk *Front-End* dan *Machine Learning* dengan 2 sesi ILT di antaranya untuk konsultasi capstone project. Selain 8 sesi tersebut, dicoding pula mempersiapkan 4 sesi untuk persiapan soft skill peserta ditambah dengan sesi konsultasi setiap minggu nya yang umumnya berdurasi 1 jam.

Adapun Capstone yang saya buat adalah aplikasi untuk memprediksi harga dari suatu barang yang saya dan mitra saya, Rizal Sihombing namakan “Inflativa”.

Pada aplikasi ini, kami menggunakan model *Machine Learning* Facebook Prophet yang ditujukan untuk memprediksi suatu target berdasarkan waktu atau seringkali disebut Time Series.

Kesimpulan yang saya dapatkan dari mengikuti program Studi Independen Bersertifikat Pengembang *Machine Learning* dan *Front-End Web* adalah saya dapat menyelesaikan masalah – masalah yang seringkali dihadapi masyarakat Indonesia melalui machine learning. Lewat *Machine Learning* kita dapat mencari solusi bukan dengan jalur tradisional namun kita membiarkan mesin untuk memberikan solusi yang terbaik untuk kita.

Kata kunci: *Machine Learning, Data, Dicoding, Front-End*

Kata Pengantar

Saya ucapkan puji syukur kepada Tuhan Yang Maha Esa atas rahmat-Nya dan berkat-Nya yang melimpah, atas terselesaikannya laporan akhir Program MSIB (Magang dan Studi Independen Bersertifikat). Tujuan dibuatnya laporan ini yaitu untuk melaporkan segala sesuatu yang berkaitan dengan Studi Independen Bersertifikat Pengembang Machine Learning dan Front-End Web yang diselenggarakan oleh Dicoding Indonesia dan yang dilaksanakan dari Agustus 2021 hingga Januari 2022.

Dalam penyusunan laporan akhir MSIB ini tentunya tidak lepas dari bimbingan dan pengarahan dari beberapa pihak, baik dari pihak prodi Informasi Teknologi dari Universitas Presiden yang dikepalai oleh Nur Hadisukmana, M.Sc yang sudah memberikan kepada saya kesempatan untuk mengikuti program MSIB ini. Saya juga mengucapkan terimakasih kepada beberapa instructor yang telah membantu saya dalam menyelesaikan program MSIB ini, di antara nya ada kak Ziya El Arif sebagai mentor saya yang menemani saya dalam proses MSIB ini. Kemudian kak Tia Dwi S. dan kak Zanuar yang sudah membantu saya dalam materi Machine Learning. Terakhir kak Dimas Maulana dan Rully Indra L. yang sudah membantu saya dalam materi Front-End.

Penulis menyadari bahwa laporan ini jauh dari kata sempurna karena keterbatasan yang dimiliki oleh penulis, baik dari segi penyusunan, penulisan, dan bahasa. Namun, penulis sudah berusaha semaksimal mungkin untuk memberikan laporan akhir yang terbaik. Oleh karena itu, penulis sangat berharap adanya kritik dan saran yang dapat membantu penulis untuk mencapai kesempurnaan dalam laporan ini.

Pada akhirnya, penulis sekali lagi mengucapkan terimakasih dan penulis berharap dengan laporan ini dapat membantu dalam peningkatan ilmu pengetahuan.

Daftar Isi

Lembar Pengesahan Program Studi Teknik Informatika	i
Lembar Pengesahan	ii
Abstraksi	iii
Kata Pengantar	iv
Daftar Isi	v
<Daftar lain-lain>	vii
Bab I Pendahuluan	I-8
I.1 Latar belakang	I-8
I.2 Lingkup	I-9
I.3 Tujuan	I-9
Bab II <Organisasi atau Lingkungan MSIB>	II-1
II.1 Struktur Organisasi	II-1
II.2 Lingkup Project	II-3
II.3 Deskripsi Project	II-3
II.4 Jadwal MSIB (Project dan pembelajaran)	II-4
Bab III PENGEMBANG MACHINE LEARNING & FRONT-END WEB	III-1
III.1 Proyek Machine Learning & Front-End Web	III-1
III.2 Proses Pembuatan Modul dan Capstone Project	III-2
III.3 Detail Pengerjaan Modul dan Proyek Capstone	III-5
Bab IV Penutup	IV-1
IV.1 <Kesimpulan >	IV-1
IV.2 <Saran>	IV-1
Referensi	viii
Lampiran A. <TOR>	A-1
Lampiran B. Log Activity	B-1
Lampiran C. <Dokumen Teknik>	C-1

Bab I Pendahuluan

I.1 Latar belakang

Saat ini kita tengah menghadapi Revolusi Industri 4.0 yang serba digital. Seluruh aspek kehidupan manusia telah dibantu atau digantikan oleh hadirnya teknologi. Salah satu yang berperan penting dalam kehidupan yang serba digital ini adalah para pengembang aplikasi. Sumber daya manusia ini sangat dibutuhkan untuk mengakselerasi Indonesia menuju dunia digital.

Untuk mendukung transformasi digital dan pertumbuhan ekonomi di Indonesia. Salah satu tantangan terbesar bagi seluruh tlenka yang berkecimpung di dunia teknologi adalah materi pembelajaran berstandar tinggi yang sesuai dengan standar Industri. Untuk itu, Dicoding bersama perusahaan teknologi bekerja sama untuk menghadirkan materi pembelajaran berkualitas tinggi yang sesuai dengan standar Industri melalui platform Dicoding Academy.

Studi Independen Bersertifikat Pengembang Machine Learning dan Front-End Web ini diajukan untuk menghasilkan talenta berstandar tinggi yang sesuai dengan standar Industri di bidang pengembangan *Machine Learning* dan *Front-End Web*. Proses pembelajaran yang dilakukan adalah online learning, dimana peserta harus mengimplementasikan materi yang diperolehnya secara langsung melalui project dan tugas-tugas yang harus diselesaikan untuk menyelesaikan setiap materinya.

Materi diberikan secara asynchronous (online melalui modul belajar di Dicoding Academy) dan akan di-review setiap interval waktu tertentu oleh pembimbing non-akademik dan expert. Selain project dan tugas, pemberian materi juga akan dilengkapi dengan kuis dan atau ujian pilihan ganda untuk memastikan pemahaman peserta.

Selain hard skill di bidang pengembangan solusi Machine Learning dan Front-End Web, soft skill juga menjadi target kompetensi peserta studi independen yaitu untuk penyiapan karir sebagai developer, termasuk namun tidak terbatas pada *self-branding, problem solving, design thinking*, serta kolaborasi.

Studi independen akan ditutup dengan project akhir, dimana peserta akan bekerja dalam kelompok dan mengembangkan solusi berbasis Machine Learning dan Front-End Web.

I.2 Lingkup

Aktivitas Studi Independen Pengembang Machine Learning dan Front-End Web meliputi pembelajaran individu dan project akhir dalam bentuk tim. Pada pembelajaran individu, setiap peserta akan mengikuti kelas dalam bentuk asynchronous (online melalui modul belajar di Dicoding Academy) dimana peserta dapat berkonsultasi dengan expert terkait materi yang dipelajarinya melalui forum diskusi.

I.3 Tujuan

Studi Independen Bersertifikat ini bertujuan untuk menghasilkan talenta berstandar tinggi yang sesuai dengan standar Industri. Proses pembelajaran yang dilakukan adalah online learning, dimana peserta harus mengimplementasikan materi yang diperolehnya secara langsung melalui project dan tugas-tugas yang harus diselesaikan untuk menyelesaikan setiap materinya.

Bab II Lingkungan Organisasi PT Presentologics

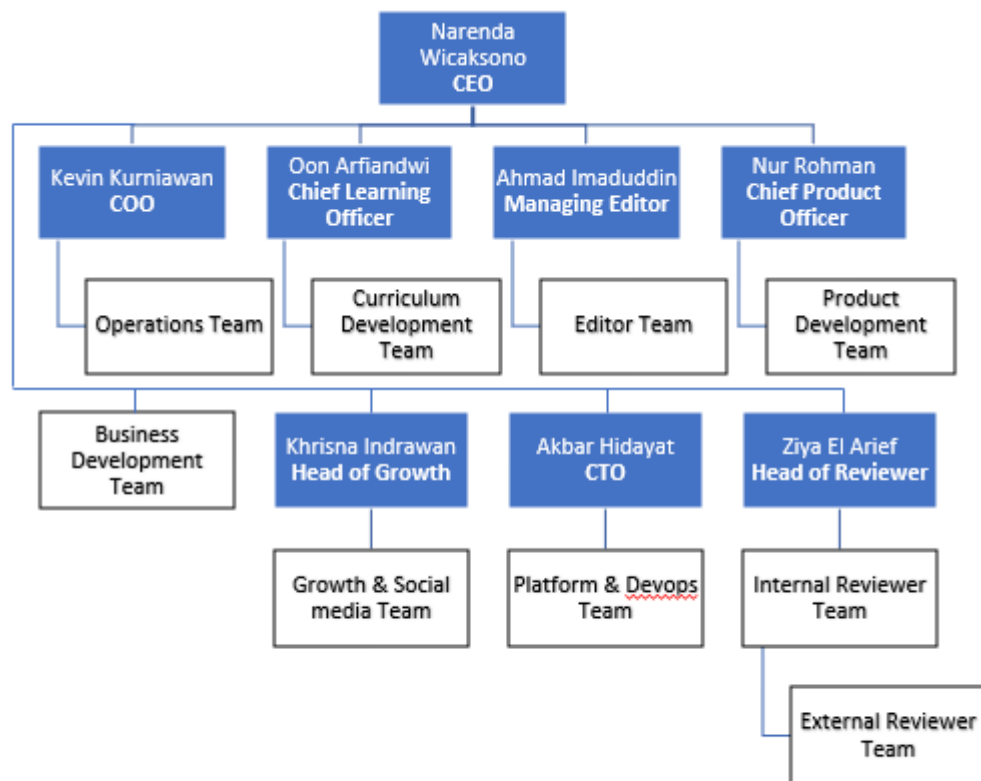
II.1 Struktur Organisasi

Dicoding secara resmi diluncurkan tanggal 5 Januari 2015 untuk menjembatani developer Indonesia dengan kebutuhan dan permintaan pasar yang semakin kompetitif. Dicoding hadir sebagai platform pendidikan teknologi yang membantu menghasilkan talenta digital berstandar global. Semua demi akselerasi Indonesia agar menjadi yang terdepan.

Saat ini, lebih dari 470 ribu developer dan calon developer telah tergabung di Dicoding. 290 ribu individu pembelajar telah dan sedang terdaftar dalam lebih dari 80 kelas yang disediakan oleh Dicoding.

Saat ini, Dicoding bermitra dengan perusahaan teknologi kelas dunia. Dicoding juga merupakan Google Authorized Training Partner dan memiliki komitmen kemitraan dengan pemilik teknologi, perusahaan multinasional, Kementerian/Lembaga Pemerintahan, serta perusahaan dengan skala nasional. Dicoding juga adalah mitra penyelenggara Bangkit, Indosat Ooredoo Digital Camp, Lintasarta Digischool, Baparekraf Digital Talent, dan Cloud and Back-End Developer Scholarship Program with content from AWS pada tahun 2021.

Adapun struktur organisasi merupakan sebuah garis penugasan formal yang menunjukkan alur tugas dan tanggung jawab setiap anggota perusahaan, perusahaan serta hubungan antar pihak dalam organisasi yang bekerja sama untuk mencapai suatu tujuan organisasi. Struktur organisasi dari PT Presentologics.



II.2 Lingkup Pekerjaan

Lingkup pekerjaan atau lingkup belajar yang kami lakukan pada Aktivitas Studi Independen Pengembang Machine Learning dan Front-End Web meliputi teori dan praktek mengenai Machine Learning dan Front-End Web. PT Presentologics, yaitu Dicoding juga telah mempersiapkan 14 modul untuk dapat dipelajari oleh peserta Aktivitas Studi Independen Pengembang Machine Learning dan Front-End Web. Pada Capstone Project, pihak PT Presentologics memberikan beberapa sugesti untuk dijadikan tema pengerjaan Capstone. Capstone project ini merupakan suatu aplikasi yang dapat membantu masyarakat Indonesia dari data – data yang sudah ada.

II.3 Deskripsi Pekerjaan

Aktivitas Studi Independen Pengembang Machine Learning dan Front-End Web meliputi pembelajaran individu dan project akhir dalam bentuk tim. Pada pembelajaran individu, setiap peserta akan mengikuti kelas dalam bentuk asynchronous (online melalui modul belajar di Dicoding Academy) dimana peserta dapat berkonsultasi dengan expert terkait materi yang dipelajarinya melalui forum diskusi.

Selain itu, setiap peserta akan memiliki pembimbing sebagai tempat konsultasi jika ditemui

kesulitan non-akademik dalam mengikuti pembelajaran. Pada program studi independen ini, terdapat dua buah learning path yang disediakan yaitu Machine Learning dan Front-End Web. Peserta akan memperoleh sertifikat kompetensi di setiap kelas di dalam Learning Path Machine Learning dan Front-End Web jika peserta berhasil lulus dari setiap ujian/penilaian yang diadakan untuk setiap kompetensi. Setelah mengikuti program ini, peserta juga dipersiapkan untuk mengikuti ujian sertifikasi global TensorFlow Developer Certification dari Google yang dapat diambil secara mandiri setelah mengikuti kegiatan Studi Independen ini.

Pada project akhir, peserta akan dibagi menjadi kelompok, dimana satu kelompok terdiri atas 2 orang dengan tema yang ditentukan oleh masing-masing kelompok dan harus mendapatkan persetujuan dari pembimbing atau expert.

II.4 Jadwal Kerja

Pelaksanaan *Studi Independen* Bersertifikat bersama Dicoding ini berjalan kurang lebih selama 22 minggu, dimulai dari tanggal 23 Agustus 2021 sampai tanggal 19 Januari 2022. Dengan rincian proses pembelajaran sebagai berikut:

- **Minggu ke-1** : Memulai Dasar Pemrograman untuk Menjadi Pengembang Software, Pengenalan ke Logika Pemrograman, dan Belajar Dasar Git dengan Github.
- **Minggu ke-2** : Pengenalan Data pada Pemrograman, Belajar Dasar Visualisasi Data, dan Memulai Pemrograman dengan Python.
- **Minggu ke-3** : Memulai Pemrograman dengan Python, Belajar Machine Learning untuk Pemula, dan sesi ILT *machine learning* bersama expert ke-1.
- **Minggu ke-4** : Belajar Pengembangan Machine Learning, dan Belajar Dasar Pemrograman Web.
- **Minggu ke-5** : Belajar Pengembangan Machine Learning, Belajar Dasar Pemrograman Web, ILT *frontend web* bersama expert ke-2, dan ILT *soft skill* ke-1.
- **Minggu ke-6** : Machine Learning Terapan, dan Belajar Dasar Pemrograman Web.
- **Minggu ke-7** : Machine Learning Terapan, dan sesi ILT *machine learning* bersama expert ke-3.
- **Minggu ke-8** : Evaluasi Penguasaan Machine Learning, dan Belajar Membuat Front-End Web untuk Pemula
- **Minggu ke-9** : Belajar Membuat Front-End Web untuk Pemula dan ILT *frontend web* bersama expert ke-4.
- **Minggu ke-10** : Tensorflow Developer Certification Prep dan ILT *soft skill* ke-2.

- **Minggu ke-11** : Belajar Fundamental Front-End Web Development dan *ILT machine learning* bersama expert ke-5.
- **Minggu ke-12** : Belajar Fundamental Front-End Web Development dan Karir: Meniti Karier sebagai Software Developer.
- **Minggu ke-13** : Belajar Fundamental Front-End Web Development, dan *ILT frontend web* bersama expert ke-6.
- **Minggu ke-14** : Pengerjaan *capstone project* dan sesi *ILT soft skill* ke-3.
- **Minggu ke-15** : Pengerjaan *capstone project* dan mentoring *capstone project* ke-1.
- **Minggu ke-16** : Pengerjaan *capstone project*.
- **Minggu ke-17** : Pengerjaan *capstone project* dan mentoring *capstone project* ke-2.
- **Minggu ke-18** : Pengerjaan *capstone project* dan *ILT soft skill* ke-4.
- **Minggu ke-19** : Penilaian *capstone project*.
- **Minggu ke-20** : Presentasi *capstone project*.

Minggu ke-21: Akhir program *Studi Independent Bersertifikat* bersama Dicoding.

Bab III PENGEMBANG MACHINE LEARNING & FRONT-END WEB

III.1 Proyek Machine Learning & Front-End Web

- Machine Learning

Machine learning adalah subbidang AI yang berfokus pada analisis data untuk menemukan hubungan antara input dan output yang diinginkan. Proses Machine Learning termasuk mengajar mesin untuk memprediksi dari data yang kita berikan. Sebagai syarat agar lulus dari semua kelas yang telah diberikan saya membuat beberapa model machine learning, diantaranya adalah :

1. Klasifikasi Gambar. Saya membuat sebuah model program yang bisa mendeteksi suatu gambar apakah suatu gambar itu gunting, atau kertas atau batu.
2. Natural Language Processing. Saya membuat sebuah model program untuk mendeteksi sebuah apakah sebuah *tweet* dinyatakan *tweet* yang serius atau tidak berdasarkan dari kata – kata yang digunakan.
3. Data Time Series. Saya membuat sebuah model machine learning perhitungan *sunspot* atau bintik matahari pada tiap harinya.
4. Klasifikasi Gambar dan Model Deployment. Saya membuat sebuah model machine learning untuk mendeteksi dan pengelompokan gambar berdasarkan angka yang tertera melalui *pixel* yang ada pada tiap – tiap gambar.
5. Regression dan Klasifikasi. Saya membuat sebuah model machine learning untuk mengetahui penumpang kapal titanic yang selamat dan tidak.
6. Sistem Rekomendasi. Saya membuat sebuah model machine learning untuk rekomendasi sebuah buku kepada pembaca melalui metode *collaborative* dan *content-based*.

- Front-End

Front-end web developer adalah developer yang bertanggung jawab untuk merancang dan membangun berbagai aplikasi web secara responsif, interaktif, dan juga user friendly. Seorang Front-End Web Developer memiliki peranan besar dalam pengembangan aplikasi web karena bertanggung jawab langsung kepada pengalaman pengguna. Di sini siswa diajarkan mulai dari dasar seperti pengenalan HTML, CSS kemudian menyusun layout yang responsif dengan Flexbox hingga Javascript. Sebagai syarat agar lulus dari semua kelas yang telah diberikan, saya telah menyelesaikan beberapa proyek, diantaranya adalah:

1. Membuat website statis yang bertujuan untuk memperlihatkan list café – café terbaik di Jakarta.
2. Membuat website pengelolaan data menggunakan DOM dan web storage, yaitu website daftar buku sederhana.
3. Membuat aplikasi web menggunakan ES6, Custom Element, NPM, Webpack, dan AJAX

- Capstone

Proyek akhir yang bertujuan untuk syarat lulus dari program ini dan merupakan kombinasi dari kedua path yaitu machine learning dan front-end, yang perlu untuk dikerjakan dengan durasi 100 jam. Saya dan teman kelompok saya membuat sebuah aplikasi machine learning berbasis website, yaitu inflativa. Sebuah aplikasi yang mendeteksi tingkat inflasi suatu sektor berdasarkan kota dan sektor barang yang ada di Indonesia.

III.2 Proses Pembuatan Modul dan Capstone Project

- Machine Learning

1. Klasifikasi gambar kertas gunting batu. Pada pembuatan model machine learning ini saya menggunakan dataset yang telah disediakan oleh pihak Dicoding, yang terdapat 3 folder yaitu rock, paper, scissors yang berisi gambar-gambar. Data dibagi menjadi 2 bagian yaitu 0,6 sebagai data latih dan

0,4 sebagai data validasi, data gambar dilatih menggunakan model keras sequential.

