LAPORAN AKHIR

MAGANG & STUDI INDEPENDEN BERSERTIFIKAT

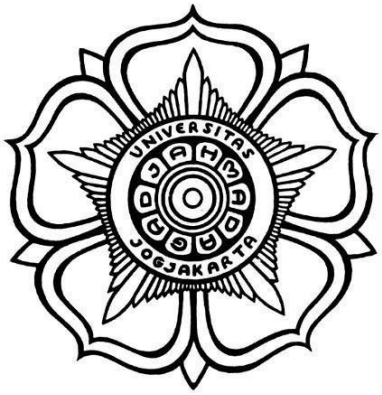
ANDROID LEARNING PATH

di Bangkit Academy 2022 by Google, GoTo, Traveloka

PT Presentologics

Diajukan untuk memenuhi persyaratan kelulusan

Program MSIB MBKM



Oleh:

Daffa Haj Tsaqif

18/427489/PA/18449

PROGRAM STUDI ELEKTRONIKA DAN INSTRUMENTASI

DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS GADJAH MADA 2021

HALAMAN PENGESAHAN

STUDI INDEPENDEN BERSERTIFIKAT MACHINE LEARNING PATH
di Bangkit Academy 2022 by Google, GoTo, Traveloka
PT Presentologics

oleh :
Daffa Haj Tsaqif / 18/427489/PA/18449

disetujui dan disahkan sebagai
Laporan Akhir Semester Studi Independen Bersertifikat Kampus Merdeka

Yogyakarta, 27 Juni 2022
Pembimbing Studi Independen Elektronika dan Instrumentasi
Universitas Gadjah Mada

Danang Lelono, S.Si, M. T.,Dr.
NIP: 196705171998031001

HALAMAN PENGESAHAN

STUDI INDEPENDEN BERSERTIFIKAT MACHINE LEARNING PATH
di Bangkit Academy 2022 by Google, GoTo, Traveloka
PT Presentologics

oleh :
Daffa Haj Tsaqif / 18/427489/PA/18449

disetujui dan disahkan sebagai
Laporan Akhir Semester Studi Independen Bersertifikat Kampus Merdeka

Bandung, 27 Juni 2022
Learning Support Manager
Bangkit Academy 2022

Adrianus Yoza Aprilio
ID. 01032015004

ABSTRAKSI

Program Studi Indepen Bersertifikat di Bangkit Academy 2022 oleh Google, GoTo, Traveloka dengan jalur pembelajaran Machine Learning sedang dilaksanakan dari tanggal 21 Februari hingga sekarang secara daring dengan teknis pembelajaran yang terdiri dari tiga metode sinkron, asinkron, dan belajar mandiri melalui 3 platform yakni: Dicoding, Qwiklabs, dan Coursera. Pada jalur pembelajaran Machine Learning sejauh ini telah mempelajari empat topik wajib Dicoding's Python, IT Automation wtih Python, Mathematics for Machine Learning, TF Developer Professional. Pada akhir program nantinya akan terdapat capstone project, yakni proyek yang akan memecahkan permasalahan pada dunia nyata. Selain itu akan terdapat persiapan untuk melakukan ujian sertifikasi TensorFlow, dan kelas tambahan.

Kata kunci: Studi Independen Bersertifikat, Google, Machine Learning

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah S.W.T., atas berkat, rahmat, dan karunia-Nya sehingga penulis dapat menyelasaikan Laporan Tengah Semester Studi Independen Bersertifikat di Bangkit Academy.

Penulis menyadari bahwa penyusunan laporan tidak dapat selesai tanpa bimbingan, arahan, bantuan, serta dukungan dari berbagai pihak selama menjalani Program Studi Independen Bersertifikat di Bangkit Academy. Untuk itu pada kesempatan yang baik ini, dengan segenap rasa hormat dan kerendahan hati, penulis mengucapkan rasa terima kasih kepada:

1. Allah S.W.T. yang telah melipahkan segala nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan tengah semester ini.
2. Ibu yang selalu mendukung, memberikan semangat dan motivasi bagi penulis.
3. Arya Wijna Astungkara yang dengan rendah hati meminjamkan komputer bagi penulis, yang memungkinkan penulis untuk melakukan Studi Independen ini.
4. Ibu Anny Kartika sari, S.Si., M.Sc., Ph.D. selaku Ketua Departmen Ilmu Komputer dan Elektronika
5. Bapak Yohanes Suyanto, Drs., M.I.Kom., Dr. selaku Ketua Program Studi S1 Elektronika dan Instrumentasi.
6. Danang Lelono, S.Si, M. T.,Dr. selaku dosen pembimbing dari Program Studi Elektronika dan Instrumentasi.
7. Kak Lukas Purba Wisesa selaku facilitator kelompok ML-17 di Bangkit Academy 2022.
8. Semua pihak yang telah membantu penulis dalam pelaksanaan program Studi Independen Bersertifikat.

Penulis menyadari akan ketidaksempurnaan dalam penulisan laporan tengah semester ini. Akhir kata, mohon maaf apabila terdapat banyak kesalahan dan kekurangan dalam penyusunan laporan ini.

Yogyakarta, 31 Maret 2022

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
Pendahuluan

1.1 Latar Belakang

Perkembangan industri teknologi telah berkembang pesat sejak beberapa tahun lalu, hal ini dapat dilihat dengan banyaknya perusahaan *startup* di berbagai negara termasuk Indonesia. Hal ini menyebabkan sertifikasi spesialisasi menjadi penunjang karir yang dapat membantu para mahasiswa dapat mendapatkan pekerjaan yang diinginkan. Bagi perusahaan, sertifikasi yang dimiliki oleh mahasiswa juga menjadi nilai penting bagi perusahaan untuk memiliki bakat bakat yang dapat bermanfaat dalam mengembangkan berbagai teknologi untuk perusahaan, salah satunya seperti keterampilan teknologi bagi perusahaan yang dibangun dan berkembang menggunakan teknologi *cloud* seperti di Google.

Program Studi Independen Bersertifikat (SIB) merupakan salah satu inisiatif dari Kementrian Pendidikan, Kebudayaan, Riset, dan Teknologi(Kemendikbud) untuk mengembangkan mahasiswa Indonesia dalam mengembangkan keterampilan digital yang menjadi salah satu pilar dalam transformasi digital yang sesuai dengan Roadmap Digital Indonesia 2021-2024.

Bangkit Academy adalah program yang diluncurkan oleh Google dengan GoTo dan Traveloka sebagai partner untuk mengembangkan bakat bakat dari Indonesia dengan memberikan media pembelajaran yang berfokus pada permasalahan di dunia nyata bagi 3000 mahasiswa terpilih dari berbagai universitas di seluruh Indonesia di 3 jalur pembelajaran yakni Cloud Computing, Machine Learning, dan Mobile Development untuk membantu mereka mengembangkan keterampilan yang dibutuhkan di bidang teknologi sepanjang semester genap 2022.

Bangkit didesain untuk mempersiapkan peserta dengan kemampuan yang relevan dengan perkembangan teknologi saat ini serta kemampuan untuk bertahan di

dunia kerja. Bangkit didesain untuk mempersiapkan peserta dengan kecakapan (skills) yang relevan dan dibutuhkan berdasarkan sertifikasi teknikal.

Sebagai bagian dari inisiatif Kampus Merdeka Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi, mahasiswa dapat melakukan mengkonversi waktu yang dihabiskan dalam program ini menjadi indeks prestasi(IP) yang setara dengan 16-20 Satuan Kredit Semester (SKS), dengan kelulusan berdasarkan keberhasilan penyelesaian program, dari sertifikasi hingga akhir masa studi, serta sebagai capstone project yang merupakan waktu dimana peserta Bangkit Academy dapat memimplementasikan ilmu yang telah didapat untuk memecahkan permasalahan di dunia nyata. Program SIB Bangkit bertujuan untuk membantu meningkatkan keterampilan digital di era industri 4.0, karena Indonesia masih membutuhkan sembilan juta talenta digital pada tahun 2035, atau sekitar 600.000 talenta digital per tahun.

1.2 Lingkup

Program Magang dan Studi Independen Bersertifikat (MSIB) merupakan bagian implementasi dari program Merdeka Belajar Kampus Merdeka (MBKM) yang dicanangkan oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbud Ristek) RI. Program ini bertujuan memberikan kesempatan pada mahasiswa untuk mengembangkan diri melalui pembelajaran di kelas yang dirancang dan dibuat khusus berdasarkan tantangan nyata yang dihadapi oleh industri sehingga mahasiswa bisa mendapatkan pengalaman terkait dunia profesi dan bisa bekerja secara profesional.

Kegiatan MSIB di Bangkit, lingkup pemebelajaran yang dilaksanakan terdapat tiga jalur pembelajaran, yaitu Machine Learning, Mobile Development (Android), dan Cloud Computing. Masing-masing alur pembelajaran dapat dikonversikan ke dalam SKS (Satuan Kredit Semester) yakni berjumlah 20 sks. Program Studi

Independen pada alur belajar Machine Learning mengikuti beberapa tipe pembelajaran seperti sinkron, asinkron, dan pembelajaran secara mandiri pada berbagai topik yang akan ditempuh selama enam bulan dari Februari – Juli 2022, seperti:

- Dicoding's Python
- IT Automation with Python
- Mathematics for Machine Learning
- TF Developer Professional Certificate
- Structuring Machine Learning Project
- TF Data and Deployment

Pada akhir modul, online assessment akan diberian sebagai bagian untuk menguji pemahaman mahasiswa ata materi program stufi independent yang telah dipelajari. Mahasiswa wajib mengikuti seluruh rangkaian online assessment yang diselenggarakan. Mahasiswa menyesuaikan jadwal pembelajaran yang disediakan oleh Bangkit. Bangkit menyediakan ruang kelas virtual di Goggle Classroom sebagai sarana untuk pelaksanaan program studi independen serta menyediakan platform pembelajaran dari Coursera.

Pada akhir program, mahasiswa diwajibkan untuk menyusun capstone project berupa solusi atas permasalahan atas scenario yang telah ditetapkan serta mahasiswa diberikan fasilitas sertifikasi internasional sebanyak 2 pada level intermediate atas materi yang dipelajari.

1.3 Tujuan

Adapun tujuan dari Bangkit Academy 2022 yaitu:

- Siswa mengerti terkait Critical Thinking, Digital Branding & Interview Communication, Time Management, Professional Communication, Adaptability, Idea Generation dan MVP Planning, serta Startup Valuation.
- Siswa mampu menceritakan kembali dan melaporkan hal yang didapatkan

selama proses pembelajaran dalam bentuk lisan dan tulisan.

- Siswa dapat mendeploy model Machine Learning pada Web.
- Siswa dapat melakukan end-to-end workflow dari Project Machine Learning
- Siswa paham membuat program python dan bagaimana menggunakan python untuk otomasi tugas administrasi secara umum.
- Siswa dapat mengelola kumpulan data/kode mereka sendiri dalam repository Github. Serta dapat berkolaborasi dengan developer lain pada repository yang sama.
- Siswa mampu menyelesaikan proyek akhir, yakni pengembangan aplikasi/solusi yang dikerjakan untuk memvalidasi skill pengembangan produk dan menambah portfolio.
- Siswa dapat berkomunikasi dan memahami materi berbahasa Inggris dengan lancar dan efektif.
- Siswa dapat memahami logika pemrograman dasar dan menerapkannya dalam pemecahan masalah yang ada di bidang pekerjaan Software Developer.
- Siswa mampu memodifikasi aplikasi perangkat lunak menggunakan panduan diagram alur dan pemrograman dengan teknologi HTML, CSS, dan JavaScript tingkat dasar secara tepat sesuai persyaratan spesifikasi dan fungsionalitas aplikasi.
- Siswa dapat matang mempersiapkan diri mengikuti ujian TensorFlow Developer Certificate.
- Siswa dapat memperoleh pengetahuan matematika prasyarat untuk melanjutkan perjalanan dan mengambil kursus yang lebih maju dalam pemelajaran mesin.
- Siswa dapat menyimpulkan dan memilih jalur karier pada bidang Software Development yang sesuai dengan diri mereka beserta mengerti hal-hal yang

harus mereka persiapkan untuk mencapai dan menjalani karier tersebut serta siswa mendapatkan gambaran karir sebagai software engineer/developer atau wawasan terkait startup & bisnis.

- Siswa dapat menerapkan keterampilan TensorFlow ke berbagai masalah dan proyek.

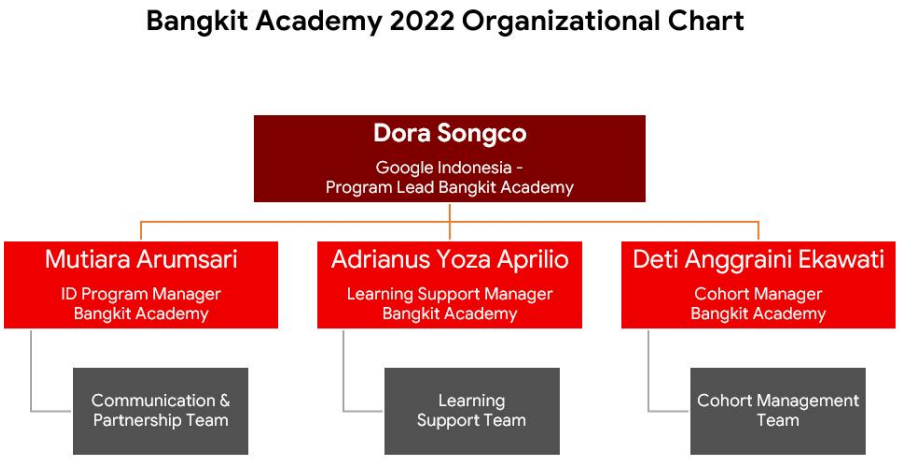
BAB II
Lingkungan Bangkit Academy

2.1 Struktur Organisasi

Bangkit didesain untuk mempersiapkan peserta dengan kecakapan (skills) yang relevan dan dibutuhkan berdasarkan sertifikasi teknikal. Tahun ini Bangkit kembali menyelenggarakan 3 (tiga) alur belajar multidisiplin - Machine Learning, Mobile Development (Android), dan Cloud Computing. Dengan mengikuti Bangkit, peserta akan memiliki pengalaman dan terekspos dengan serba-serbi karir di industri dan pekerjaan di ekosistem teknologi Indonesia.

Bangkit merupakan program pembelajaran yang dipimpin oleh Google dengan dukungan GoTo, Traveloka, dan DeepTech Foundation. Dengan dukungan Kampus Merdeka, Bangkit akan menawarkan 3.000 tempat untuk mahasiswa Indonesia untuk memastikan mereka relevan dengan kecakapan yang dibutuhkan oleh industri pada semester genap, tahun 2021/2022.

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 Bangkit Academy.



Gambar 2.1 Diagram organisasi bangkit academy

2.2 Lingkup Pembelajaran

Kegiatan MSIB dilaksanakan di Bangkit dengan Learning Path Machine learning. Pokok materi yang dipelajari terdapat enam topik pembelajaran yaitu:

1. Dicoding Python

Python adalah bahasa pemrograman umum yang bersifat interperatif yang merupakan salah satu bahasa pemrograman paling populer, bahasa pemrograman ini berfokus untuk mudah dipahami dan mudah untuk dibaca, dengan banyaknya library yang ada membuat bahasa ini menjadi bahasa pemrograman pilihan automasi pada komputer, dan juga bidang data science, machine learning, dan lainnya.

Materi pembelajarannya terdiri dari:

-Dasar Python

-Tipe Data pada Python

-Input/Output dan Operasi pada Python

-Style Guide pada Python

-Control Flow

-Penanganan Kesalahan

-Fungsi dan Method

-Pemrograman Berorientasi Objek

-Unit Testing

-Library Populer pada Python

-Final Exam

2. IT Automation with Python

Kursus ini dirancang dan diajarkan oleh Google untuk membekali profesional IT dengan skill mengenai Python, Git, dan juga automasi IT, dan juga mengajarkan keterampilan yang bersifat non-teknikal yang dapat berguna untuk memecahkan permasalahan yang dapat terjadi di lapangan pekerjaan. Kursus ini dirancang untuk memungkinkan mahasiswa menguasai bahasa pemrograman Python dan juga aplikasinya yang umum digunakan di lapangan pekerjaan, mahasiswa juga nanti akan belajar Git dan juga Github yang nantinya berguna dalam menyimpan pekerjaan dan juga untuk bekerja di dalam sebuah tim, dan nantinya mahasiswa juga akan belajar dalam memecahkan permasalahan yang dapat terjadi di dunia nyata.

Adapun tingkat kursus yang dipelajari yaitu:

-Kursus Singkat di Python

-Menggunakan Python untuk Berinteraksi dengan Sistem Operasi

-Pengantar Git dan GitHub

-Teknik Pemecahan Masalah dan Debugging

-Manajemen Konfigurasi dan Cloud

-Mengotomatiskan Tugas Dunia Nyata dengan Python

3. Mathematics of Machine Learning

Kursus ini dirancang untuk membekali mahasiswa dengan pemahaman matematika yang menjadi pondasi dari topik machine learning yang nantinya akan dipelajari kedepannya, hal ini dilakukan supaya mahasiswa tidak hanya memahami cara merancang model machine learning tetapi juga konsep dasarnya, serta dilakukan agar mahasiswa kedepannya juga dapat melakukan perhitungan matematis menggunakan python kedepannya. Kursus ini menggunakan Python dan juga library Numpy untuk memungkinkan mahasiswa melakukan perhitungan matematika menggunakan bahasa pemrograman.