2. Natural Language Processing. Pada pembuatan model machine learning ini saya menggunakan dataset yang diperoleh dari platform Kaggle, yang didalamnya terdapat 1 file CSV dengan banyak 7613 baris dan 5 buah kolom yaitu pesan, spam, dan bukan spam. Dataset ini berbicara tentang *tweet – tweet* yang diberikan oleh sejumlah pengguna twitter yang sudah diberikan label untuk menentukan apakah tweet – tweet yang ada bersifat serius atau tidak. Data dibagi menjadi 2 bagian yaitu 0,8 sebagai data latih dan 0,2 sebagai data test, data text diencode menjadi tokenizer diubah menjadi kumpulan sebuah angka lalu menggunakan model keras sequential data dicompile.
3. Data Time Series. Pada pembuatan model machine learning ini saya menggunakan dataset yang diperoleh dari platform Kaggle, yang didalamnya terdapat 1 file CSV dengan banyak 3265 baris dan 3 buah kolom. Dataset ini berbicara tentang jumlah bintik matahari semenjak tahun 1791. Data dibagi menjadi 2 bagian yaitu 0,8 sebagai data latih dan 0,2 sebagai data test dan kemudian model machine learning dilatih berdasarkan dataset yang diberikan.
4. Klasifikasi Gambar dan Model Deployment. Pada pembuatan model machine learning ini saya menggunakan dataset yang diperoleh dari platform Kaggle, yang didalamnya terdapat 1 file CSV dengan banyak 42000 baris dan 785 buah kolom. Data dibagi menjadi 2 bagian yaitu 0,8 sebagai data latih dan 0,2 sebagai data validasi, data gambar dilatih menggunakan model keras sequential. Pembuatan model memberikan hasil akurasi sebesar 90%, dan dideploy kedalam sebuah file dengan extension h5.
5. Regression dan Klasifikasi penumpang kapal Titanic. Pada pembuatan model machine learning ini saya menggunakan dataset yang didapat pada platform Kaggle, yang terdapat 2 file CSV yaitu data test, dan training yang berisi data penumpang kapal titanic. Dari 14 kolom saya hanya memakai 8 kolom yang menjadi fitur utama, dikarenakan pembuatan model yang sangat kompleks. Saya menggunakan 4 algoritma dan salah satu dari algoritma tersebut menghasilkan 70% akurasi dengan nilai akurasi tertinggi.

6. Sistem Rekomendasi. Pada pembuatan model machine learning ini saya menggunakan dataset yang didapat pada platform Kaggle, yang terdapat 1 file CSV yaitu data buku yang berisi data – data seperti nama buku dan penulis buku. Dari saya memakai 8 kolom yang menjadi fitur utama, dikarenakan pembuatan model yang sangat kompleks. Saya menggunakan 2 algoritma diantaranya adalah Content Based Filtering, dan Collaborative Filtering.

- Front-End

1. Membuat website statis yang berisi daftar nama – nama café. Dengan menggunakan HTML5, CSS3. Proses pembuatan relatif cepat dikarenakan webnya sederhana dan juga responsif terhadap ukuran handphone.
2. Membuat website pengelolaan data menggunakan DOM dan web storage, yaitu website buku. Dengan menggunakan HTML5, CSS3, dan javascript DOM. Proses pembuatan dimulai dari merancang sebuah tampilan pada website. Proses pengerjaan website ini relatif cepat dikarenakan mengikuti pengerjaan website yang sudah ada sebelumnya.
3. Membuat Aplikasi Web dengan ES6, Custom Element, NPM, Webpack, dan AJAX, saya membuat sebuah website yang berisi daftar nama – nama cafe. Proses pembuatan dimulai dari merancang sebuah tampilannya dengan menggunakan warna biru muda, lalu dilanjutkan dengan pemilihan fonts. Lalu saya mulai menggunakan dalam ES6, Custom Element, NPM, Webpack, dan AJAX website saya.

- Capstone

Proses pembuatan capstone ini dibagi menjadi 2 bagian yaitu machine learning dan front-end. Di hari pertama saya dan rekan saya mengumpulkan data pada halaman website Badan Pusat Statistika (BPS) tentang tingkat inflasi berdasarkan kota dan sektor barang. Kami mengambil 90 kota yang ada di Indonesia yang berasal dalam periode Januari 2014 hingga Oktober 2021. Lalu dilanjutkan dengan merancang aplikasi web dan model machine learning dibentuk dengan model Facebook Prophet. Lalu model di-*deploy* ke dalam

web menggunakan framework flask, dan dihosting kedalam platform python anywhere.

III.3 Detail Pengerjaan Modul dan Proyek Capstone

MODUL

1. Memulai Dasar Pemrograman untuk Menjadi Pengembang Software

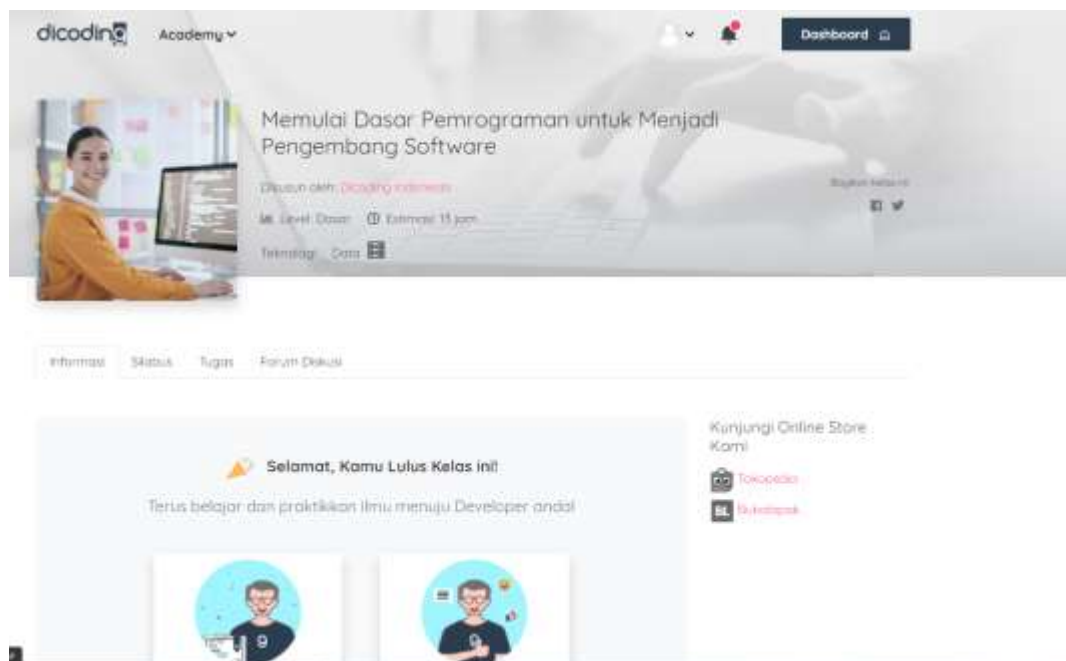


Figure 1: Modul “Memulai Dasar Pemrograman untuk Menjadi Pengembang Software” Selesai

2. Pengenalan ke Logika Pemrograman

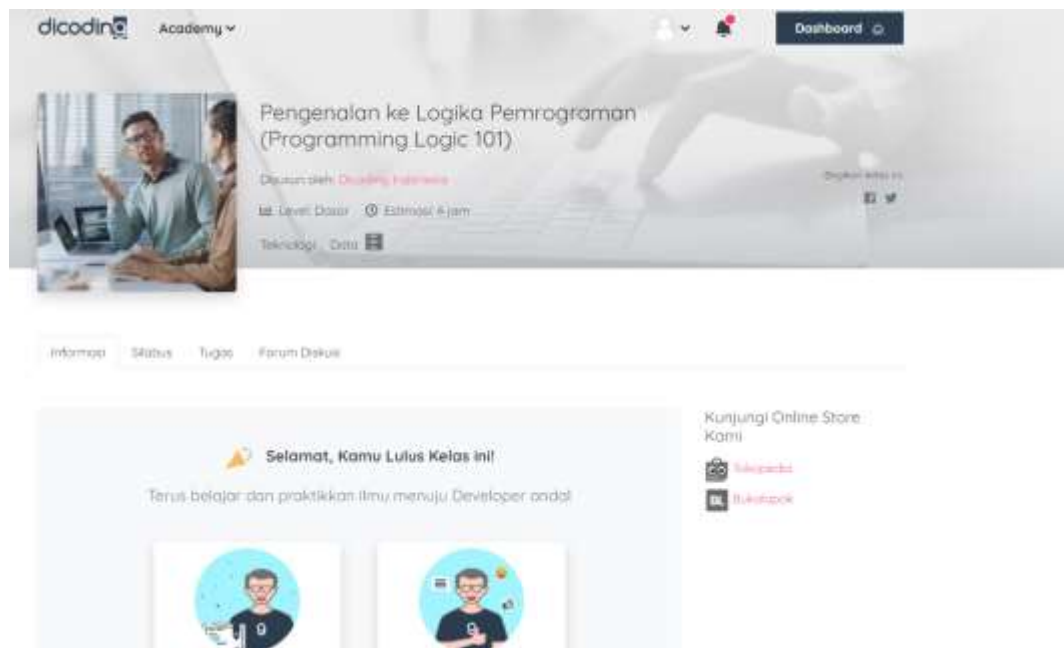


Figure 2: Modul “Pengenalan ke Logika Pemrograman” Selesai

3. Belajar Dasar Git dengan Github

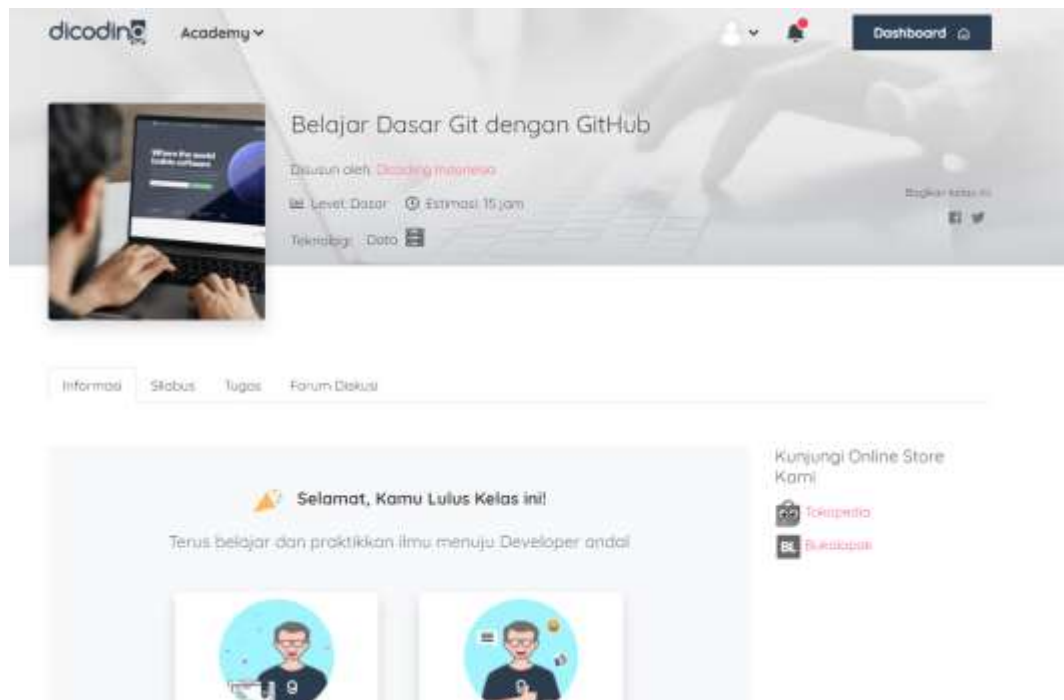


Figure 3: Modul “Belajar Dasar Git dengan Github” Selesai

4. Pengenalan Data pada Pemrograman

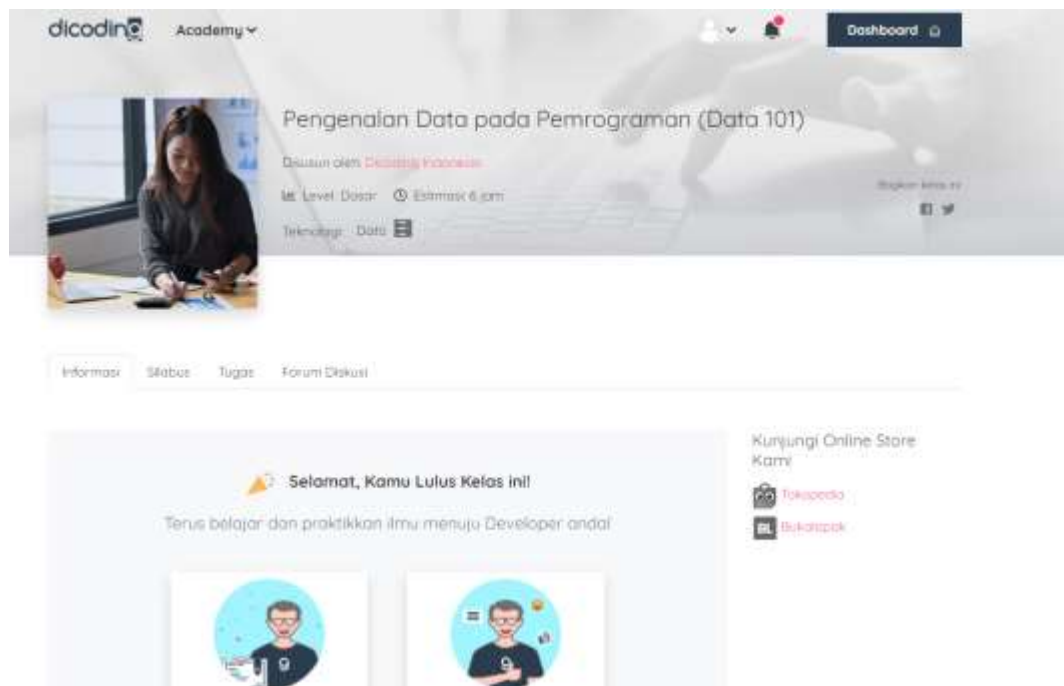


Figure 4: Modul “Pengenalan Data pada Pemrograman” Selesai

5. Belajar Dasar Visualisasi Data

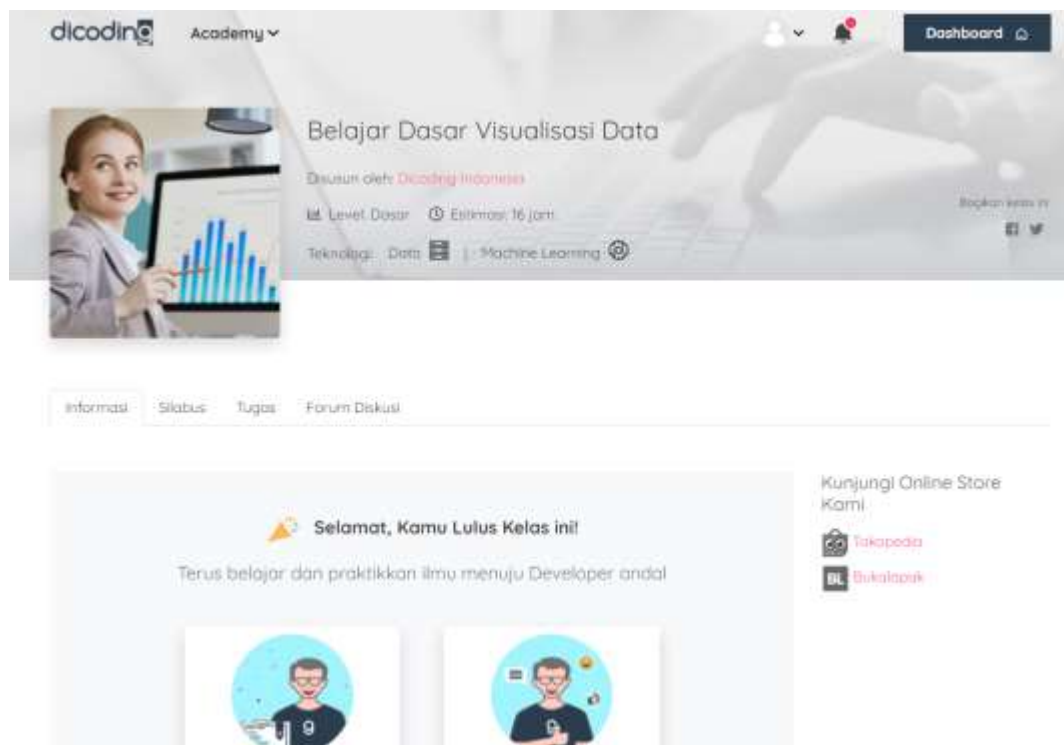


Figure 5: Modul “Belajar Dasar Visualisasi Data” Selesai

6. Memulai Pemrograman dengan Python

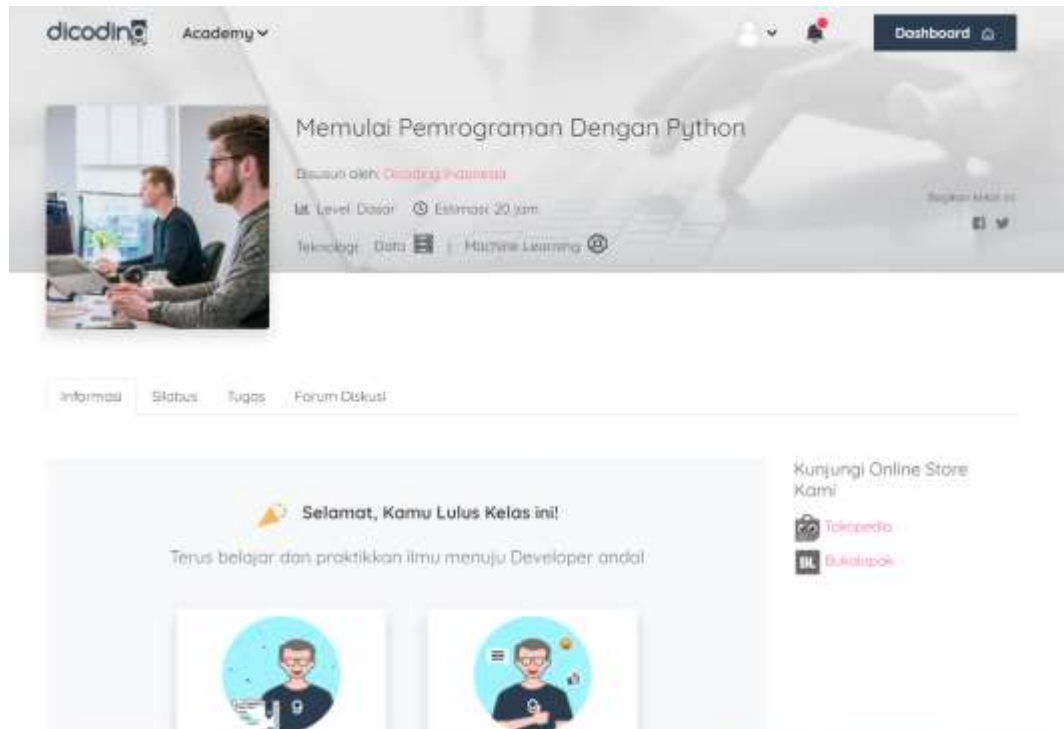


Figure 6: Modul “Memulai Pemrograman dengan Python” Selesai

7. Belajar Machine Learning untuk Pemula

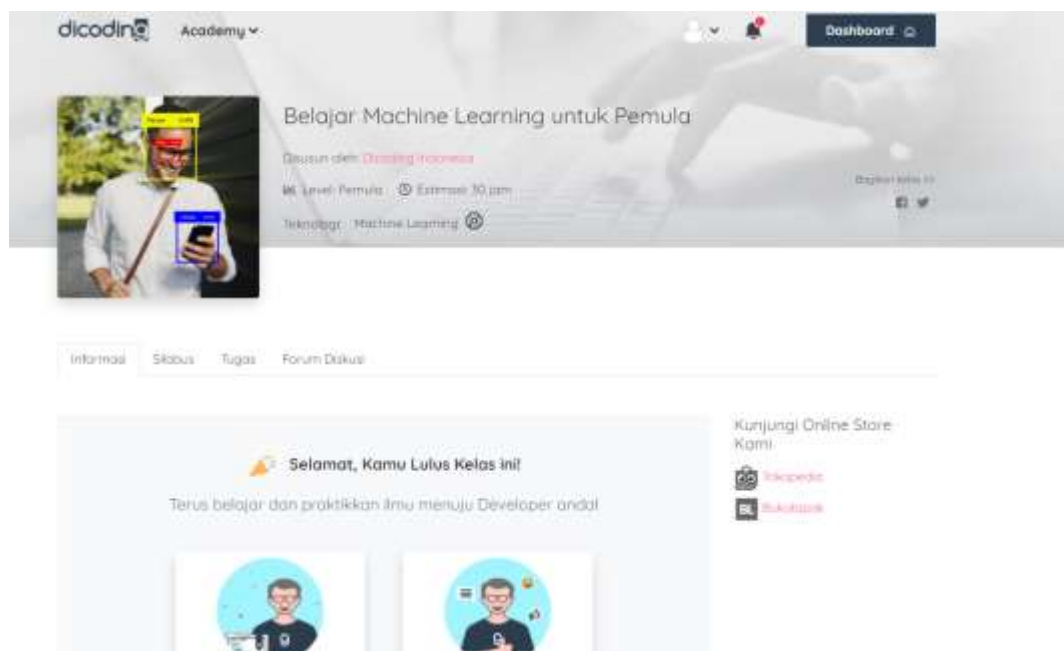
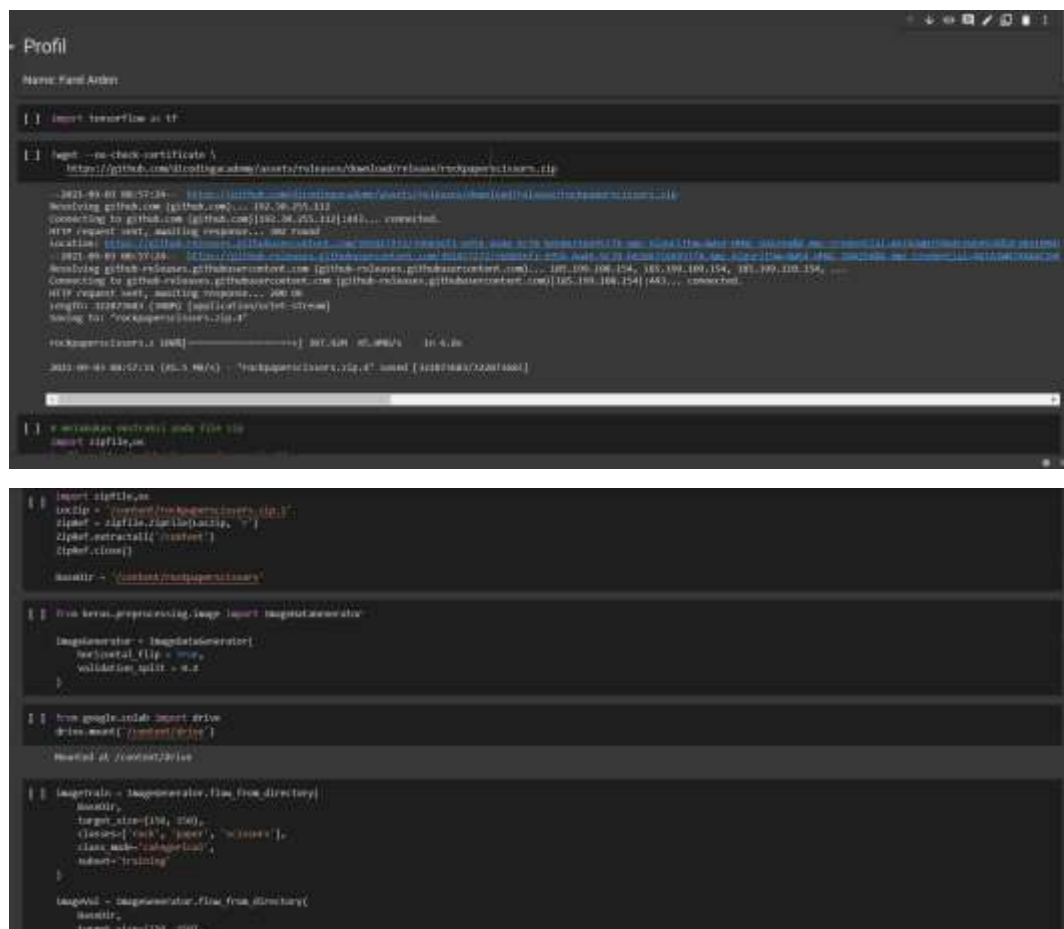


Figure 7: Modul “Belajar Machine Learning untuk Pemula” Selesai

Proyek Akhir:

Pada proyek akhir ini saya diminta untuk membuat mesin yang mengenal gambar yang diberikan apakah tangan itu berbentuk batu, kertas, atau gunting dengan menggunakan Deep Learning.



```

In [ ]: import tensorflow as tf

In [ ]: !git --no-check-certificate \
https://github.com/donlogica/adam/assets/releases/download/v1.0.0/tfrockpaperscissors.zip

--2021-09-09 08:57:38-- https://github.com/donlogica/adam/assets/releases/download/v1.0.0/tfrockpaperscissors.zip
Resolving github.com [github.com]... 192.30.255.112
Connecting to github.com [github.com]:443... connected.
HTTP request sent, awaiting response... 200 OK
location: https://github.com/donlogica/adam/assets/releases/download/v1.0.0/tfrockpaperscissors.zip
--2021-09-09 08:57:38-- https://github.com/donlogica/adam/assets/releases/download/v1.0.0/tfrockpaperscissors.zip
Resolving github.com [github.com]... 192.30.255.112
Connecting to github.com [github.com]:443... connected.
HTTP request sent, awaiting response... 200 OK
length: 11287000 (10.7M) [application/octet-stream]
Saving to: 'tfrockpaperscissors.zip.d'

tfrockpaperscissors.zip.d 100% |#####| 107.40M  85.0MB/s  in 4.1s

2021-09-09 08:57:41 (25.5 MB/s) - 'tfrockpaperscissors.zip.d' saved [11287000/11287000]

In [ ]: # extract the zip file to
import zipfile, os

In [ ]: ! unzip tfrockpaperscissors.zip.d

In [ ]: ! mv tfrockpaperscissors tfrockpaperscissors

In [ ]: from keras.preprocessing.image import ImageDataGenerator

ImageDataGenerator = ImageDataGenerator(
    rescale=1./255,
    validation_split = 0.2
)

In [ ]: from google.colab import drive
drive.mount('/content/drive')

model_dir = '/content/drive'

In [ ]: ImageTrain = ImageDataGenerator.flow_from_directory(
    model_dir,
    target_size=(150, 150),
    classes= ['rock', 'paper', 'scissors'],
    class_mode='categorical',
    subset='training'
)

ImageVal = ImageDataGenerator.flow_from_directory(
    model_dir,
    target_size=(150, 150),

```

```
[1] ImageDataGenerator.flow_from_directory(
    root_dir,
    target_size=(150, 150),
    classes=['rock', 'paper', 'scissors'],
    class_mode='categorical',
    subset='training'
)

ImageDataGenerator.flow_from_directory(
    root_dir,
    target_size=(150, 150),
    classes=['rock', 'paper', 'scissors'],
    class_mode='categorical',
    subset='validation'
)

Found 150 images belonging to 3 classes.
Found 50 images belonging to 3 classes.

[1] ImageModel = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(256, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(512, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(3, activation='softmax')
])
```

```
[1] ImageModel.compile(loss='categorical_crossentropy',
    optimizer=tf.optimizers.Adam(),
    metrics=['accuracy'])

[1] ImageModel.fit(
    ImageData,
    steps_per_epoch=10,
    epochs=10,
    validation_data=ImageVal,
    validation_steps=10,
    verbose=1)

Epoch 1/10 -----] - 10s 2s/step - loss: 0.8181 - accuracy: 0.0733 - val_loss: 0.1044 - val_accuracy: 0.8531
Epoch 2/10 -----] - 10s 2s/step - loss: 0.8663 - accuracy: 0.0759 - val_loss: 0.1182 - val_accuracy: 0.8606
Epoch 3/10 -----] - 10s 2s/step - loss: 0.8694 - accuracy: 0.0803 - val_loss: 0.1172 - val_accuracy: 0.8655
Epoch 4/10 -----] - 10s 2s/step - loss: 0.8257 - accuracy: 0.0934 - val_loss: 0.1194 - val_accuracy: 0.8625
Epoch 5/10 -----] - 10s 2s/step - loss: 0.8133 - accuracy: 0.0934 - val_loss: 0.1189 - val_accuracy: 0.8636
Epoch 6/10 -----] - 10s 2s/step - loss: 0.8666 - accuracy: 0.0934 - val_loss: 0.1205 - val_accuracy: 0.8668
Epoch 7/10 -----] - 10s 2s/step - loss: 0.8125 - accuracy: 0.0912 - val_loss: 0.1075 - val_accuracy: 0.8719
Epoch 8/10 -----] - 10s 2s/step - loss: 0.8158 - accuracy: 0.0909 - val_loss: 0.1067 - val_accuracy: 0.8691
Epoch 9/10 -----] - 10s 2s/step - loss: 0.8188 - accuracy: 0.0916 - val_loss: 0.1192 - val_accuracy: 0.8631
Epoch 10/10 -----] - 10s 2s/step - loss: 0.8188 - accuracy: 0.0916 - val_loss: 0.1192 - val_accuracy: 0.8631
```

```
[1] import numpy as np
from google.colab import files
from keras.preprocessing import image
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import matplotlib

uploader = files.upload()

for fn in uploader.keys():
    # predicting images
    path = fn
    img = image.load_img(path, target_size=(150, 150))
    imgplot = plt.imshow(img)
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

    images = np.vstack([x])
    classes = ImageModel.predict(images, batch_size=1)

    print(classes)

    if classes[0][0] > classes[0][1] and classes[0][0] > classes[0][2]:
        print('rock')
    elif classes[0][1] > classes[0][0] and classes[0][1] > classes[0][2]:
        print('paper')
    elif classes[0][2] > classes[0][0] and classes[0][2] > classes[0][1]:
        print('scissors')
```

to the chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving 0a3f0b1513a0b8.png to 0a3f0b1513a0b8.png

0a3f0b1513a0b8.png

data:



Figure 8: Script untuk Proyek Akhir Modul “Belajar Machine Learning untuk Pemula”

8. Belajar Dasar Pemrograman Web

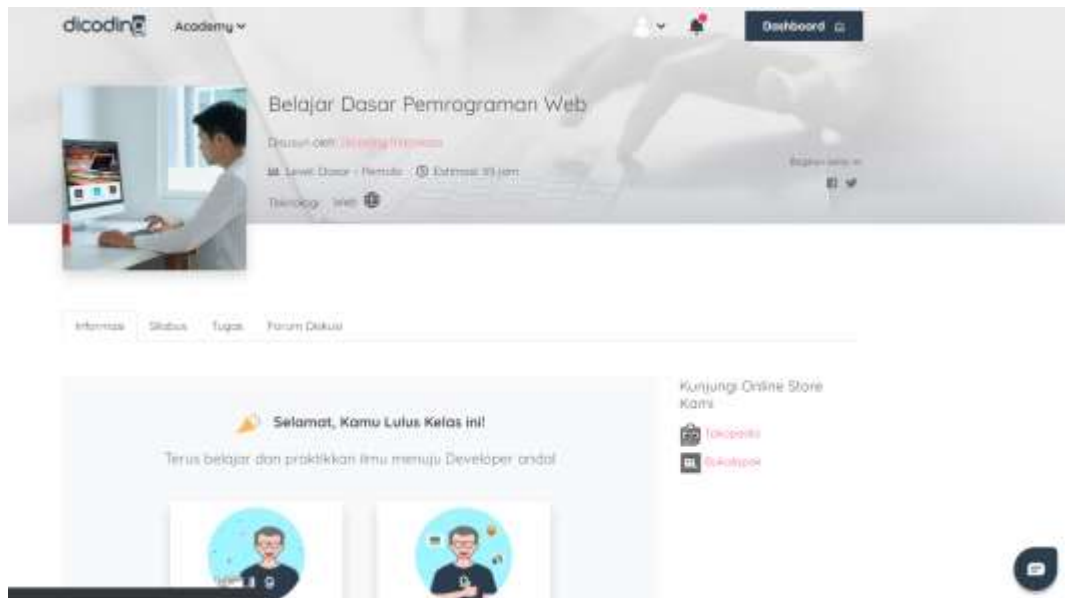


Figure 9: Modul “Belajar Dasar Pemrograman Web” Selesai

Proyek Akhir:

Berikut adalah kriteria untuk proyek akhir:

- Terdapat elemen `<header>`, `<footer>`, `<main>`, `<article>`, dan `<aside>` di berkas HTML.
- Masing-masing elemen wajib berisi konten yang peruntukannya sesuai dengan elemen tersebut (menerapkan konsep semantic HTML dalam menyusun struktur website).
- Sebagai contoh: Header berisi judul dan navigation. Sedangkan konten artikel tidak boleh berada pada Header.
- Wajib menampilkan identitas diri (biodata diri) yang minimal harus berisi foto asli diri dan nama sesuai profil Dicoding. Identitas diri wajib ditampilkan dalam elemen `<aside>`.
- Menyusun layout dengan menggunakan float atau flexbox.
- Tema yang ditampilkan bebas, kecuali tema Bandung.

- Semakin detail dan lengkap website Anda maka nilai submission bisa lebih tinggi.

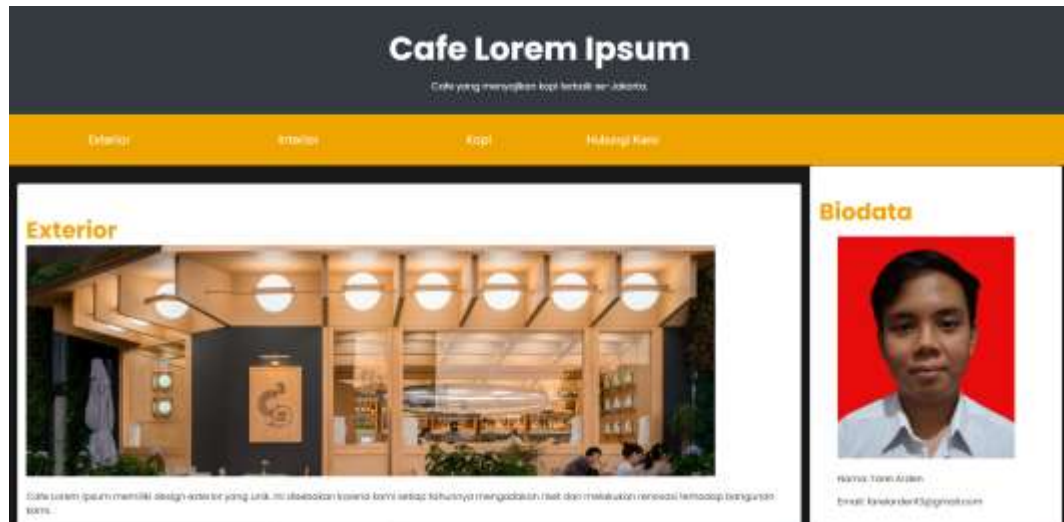


Figure 10: Script untuk Proyek Akhir Modul “Belajar Dasar Pemrograman Web”

9. Belajar Pengembangan Machine Learning

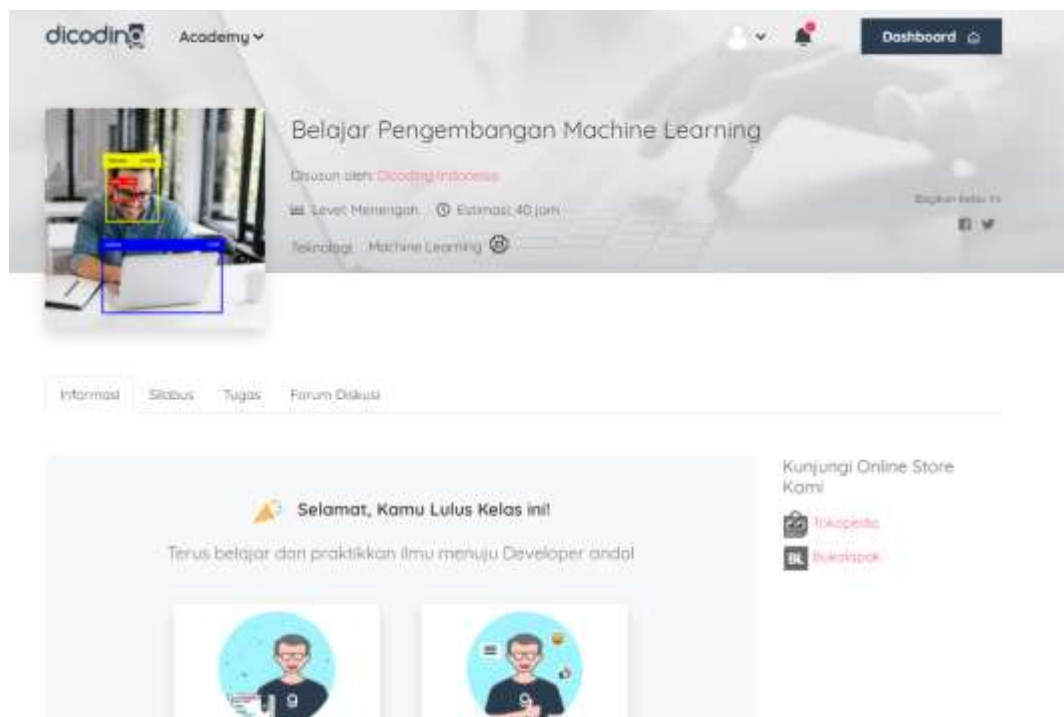


Figure 11: Modul “Belajar Pengembangan Machine Learning” Selesai

Proyek Pertama:

Kriteria Proyek Pertama:

- Dataset yang akan dipakai bebas, namun minimal memiliki 1000 sampel.
- Harus menggunakan LSTM dalam arsitektur model.
- Harus menggunakan model sequential.
- Validation set sebesar 20% dari total dataset.
- Harus menggunakan Embedding.
- Harus menggunakan fungsi tokenizer.
- Akurasi dari model minimal 75%.