Adapun tingkat kursus yang dipelajari yaitu:

-Matematika untuk Machine Learning: Aljabar Linier

-Matematika untuk Machine Learning: Kalkulus Multivariate

-Matematika untuk Machine Learning: PCA

4. Tensorflow Developer Professional Certificate

Tensorflow merupakan salah satu framework yang paling populer untuk pengembangan deep learning, dengan sifat open-source nya menjadikan framework dengan perkembangan yang pesat dalam 5 tahun terakhir. Program Sertifikat Profesional Pengembang TensorFlow dari DeepLearning.AI dirancang untuk mahasiswa mempelajari dasar dasar dari Tensorflow dan juga aplikasinya yang dapat digunakan untuk memecahkan permasalahan di dunia nyata. Mahasiswa nantinya akan diberikan pembelajaran mengenai framework Tensorflow itu sendiri, merancang model deep learning yang sesuai dengan permasalahan yang diberikan, melakukan pre-processing dataseet, dan juga langkah langkah dalam memecahkan masalah yang nantinya akan dihadapi. Program ini juga dirancang untuk mempersiapkan mahasiswa untuk mengerjakan ujian yang nantinya akan digunakan untuk mendapatkan Tensorflow Developer Certificate.

Kursus ini terdiri dari:

-Pengantar TensorFlow untuk Kecerdasan Buatan, Pembelajaran Mesin, dan Pembelajaran Mendalam

-Jaringan Saraf Konvolusional di TensorFlow

-Pemrosesan Bahasa Alami di TensorFlow

-Urutan, Deret Waktu, dan Prediksi

5. Structuring Machine Learning Projects

Kursus ini dirancang oleh Andrew Ng yang merupakan salah satu pendiri dari Coursera dna juga Deeplearning.ai untuk membekali mahasiswa dengan studi kasus yang nantinya akan dihadapi ketika berada di lapangan pekerjaan. Kursus ini nantinya digunakan agar mahasiswa dapat mengantisipasi apabila terdapat permasalahan yang nantinya dapat terjadi seperti diagnosa error, strategi yang digunakan, pengaturran dataset dan sebagainya.

6. Tensorflow: Data and Deployment

Kursus ini merupakan kursus spesialisasi yang dirancang untuk menyebarkan model yang sebelumnya telah dirancang tersebut ke beberapa platform, yaitu web, mikrokontroler, aplikasi mobile, dan juga server. Kursus ini juga akan mempelajari proses data dan juga akan melatih kembali model yang sebelumnya telah dilatih, nantinya akan mempelajari komponen lain dari Tensorflow seperti TensorFlow Serving, TensorFlow Hub, TensorBoard, dan lainnya.

Tingkatan kursus yang dipelajari diantaranya:

-Model Berbasis Browser dengan TensorFlow.js

-Model Berbasis Perangkat dengan TensorFlow Lite

-Pipeline Data dengan Layanan Data TensorFlow

-Skenario Penerapan Lanjutan dengan TensorFlow

2.3 Definisi Pembelajaran

Matrikulasi pembelajaran selama setengah semester di program MSIB di Bangkit Academy ditunjukan Tabel 2.1.

Learning Objective	**Tingkat Kompetensi**	**Detil Pembelajaran**	**Durasi Pembelajaran**
Memulai Dasar Pemrograman untuk Menjadi Pengembang Software	Di akhir kelas, siswa mampu memodifikasi aplikasi perangkat lunak menggunakan panduan diagram alur dan pemrograman dengan teknologi HTML, CSS, dan JavaScript tingkat dasar secara tepat sesuai persyaratan spesifikasi dan fungsionalitas aplikasi.	1. Siswa mampu meneliti, menganalisis, dan mengevaluasi persyaratan untuk aplikasi perangkat lunak dengan memahami kebutuhan aplikasi dari sisi pengguna dan spesifikasi teknis aplikasi. 2. Siswa mampu membuat perencanaan modifikasi aplikasi perangkat lunak dengan pembuatan requirement aplikasi dan diagram alur. 3. Siswa mampu memodifikasi aplikasi perangkat lunak menggunakan pemrograman HTML, CSS, dan JavaScript tingkat dasar. 4. Siswa mampu mengarahkan dokumentasi pemrograman dan pengembangan perangkat lunak menggunakan metode pengarsipan.	13 Jam (1 hari)

Pengenalan Ke Logika Pemrograman	Di akhir kelas, siswa dapat memahami logika pemrograman dasar dan menerapkannya dalam pemecahan masalah yang ada di bidang pekerjaan Software Developer.	1. Mengerti apa itu logika pemrograman. 2. Mengetahui apa itu gerbang logika beserta jenis-jenisnya. 3. Memahami cara pemecahan masalah dengan computational thinking.	6 Jam (1 hari)
Belajar Dasar Git dengan Github	Di akhir kelas, siswa dapat mengelola kumpulan data/kode mereka sendiri dalam repository Github. Serta dapat berkolaborasi dengan developer lain pada repository yang sama.	1. Memahami git sebagai version control system. 2. Memahami Github sebagai tools untuk mengelola kumpulan data/kode. 3. Memahami cara mengelola kumpulan data/kode, mulai dari membuat repository, melakukan perubahan, membuat branch lain, hingga melakukan pull request. 4. Memahami cara berkolaborasi dengan developer lain pada repository yang sama. 5. Memahami penggunaan GitHub sebagai portfolio.	15 Jam (1 hari)
Subtotal Matrikulasi			34 Jam, 3 Hari
Google IT Automation with Python	Di akhir kelas, siswa paham membuat program python dan	1. Memanfaatkan praktik terbaik untuk memilih perangkat keras,	132 Jam (15 hari)

	bagaimana menggunakan python untuk otomasi tugas administrasi secara umum.	vendor, dan layanan untuk organisasi Anda. 2. Memahami bagaimana layanan infrastruktur yang paling umum yang menjaga sebuah organisasi menjalankan pekerjaan, dan cara mengelola server infrastruktur. 3. Memahami cara memaksimalkan cloud untuk organisasi Anda. 4. Mengelola komputer dan pengguna dalam menggunakan layanan direktori, Aktif Direktori, dan OpenLDAP. 5. Memilih dan mengelola alat yang akan digunakan organisasi Anda. 6. Membackup data organisasi Anda dan mengetahui cara memulihkan infrastruktur TI Anda jika terjadi kendala. 7. Memanfaatkan pengetahuan administrasi sistem untuk merencanakan dan meningkatkan proses untuk IT environments.	
Mathematics for Machine Learning	Di akhir kelas, siswa dapat memperoleh pengetahuan matematika	1. Memahami vektor dan matriks yang akan membantu Anda menjembatani	55 Jam (6 hari)

	prasyarat untuk melanjutkan perjalanan dan mengambil kursus yang lebih maju dalam pemelajaran mesin.	kesenjangan ke dalam masalah aljabar linier, dan cara menerapkan konsep ini ke pemelajaran mesin. 2. Dapat mengoptimalkan fungsi pemasangan agar sesuai dengan data 3. Memahami konsep matematika penting dan Anda dapat mengimplementasikan PCA sendiri	
DeepLearning.AITensorFlowDeveloperProfessionalCertificate	Di akhir kelas, siswa dapat menerapkan keterampilan TensorFlow ke berbagai masalah dan proyek.	1. Membuat dan melatih Neural Network menggunakan Tensorflow 2. Meningkatkan performa network menggunakan Convolution dengan melatih dan identifikasi gambar nyata. 3. Melatih machine untuk memahami, menganalisa, dan merespon human speech dengan sistem NLP. 4. Memroses teks, menggambarkan kalimat sebagai vector, dan melatih model untuk menciptakan puisi original	83 Jam (9 hari)
Structuring Machine Learning Projects	Di akhir kelas, siswa dapat melakukan end-to-	1. Memahami cara mendiagnosis kesalahan dalam	6 Jam (1 hari)

	end workflow dari Project Machine Learning	sistem pemelajaran mesin, dan 2. Mampu memprioritaskan arah yang paling menjanjikan untuk mengurangi kesalahan 3. Memahami pengaturan ML yang kompleks, seperti set pelatihan/pengujian yang tidak cocok, dan membandingkan dengan dan/atau melampaui kinerja tingkat manusia 4. Mengetahui bagaimana menerapkan pembelajaran end-to-end, pembelajaran transfer, dan pembelajaran multi-tugas.	
DeepLearning.AI Tensorflow Data and Deployment	Di akhir kelas, siswa dapat mendeploy model Machine Learning pada Web	1. Memahami bagaimana melatih dan menjalankan model machine learning di Web browser dan aplikasi mobile 2. Mempelajari bagaimana memanfaatkan built-in datasets dengan baris code yang sedikit. 3. Mempelajari tentang data pipeline dengan servis data Tensorflow	53 Jam (6 hari)