Natural Language Processing with Disaster Tweets

Notebook Data Logs Comments (0) Settings

Nama: Farel Arden

```
In [3]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load
# downloading a tokenization script created by the Google team
!wget --quiet https://raw.githubusercontent.com/tensorflow/models/master/official/nlp/bert/tokenization.py
import seaborn as sns
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import nltk, re, string

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 2GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
In [2]: train_df = pd.read_csv("/kaggle/input/nlp-getting-started/train.csv")
test_df = pd.read_csv("/kaggle/input/nlp-getting-started/test.csv")
```

```
In [3]: train_df.head()
```

```
Out[3]:
```

	id	keyword	location	text	target
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
2	5	NaN	NaN	All residents asked to 'shelter in place' are...	1
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1

```
In [4]: import pandas_profiling as pp
pp.ProfileReport(train_df)
```

Summarize dataset: 100%  18/18 [00:06<00:00, 2.10it/s, Completed]

Generate report structure: 100%  1/1 [00:04<00:00, 4.87s/it]

Render HTML: 100%  1/1 [00:00<00:00, 1.44it/s]

Pandas Profiling Report

Overview Variables Interactions Correlations Missing values Sample

Overview

Overview Warnings **8** Reproduction

Dataset statistics

Number of variables	5
Number of observations	7813
Missing cells	2594
Missing cells (%)	6.8%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	297.5 KiB
Average record size in memory	40.0 B

Variable types

Numeric	1
Categorical	4

```
In [5]: train_df.shape
```

```
Out[5]: (7613, 5)
```

```
In [6]: train_df.location.unique()
```

```
Out[6]: array([nan, 'Birmingham', 'Est. September 2012 - Bristol', ...,
        'Vancouver, Canada', 'London ', 'Lincoln'], dtype=object)
```

```
In [7]: train_df.keyword.unique()
```

```
Out[7]: array([nan, 'ablaze', 'accident', 'aftershock', 'airplane%20accident',
        'ambulance', 'annihilated', 'annihilation', 'apocalypse',
        'armageddon', 'army', 'arson', 'arsonist', 'attack', 'attacked',
        'avalanche', 'battle', 'bioterror', 'bioterrorism', 'blaze',
        'blazing', 'bleeding', 'blew%20up', 'blight', 'blizzard', 'blood',
        'bloody', 'blown%20up', 'body%20bag', 'body%20bagging',
        'body%20bags', 'bomb', 'bombed', 'bombing', 'bridge%20collapse',
        'buildings%20burning', 'buildings%20on%20fire', 'burned',
        'burning', 'burning%20buildings', 'bush%20fires', 'casualties',
        'casualty', 'catastrophe', 'catastrophic', 'chemical%20emergency',
        'cliff%20fall', 'collapse', 'collapsed', 'collide', 'collided',
        'collision', 'crash', 'crashed', 'crush', 'crushed', 'curfew',
        'cyclone', 'damage', 'danger', 'dead', 'death', 'deaths', 'debris',
        'deluge', 'deluged', 'demolish', 'demolished', 'demolition',
        'derail', 'derailed', 'derailment', 'desolate', 'desolation',
        'destroy', 'destroyed', 'destruction', 'detonate', 'detonation',
        'devastated', 'devastation', 'disaster', 'displaced', 'drought',
        'drown', 'drowned', 'drowning', 'dust%20storm', 'earthquake',
        'electrocute', 'electrocuted', 'emergency', 'emergency%20plan',
        'emergency%20services', 'engulfed', 'epicentre', 'evacuate',
        'evacuated', 'evacuation', 'explode', 'exploded', 'explosion',
        'eyewitness', 'famine', 'fatal', 'fatalities', 'fatality', 'fear',
        'fire', 'fire%20truck', 'first%20responders', 'flames',
        'flattened', 'flood', 'flooding', 'floods', 'forest%20fire',
        'forest%20fires', 'hail', 'hailstorm', 'harm', 'hazard',
        'hazardous', 'heat%20wave', 'hellfire', 'hijack', 'hijacker',
        'hijacking', 'hostage', 'hostages', 'hurricane', 'injured',
        'injuries', 'injury', 'inundated', 'inundation', 'landslide',
        'lava', 'lightning', 'loud%20bang', 'mass%20murder',
        'mass%20murderer', 'massacre', 'mayhem', 'meltdown', 'military',
        'mudslide', 'natural%20disaster', 'nuclear%20disaster',
        'nuclear%20reactor', 'obliterate', 'obliterated', 'obliteration',
        'oil%20spill', 'outbreak', 'pandemonium', 'panic', 'panicking',
        'police', 'quarantine', 'quarantined', 'radiation%20emergency',
        'rainstorm', 'razed', 'refugees', 'rescue', 'rescued', 'rescuers',
        'riot', 'rioting', 'rubble', 'ruin', 'sandstorm', 'screamed',
        'screaming', 'screams', 'seismic', 'sinkhole', 'sinking', 'siren',
```

```
In [8]: dup_train = train_df['text'].duplicated().sum()
print(f'there are {dup_train} tweets duplicated in train_df.')
```

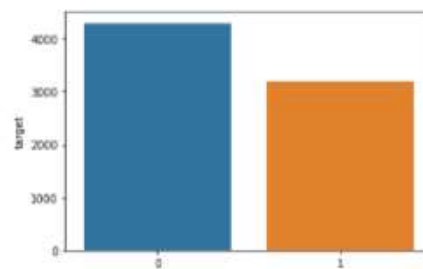
```
there are 110 tweets duplicated in train_df.
```

```
In [9]: train_df = train_df.drop_duplicates(subset=['text'], keep='first')
```

```
In [10]: # check the distribution of the disaster and no-disaster tweets
count = train_df['target'].value_counts()
sns.barplot(count.index, count)
count
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be 'data', and passing other arguments without an explicit keyword will result in an error or misinterpretation.
FutureWarning
```

```
Out[10]:
0    4305
1    3198
Name: target, dtype: int64
```



```
In [11]: def Data_Cleaning(text):
text = text.lower()
text = re.sub("won't", "will not", text)
text = re.sub("can't", "can not", text)
text = re.sub("don't", "do not", text)

text = re.sub('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&#]|!*\(\)|)\.'\n
              '(?:%[(0-9a-fA-F][0-9a-fA-F]))+', ' ', text)
text = re.sub(r'&?', ' ', text)
text = re.sub(r'&lt;', ' ', text)
text = re.sub(r'&gt;', ' ', text)
```

In [11]:

```
def Data_Cleaning(text):
    text = text.lower()
    text = re.sub("won't", "will not", text)
    text = re.sub("can't", "can not", text)
    text = re.sub("don't", "do not", text)

    text = re.sub('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&#]|!*\(\)|)\.?!|'
                  '(?:%[\0-9a-fA-F][0-9a-fA-F]))+', ' ', text)
    text = re.sub(r'&?', ' ', text)
    text = re.sub(r'&lt;', ' ', text)
    text = re.sub(r'&gt;', ' ', text)

    text = re.sub(r'\d{2}:\d{2}:\d{2}', ' ', text)
    text = re.sub(r'UTC', ' ', text)
    text = re.sub(r'\d{2}km', ' ', text)
    text = re.sub(r'\b\d+\b', ' ', text) # removing the numbers

    text = re.sub(r"#", "", text)
    text = re.sub(r"(?:\@)\w+", ' ', text)
    text = re.sub(r'\n', ' ', text)

    text = text.translate(str.maketrans(string.punctuation, ' '*len(string.punctuation)))
    text = re.sub(' +', ' ', text) # remove multiple spaces

    text = [word for word in word_tokenize(text) if not word in stopwords.words('english')]
    text = ' '.join(text)

    return text
```

In [12]:

```
train_df['text'] = train_df['text'].apply(lambda x: Data_Cleaning(x))
test_df['text'] = test_df['text'].apply(lambda x: Data_Cleaning(x))
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
"""Entry point for launching an IPython kernel.
```

In [13]:

```
text = train_df['text'].values
target = train_df['target'].values
```

In [14]:

```
from sklearn.model_selection import train_test_split
text_train, text_valid, target_train, target_valid = train_test_split(text, target, test_size=
0.2)
```


Natural Language Processing with Disaster Tweets

Notebook Data Logs Comments (0) Settings

In [1]:

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

tokenizer = Tokenizer(num_words=5000, oov_token='x')
tokenizer.fit_on_texts(text_train)
tokenizer.fit_on_texts(text_valid)

sekuens_latih = tokenizer.texts_to_sequences(text_train)
sekuens_test = tokenizer.texts_to_sequences(text_valid)

padded_latih = pad_sequences(sekuens_latih)
padded_test = pad_sequences(sekuens_test)
```

```
2021-09-12 13:14:15.835525: W tensorflow/stream_executor/platform/default/dso_loader.cc:60]
Could not load dynamic library 'libcudart.so.11.0'; dLError: libcudart.so.11.0: cannot open
shared object file: No such file or directory; LD_LIBRARY_PATH: /opt/conda/lib
2021-09-12 13:14:15.835664: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above
cudart dLError if you do not have a GPU set up on your machine.
```

In [16]:

```
import tensorflow as tf
class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if(logs.get('accuracy')>0.9):
            print("\nAkurasi telah mencapai >90%!")
            self.model.stop_training = True
callbacks = myCallback()
```

In [17]:

```
import tensorflow as tf
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(input_dim=5000, output_dim=16),
    tf.keras.layers.LSTM(64),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(24, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
2021-09-12 13:14:22.612106: I tensorflow/compiler/jit/xla_cpu_device.cc:41] Not creating XLA
devices, tf_xla_enable_xla_devices not set
2021-09-12 13:14:22.618298: W tensorflow/stream_executor/platform/default/dso_loader.cc:60]
Could not load dynamic library 'libcuda.so.1'; dLError: libcuda.so.1: cannot open shared obj
ect file: No such file or directory; LD_LIBRARY_PATH: /opt/conda/lib
2021-09-12 13:14:22.618350: W tensorflow/stream_executor/cuda/cuda_driver.cc:326] failed cal
l to cuInit: UNKNOWN ERROR (383)
2021-09-12 13:14:22.618385: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kerne
l driver does not appear to be running on this host (07a668a6ccf3): /proc/driver/nvidia/vers
ion does not exist
2021-09-12 13:14:22.619539: I tensorflow/core/platform/cpu_feature_guard.cc:142] This Tensor
Flow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the followi
ng CPU instructions in performance-critical operations: AVX2 FMA
```

```
In [18]: target_train.shape
```

```
Out[18]: (6002,)
```

```
In [19]: target_train = np.asarray(target_train).astype('float32').reshape((-1,1))
         target_valid = np.asarray(target_valid).astype('float32').reshape((-1,1))
```

```
In [20]: paddedlatih.shape
```

```
Out[20]: (6002, 22)
```

```
In [21]: padded_test.shape
```

```
Out[21]: (1501, 23)
```

```
In [22]: target_train.shape
```

```
Out[22]: (6002, 1)
```

```
In [23]: target_valid.shape
```

```
Out[23]: (1501, 1)
```

```
In [24]: model.summary()
```

```
Model: "sequential"
-----
Layer (type)                 Output Shape              Param #
-----
embedding (Embedding)        (None, None, 16)         80000
-----
lstm (LSTM)                   (None, 64)                20736
-----
dense (Dense)                 (None, 128)               8320
-----
dense_1 (Dense)               (None, 64)                8256
-----
dense_2 (Dense)               (None, 24)                1560
-----
dense_3 (Dense)               (None, 1)                 25
-----
Total params: 118,897
```

```
In [25]: num_epochs = 30
history = model.fit(padded_latih, target_train, epochs=num_epochs,
                    validation_data=(padded_test, target_valid), verbose=2, callbacks=[callbacks])
```

```
2021-09-12 13:14:24.059907: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116]
None of the MLIR optimization passes are enabled (registered 2)
2021-09-12 13:14:24.070863: I tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU F
requency: 2199995000 Hz
```

```
Epoch 1/30
188/188 - 5s - loss: 0.5523 - accuracy: 0.7124 - val_loss: 0.4712 - val_accuracy: 0.7828
Epoch 2/30
188/188 - 2s - loss: 0.3496 - accuracy: 0.8564 - val_loss: 0.4725 - val_accuracy: 0.7868
Epoch 3/30
188/188 - 2s - loss: 0.2627 - accuracy: 0.8999 - val_loss: 0.5607 - val_accuracy: 0.7675
Epoch 4/30
188/188 - 2s - loss: 0.1953 - accuracy: 0.9304 - val_loss: 0.6031 - val_accuracy: 0.7588

Akurasi telah mencapai >90%
```

```
In [26]: import matplotlib.pyplot as plt
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Akurasi Model')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

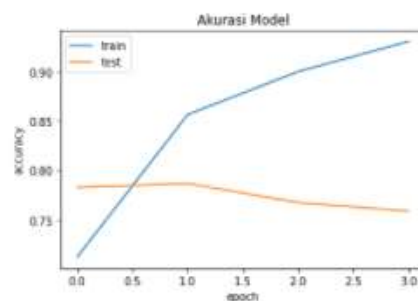


Figure 12: Script untuk Proyek Pertama Modul “Belajar Pengembangan Machine Learning”

Proyek Kedua:

Kriteria Proyek Kedua:

- Dataset yang akan dipakai bebas, namun minimal memiliki 1000 sampel.
- Harus menggunakan LSTM dalam arsitektur model.

- Validation set sebesar 20% dari total dataset.
- Model harus menggunakan model sequential.
- Harus menggunakan Learning Rate pada Optimizer.
- $MAE < 10\%$ skala data.

Sunspots

Notebook Data Logs Comments (1) Settings

Name: Farel Arden

Table of Contents

Name: Farel Arden

```
In [1]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session

/kaggle/input/daily-sun-spot-data-1818-to-2019/sunspot_data.csv
/kaggle/input/sunspots/Sunspots.csv
```

Fungsi dibawah ini digunakan untuk mem-plot grafik

```
In [2]: def plot_series(time, series, format="-", start=0, end=None):
plt.plot(time[start:end], series[start:end], format)
plt.xlabel("Time")
plt.ylabel("Value")
plt.grid(True)
```

```
In [3]: dataset=pd.read_csv("../input/sunspots/Sunspots.csv")
dataset.head()
```

```
Out[3]:
```

	Unnamed: 0	Date	Monthly Mean Total Sunspot Number
0	0	1749-01-31	66.7
1	1	1749-02-28	104.3
2	2	1749-03-31	116.7
3	3	1749-04-30	92.8
4	4	1749-05-31	141.7

Sunspots

Notebook

Data

Logs

Comments (1)

Settings

4

Edit

In []:

```
import pandas_profiling as pp
pp.ProfileReport(dataset)
```

Table of Contents

>

Name: Farel Arden

Summarize dataset: 100%

16/16 [00:04:00:00, 2.38it/s, Completed]

Generate report structure: 100%

1/1 [00:02:00:00, 2.36s/it]

Render HTML: 100%

1/1 [00:00:00:00, 1.00it/s]

Pandas Profiling Report

Overview

Variables

Interactions

Correlations

Missing values

Sample

Overview

Overview

Warnings 6

Reproduction

Dataset statistics

Number of variables	3
Number of observations	3265
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	78.8 KiB
Average record size in memory	24.0 B

Variable types

Numeric	2
Categorical	1

Variables

Unnamed: 0

Distinct 3265

Minimum 0

Maximum 3264

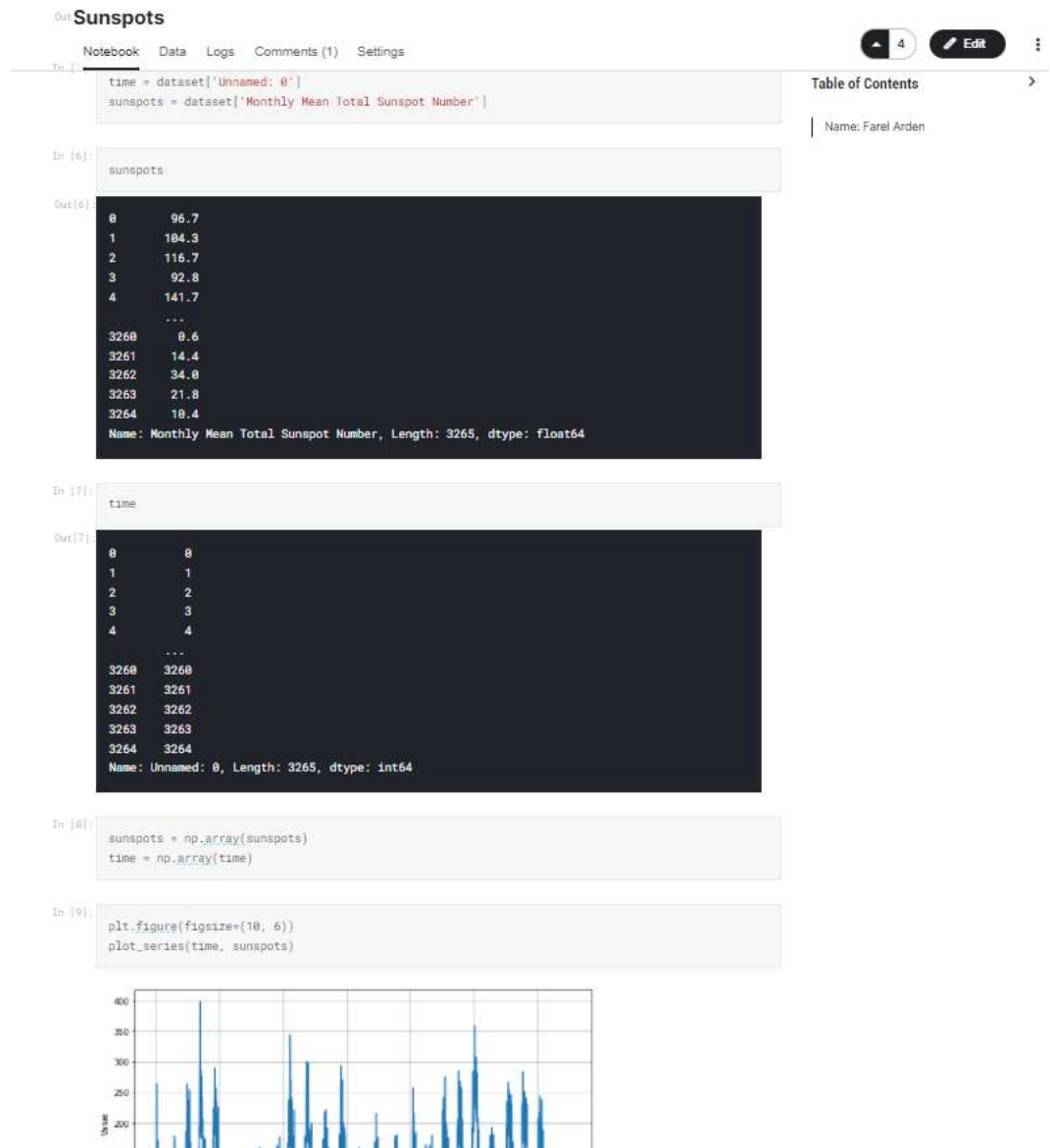
Real number (float64)

Distinct (%) 100.0%

Out[4]:

In [5]:

```
time = dataset['Unnamed: 0']
sunspots = dataset['Monthly Mean Total Sunspot Number']
```



Sunspots

Notebook Data Logs Comments (1) Settings

```
In [10]: # 88% dari data yang ada = 2612, sehingga kita hanya membutuhkan 653 rows untuk memvalidasi data.
split_time = 2612
time_train = time[:split_time]
x_train = sunspots[:split_time]
time_valid = time[split_time:]
x_valid = sunspots[split_time:]

window_size = 30
batch_size = 32
```

```
In [11]: x_train
```

```
Out[11]: array([ 96.7, 104.3, 116.7, ...,  67.7,  80.2,  72.6])
```

```
In [12]: time_train
```

```
Out[12]: array([  0,   1,   2, ..., 2609, 2610, 2611])
```

```
In [13]: def windowed_dataset(series, window_size, batch_size, shuffle_buffer):
series = tf.expand_dims(series, axis=-1)
ds = tf.data.Dataset.from_tensor_slices(series)
ds = ds.window(window_size + 1, shift=1, drop_remainder=True)
ds = ds.flat_map(lambda w: w.batch(window_size + 1))
ds = ds.shuffle(shuffle_buffer)
ds = ds.map(lambda w: (w[:-1], w[1:]))
return ds.batch(batch_size).prefetch(1)
```

```
In [14]: import tensorflow as tf
train_set = windowed_dataset(x_train, window_size=60, batch_size=100, shuffle_buffer=1000)
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv1D(filters=60, kernel_size=5,
                           strides=1, padding="causal",
                           activation="relu",
                           input_shape=[None, 1]),
    tf.keras.layers.LSTM(60, return_sequences=True),
    tf.keras.layers.LSTM(60, return_sequences=True), #declaring the LSTM layers
    tf.keras.layers.Dense(10, activation="relu"),
    tf.keras.layers.Dense(10, activation="relu"),
    tf.keras.layers.Dense(1),
    tf.keras.layers.Lambda(lambda x: x * 400)
])
```

```
2021-09-17 02:48:21.815280: I tensorflow/stream_executor/platform/default/dso_loader.cc:49]
Successfully opened dynamic library libcudart.so.11.0
2021-09-17 02:48:25.535938: I tensorflow/compiler/jit/xla_cpu_device.cc:41] Not creating XLA
devices, tf_xla_enable_xla_devices not set
2021-09-17 02:48:25.539610: I tensorflow/stream_executor/platform/default/dso_loader.cc:49]
```

```

1 optimizer=tf.keras.optimizers.SGD(lr=1.0000e-04, momentum=0.9)
2 model.compile(loss=tf.keras.losses.Huber(),
3               optimizer=optimizer,
4               metrics=["mae"])
5 history = model.fit(train_set, epochs=100)

```

Figure 13: Script untuk Proyek Kedua Modul “Belajar Pengembangan Machine Learning”

Proyek Akhir:

Kriteria Proyek Akhir:

- Dataset yang akan dipakai bebas, namun minimal memiliki 1000 buah gambar.
- Dataset tidak pernah digunakan pada submission kelas machine learning sebelumnya.
- Dataset dibagi menjadi 80% train set dan 20% test set.
- Model harus menggunakan model sequential.
- Model harus menggunakan Conv2D Maxpooling Layer.
- Akurasi pada training dan validation set minimal sebesar 80%.
- Menggunakan Callback.
- Membuat plot terhadap akurasi dan loss model.
- Menulis kode untuk menyimpan model ke dalam format TF-Lite.

Digit Recognizer - Farel Arden

Notebook Data Logs Comments (0) Settings

```
In [1]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import seaborn as sns
%matplotlib inline
from tensorflow.keras.utils import to_categorical

from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Flatten, Dense, Dropout, MaxPooling2D, BatchNormali
zation

from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau
from tensorflow.keras import regularizers, optimizers
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under t
he input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as out
put when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the cur
rent session
```

```
2021-09-13 13:58:10.813990: I tensorflow/stream_executor/platform/default/dso_loader.cc:49]
Successfully opened dynamic library libcudart.so.11.0
```

```
/kaggle/input/digit-recognizer/sample_submission.csv
/kaggle/input/digit-recognizer/train.csv
/kaggle/input/digit-recognizer/test.csv
```

```
In [2]: train_data=pd.read_csv("../input/digit-recognizer/train.csv")
test_data=pd.read_csv("../input/digit-recognizer/test.csv")
sample_submission=pd.read_csv("../input/digit-recognizer/sample_submission.csv")
```

Digit Recognizer - Farel Arden

Notebook Data Logs Comments (0) Settings

In []: `train_data.head()`

Out[3]:

	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774	pixel775	pixel776	pixel7
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0
3	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0

5 rows × 785 columns

In [4]: `train_data.shape`

Out[4]: `(42000, 785)`

In [5]: `train_data.isnull().any()`

Out[5]:

```
label      False
pixel0     False
pixel1     False
pixel2     False
pixel3     False
...
pixel779   False
pixel780   False
pixel781   False
pixel782   False
pixel783   False
Length: 785, dtype: bool
```

In [6]:

```
y = train_data["label"]
x = train_data.drop(labels = ["label"],axis = 1)
```

In [7]:

```
x = x.values.reshape(-1,28,28,1)
test_data = test_data.values.reshape(-1,28,28,1)
```

In [8]:

```
x = x / 255.0
x_test = test_data / 255.0
```

In [9]: `y = to_categorical(y, num_classes=10)`

Digit Recognizer - Farel Arden

Notebook Data Logs Comments (0) Settings

```
In [8]:
x = x / 255.0
x_test = test_data / 255.0
```

```
In [9]:
y = to_categorical(y, num_classes=10)
```

```
In [10]:
import tensorflow as tf
class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if(logs.get('accuracy')>0.98):
            print("\n\nThe accuracy has reached 99%!")
            self.model.stop_training = True
callbacks = myCallback()
```

Training and Building the model

```
In [11]:
# model = Sequential()
# model.add(Conv2D(filters=32, kernel_size=3, activation="relu", input_shape=(28, 28, 1)))
# model.add(Conv2D(filters=32, kernel_size=3, activation="relu"))
# model.add(Flatten())
# model.add(Dense(128, activation="relu"))
# model.add(Dense(10, activation="softmax"))
```

```
In [12]:
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation="relu", input_shape=(28, 28,
1)),
    tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation="relu"),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation="relu"),
    tf.keras.layers.Dense(10, activation="softmax")
])
```

Digit Recognizer - Farel Arden

Notebook Data Logs Comments (0) Settings

In [13]:

```
x.shape
```

Out[13]:

```
(42000, 28, 28, 1)
```

In [14]:

```
y.shape
```

Out[14]:

```
(42000, 10)
```

In [15]:

```
x_train, x_valid, y_train, y_valid = train_test_split(x, y, test_size=0.2)
```

In [16]:

```
x_train.shape
```

Out[16]:

```
(33600, 28, 28, 1)
```

In [17]:

```
y_train.shape
```

Out[17]:

```
(33600, 10)
```

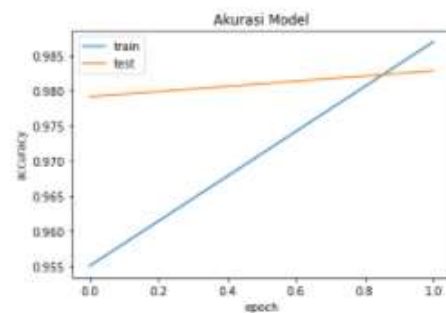
In [18]:

```
model.compile(loss="categorical_crossentropy", metrics=["accuracy"], optimizer="adam")

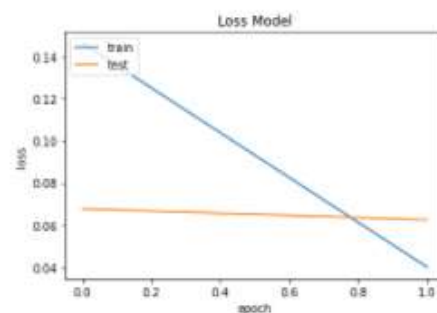
history = model.fit(x_train, y_train, epochs=10,
                    validation_data=(x_valid, y_valid), verbose=2, callbacks=[callbacks], batch_
                    size=32)
```

Plotting the val accuracy and the accuracy of the model

```
In [19]: import matplotlib.pyplot as plt
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Akurasi Model')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



```
In [20]: plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Loss Model')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



TF-Lite formatting

```
In [21]: import tensorflow as tf
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
```

```
In [22]: with tf.io.gfile.GFile('model.tflite', 'wb') as f:
f.write(tflite_model)
```

Figure 14: Script untuk Proyek Akhir Modul “Belajar Pengembangan Machine Learning”

10. Belajar Dasar Pemrograman Web

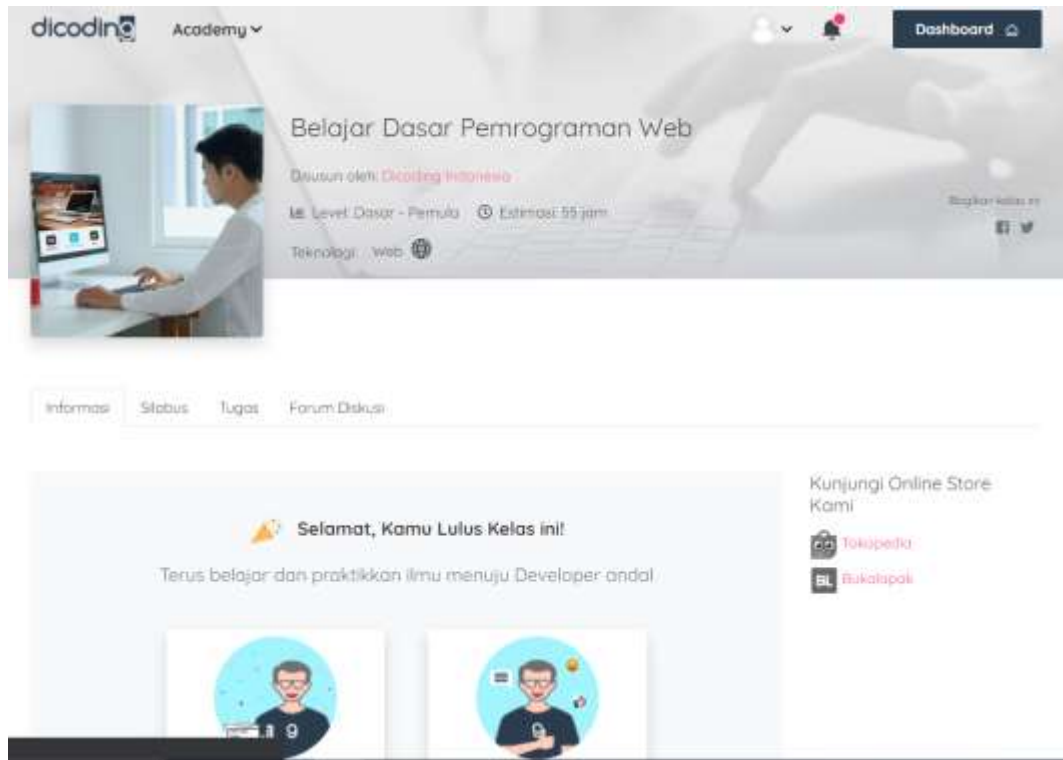


Figure 15: Modul “Belajar Dasar Pemrograman Web” Selesai

Proyek Akhir:

Kriteria Proyek Akhir:

- dasar HTML (dapat dipelajari mulai dari modul Pengenalan ke HTML),
- penerapan elemen semantic (dapat dipelajari mulai dari modul Semantic HTML - Mengorganisasikan Halaman Konten),
- menerapkan styling menggunakan CSS (dapat dipelajari mulai dari modul Pengenalan ke Styling); dan
- layouting menggunakan float ataupun flexbox (dapat dipelajari mulai dari modul Floating).

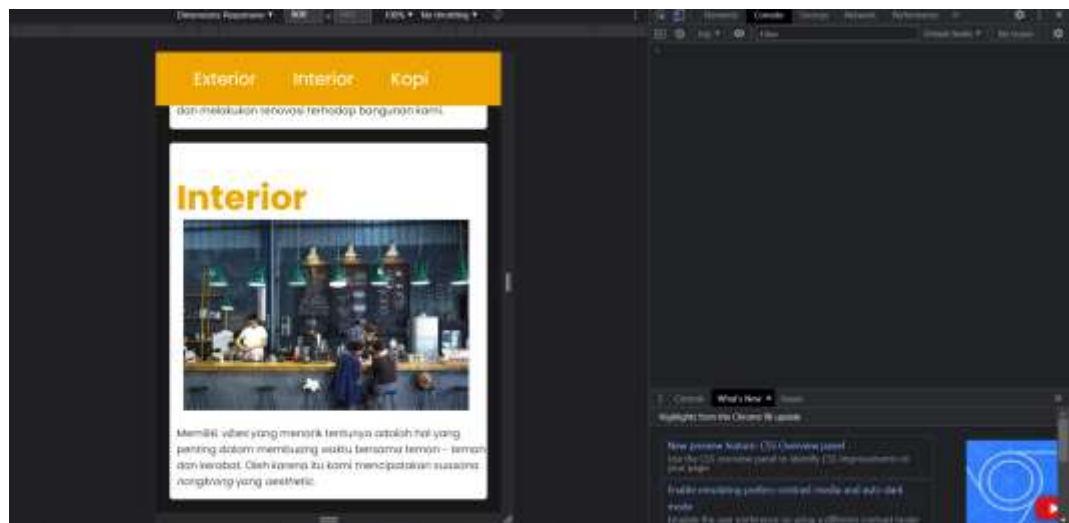


Figure 16: Script untuk Proyek Akhir Modul “Belajar Dasar Pemrograman Web”

11. Machine Learning Terapan

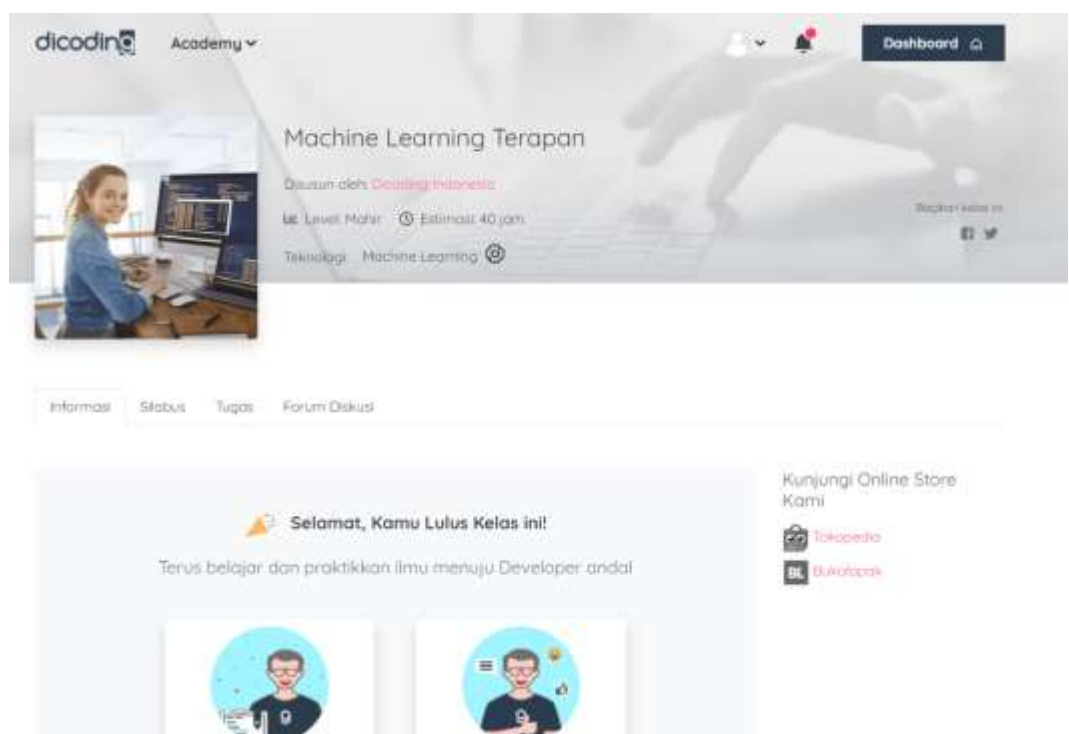


Figure 17: Modul “Machine Learning Terapan” Selesai

Proyek Pertama:

- Project merupakan hasil pekerjaan sendiri.
- Project belum pernah digunakan untuk submission kelas Machine Learning di Dicoding dan belum pernah dipublikasikan di platform manapun.
- Dataset yang dipakai merupakan data kuantitatif (minimum 500 sampel data).
- Memberikan dokumentasi menggunakan text cell pada notebook (.ipynb) untuk menjelaskan setiap tahapan proyek.
- Menentukan solusi permasalahan menggunakan pendekatan machine learning atau deep learning dengan memilih salah satu dari penyelesaian berikut:
 - Klasifikasi
 - Regresi
 - Clustering
 - Time series dan forecasting
- Membuat draf laporan proyek machine learning yang menjelaskan alur proyek Anda dari mulai pemilihan domain permasalahan (problem domain), data understanding, data preparation, modeling, hingga tahap evaluasi. Ketentuan draf laporan proyek machine learning dapat Anda lihat pada sub modul berikutnya tentang Detail Laporan.

Untuk hasil pengerjaan dapat dilihat pada repository di bawah ini:

<https://github.com/farelarden/Dicoding-SIB/tree/main/Machine%20Learning%20Terapan/Titanic>

Proyek Akhir:

Kriteria Proyek Akhir:

- Project merupakan hasil pekerjaan sendiri.
- Project belum pernah digunakan untuk submission kelas Machine Learning di Dicoding dan belum pernah dipublikasikan di platform manapun.

- Dataset yang dipakai bebas, asal bisa digunakan untuk membuat sistem rekomendasi.
 - Memberikan dokumentasi menggunakan text cell pada notebook (.ipynb) untuk menjelaskan setiap tahapan proyek.
 - Menentukan solusi permasalahan dengan memilih pendekatan berikut:
 - Content-based Filtering
 - Collaborative Filtering
 - Membuat draf laporan proyek machine learning yang menjelaskan alur proyek Anda mulai dari project overview, business understanding, data understanding, data preparation, modeling, hingga tahap evaluasi.
- Ketentuan draf laporan proyek machine learning dapat Anda lihat pada sub modul berikutnya tentang Detail Laporan.

Untuk hasil pengerjaan dapat dilihat pada repository di bawah ini:

<https://github.com/farelarden/Dicoding-SIB/tree/main/Machine%20Learning%20Terapan/Rekomendasi%20Buku>

12. Evaluasi Penguasaan Machine Learning

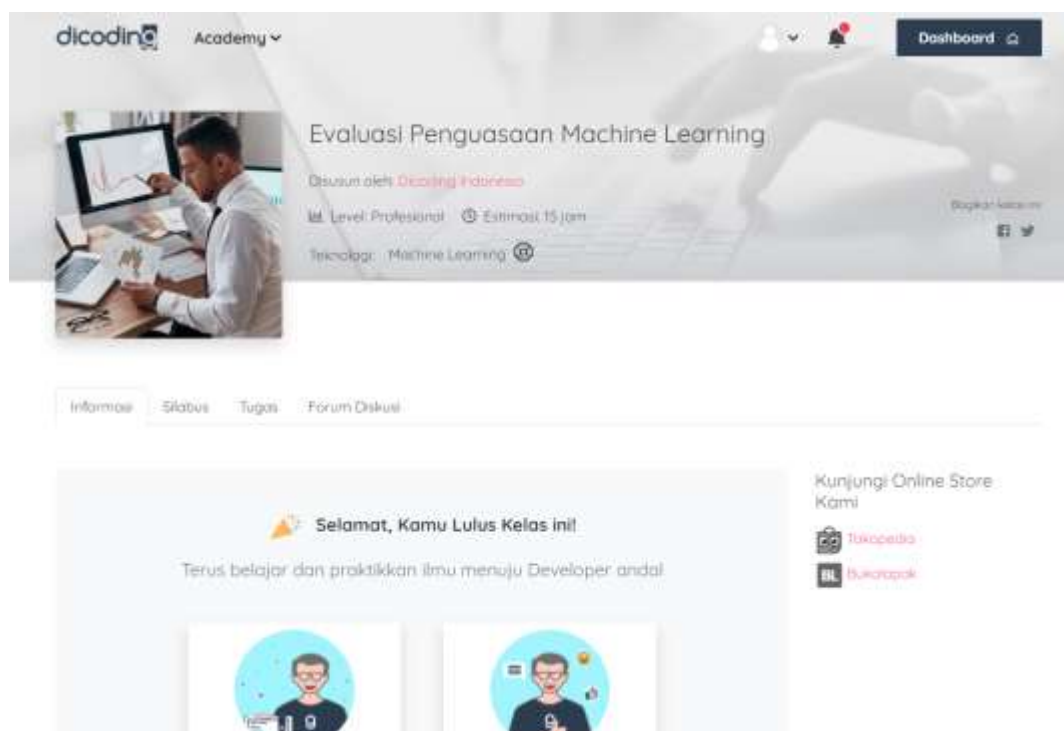


Figure 18: Modul “Evaluasi Penguasaan Machine Learning” Selesai

13. Belajar Membuat Front-End Web untuk Pemula

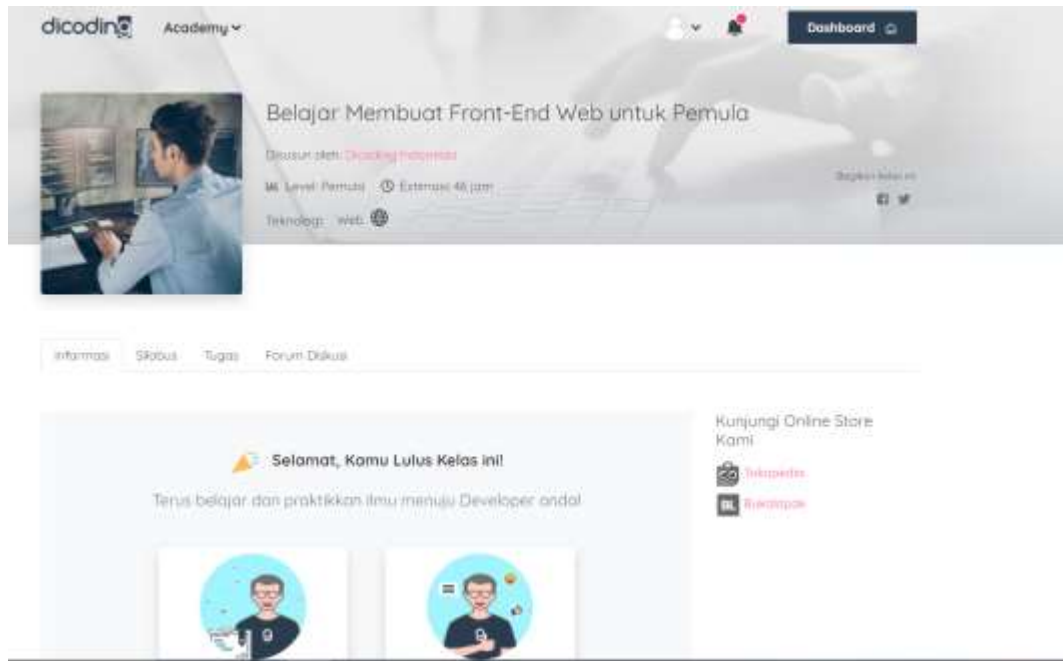


Figure 19: Modul “Belajar Membuat Front-End Web untuk Pemula” Selesai

Proyek Akhir:

Kriteria Proyek Akhir:

- Tambahkan fitur pencarian untuk mem-filter buku yang ditampilkan pada rak sesuai dengan title buku yang dituliskan pada kolom pencarian.
- Berkreasilah dengan membuat proyek Bookshelf Apps tanpa menggunakan project starter.
- Menuliskan kode dengan bersih.
- Bersihkan comment dan kode yang tidak digunakan.
- Indentasi yang sesuai.
- Terdapat improvisasi fitur seperti (pilih satu):
- Custom Dialog ketika menghapus buku.
- Dapat meng-edit buku.
- dsb.