		4. Menggunakan API untuk mengontrol data splitting, memproses semua tipe data yang tidak terstruktur 5. Melatih kembali model yang sudah di deploy dengan data user dan tetap menjaga privasi data. 6. Menerapkan ilmu di berbagai skenario 7. Pengenalan pada TensorFlow Serving, TensorFlow, Hub, TensorBoard, dan banyak lagi.	
Simulasi Ujian TensorFlow Developer Certificate	Di akhir kelas, siswa dapat matang mempersiapkan diri mengikuti ujian TensorFlow Developer Certificate.	1. Memahami bagaimana cara membuat program perangkat lunak menggunakan TensorFlow dan menemukan informasi yang dibutuhkan untuk bekerja sebagai praktisi Machine Learning. 2. Memahami prinsip dasar Machine Learning dan Deep Learning menggunakan TensorFlow 2.x sehingga mampu membuat dan melatih model Jaringan Syaraf Tiruan menggunakan TensorFlow. 3. Memahami cara	25 Jam (3 hari)

		membuat model pengenalan gambar dan deteksi objek dengan Deep Neural Networks dan Convolutional Neural Networks menggunakan TensorFlow 2.x. 4. Memahami cara menggunakan Jaringan Syaraf Tiruan untuk menyelesaikan masalah pemrosesan Natural Language menggunakan TensorFlow. 5. Memahami cara menyelesaikan soal Time Series dan masalah perkiraan dengan menggunakan TensorFlow.	
Subtotal Pembelajaran Machine Learning Learning Path	354 jam (40 hari)	5.	
Capstone Project / Proyek Akhir	Siswa mampu menyelesaikan proyek akhir, yakni pengembangan aplikasi/solusi yang dikerjakan untuk memvalidasi skill pengembangan produk dan menambah portfolio.	Siswa akan dikelompokkan dalam kelompok grup untuk mengerjakan proyek tematik pada dunia nyata yang dapat membantu masyarakat.	200 Jam (20 hari)

Tabel 2.1 Matrikulasi pembelajaran

2.4 Jadwal MSIB

Jadwal kegiatan selama mengikuti program MSIB di Bangkit Academy ditunjukan Tabel 2.2.

Tabel 2.2 Jadwal MSIB

Bulan	Minggu ke-	Waktu Sesi		Durasi Pembelajaran	Learning Objecive	Topik
		Sinkron	Asinkron			
Februari s.d Maret	1		Selasa, 8 Februari 2022	Asinkron: 15 Jam	Belajar Dasar Git dengan GitHub	1. Git dan GitHub 2. Dasar Git 3. Studi Kasus Pengalaman Belajar 4. Git Branches 5. Kolaborasi dengan Tim 6. Studi Kasus Kolaborasi dengan Tim 7. GitHub sebagai Portofolio
			Kamis, 10 Februari 2022	Asinkron: 13 Jam	Memulai Dasar Pemrograman untuk Menjadi Pengembang Software	1. Memahami Kebutuhan Aplikasi 2. Perencanaan Modifikasi Aplikasi 3. Mengerti Konsep Dasar Pemrograman 4. Modifikasi Aplikasi Perangkat Lunak 5. Dokumentasi Pemrograman dan Pengembangan Aplikasi Perangkat Lunak
	2		Rabu, 16 Februari 2022	Asinkron: 20 jam	Memulai Pemrograman Dengan Python	1. Pendahuluan 2. Dasar Python 3. Tipe Data Pada Python 4. Input/Output dan Operasi pada Python

							5. Style Guide pada Python 6. Control Flow 7. Penanganan Kesalahan 8. Fungsi dan Method 9. Pemrograman Berorientasi Objek 10. Unit Testing 11. Library Populer
Bulan	**Minggu ke-**	**Waktu Sesi**		**Durasi Pembelajaran**		**Learning Objective**	**Topik**
		Sinkron	**Asinkron**	**Sinkron**	**Asinkron**		
	2		Senin-Selasa, 21-22 Februari 2022		28 Jam	Google IT Automation with Python: Crash Course with Python	1. Hello Python! 2. Basic Python Syntax 3. Loops 4. Strings, Lists and Dictionaries 5. Object Oriented Programming 6. Final Project
			Selasa-Rabu 22-23 Februari 2022		27 Jam	Google IT Automation with Python: Using Python to Interact with the Operating System	1. Getting Your Python On 2. Managing Files with Python 3. Regular Expressions 4. Managing Data and Processes 5. Testing in Python 6. Bash Scripting 7. Final Project
		Kamis, 24 Februari 2022 pk.15.30-17.30		2 Jam		ILT-ML-01-A Python IT Automation	1. Python 2. Regex 3. Bash Scripting
			Kamis-Jumat 24-25 Februari 2022		16 Jam	Google IT Automation with Python: Introduction to Git and GitHub	1. Introduction to Version Control 2. Using Git Locally 3. Working with Remotes 4. Collaboration
	3		Selasa-Kamis 29 Februari -		16 Jam	Google IT Automation	1. Troubleshooting Concepts

			3 Maret 2022			with Python: Troubleshooting and Debugging Techniques	2. Slowness 3. Crashing Programs 4. Managing Resources
Maret s.d April		Jumat 4 Maret 2022 15.30-17.30		2 jam		ILT-SS-01-AQ Time Management	1. Belajar waktu 2. Skala prioritas 3. 4 Kuadran teknik
	4		Senin-Selasa 7-8 Maret 2022		15 Jam	Google IT Automation with Python: Configuration Management and the Cloud	1. Automating with Configuration Management 2. Deploying Puppet 3. Automation in the Cloud 4. Managing Cloud Instances at Scale
Bulan	**Minggu ke-**	**Waktu Sesi**		**Durasi Pembelajaran**		**Learning Objective**	**Topik**
		Sinkron	**Asinkron**	**Sinkron**	**Asinkron**		
Maret s.d April	4	Rabu, 9 Maret 2022 pk.15.30-17.30		2 Jam		ILT-ML-02-S Python IT Automation	1. Git Collaboration 2. Troubleshooting 3. Intro to Cloud
		Jumat, 11 Maret 2022 pk. 13.00 - 14.30		1,5 Jam		English Session EN1-130 Spoken Correspondence	1. Using right word for replying question 2. Using right sentence when refusing 3. Using right word when asking
			Rabu-Kamis 9-10 Maret 2022		13 Jam	Google IT Automation with Python: Automating Real-World Tasks with Python	4. Manipulating Images 5. Interacting with Web Services 6. Automatic Output Generation 7. Putting It All Together
	5	Rabu, 16 Maret 2022 pk.15.30 - 17.00		2 Jam		ILT-SS-02-V Professional Branding & Interview	1. Creating Resume 2. Creating CV 3. Know our value

			Rabu-Jumat, 16-18 Maret 2022		19 Jam	Mathematics for Machine Learning: Linear Algebra	1. Introduction to Linear Algebra and to Mathematics for Machine Learning 2. Vectors are objects that move around space 3. Matrices in Linear Algebra: Objects that operate on Vectors 4. Matrices make linear mappings 5. Eigenvalues and Eigenvectors: Application to Data Problems
Bulan	**Minggu ke-**	**Waktu Sesi**		**Durasi Pembelajaran**		**Learning Objective**	**Topik**
		Sinkron	**Asinkron**	**Sinkron**	**Asinkron**		
Maret s.d April	6		Sabtu-Selasa 19-22 Maret 2022		18 Jam	Mathematics for Machine Learning: Multivariate Calculus	1. What is calculus? 2. Multivariate calculus 3. Multivariate chain rule and its applications 4. Taylor series and linearization 5. Intro to optimization 6. Regression
		Senin, 21 Maret 2022 pk.09.00-11.00		2 Jam		ILT-ML-03-B Mathematics for Machine Learning	1. Linear Algebra 2. Calculus for ML 3. PCA
			Rabu-Jumat 23-25 Maret 2022		18 Jam	Mathematics for Machine	1. Statistics of Datasets

						Learning: PCA	2. Inner Products 3. Orthogonal Projections 4. Principal Component Analysis
	7		Senin-Selasa 28-29 Maret 2022		18 Jam	Introduction to TensorFlow for Artificial Intelligence, Machine Learning, and Deep Learning	1. A New Programming Paradigm 2. Introduction to Computer Vision 3. Enhancing Vision with Convolutional Neural Networks 4. Using Real-world Images
			Selasa-Rabu 29-30 Maret 2022		18 Jam	Convolutional Neural Networks in TensorFlow	1. Exploring a Larger Dataset 2. Augmentation: A technique to avoid overfitting 3. Transfer Learning 4. Multiclass Classifications
		Rabu 30 Maret 2022 pk.15.30-17.00		2 Jam		ILT-SS-03-AH Critical Thinking	1. Teknik 5 why's 2. MECE 3. Find root of problem
			31 Maret - 3 April 2022		25 Jam	Natural Language Processing in TensorFlow	1. Sentiment in Text 2. Word Embeddings 3. Sequence Models 4. Sequence Models and Literature
		Senin 4 April 2022 pk. 09.00-11.00		2 Jam		English Session EN2-004 Expressing Opinions	1. How to deliver opinion in various situations 2. How handle disagreements properlu 3. Delivering Feedback
			Rabu-Kamis 6-7 April 2022		24 Jam	Sequences, Time Series and	1. Sequences and Prediction 2. Deep Neural Networks for