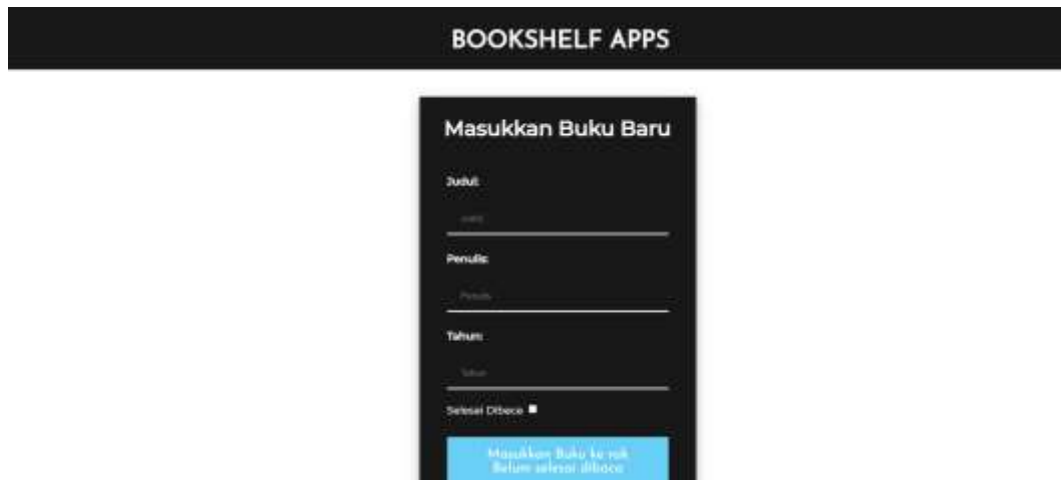


Figure 20: Script untuk Proyek Akhir Modul “Belajar Membuat Front-End Web untuk Pemula”

14. Tensorflow Developer Certification Prep (Optional)

15. Belajar Fundamental Front-End Web Development

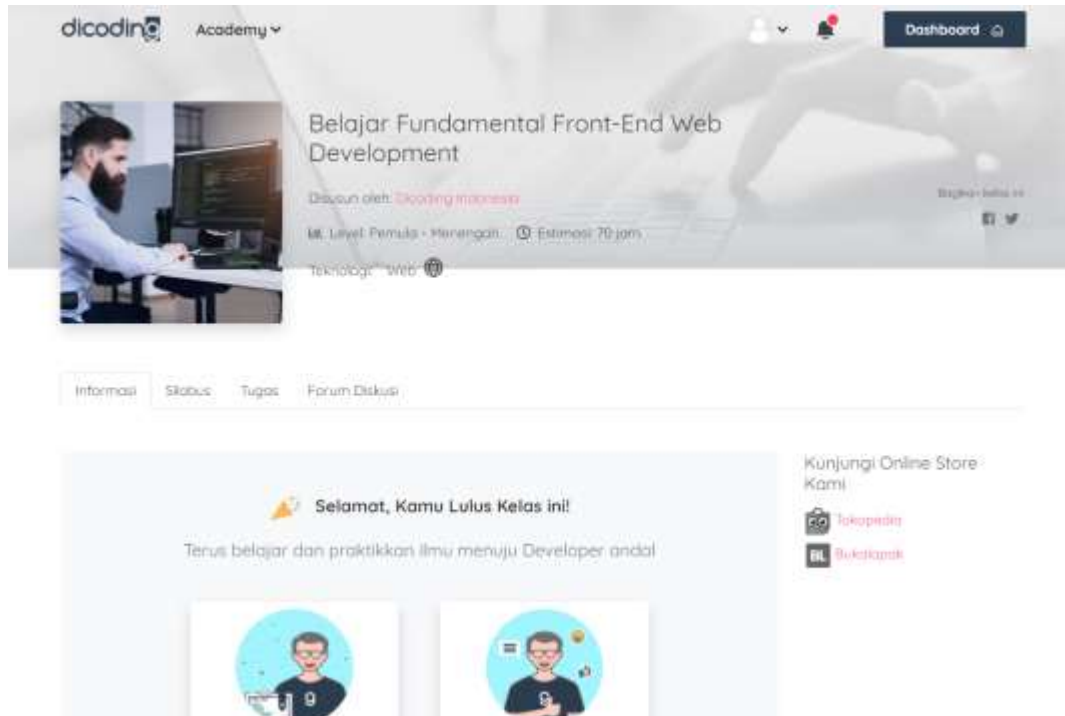


Figure 21: Modul “Belajar Fundamental Front-End Web Development”
Selesai

Proyek Akhir:

Kriteria Proyek Akhir:

- Sintaks ES6 (dapat dipelajari mulai dari modul ECMAScript 6),
- Web Components (dapat dipelajari mulai dari modul Web Component),
- Node Package Manager (dapat dipelajari mulai dari modul Node Package Manager),
- Webpack (dapat dipelajari mulai dari modul Webpack), dan
- Asynchronous JavaScript Request menggunakan XHR dan Fetch (dapat dipelajari mulai dari modul Asynchronous JavaScript Request).

CAPSTONE

JUDUL APLIKASI:

Inflativa (Inflation Predictor)

METODE PENGERJAAN:

- Melakukan pencarian dataset
- Data Pre-Processing
- Data Modeling
- Conclusion

HASIL PENGERJAAN

Pencarian Data

Langkah pertama yang saya dan anggota tim saya lakukan adalah mencari dataset. Dataset yang kami gunakan pada capstone project ini berasal dari Badan Pusat Statistika (BPS) tentang tingkat inflasi berdasarkan kota dan sektor barang.

Sumber: <https://www.bps.go.id/indicator/3/1918/1/inflasi-tahun-kalender-2018-100-menurut-kelompok-dan-sub-kelompok-02-pakaian-dan-alas-kaki.html>

Data Preprocessing

Dari data yang sudah ada, kami olah sehingga dapat digunakan untuk melatih model kami. Dapat dilihat dalam tabel di bawah ini, tanggal yang berjarak sebulan terletak pada kolom kedua setelah kolom “Urutan” disertai dengan nama – nama kota diisi dengan tingkat inflasi setiap kota tersebut.

JURATAN	BULAN	KOTA MELAKA	KOTA BANGKAWATI	KOTA SUBANG	KOTA KEMAMAN	KOTA PADANG	KOTA PADANG	KOTA BUKIT MELANGKAR	KOTA PADANG	KOTA BUNGO	KOTA JAMPAH	KOTA PALEMBANG	KOTA LUBUKALINGGA	KOTA BANGKAWATI	METANING	KOTA						
0	1/1/2014	1.1	0.25	0.1	0	0.10	0.20	1.50	0.15	0.10	0	0.10	0.08	0.08	0.34	0.37	0.08	0				
1	1/1/2014	0.86	0	0	0.29	0.39	0.20	0.17	0.15	0.60	0	0.03	0.34	0.03	0	0	0	0				
2	1/1/2014	1.42	0.1	0.78	0.39	0.56	0.22	0	0.05	0	0.06	0.06	0.07	0.03	0	0.05	0.06	0.16				
3	4/2/2014	0.17	0.06	1.06	0.12	0.67	1.01	0.91	0.06	0.06	0.16	0.10	0.37	0.06	0.07	0	0.36	0	0.07	0		
4	5/1/2014	0.61	0	0	0	0	0.06	0	0.06	0	0	0	0	0.17	0	0.54	0	0	0.06	0		
5	6/1/2014	0	0.16	0	0.03	0.62	0.44	0.05	0	0.25	0.02	0.18	0.06	0.07	0.07	0.39	0.11	0.05	0.02	0		
6	7/1/2014	0.26	0.01	0	0.25	0.58	1.24	0.06	0.03	0.38	0.43	0.04	0.09	0.1	0.07	0.34	0	0.09	1	0.09	0	
7	8/1/2014	1.09	0.02	0	0	0.1	0.32	0.31	0.31	0.1	0	0.18	0.04	0	0.3	0.18	0.43	0.31	2.54	0.06	0	
8	9/1/2014	0.25	0.16	0	0	0	0.51	0.19	0.42	1.39	0	0.06	1.06	0	0.1	0.07	0.43	0.09	0.1	0.03	0	
9	10/1/2014	0.3	0	0.06	0	0.06	0.15	0.44	0	0.37	0	0.45	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.12	0	
10	11/1/2014	0	0.01	0.24	0.68	0	0.06	1.05	0.01	0.01	0.5	0.14	1.3	0.06	0.1	1.52	0.06	0	0.07	0	0.10	
11	12/1/2014	0	0	0.26	1.62	0.31	1.05	0.03	1.1	0.12	0	1.74	0.14	0.21	1.33	0.3	1.07	0.34	0.21	0.11	0.44	0
12	1/2/2014	0.62	0.05	0.14	0.05	0.77	0.45	0	0.68	0.67	0.61	0.05	0.44	0.37	1.54	2.26	0.05	1.57	0.06	0.22	0.67	0
13	2/2/2014	0	0.05	0.36	0.14	1.5	0.2	0	0.06	0.18	0	0.06	0.06	0	0.40	0.22	0	0.02	0.13	0	1.37	0
14	3/2/2014	0	0	0.01	0.31	0.69	0.01	0.21	0.33	0	0	0.07	0	0.06	0.43	2.11	0.06	0.21	0.33	0.17	0.21	0
15	4/2/2014	0	0.02	0	0.03	0.24	0.01	0.02	0.33	0.01	0	0.07	0.33	0.34	0.77	0.06	0.76	2.07	0.06	0	0	0
16	5/2/2014	0	0.12	0.03	0.03	0.09	0.11	0.05	0.08	0.37	0.34	0.01	0.02	0	0.07	0	0.1	0.04	2.72	0.4	0.4	0
17	6/2/2014	0	0.02	0.07	0	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0
18	7/2/2014	0	0	0.29	0	0.06	0.17	0.01	0.18	0.07	0	0.04	0.13	0.25	0.01	0.06	0.06	0.06	0.06	0.06	0.06	0
19	8/2/2014	0	0.08	1.22	0.03	0.77	0.13	0.06	0.08	0.75	0	0.64	2.18	0	1.19	0.52	0	0.06	0.73	0.43	0.72	0
20	9/2/2014	0	0.05	0.29	0.03	0.82	0.03	0	0.17	0.05	0	0.06	0.18	0	0.21	0.97	0.21	0.54	0.73	0.43	0	0
21	10/2/2014	0	0.04	0.03	0.83	0.81	0.27	0	1.18	1.53	0	0.05	0.21	0	0.34	0.12	0	0.04	0	0	0.13	0
22	11/2/2014	0	0	0	0	0.2	0.1	0.10	0.06	0	0.06	0	0.43	0.05	0.05	0	0.04	0	0	0.63	0	
23	12/2/2014	0	0.11	0	0.68	0.45	0.34	0	0	0.04	0.07	0.08	0.05	0	0.03	0.73	0	0.01	0.01	0	0	0
24	1/1/2015	0	0	0.88	0.47	1.71	0.09	0.01	0.04	0	0.07	0	0.38	0.06	0.16	0.09	0.33	0	0	0	2.85	0
25	2/1/2015	0.84	0	0	0.1	0.38	0.22	0.01	0.09	0.82	0.87	0	0.09	0.09	0.02	0	0.47	0	1.29	0	0	0
26	3/1/2015	0	0.6	0.47	0.28	0.14	0.75	0	0.07	0.11	0	0.19	0.01	0.01	0.77	0	0	0	0	0	0	0
27	4/1/2015	0.73	0.01	0	0.05	0.05	0.4	0.29	0	0.02	0.04	0	0.04	0	0.42	0.06	0.75	0	0.06	0	0.12	0

Figure 23: Clean Dataset

Pada Python Notebook, kami kembali melakukan analisa terhadap dataset yang ada.

- Pertama, kami melakukan impor pada library – library yang akan kami gunakan dalam notebook ini

```
[1] import os
import pandas as pd
from fbprophet import Prophet

import warnings;
warnings.simplefilter('ignore')
```

Figure 23: Import Libraries

- Kedua, kami melakukan load pada dataset dengan melakukan clone dari github repository kami.

```
!git clone https://github.com/farelarden/Dicoding-Capstone-ML.git
```

Figure 25: Clone Github Repository

```
df_Makanan = pd.read_csv('/content/Dicoding-Capstone-ML/Dataset/Fix Dataset/fix 02 - Makanan.csv')
df_Minuman = pd.read_csv('/content/Dicoding-Capstone-ML/Dataset/Fix Dataset/fix 02 - Minuman.csv')
df_Rumah = pd.read_csv('/content/Dicoding-Capstone-ML/Dataset/Fix Dataset/fix 03 - Rumah.csv')
df_Sandang = pd.read_csv('/content/Dicoding-Capstone-ML/Dataset/Fix Dataset/fix 04 - Sandang.csv')
df_Transportasi = pd.read_csv('/content/Dicoding-Capstone-ML/Dataset/Fix Dataset/fix 07 - Transportasi.csv')
```

Figure 26: Read the dataset

- Ketiga, kami melakukan eksplorasi pada dataset yang ada

```
[7] df_Makanan.dtypes
```

Unnamed: 0	int64
BULAN	object
KOTA MEULABOH	float64
KOTA BANDA ACEH	float64
KOTA LHOKSEUMAWA	float64
	...
KOTA TERNATE	float64
MANOKWARI	float64
KOTA SORONG	float64
MERAUKE	float64
KOTA JAYAPURA	float64
Length: 84, dtype: object	

Figure 27: Menampilkan datatype dari kolom – kolom

- Keempat, kami memilih kolom bulan dan satu kolom yaitu satu nama kota untuk dijadikan target dalam proses pelatihan mesin.

```
[8] df_Makanan = df_Makanan[['KOTA MEULABOH', 'BULAN']]
df_Makanan.columns = ['y', 'ds']
```

Figure 28: Pemilihan Kota

Kita juga melakukan perubahan tipe data dari object ke datetime pada kolom “Bulan”.

```
df_Makanan.ds = pd.to_datetime(df_Makanan.ds)
```

Figure 29: Perubahan Datatype

Data Modeling

Pada proses pengerjaan ini, kami menggunakan metode Time Series dengan model yang digunakan adalah Facebook Prophet.

```
[16] m = Prophet(interval_width=0.95, daily_seasonality=True)
model = m.fit(df_Makanan)
```

Figure 30: Facebook Prophet

Kemudian kami melakukan forecasting pada data yang ada. Forecasting ini kami lakukan hingga tahun 2030.

```
[17] future = m.make_future_dataframe(periods=100,freq='M')
```

Figure 31: Forecasting

Evaluation

Facebook prophet tidak hanya memberikan nilai prediksi saja, tetapi juga upper bound dan lower bound, trend tiap tahun dan bulan. Hal ini tentunya akan sangat berguna bila kita memiliki tujuan yang menggunakan informasi – informasi tersebut.

```
[10] forecast = m.predict(future)
forecast.head()
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	daily	daily_lower	daily_upper	yearly	yearly_lower	yearly_upper
0	2018-01-01	0.040172	0.494008	0.777810	0.040170	0.040172	1.111738	1.111736	1.111740	0.000004	0.000004	0.000004	0.000004	0.000004	0.000004
1	2018-02-01	0.030308	1.000000	1.399999	0.030306	0.030308	0.231101	-0.231101	-0.231101	0.000004	0.000004	0.000004	0.000004	0.000004	-0.000004
2	2018-03-01	0.020394	1.478137	1.788019	0.020392	0.020394	0.068726	0.068726	0.068726	0.000004	0.000004	0.000004	0.000004	0.000004	0.000004
3	2018-04-01	0.010480	1.868130	1.868130	0.010478	0.010480	0.000000	0.000000	0.000000	0.000004	0.000004	0.000004	0.000004	0.000004	0.000004
4	2018-05-01	0.000566	1.354148	1.354148	0.000564	0.000566	0.000000	0.000000	0.000000	0.000004	0.000004	0.000004	0.000004	0.000004	0.000004

Figure 32: Forecasting

Kemudian kami juga menggunakan grafik untuk menganalisa prediksi tingkat inflasi di masa depan.

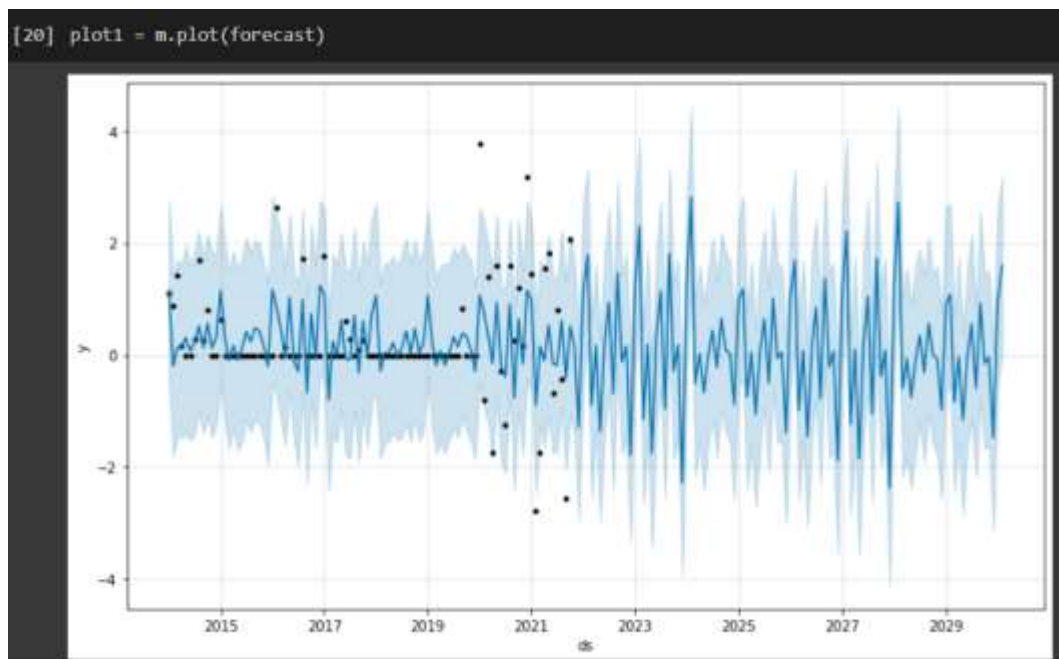


Figure 33: Grafik Forecast 1

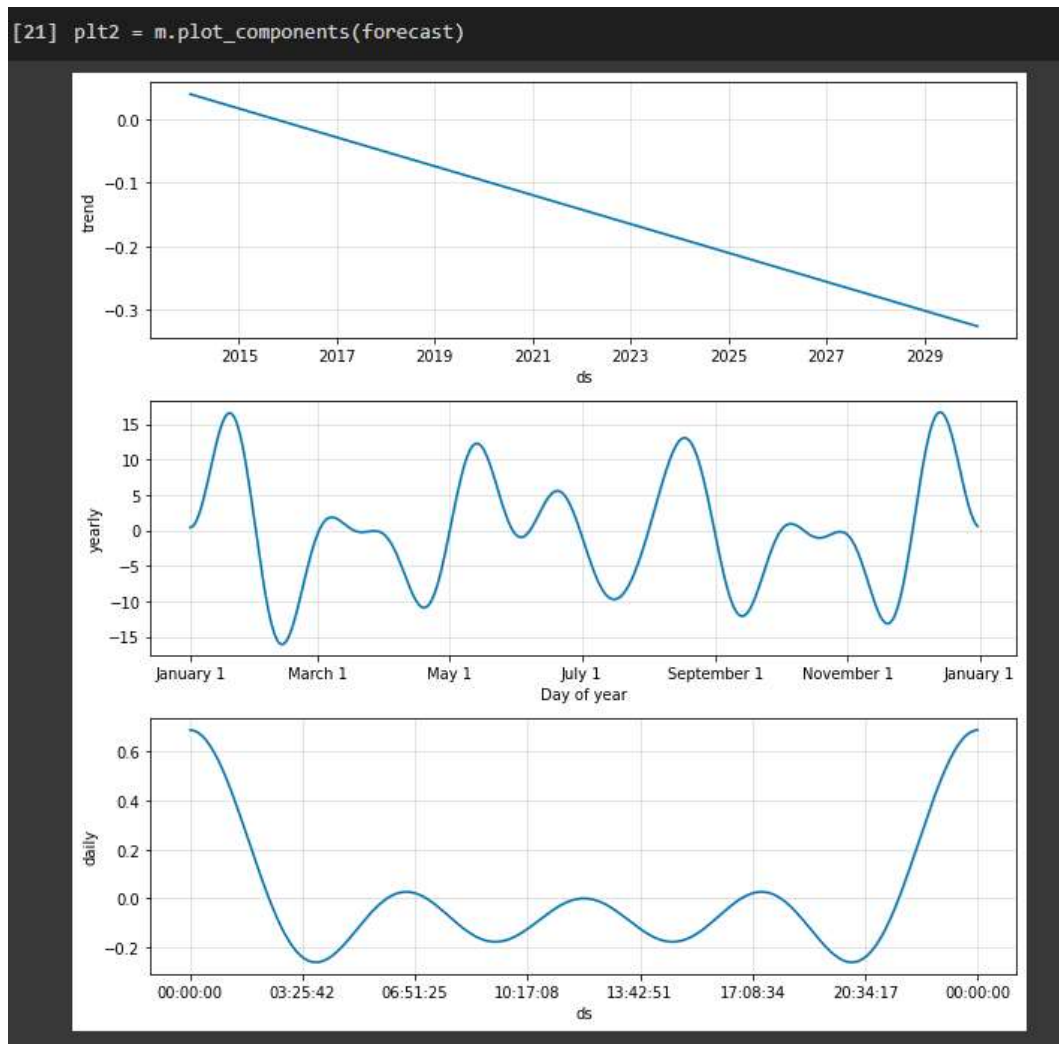


Figure 34: Grafik Forecast 2

Melalui gambar grafik di atas, ini menunjukkan bahwa inflasi akan selalu bersifat fluktuatif. Jika diperhatikan pada grafik pertama, maka peningkatan dan penurunan inflasi terjadi kapan saja dengan tingkat kenaikan yang berbeda – beda. Kemudian pada ketiga grafik terakhir menunjukkan tingkat inflasi yang bermacam – macam secara trend, bulanan, dan harian.