						Prediction	Time Series 3. Recurrent Neural Networks for Time Series 4. Real-world time series data
		Jumat, 8 April 2022 pk. 15.30-17.00		2 Jam		ILT-ML-04-V Tensorflow in Practice	1. Computer vision 2. CNN 3. Transfer learning
	9	Kamis, 14 April 2022 pk. 15.30-17.00		2 Jam		ILT-SS-04-AJ Adaptability	1. Anticipating Changes in the Workplace 2. Ways of thinking 3. Fixed mindset and growth mindset
			Kamis-Sabtu 14-16 April 2022		10 Jam	Structuring Machine Learning Projects	1. Train/Dev/Test Distributions 2. Understanding Human-level Performance 3. Surpassing Human-level Performance 4. Improving your Model Performance. 5. Error analysis 6. multi-task, transfer, dan end-to-end deep learning.
	10		Senin-Rabu, 18-20 April 2022		22 Jam	Browser-based Models with TensorFlow.js	1. Introduction to TensorFlow.js 2. Image Classification In the Browser 3. Converting Models to JSON Format 4. Transfer Learning with Pre-Trained Models
		Kamis, 21 April 2022 pk. 15.30-17.00		2 Jam		ILT-ML-05-Q Tensorflow Data & Deployment	1. Deploy on the mobile 2. Deploy on the cloud 3. Federate learning
			Kamis-		15 Jam	Device-based	1. Device-based

			Jumat, 21-22 April 2022			Models with TensorFlow Lite	models with TensorFlow Lite 2. Running a TF model in an Android App 3. Building the TensorFLow model on IOS 4. TensorFlow Lite on devices
	11		Senin-Selasa, 25-26 April 2022		16 Jam	Data Pipelines with TensorFlow Data Services	1. Data Pipelines with TensorFlow Data Services 2. Splits and Slices API for Datasets in TF 3. Exporting Your Data into the Training Pipeline 4. Performance
			Rabu, 27 April 2022 pk. 13.00-14.00		1 Jam	Traveloka Capstone Onboarding	1. Welcoming & Onboarding for Traveloka Company Capstone Groups
			Rabu-Kamis. 27-28 April 2022		18 Jam	Advanced Deployment Scenarios with TensorFlow	1. TensorFlow Extended 2. Sharing pre-trained models with TensorFlow Hub 3. Tensorboard: tools for model training 4. Federated Learning
Mei s.d. Juni	12	Hari Raya Idul Fitri					
	13	Rabu, 11 Mei 2022 pk. 15.30-		2 Jam		ILT-SS-05-AX Idea Generation &	1. Entrepreneurship dan Intrapreneurship 2. Minimum

		17.00				MVP Planning	Viable Product 3. Create and Scale MVP 4. Indentifiying Vision Statement
		Kamis, 12 Mei 2022 pk. 13.00-15.00		2 Jam		Bangkit 2022 - Team Meeting 3	1. Google Certification 2. 3rd Softskill Challenge 3. Bangkit 2022 Career Fair 4. 3rd & 4th Milestone recap
	14	Selasa, 17 Mei 2022 pk. 09.30-11.00		1,5 Jam		English Session EN3-008 Business Presentation	1. Basic Structure of a Business Presentation 2. Tips for presentation 3. Explaining Visual Information 4. Dealing With Difficult Question
Juni s.d. Juli	17	Selasa, 7 Juni 2022 pk. 13.00-15.00		2 Jam		Bangkit 2022 - Team Meeting 4	1. Monthly Milestone Recap 2. Capstone Update 3. Softskills Challenge 4. QA
	18	Minggu, 12 Juni 2022 pk. 15.30-17.30		2 Jam		Capstone Team Traveloka - Check Point Presentation - Session 2	1. Presentasi perkembagan capstone 2. Evaluasi dan masukan untuk final productnya.
		Rabu, 15 Juni 2022 pk. 10.00-12.00		2 Jam		Mentoring Session C22CB-CB03	1. Masukan untuk final product 2. Menambahkan fitur tunanetra
		Jumat, 17 Juni 2022 pk. 13.00-17.00		4 Jam		Rekam Video & Penyelesaian Final Deliverables	1. Rekam video presentasi 2. Menyelesaikan ppt

			Sabtu, 18 Juni 2020		5 Jam	Simulasi Ujian TensorFlow Developer Certificate A	1. Membuat model dengan berbaagai contoh yang sederhana untuk simulasi ujian TF
			Minggu, 19 Juni 2022		5 Jam	Simulasi Ujian TensorFlow Developer Certificate B	1. Membuat model dengan berbaagai contoh yang sederhana untuk simulasi ujian TF
	19	Senin, 20 Juni 2022 pk. 09.00-11.00		2 Jam		ILT-SS-06-D Startup Valuation & Investment Pitch	1. valuasi startup 2. cara mencari capital untuk start up 3. aspek yang dapat diperhitungkan kalau ingin invest ke startup
			Senin, 20 Juni 2022		1 Jam	Bangkit 2022 - English Post-Class Assessmen	1. Post test mengenai materi pelajaran Inggris.
			Selasa, 21 Juni 2022		5 Jam	Simulasi Ujian TensorFlow Developer Certificate C	1. Membuat model dengan berbaagai contoh yang sederhana untuk simulasi ujian TF
		Jumat, 24 Juni 2022, pk 19.00-21.30		2,5 Jam		Bangkit Judging for Company Capstone - Traveloka Session 2	1. Presentasi hasil kerja capstone 2. Evaluasi hasil kerja dari judges

BAB III
Machine Learning Path

3.1 Machine Learning

Mengutip dari Dicoding (2020) Teknologi machine learning (ML) adalah mesin yang dikembangkan untuk bisa belajar dengan sendirinya tanpa arahan dari penggunanya. Pembelajaran mesin dikembangkan berdasarkan disiplin ilmu lainnya seperti statistika, matematika dan data mining sehingga mesin dapat belajar dengan menganalisa data tanpa perlu di program ulang atau diperintah. Machine learning memungkinkan untuk mesin melakukan tugas tanpa perlu menyatakan logika dari tugas itu sendiri, dan hanya perlu mempelajari data masukan dan data akhir.

Deep Learning adalah subbidang dari Machine Learning yang dipelajari di Bangkit Academy, mengutip dari Dicoding (2021) Deep learning merupakan subbidang machine learning yang algoritmanya terinspirasi dari struktur otak manusia. Struktur tersebut dinamakan Artificial Neural Networks atau disingkat ANN. Pada dasarnya, ia merupakan jaringan saraf yang memiliki tiga atau lebih lapisan ANN. Ia mampu belajar dan beradaptasi terhadap sejumlah besar data serta menyelesaikan berbagai permasalahan yang sulit diselesaikan dengan algoritma machine learning lainnya.

Program Bangkit Academy membekali peserta dengan beberapa jenis machine learning yang dipelajar, salah satunya merupakan *image classification* yang merupakan bagian dari machine learning yang digunakan untuk mempelajari dan mengenali citra, salah satu contoh permasalahan dari image classsification adalah membedakan kucing dan anjing, bagi peserta sendiri hal ini merupakan sesuatu yang dianggap sepele, tetapi perancangan machine learning untuk dapat image classification perlu adanya beberapa proses, dan di dalam program ini peserta mempelajari perancangan arsitektur model, *preprocessing*, dan juga augmentasi data untuk memperbanyak data yang nantinya digunakan untuk melatih model.

Jenis machine learning yang dipelajari oleh peserta bangkit lainnya adalah Time-Series Forecasting, yaitu teknik untuk memprediksi nilai yang nantinya akan terjadi dengan mempelajari pola yang telah terjadi sebelumnya. Dalam program ini peserta juga akan mempelajari konsep Long Short Term Memory layer, Bidirectional, dan sebagainya

3.2 Instructor-Led Training

Intructor-Led Training atau ILT adalah sebuah kelas yang diberikan oleh tim Bangkit Academy untuk membekali peserta dengan pengetahuan mengenai Machine Learning, Softskill, dan juga bahasa inggris. Kelas tersebut diajar oleh ahli ahli yang bekerja di industri.

ILT Machine Learning mengajarkan konsep mengenai machine learning, seperti matematika, otomasi, dan juga penggunaan Tensorflow. Kelas ILT-ML yang telah dijalani dari awal hingga waktu penulisan diantaranya:

1. ILT-ML-01 Python IT Automation - Intro to Python, Regex, and Bash Scripting
2. ILT-ML-02 Python IT Automation - Git Collaboration, Troubleshooting, and Intro to Cloud
3. ILT-ML-03 Mathematics for Machine Learning
4. ILT-ML-04 Tensorflow in Practice
5. ILT-ML-05 Tensorflow Data & Deployment

ILT Softskill mengajarkan kemampuan kemampuan yang tidak berkaitan dengan kemampuan teknikal tetapi sangat penting dalam pekerjaan, industri, dan lainnya. Kelas ILT-SS yang telah dijalani dari awal hingga waktu penulisan diantaranya:

1. ILT-SS-01 Time Management
2. ILT-SS-02 Professional Branding & Interview
3. ILT-SS-03 Critical Thinking
4. ILT-SS-04 Adaptability

5. ILT-SS-05 Idea Generation & MVP Planning
6. ILT-SS-06 Startup Valuation & Investment Pitch

English Session merupakan kelas dimana peserta akan belajar bahasa inggris dengan pengajar internasional, dan nantinya akan mempelajari kemampuan berbahasa inggris yang dapat berguna di dalam industri. Kelas English Session yang telah dijalani dari awal hingga waktu penulisan diantaranya:

1. English Session EN1 Spoken Correspondence
2. English Session EN2 Expressing Opinions
3. English Session EN3 Business Presentation

3.3 Capstone Project MSIB

Capstone Project adalah proyek mandiri yang dilakukan pada akhir masa Bangkit Academy dan dilakukan secara berkelompok, Capstone Project dilakukan pada minggu ke 12 hingga minggu ke 17 dari program ini. Capstone Project pada tahun ini dibagi oleh tim Bangkit Academy menjadi 2 jenis, yakni Product Based Capstone yang membebaskan peserta memecahkan permasalahan yang mereka ingin selesaikan dengan menggabungkan ilmu dari 3 Learning Path yang ada di program ini. Company Based Capstone merupakan jenis Capstone baru yang dimulai tahun ini dan tim Bangkit bekerja sama dengan beberapa perusahaan untuk memecahkan permasalahan dari mereka dan peserta ditantang untuk memecahkan permasalahan mereka, perbedaan lainnya adalah capstone ini mewajibkan untuk kelompok memiliki komposisi Learning Path yang telah ditetapkan oleh perusahaan yang bersangkutan. Keluaran dari kedua jenis Capstone Project adalah video dan juga purwarupa hasil kerja dari kelompok.

Penulis memilih Company Capstone dengan permasalahan yang ditawarkan oleh perusahaan Traveloka Singapura. Proyek yang dilakukan adalah *chatbot* yang digunakan untuk pelanggan. Kelompok yang disusun untuk capstone project ini terdiri dari 3 orang dari Machine Learning, 3 orang dari Mobile Development, dan

juga 3 orang dari Cloud Computing. Solusi yang penulis dan kelompok kerjakan adalah chatbot yang digunakan untuk menjawab pertanyaan dari pelanggan. Kelompok penulis melihat bagaimana pelanggan dari suatu website atau pelayanan enggan melihat *Frequently Asked Question(FAQ)*, hal ini juga ditambah dengan tampilan dari FAQ yang dimiliki oleh website Traveloka terlihat kurang menarik.

3.4 Pelaksanaan, Hasil, dan Pembahasan Capstone Project

Pelaksanaan capstone project pertama dilakukan dengan melakukan diskusi bersama sama dengan 8 anggota lainnya untuk solusi yang ingin digunakan pada setiap Learning Path, seperti infrastruktur cloud computing yang ingin dibuat, desain kasar dari aplikasi, dan juga model machine learning yang ingin dirancang. Subkelompok Machine Learning memutuskan untuk melakukan *transfer learning* hal ini dilakukan dengan menggunakan model yang sudah pernah dibuat sebelumnya lalu dilakukan modifikasi untuk memenuhi kebutuhan kelompok penulis, hal tersebut kami pertimbangkan karena batasan waktu dan juga fleksibitas dari hasil akhir yang didapatkan dapat cukup fleksibel untuk digunakan pada permasalahan yang berbeda tetapi masih berkaitan dengan tanya jawab.

Model yang akan kita rancang adalah model tanya jawab yang berbasis dari model yang dirancang oleh Cahya Irawan bernama Indonesian BERT base model (uncased),berdasarkan artikel dari mti.binus(2020) BERT atau Bidirectial Encoder Representations from Transformers adalah model yang dirancang dan digunakan oleh Google untuk memahami konteks kata dalam permintaan pencarian, model tersebut dilatih menggunakan dataset berbahasa indonesia dari Wikipedia, dan hasil akhir dari model nya digunakan untuk aplikasi *masked language modelling*, dan model ini merupakan model yang bersifat uncased yang berarti model ini tidak peduli dengan huruf besar atau huruf kecil dari suatu kata.

Dataset yang kita gunakan adalah Stanford Question Answering Dataset atau SQuAD versi 2.0, dataset ini adalah dataset tanya jawab yang cukup populer, dataset

ini berisi konteks yang digunakan model machine learning untuk memahami maksud dari pertanyaan yang ditanyakan, pertanyaan itu sendiri, dan juga jawaban. Sebelum dataset tersebut digunakan, isi dari dataset tersebut perlu diterjemahkan ke bahasa indonesia dikarenakan model yang akan kami rancang menggunakan bahasa indonesia dan dataset yang digunakan wajib berbahasa indonesia.

Proses menerjemahkan dataset menggunakan library deep_translator yang digunakan untuk menghubungkan program dengan layanan terjemahan seperti Google Transalate, file dataset yang berbentuk JSON akan disortir isinya dan nantinya akan diterjemahkan ke bahasa indonesia, lalu dataset yang sudah diterjemahkan akan disusun menjadi format yang digunakan oleh huggingface, yang merupakan website yang berisi banyak sekali model pre-trained dan juga dataset. Setelah dataset telah diterjemahkan dan diformat ulang maka proses transfer learning dapat dimulai.

Proses transfer learning pertama yang dilakukan adalah impor dua library yang dibutuhkan yakni tensorflow versi 2.8.1 dan juga transformers 4.18.0, versi yang digunakan perlu spesifik untuk mencegah permasalahan dalam pengerjaan. Setelah library sudah terimpor, maka langkah selanjutnya adalah memuat model dan juga tokenizer yang akan digunakan untuk melakukan transfer learning.

```
# loading base model

model_name = "cahya/bert-base-indonesian-522M"
batch_size = 16

from transformers import AutoTokenizer, TFAutoModel # make sure use tensorflow model
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = TFAutoModel.from_pretrained(model_name) # if not specified, it will use torch model
```

Gambar 3.1 Memuat model dari huggingface

Lalu dataset dapat dimasukan kedalam program, dan sebelum dataset tersebut dapat digunakan perlu adanya proses pre-processing, hal ini dilakukan untuk