```
[27] mask = (forecast['ds'] > '2021-12-01') & (forecast['ds'] <= '2021-12-31')
forecast_mask = forecast.loc[mask]
```

Figure 35: Masking

Sebelum masuk ke deployment, kita perlu melakukan tes pada dataset kita dengan melakukan masking, karena pada tahap deployment, user akan diberikan kebebasan untuk memilih waktu target.

Deployment

Deployment pada aplikasi web-based kami menggunakan flask.

```
def marks_prediction(value,city, sector, start_date, end_date):
    df_Makanan = pd.read_csv('Dataset/Fix 02 - Makanan.csv')
    df_Minuman = pd.read_csv('Dataset/Fix 02 - Minuman.csv')
    df_Rumah = pd.read_csv('Dataset/Fix 03 - Rumah.csv')
    df_Sandang = pd.read_csv('Dataset/Fix 04 - Sandang.csv')
    df_Transportasi = pd.read_csv('Dataset/Fix 07 - Transportasi.csv')

    sector = sector_chosen(sector)

    sector = sector[[city,'BULAH']]
    sector.columns = ['y', 'ds']

    sector.ds = pd.to_datetime(sector.ds)

    m = Prophet(interval_width=0.95, daily_seasonality=True)
    model = m.fit(sector)

    future = m.make_future_dataframe(periods=100,freq='M')
    forecast = m.predict(future)
    start_date = start_date + dateutil.relativedelta.relativedelta(months=1)
    end_date = end_date + dateutil.relativedelta.relativedelta(months=1)
    mask = (forecast['ds'] > start_date) & (forecast['ds'] <= end_date)

    graph_mask = forecast['ds'] <= end_date

    forecast_mask = forecast.loc[mask]

    forecast_graph_mask = forecast.loc[graph_mask]
    forecast_graph_mask['value'] = forecast_graph_mask['yhat']*value

    if forecast_mask.empty:
        return value

    inflation_average_rate = (forecast_mask['yhat'].sum())/forecast_mask.shape[0]

    value += inflation_average_rate*0.01*value

    forecast_graph_mask['value'] = value + forecast_graph_mask['yhat']*value*0.01

    forecast_graph_mask.value = forecast_graph_mask.value.astype(int)

    chart_from_python_city_1=my_plot(forecast_graph_mask,"value",city)

    value = int(value)

    return value,chart_from_python_city_1
```

Figure 36: Fungsi Prediksi

Gambar di atas merupakan fungsi machine learning yang akan dipakai dalam memprediksi tingkat inflasi dan nilai yang ingin diprediksi oleh pengguna. Fungsi

tersebut pada dasarnya merupakan script yang ada pada Notebook yang sudah ditampilkan sebelumnya.

Ada 3 fitur tambahan selain dari menampilkan nilai prediksi akhir yang dapat membantu user dalam menggunakan aplikasi ini, yang pertama ada fitur di mana user dapat memilih barang apa yang ingin diprediksi. Barang – barang yang bisa diprediksi oleh user ada bermacam – macam, namun saat ini kami hanya mempersiapkan 20 barang yang dapat digunakan oleh user.

```
def get_sector_value(product):

    dataset= pd.read_csv('Dataset/Product Names - Sheet1.csv')
    dataset=dataset.rename(columns = {'Nama Produk':'Nama_Produk'})
    new = dataset.query('Nama_Produk.str.contains(@product)', engine='python')
    if new.empty:
        return 0,0
    else:
        value = new.at[0,'Harga(IDR)']
        sector = new.at[0,'Sektor']
        return value,sector
```

Figure 37: Fungsi Prediksi

Fitur tambahan kedua, adalah fitur untuk membandingkan tingkat inflasi dari kota yang dipilih dengan 2 kota terdekat lainnya. Kami menggunakan Haversine formula untuk menemukan kedua kota terdekat. Kami pula menambahkan satu dataset lagi untuk menyimpan letak titik lintang dan bujur setiap kota yang nanti nya digunakan untuk mengukur kedekatan dari tiap – tiap kota. Berikut

Adalah 3 fungsi yang ditunjukkan untuk mencari 2 kota terdekat dari kota yang dipilih oleh pengguna.

```

def distance(lat1, lon1, lat2, lon2):
    p = 0.017453292519943295
    hav = 0.5 - cos((lat2-lat1)*p)/2 + cos(lat1*p)*cos(lat2*p) * (1-cos((lon2-lon1)*p)) / 2
    return 12742 * asin(sqrt(hav))

def closest(data, v):
    return min(data, key=lambda p: distance(v['lat'],v['lon'],p['lat'],p['lon']))

def city_comparison(city):
    dataset_1 = pd.read_csv('Dataset/longitude, latitude.csv')
    dataset_2 = pd.read_csv('Dataset/longitude, latitude.csv')
    dataset_1=dataset_1.rename(columns = {'Latitude':'lat','Longitude':'lon'})
    dataset_2=dataset_2.rename(columns = {'Latitude':'lat','Longitude':'lon'})
    Kota_Pilihan = city
    dataset_2_1 = dataset_2[dataset_2.Kota != Kota_Pilihan]
    dataset_2_1_1 = dataset_2.set_index("Kota")
    dataset_2_1 = dataset_2_1.to_dict('records')
    dataset_2_1_2 = dataset_2_1_1.loc[[Kota_Pilihan]]

    v = {'lat': dataset_2_1_2['lat'], 'lon': dataset_2_1_2['lon']}
    closest_1 = closest(dataset_2_1, v)

    First_city = closest_1.get('Kota')

    for i in range(len(dataset_2_1)):
        if dataset_2_1[i]['Kota'] == First_city:
            del dataset_2_1[i]
            break

    v = {'lat': dataset_2_1_2['lat'], 'lon': dataset_2_1_2['lon']}
    closest_2 = closest(dataset_2_1, v)

    Second_city = closest_2.get('Kota')

    return First_city, Second_city

```

Figure 38: City Comparison

Fitur tambahan terakhir adalah fitur untuk menampilkan grafik harga dari tahun ke tahun. Berikut adalah fungsi yang digunakan untuk menampilkan grafik inflasi sehingga pengguna dapat memiliki gambaran akan inflasi yang akan ditemui nantinya.

```
def my_plot(data,plot_var,city):
    data_plot = go.Scatter(x=data["ds"],y=data[plot_var], line=dict(color="#CE285E",width=2))
    layout=go.Layout(title=dict(text="The value of your item in "+" "+str(city),x=0.5),
        xaxis_title="Date", yaxis_title="Values"
    )
    fig =go.Figure(data=data_plot, layout=layout)