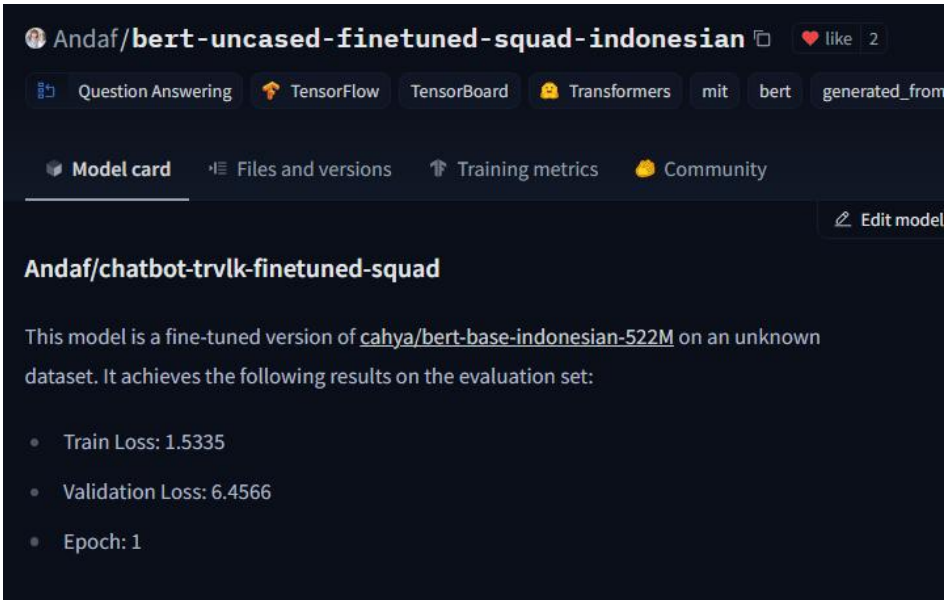
mengubah dataset menjadi nilai yang dapat diterima oleh model machine learning nantinya. Proses yang dilakukan pertama kali adalah tokenizing, yakni mengubah teks menjadi bagian kecil yakni token, hal ini dilakukan untuk membuat teks dapat diproses oleh model nantinya, tokenizer juga digunakan untuk membatasi jumlah kata yang diproses nantinya. Lalu dilakukan mapping untuk mengatur fitu dari teks yang panjang, lalu dilakukan processing lainnya agar dataset dapat dimengerti oleh model nantinya. Apabila preprocessing sudah dilakukan, maka model sebelumnya diunduh kembali tetapi kelas model yang digunakan diganti untuk kebutuhan tanya jawab.

```
from transformers import TFAutoModelForQuestionAnswering

model = TFAutoModelForQuestionAnswering.from_pretrained(model_name)
```

```
All model checkpoint layers were used when initializing TFBertForQuestionAnswering.

Some layers of TFBertForQuestionAnswering were not initialized from the model checkpoint at cahya/bert-base-indonesian-522M and are newly initialized:
['qa_outputs']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

Gambar 3.2 Memuat model untuk tugas tanya jawab

Setelah model diunduh ulang, akan terdapat pemberitahuan bahwa beberapa bobot telah diubah nilainya atau dibuang, hal ini dikarenakan model yang digunakan berbeda dengan tugas sebenarnya dan perlu adanya pelatihan ulang. Sebelum model dapat dilatih perlu adanya pengaturan terhadap akun yang akan digunakan untuk mendorong model ke huggingface, callback, optimizer, loss function dan pengaturan untuk training juga perlu diatur sebelum model dapat dilatih, setelah itu model nya dapat dicompile dan dilatih dengan waktu total kurang lebih 5 jam, lalu modelnya akan disimpan dengan nama “Andaf/bert-uncased-finetuned-squad-indonesian”.



Gambar 3.3 Model akhir yang telah diunggah

Model yang dihasilkan hanya dapat menghasilkan satu jawaban untuk setiap pertanyaan, hal ini dikarenakan model yang dihasilkan mengeluarkan jawaban dengan persentase *confidence* tertinggi, hal lain yang dapat terjadi adalah jawaban salah apabila konteks yang digunakan banyak menggunakan bahasa inggris, hal ini dikarenakan model yang digunakan menggunakan bahasa indonesia.

Model akhir nantinya akan digunakan oleh server yang telah disiapkan oleh subkelompok Cloud Computer, server yang rancang berbasis framework Flask yaitu framework web yang berbasis bahasa pemrograman Python, dan karena bahasa pemrograman yang digunakan untuk merancang model machine learning juga menggunakan Python, hal ini juga memudahkan dalam diskusi apabila terdapat kesulitan. *Deployment* webapp menggunakan Google Cloud Run dikarenakan pengaturannya yang minim.

Aplikasi Mobile yang dirancang oleh subkelompok berbasis bahasa pemrograman Kotlin, aplikasi ini dirancang untuk terlihat seperti aplikasi yang dimiliki traveloka langsung, aplikasi ini memiliki fungsi utama untuk berkomunikasi

dengan server, dan juga aplikasi memiliki fitur speech-to-text untuk membantu tunanetra dapat menggunakan aplikasi.

Gambar 3.4 Tampilan aplikasi

BAB IV
Penutup

4.1 Kesimpulan

Melihat seluruh kegiatan yang telah dilakukan di Program Studi Independen Bersertifikat di Bangkit Academy 2022 dapat disimpulkan bahwa:

1. Pelaksanaan Program Studi Independen Bersertifikat di Bangkit Academy 2022 pada Machine Learning path dibagi menjadi 3 jenis pembelajaran, yaitu: pembelajaran mandiri menggunakan platform Dicoding dan Coursera, Instructor-Led Training atau ILT, dan juga Capstone Project.
2. Pembelajaran mandiri di program Studi Independen Bersertifikat di Bangkit Academy 2022 pada Machine Learning path terdapat beberapa topik, yaitu: Dicoding's Python, IT Automation with Python, Mathematics for Machine Learning, TF Developer Professional Certificate, Structuring Machine Learning Project, TF Data and Deployment, TensorFlow Certification Preparation.
3. Program Studi Independen Bersertifikat di Bangkit Academy 2022 menjadi pintu bagi mahasiswa untuk mempelajari keahlian dalam pengembangan machine learning.
4. Pelaksanaan Capstone Project memberikan peserta program pengalaman dalam pengerjaan proyek yang dapat mencerminkan pengerjaan proyek yang nantinya akan dilakukan di industri.
5. Sertifikasi dan juga tugas tugas yang diberikan menjadi tolak ukur bahwa peserta memahami materi materi yang telah dipelajari sebelumnya.

4.2 Saran

Adapun saran untuk Program Studi Independen Bersertifikat di Bangkit Academy 2022, yaitu:

1. Program ILT-ML akan lebih baik apabila diisi dengan materi yang setidaknya berbeda dengan materi yang sebelumnya telah dipelajari secara mandiri, hal ini membuat program tersebut terasa membosankan karena harus mengulang materi.
2. Template dokumen dan juga urusan administrasi lainnya perlu disiapkan terlebih dahulu, hal ini untuk mempermudah peserta untuk mempersiapkan.
3. Penggunaan logbook dirasa kurang efektif apabila digunakan secara bersamaan dengan laporan akhir, karena dirasa melakukan dua kegiatan yang sama.
4. Bantuan ketika mengerjakan Capstone Project, seperti Google Cloud Platform Credit atau Colab Pro perlu disegerakan, penundaan serta ketidakjelasan ketika pengerjaan dapat memperlambat kemajuan dari proyek.

Daftar Pustaka

Takdirillah, R., 2020. Apa itu Machine Learning? Beserta Pengertian dan Cara Kerjanya. [online] Dicoding. Available at: <https://www.dicoding.com/blog/machine-learning-adalah/> [Accessed 25 June 2022].

Setiawan, R., 2021. Mengenal Deep Learning Lebih Jelas. [online] Dicoding. Available at: <https://www.dicoding.com/blog/mengenal-deep-learning/> [Accessed 25 June 2022].

Alaydrus, A., 2020. Penerapan Algoritma BERT dalam Search Engine Google. [online] MTI. Available at: <https://mti.binus.ac.id/2020/09/03/penerapan-algoritma-bert-dalam-search-engine-google/> [Accessed 27 June 2022].

Lampiran A. Terms of Service

Completion Requirements

Careful planning has gone into designing the curriculum for this program from beginning to end. At the conclusion of the program, all participants who meet the completion criteria, will be regarded as Bangkit Graduates and given certificate of accomplishment/completion and a complete transcript. Those who didn't complete all the Bangkit will get Certificate of Attendance and partial transcript. Bangkit Graduates will also receive a voucher for the certification exam of their respective Learning Path. The requirements for graduation from Bangkit 2022 are as follows:

- **Attending and actively participating in mandatory sessions**, including but not limited to:
    - Bangkit 2022 Opening Session
    - 80% of the Instructor-led sessions for Tech*
    - 80% of the Instructor-led sessions for Soft Skills*
    - 90% of mandatory guest/special lectures*