    fig_json = fig.to_json()
    graphJSON = json.dumps(fig_json, cls=plotly.utils.PlotlyJSONEncoder)
    return graphJSON
```

Figure 39: Fungsi Plot

Selain dari python, kami juga menggunakan HTML, CSS, dan Javascript untuk memberikan tampilan web yang menarik.

Hasil Akhir

Predict the **value** of your item

You can fill in either value or product name

Value (in Rupiah)	<input type="text" value="input your value"/>
Product Name	<input type="text" value="input your product"/>
Sector	<input type="text" value="Choose your sector"/>
City	<input type="text" value="Choose your city"/>
Date	<input type="text" value="mm/dd/yyyy"/>

Figure 40: Halaman Awal

Gambar di atas merupakan tampilan awal ketika pengguna memasuki website kami.

Kemudian pengguna dapat memilih untuk mengisi harga dari barang yang akan pengguna prediksi atau nama produk.

Predict the **value** of your item

You can fill in either value or product name

Value (in Rupiah)	100000000
Product Name	-input your product-
Sector	Rumah
City	DKI Jakarta
Date	04/16/2022

Check!

Figure 41: Opsi Pertama

Gambar di atas merupakan tampilan opsi pertama, yaitu ketika pengguna memasuki harga dari barang pengguna, namun ketika pengguna sudah memasuki harga dari barang pengguna, maka pengguna tidak bisa memasuki sektor dari barang pengguna.

Perlu digaris bawahi: tanggal yang diberikan tidak boleh pada hari yang sama maupun pada hari sebelum hari ini.

Predict the **value** of your item

You can fill in either value or product name

Value (in Rupiah)	-input your value-
Product Name	Rogot Api 200 g
Sector	-Choose your sector-
City	Semarang
Date	03/29/2022

Check!

Figure 42: Opsi Kedua

Gambar di atas merupakan tampilan opsi kedua, yaitu ketika anda memasuki nama produk dari barang anda, namun ketika anda sudah memasuki nama produk, anda tidak dapat mengisi harga maupun sektor dari barang tersebut

Perlu digaris bawahi: tanggal yang diberikan tidak boleh pada hari yang sama maupun pada hari sebelum hari ini



Figure 43: Grafik Kota Pertama

Gambar di atas merupakan tampilan dari hasil prediksi beserta grafik yang memunculkan pergerakan harga dari tahun ke tahun (dari 2014 hingga tanggal prediksi).

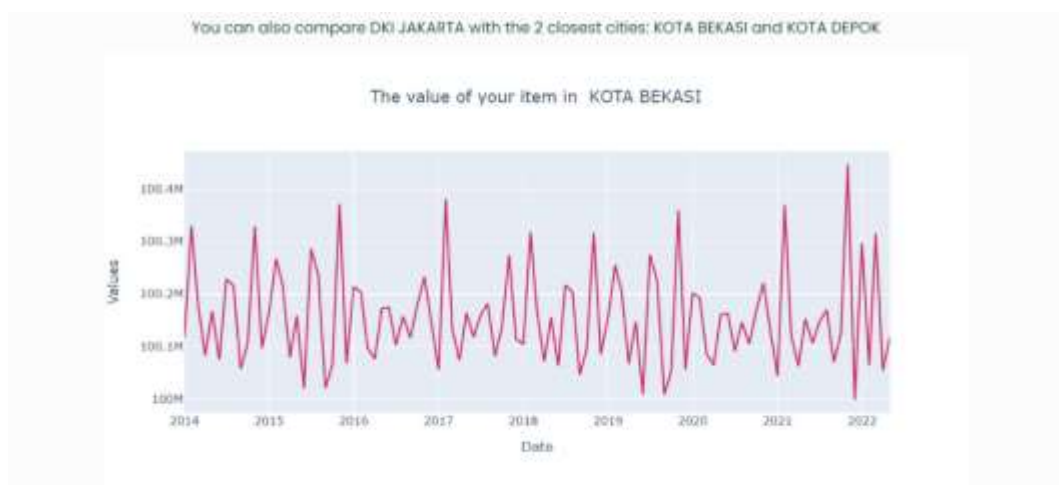




Figure 44: Grafik Kota Kedua dan Ketiga

Kami pula memberikan perbandingan terhadap 2 kota terdekat dari kota yang dipilih oleh pengguna, sehingga dapat menjadi pertimbangan pengguna untuk membuat keputusan.

Bab IV Penutup

IV.1 Kesimpulan

Pelaksanaan studi independen dengan mitra PT Presentologics pada hakikatnya memberikan suatu pembelajaran baru kepada para peserta nya. Dalam hal ini mitra PT Presentologics sudah menyediakan fasilitas – fasilitas dan timeline yang sesuai dengan proses pembelajaran dan perancangan yang sudah dibuat sebelumnya. Hal ini demi mewujudkan manusia yang berwawasan unggul dalam menghadapi kegiatan industri yang dinamis. Pembelajaran selama proses pelaksanaan Program Studi Independen Bersertifikat Pengembang Machine Learning dan Front-End Web dilakukan secara asynchronous dengan peserta menyelesaikan 14 modul secara online dari platform Dicoding academy. Pengenalan materi dan pengumpulan proyek semua dilakukan secara bertahap dan peserta dinyatakan lolos dari setiap modul bisa proyek atau ujian akhir telah memenuhi persyaratan kelulusan. Materi lebih lanjut akan diberikan pada sesi ILT (Instructor Led Training). Para instruktur telah dipilih oleh pihak Dicoding untuk memberikan materi machine learning dan front-end yang terbaik.

Pada bidang Machine Learning dan Front-End Web, pembelajaran dimulai dengan pengenalan tentang logika pemograman, dasar pemograman, dan cara menggunakan github. Kemudian dilanjutkan dengan Belajar Dasar Visualisasi Data, Memulai Pemrograman dengan Python, disertai dengan 3 modul yang membawakan materi tentang Machine Learning dari dasar machine learning hingga machine learning expert yang membawakan materi tentang deployment pada aplikasi yang sudah kita buat. Pada bidang front end, peserta dipersiapkan dengan materi dasar hingga web component pada website, di dalamnya termasuk AJAX dan Javascript.

Peserta juga iberikan kesempatan untuk mengikuti simulasi dari ujian sertifikasi internasional, TensorFlow Developer Certification dari Google dengan dipersiapkan dengan 15 soal untuk dikerjakan oleh peserta. Pada akhir

pembelajaran, peserta mengerjakan sebuah capstone project yang bertujuan untuk memecahkan masalah yang sedang dihadapi dunia. Melalui pembelajaran ini, peserta diharapkan dapat menjadi lebih berkompeten dalam memenuhi standar industri. Diharapkan juga peserta dapat mengembangkan ilmu lebih baik dan dapat memimpin industry 4.0.

IV.2 Saran

Sebagai salah satu bagian penutup laporan akhir program ini, penulis akan memberikan beberapa saran, baik untuk program Studi Independen Bersertifikat Kampus Merdeka dan PT Presentologics sebagai mitra.

Untuk Kampus Merdeka:

1. Pemberian insentif sebaiknya dilakukan tepat waktu dan sesuai jadwal

Untuk PT Presentologics:

1. Sebaiknya pelaksanaan ILT diberitahu 2 minggu sebelumnya, sehingga peserta dapat memberikan informasi tentang kehadiran maupun ketidakhadirannya.
2. Sebaiknya pelaksanaan Mentoring pada Capstone project dilakukan lebih awal sehingga peserta dapat mendapatkan pengarahan yang baik dari pihak panitia.

Referensi

Lampiran A. <TOR>

Informasi Program SIB dan Dicoding

Tentang Kampus Merdeka

Program persiapan karir yang komprehensif guna mempersiapkan generasi terbaik Indonesia. Kampus Merdeka merupakan bagian dari kebijakan Merdeka Belajar oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia yang memberikan kesempatan bagi mahasiswa/i untuk mengasah kemampuan sesuai bakat dan minat dengan terjun langsung ke dunia kerja sebagai persiapan karir masa depan. - *Web Kampus Merdeka*

Tentang Dicoding

Dicoding secara resmi diluncurkan tanggal 5 Januari 2015 untuk menjembatani developer Indonesia dengan kebutuhan dan permintaan pasar yang semakin kompetitif. Dicoding hadir sebagai platform pendidikan teknologi yang membantu menghasilkan talenta digital berstandar global. Semua demi mengakselerasi Indonesia agar menjadi yang terdepan. Saat ini, lebih dari 470 ribu developer dan calon developer telah tergabung di Dicoding. 290 ribu individu pembelajar telah dan sedang terdaftar dalam lebih dari 80 kelas yang disediakan oleh Dicoding

Saat ini, Dicoding bermitra dengan perusahaan teknologi kelas dunia. Dicoding juga merupakan Google Authorized Training Partner dan memiliki komitmen kemitraan dengan pemilik teknologi, perusahaan multinasional, Kementerian/Lembaga Pemerintahan, serta perusahaan dengan skala nasional. Dicoding juga adalah mitra penyelenggara Bangkit, Indosat Ooredoo Digital Camp, Lintasarta Digischool, Baparekraf Digital Talent, dan Cloud and Back-End Developer Scholarship Program with content from AWS pada tahun 2021.

Akun, Credential, dan Sites

Dicoding

Gunakan hanya akun Dicoding Anda sendiri dalam program ini. Pastikan untuk menggunakan akun yang sama sepanjang program. **Pastikan tidak membuat akun baru.**

Google Workspace

Anda telah menerima **credential** @dicoding.org

1. Login melalui komputer untuk mengganti password, dan pastikan untuk login di ponsel untuk memastikan email, calendar, dan jadwal-jadwal tidak terlewatkan
2. Pertemuan tatap muka akan menggunakan Google Meet yang dijadwalkan melalui Google Calendar.
3. Akun Google Workspace hanya digunakan untuk kepentingan SIB dan akan dihapus saat program berakhir.
4. Anda diperbolehkan (tidak wajib) mengaitkan akun dicoding Anda dengan Google Login SIB. Silakan simak infonya pada dokumen cohort guide.

Microsite

<https://sib.dicoding.org>

Microsite disediakan untuk pengumuman dan informasi terbaru terkait program SIB.

Anda harus menggunakan email @dicoding.org untuk membukanya.

Dashboard

<https://dashboard.dicoding.org>

Dashboard disediakan untuk informasi progres peserta, absensi, dan tugas-tugas terkait program SIB. Anda harus menggunakan email @dicoding.org untuk membukanya.

Bapak/Ibu dosen dan pembimbing Anda akan juga dapat mengakses dashboard.

Merdeka Belajar - Kampus Merdeka dan Platform terkait Pelaporan Dikti

<https://kampusmerdeka.kemdikbud.go.id>

Sebagai peserta program Kampus Merdeka, Anda akan mendapatkan akses untuk memasukkan **logbook, aktivitas, mood, dan informasi tentang insentif**. Untuk saat ini, Anda wajib membuat rekening bank di BRI (Provinsi selain Aceh) atau BSI (Aceh) dan memasukkannya ke laman Kampus Merdeka.

Media Komunikasi

1. **Teknis Program, Penilaian, Insentif, Keluhan, dan Saran**
2. **email: mbkm@dicoding.org dan livechat: www.dicoding.com**
3. **Pertanyaan terkait materi Teknis, Kendala Tutorial, Review**
4. **Forum diskusi di masing-masing kelas dan Slido atau diskusi pada sesi tatap-muka**
5. **Pertanyaan terkait soft skill dan komunikasi/ pelaporan dengan kampus asal, Curhat dan diskusi nonteknis**
6. **WA Group per kelompok kecil (+- 25 peserta) dan Sesi Konsultasi dengan Pembimbing non akademik**
 - Pastikan sopan, menyebutkan identitas dan konteks dalam berkomunikasi.
 - Menggunakan etika dalam berdiskusi, berkomunikasi, dan menyampaikan pendapat.
 - Tidak diperkenankan menghubungi tim/mentor/expert Dicoding melalui email atau WA pribadi, kecuali dipersilakan/diminta oleh yang bersangkutan.
 - **Pelanggaran atas aspek komunikasi ini akan mengakibatkan pengurangan nilai sikap.**

Kurikulum dan Tatap Muka

Kurikulum program SIB telah disesuaikan dengan peta okupasi yang diharapkan oleh industri. Lulusan program ini diharapkan dapat masuk ke ekosistem IT dan memperkaya

ekosistem itu sendiri. Setiap paket didesain untuk dikonversi setara 20 (dua puluh) SKS. Namun pada prinsipnya, konversi SKS merupakan **hak prerogatif Jurusan/Program Studi** asal Anda.

Selama proses SIB, Anda akan menjalani:

1. Belajar Mandiri (self-paced)

- a) Didesain 8 (delapan) jam sehari atau 40 (empat puluh) jam seminggu, dengan asumsi waktu efektif 5 hari kerja, di luar hari libur nasional dan Sabtu/Minggu
- b) Anda akan menerima token yang dipergunakan untuk mengakses kelas.
- c) Dilaksanakan melalui platform dicoding.com.
- d) Dipantau setidaknya setiap minggu dalam proses SIB.
- e) Akan mendapatkan sertifikat kompetensi untuk kelas-kelas yang diluluskan kecuali simulasi sertifikasi.
- f) Wajib menyelesaikan dan lulus **seluruh materi belajar mandiri dan tugas/submission/quiz/assignment** untuk dinyatakan menyelesaikan program.

2. Belajar bersama expert / tatap muka (instructor-led)

- a) Didesain 2 (dua) jam setiap dua minggu.
- b) Sesi bersifat wajib dan dalam sesi tatap muka tersebut akan ada tes penilaian.
- c) Anda akan menerima undangan melalui Google Calendar untuk bergabung dalam sesi meet dan **wajib on-cam selama durasi instructor-led**.
- d) Mahasiswa yang tidak hadir tidak perlu memberikan surat/izin, namun penilaian pada sesi tersebut akan dinilai 0.
- e) Wajib hadir setidaknya 70% dari 6 sesi yang diadakan untuk dinyatakan menyelesaikan program.

3. Sesi Softskill & Penyiapan Karir bersama pembimbing (career-development)

- a) Didesain 2 (dua) jam, satu kali setiap bulannya.
- b) Sesi bersifat wajib dan dalam sesi tatap muka tersebut akan ada tes penilaian.
- c) Anda akan menerima undangan melalui Google Calendar untuk bergabung dalam sesi meet dan **wajib on-cam selama durasi career-development**.
- d) Mahasiswa yang tidak hadir tidak perlu memberikan surat/izin, namun penilaian pada sesi tersebut akan dinilai 0.

- e) Terdapat materi bacaan sebelum sesi maupun tugas yang perlu diselesaikan.
 - f) Wajib hadir setidaknya 70% dari 4 sesi yang diadakan untuk dinyatakan menyelesaikan program.
- 4. Sesi Konsultasi bersama pembimbing (konsultasi)**
- a) Didesain 1 (satu) jam, satu kali setiap minggunya.
 - b) Sesi bersifat wajib.
 - c) Anda akan menerima undangan melalui Google Calendar untuk bergabung dalam sesi meet dan **wajib on-cam selama durasi konsultasi**.
 - d) Kehadiran Anda dalam sesi ini akan terkait dengan insentif Anda.
- 5. Proses pengisian logbook, moodbar, dan pelaporan progress (logbook)**
- a) Diisi setidaknya satu minggu sekali pada akhir minggu.
 - b) Akan diperiksa oleh tim Pembimbing Dicoding dan Kemdikbudristek.
 - c) Konsistensi Anda mengisi logbook akan terkait dengan insentif Anda.
 - d) Wajib diisi selama durasi program untuk dinyatakan menyelesaikan program.
- 6. Capstone - Final Project (capstone)**
- a) Didesain untuk proses selama 100 (seratus) jam pada akhir program.
 - b) Wajib berkontribusi aktif menyelesaikan Capstone hingga pengumpulan produk dan hadir pada presentasi akhir untuk dinyatakan menyelesaikan program.

Aturan dan Syarat Penggunaan Dicoding

Mohon membaca kembali <https://www.dicoding.com/termsfuse>.

Pertanyaan yang sering ditanyakan terkait Dicoding

Mohon menuju <https://help.dicoding.com>.

Cohort Guide

Mohon menuju [File Panduan ini](#). (File ini hanya dapat diakses dengan akun dicoding.org)

Periode Program

Program akan berlangsung 23 Agustus 2021 - 19 Januari 2022 (\pm 5 bulan).

Tanggal-tanggal penting adalah sebagai berikut:

Registrasi dan Filtering	: 1 Juni 2021 - 31 Juli 2021
Pengumuman peserta diterima	: 8 Agustus 2021
Persiapan	: 1 Agustus 2021 - 22 Agustus 2021
Konsolidasi dengan Mitra PT	: 14 Agustus 2021
Pemberian akses credential dan kelas	: 23 Agustus 2021
Onboarding oleh Mendikbudristek	: 23 Agustus 2021
Briefing Teknis oleh tim Dicoding	: 23 Agustus 2021
Masa pembelajaran	: 24 Agustus 2021 - 20 Desember 2021
Pelaporan ke dosen pembimbing	: Minggu ketiga di setiap bulan.
Final / Capstone Project	: November 2021 - Desember 2021
Presentasi Final Project	: Januari 2022
Penyelesaian Administrasi dan Pelaporan	: Januari 2022

Nilai-nilai yang Dijunjung Tinggi dalam SIB

Integritas dan Kejujuran

Sebagai peserta SIB dan dalam lingkup pembelajaran, Integritas dan Kejujuran wajib dijunjung tinggi. Termasuk namun tidak terbatas pada setiap proses pembelajaran yang dilakukan. Plagiarisme merupakan hal tidak terpuji dan dilarang (secara khusus disebutkan pada Peraturan Menteri Pendidikan 74/P/2021 halaman 5 (lima) pasal 7 (tujuh). Peserta yang melakukan plagiarisme tidak dapat melanjutkan program SIB.

Niat Baik dan Tidak Berasumsi

Pada lingkungan pembelajaran, penting untuk dapat menyampaikan pengalaman, pemikiran, permasalahan secara terbuka dan tanpa tendensi apapun. Dicoding menganjurkan peserta dapat mencerna setiap komunikasi dengan mempertimbangkan niat baik dan tidak berasumsi. Dengan demikian, setiap interaksi dapat menjadi titik pertumbuhan dan kepercayaan bagi setiap individu. Hal ini termasuk dalam menyampaikan saran, masukan, kritik kepada siapapun, termasuk tim/mentor/expert/pembimbing Dicoding.

Profesional dan Konstruktif

Pada program SIB ini, kami ingin memberikan layanan dan *benefit* sebaik-baiknya dan sebesar-besarnya kepada setiap *stakeholder*. Begitu juga yang kami harapkan dari peserta dan stakeholder program ini. Setiap komunikasi diharapkan untuk Profesional dan Konstruktif demi menunjang pengalaman kerja secara profesional.

Responsif dan Berani Menyampaikan

Pada program ini, setiap stakeholder diharapkan responsif dan dapat menyampaikan hal yang diperlukan. Komunikasi yang baik akan terjalin saat setiap pihak memiliki ekspektasi dan waktu respons yang sama. Kami mendorong setiap stakeholder untuk menyampaikan hal-hal yang dirasa kurang baik atau mengganggu, dengan harapan program ini akan lebih baik kedepannya.

Saling membantu dan bebas gangguan.

Pada program ini, setiap individu memegang peranan yang penting. Kami berusaha untuk membuat setiap individu dapat mencapai potensi terbaiknya. Penting untuk membuat kultur saling membantu yang bebas gangguan, intimidasi, bias, maupun diskriminasi. Apabila Anda mengalami hal-hal ini atau melihat/mendengar/merasa individu lainnya mengalami hal ini dengan bukti yang dapat dipertanggungjawabkan, mohon laporkan ke pengelola program

B. Log Activity

Minggu/Tgl	Kegiatan	Hasil
1/23-08-2021	<p>Mengerjakan Modul:</p> <ul style="list-style-type: none"> Memulai Dasar Pemrograman untuk Menjadi Pengembang Software Pengenalan ke Logika Pemrograman Belajar Dasar Git dengan Github <p>Reporting & Consulting 1</p>	<p>Peserta mampu untuk mengenali dasar pemrograman, logika pemrograman, dan dasar penggunaan git dan github.</p>
2/30-08-2021	<p>Mengerjakan Modul:</p> <ul style="list-style-type: none"> Pengenalan Data pada Pemrograman Belajar Dasar Visualisasi Data Memulai Pemrograman dengan Python <p>Reporting & Consulting 2</p>	<p>Peserta mampu untuk mengenal data dan cara memvisualisasikannya. Namun belum mampu untuk melakukan pemrograman dengan python.</p>
3/06-09-2021	<p>Mengerjakan Modul:</p> <ul style="list-style-type: none"> Memulai Pemrograman dengan Python Belajar Machine Learning untuk Pemula <p>IIT 1: Machine Learning</p> <p>Reporting & Consulting 3</p>	<p>Peserta mampu untuk menggunakan bahasa pemrograman Python dan mampu untuk melakukan Machine Learning dengan level pemula.</p>

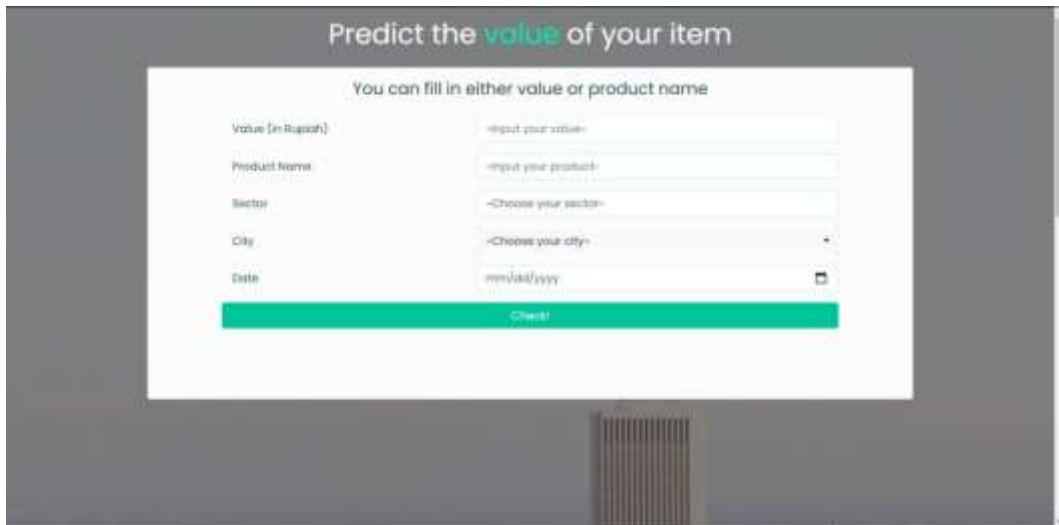
4/13-09-2021	<p>Mengerjakan Modul:</p> <ul style="list-style-type: none"> • Belajar Pengembangan Machine • Belajar Dasar Pemrograman Web (28%) <p>Reporting & Consulting 4</p>	<p>Peserta mampu menggunakan machine learning dengan lumayan baik dan mulai mempelajari dasar web</p>
5/20-09-2021	<p>Mengerjakan Modul:</p> <ul style="list-style-type: none"> • Belajar Pengembangan Machine Learning • Belajar Dasar Pemrograman Web <p>ILT 2: Front End</p> <p>Time Scheduling (Soft Skill)</p> <p>Reporting & Consulting 5</p>	<p>Peserta mampu mengerjakan masalah – masalah machine learning dan dasar – dasar pemrograman web. Peserta juga mampu melakukan time scheduling.</p>
6/27-09-2021	<p>Mengerjakan Modul:</p> <ul style="list-style-type: none"> • Machine Learning Terapan • Belajar Dasar Pemrograman Web <p>Task for Time Scheduling (Soft Skill)</p> <p>Reporting & Consulting 6</p>	<p>Peserta mampu mengerjakan masalah – masalah machine learning secara praktikal.</p>
7/04-10-2021	<p>Mengerjakan Modul:</p> <ul style="list-style-type: none"> • Machine Learning Terapan <p>ILT 3: Machine Learning</p> <p>Reporting & Consulting 7</p>	<p>Peserta mampu mengerjakan masalah – masalah machine learning secara praktikal.</p>
8/11-10-	<p>Mengerjakan Modul:</p>	<p>Peserta mampu</p>

2021	<ul style="list-style-type: none"> • Evaluasi Penguasaan Machine Learning • Belajar Membuat Front-End Web untuk Pemula Reporting & Consulting 8	mengerjakan masalah – masalah machine learning secara praktikal dan teoritis.
9/18-10-2021	Mengerjakan Modul: <ul style="list-style-type: none"> • Belajar Membuat Front-End Web untuk Pemula ILT 4: Front End Reporting & Consulting 9	Peserta mampu mengerjakan pembuatan aplikasi berbasis website dengan responsive.
10/25 – 10 - 2021	Mengerjakan Modul: <ul style="list-style-type: none"> • Tensorflow Developer Certification Prep Reporting & Consulting 10 Critical/Design Thinking (Soft Skill)	Peserta mampu mengerjakan simulasi ujian Tensorflow.
11/01 – 11 - 2021	Mengerjakan Modul: <ul style="list-style-type: none"> • Belajar Fundamental Front-End Web Development ILT 5: Machine Learning Reporting & Consulting 11 Task for Critical/Design Thinking (Soft Skill)	Peserta mampu untuk mengerjakan website dengan menggunakan DOM.
12/08 – 11 -	Mengerjakan Modul:	Peserta mampu

2021	<ul style="list-style-type: none"> Belajar Fundamental Front-End Web Development <p>Reporting & Consulting 12</p> <p>Karir: Meniti Karier sebagai Software Developer</p>	mengerjakan pembuatan website berbasis Javascript AJAX.
13/15 – 11 - 2021	<p>Mengerjakan Modul:</p> <ul style="list-style-type: none"> Belajar Fundamental Front-End Web Development <p>ILT 6: Front End</p> <p>Reporting & Consulting 13</p>	Peserta mampu mengerjakan website menggunakan Javascript Request.
14/22 – 11 - 2021	<p>Peserta mengerjakan Capstone Project</p> <p>Reporting & Consulting 14</p> <p>Effective Communication (Soft Skill)</p>	Peserta sedang mengerjakan capstone proyek akhir.
15/29 – 11 - 2021	<p>Peserta mengerjakan Capstone Project</p> <p>ILT 7: Capstone Mentoring</p> <p>Reporting & Consulting 15</p> <p>Task for Effective Communication (Soft Skill)</p>	Peserta sedang mengerjakan capstone proyek akhir.
16/06 – 12 - 2021	<p>Peserta mengerjakan Capstone Project</p> <p>Reporting & Consulting 16</p>	Peserta sedang mengerjakan capstone proyek akhir.
17/13 – 12 -	<p>Peserta mengerjakan Capstone Project</p>	Peserta sedang

2021	ILT 7: Capstone Mentoring Reporting & Consulting 17	mengerjakan capstone proyek akhir.
18/20 – 12 - 2021	Peserta mengumpulkan Capstone Project Reporting & Consulting 18 Digital Branding & Interview Preparations (Soft Skill)	Peserta menyelesaikan pembuatan capstone proyek akhir.
19/27 – 12 – 2021	Capstone Judging Reporting & Consulting 19 Task for Digital Branding & Interview Preparations (Soft Skill)	-
20/03 – 01 - 2021	Top Presentation Capstone Reporting & Consulting 20	-
21/10 – 01 - 2021	Peserta mendapatkan final transcript	-
22/17 – 01 - 2021	Penutupan MBKM	-

Lampiran C. <Dokumen Teknik>



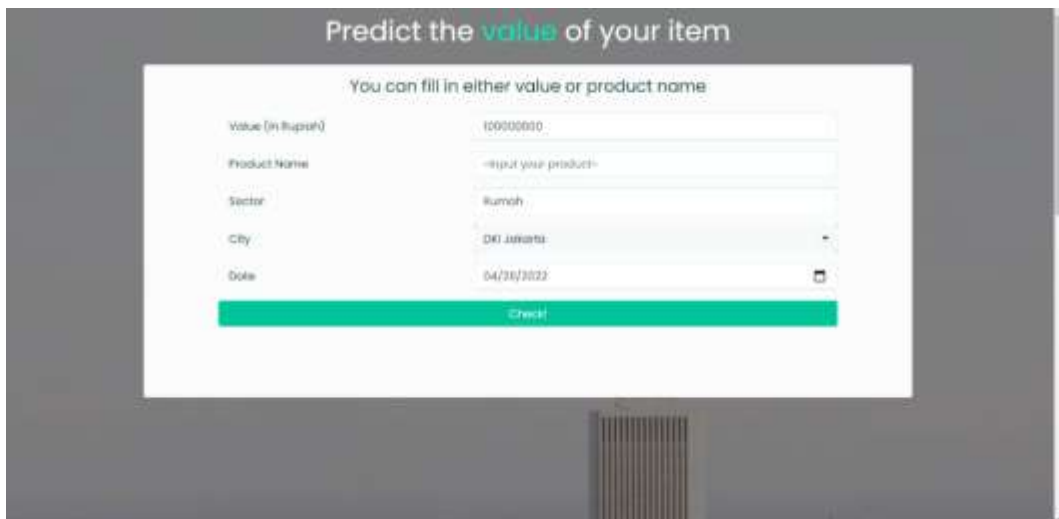
Predict the **value** of your item

You can fill in either value or product name

Value (in Rupiah)	<input your value>
Product Name	<input your product>
Sector	<Choose your sector>
City	<Choose your city>
Date	mm/dd/yyyy

Check!

Figure 40: Halaman Awal



Predict the **value** of your item

You can fill in either value or product name

Value (in Rupiah)	100000000
Product Name	<input your product>
Sector	Rumah
City	DKI Jakarta
Date	04/20/2022

Check!

Figure 45: Opsi Pertama

Predict the **value** of your item

You can fill in either value or product name

Value (in Rupiah)	<input type="text" value="-input your value-"/>
Product Name	<input type="text" value="Kopi Aji 200 g"/>
Sector	<input type="text" value="-Choose your sector-"/>
City	<input type="text" value="Semarang"/>
Date	<input type="text" value="03/25/2022"/>
<input type="button" value="Check"/>	

Figure 46: Opsi Kedua



Figure 47: Grafik Kota Pertama

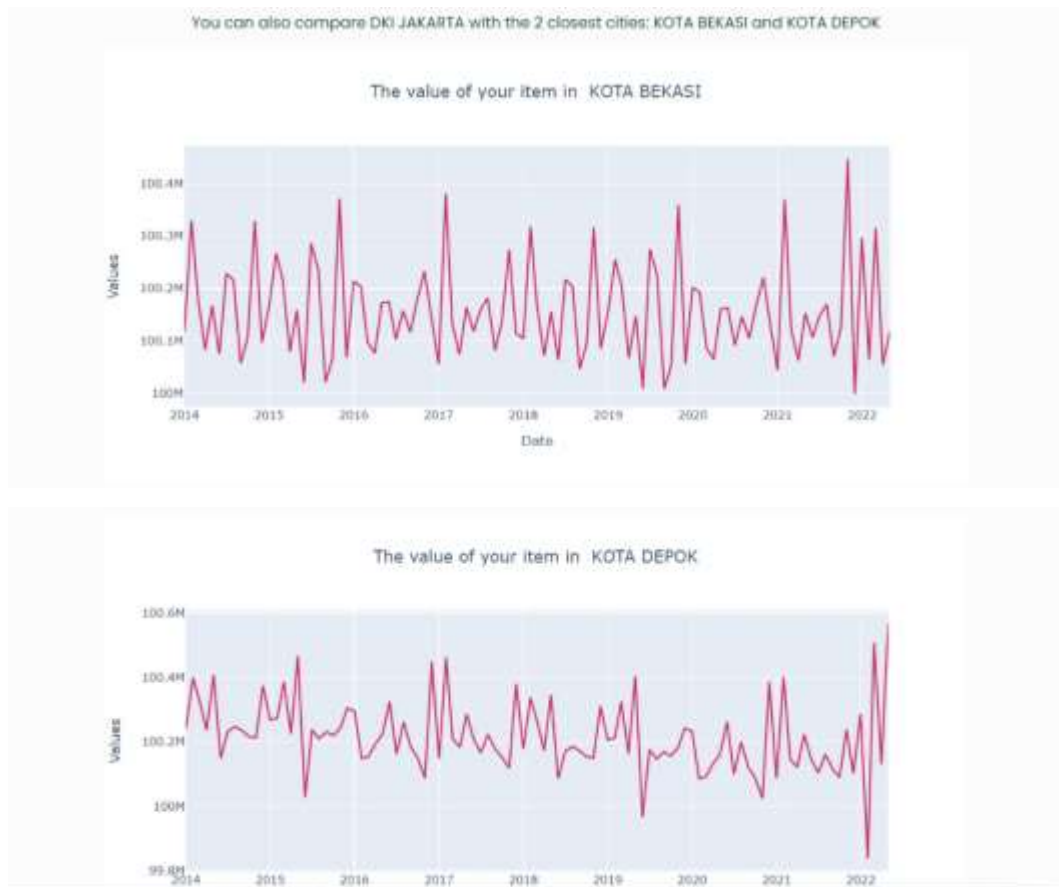


Figure 48: Grafik Kota Kedua dan Ketiga