    - and other mandatory sessions added at the discretion of the Bangkit Team

  Sessions will be informed at least 7 calendar days before. So please check your calendar on a daily basis

  * participants may skip sessions due to extraordinary & indispensable circumstances by filling this form (max. 3 day before the session). And participants have **5 chances** to skip the mandatory session. Missed sessions must be made up by joining another group's session or watching the recording and submitting an abstract.

- **For self-paced sessions, you just need to complete them in the same week**.

  If you have things to do for the allocated self-paced time, you don't need to fill the form. Just allocate another time outside Bangkit allocated time to study and adjust by yourself.

  As you're aware, the Bangkit learning method combines online self-paced study, online synchronous / instructor-led training (ILT), and project-based learning. Therefore, to help you plan your time, we have created a Bangkit learning schedule.

- **Submit your own work for assignments and projects.**

  Bangkit is part of the Kampus Merdeka program where academic honesty is upheld. You should demonstrate and uphold the highest integrity and honesty in all the academic work that you do. Plagiarism isn't permitted and score for the respective assignment will be void/canceled in the event your work is flagged for plagiarism. Our learning platform partners may ban or disable your account if you plagiarize or are dishonest based on their sole discretion.

- **Completing official Bangkit assignments** (including classroom and our learning platform partners - Dicoding, Google Cloud Skills Boost, Coursera) each in accordance with their respective standards. Late submission will be accepted, but will reduce the respective assignment score.

- **Contributing to Bangkit Capstone Project**.

  This will be scored by the Bangkit Committee and your team members and includes your attendance in the final project presentation.

- **Adhering to the Bangkit Code of Conduct**

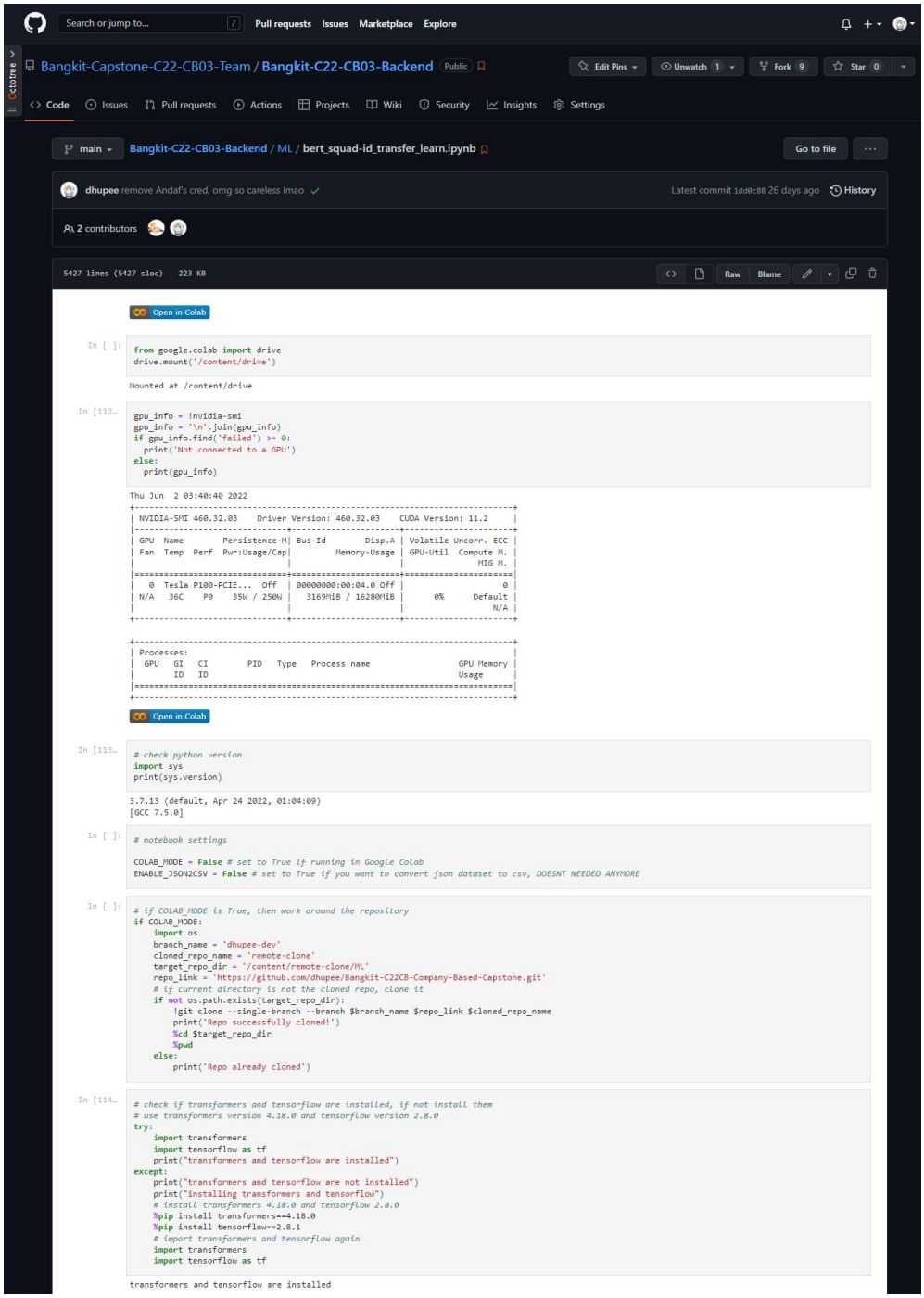
Lampiran B. Target Schedule

<table>
<tr><td colspan="4">Bangkit 2022</td></tr>
<tr><td colspan="2">Week of</td><td>Soft skills</td><td>English</td><td>Machine Learning</td></tr>
<tr><td>Week 0</td><td>7,Feb</td><td></td><td>English Pre-test</td><td>Matriculation</td></tr>
<tr><td>Week 1</td><td>14,Feb</td><td></td><td></td><td>Dicoding's Python (end of the course)</td></tr>
<tr><td rowspan="3">Week 2</td><td rowspan="3">21,Feb</td><td rowspan="3">Preread SS 1
Time Management</td><td rowspan="3"></td><td>ILT Tech 1</td></tr>
<tr><td>IT Automation with Python (Python Crash Course Final Project)</td></tr>
<tr><td>IT Automation with Python (Course 2 Python to Interact with OS)</td></tr>
<tr><td>Week 3</td><td>28,Feb</td><td>ILT SS 1</td><td></td><td>IT Automation with Python (Course 3 Intro to Git & GitHub)</td></tr>
<tr><td rowspan="3">Week 4</td><td rowspan="3">07,Mar</td><td rowspan="2">Assignment SS 1</td><td rowspan="3">English - 1 Spoken Correspondence</td><td>ILT Tech 2</td></tr>
<tr><td>IT Automation with Python (Course 4 Troubleshooting & Debugging)</td></tr>
<tr><td>Preread SS 2
Professional Branding & Interview</td><td>IT Automation with Python (Course 5 Configuration Management & the Cloud)</td></tr>
<tr><td rowspan="2">Week 5</td><td rowspan="2">14,Mar</td><td rowspan="2">ILT SS 2</td><td rowspan="2"></td><td>IT Automation with Python (Course 6 Automating Real World Task)</td></tr>
<tr><td>Mathematics for Machine Learning
(Course 1 Linear Algebra)</td></tr>
<tr><td rowspan="2">Week 6</td><td rowspan="2">21,Mar</td><td rowspan="2">Assignment SS 2</td><td rowspan="2"></td><td>ILT Tech 3</td></tr>
<tr><td>Mathematics for Machine Learning
(Course 2 Multivariate Calculus)</td></tr>
</table>

		Preread SS 3 Critical thinking		Mathematics for Machine Learning (Course 3 PCA)
Week 7	28,Mar	ILT SS 3		TF Developer Professional Certificate (Course 1 Intro to TF)
				TF Developer Professional Certificate (Course 2 Convolutional Neural Network - Week 1 Exploring a Larger Dataset)
Week 8	04,Apr	Assignment SS 3	English - 2 Expressing Opinion	ILT Tech 4
				TF Developer Professional Certificate (Course 2 Convolutional Neural Network - end of the course)
		Preread SS 4 Adaptability		TF Developer Professional Certificate (Course 3 Natural Languange Processing)
Week 9	11,Apr	ILT SS 4		TF Developer Professional Certificate (Course 4 Time Series)
				Structuring Machine Learning Project (end of the Course)
Week 10	18,Apr			ILT Tech 5
		Assignment SS 4		TF Data and Deployment (Browser based Model)
		Preread SS 5 Idea Generation & MVP Planning		TF Data and Deployment (Device based Model)
Week 11	25,Apr	ILT SS 5		TF Data and Deployment (Data Pipelines)

				TF Data and Deployment (Advanced Deployment Scenarios)
	29,Apr	IED HOLIDAY		
Week 12	09,May	Assignment SS 5		Capstone Project
Week 13	16,May		English - 3 Business Presentation	
Week 14	23,May			
Week 15	30,May			
Week 16	06,Jun			
Week 17	13, Jun	Preread SS 6 Startup Valuation & Investment Pitch		
Week 18	20,Jun	ILT SS 6 & Assignment	English Post-test	TensorFlow Certification Preparation (up to 2nd case)
Week 19	27,Jun	Preread SS 7 Professional Communications		ILT Tech 6
				TensorFlow Certification Preparation (up to last simulation)
Week 20	04,Jul	ILT SS 7 & Assignment		Expert Classes (Optional)
Week 21	11,Jul			End of Learning, Certification Offering, Merchandise
	18,Jul			Transcript & Administration
	25,Jul			Clarification, Legal & Letters, Closing

Lampiran C. Dokumen Teknik



Search or jump to... / Pull requests Issues Marketplace Explore

Bangkit-Capstone-C22-CB03-Team / Bangkit-C22-CB03-Backend Public

Edit Pins | Unwatch 1 | Fork 9 | Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main Bangkit-C22-CB03-Backend / ML / bert_squad-id_transfer_learn.ipynb Go to file

dhupee remove Andaf's cred, omg so careless lmao ✓ Latest commit 1dd0c88 26 days ago History

2 contributors

5427 lines (5427 sloc) | 223 KB Raw Blame

Open in Colab

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

```
Mounted at /content/drive
```

```
In [112…
gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
  print('Not connected to a GPU')
else:
  print(gpu_info)
```

```
Thu Jun  2 03:40:40 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:04.0 Off |                    0 |
| N/A   36C    P0    35W / 250W |   3169MiB / 16280MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
+-----------------------------------------------------------------------------+
```

Open in Colab

```
In [113…
# check python version
import sys
print(sys.version)
```

```
3.7.13 (default, Apr 24 2022, 01:04:09)
[GCC 7.5.0]
```

```
In [ ]:
# notebook settings

COLAB_MODE = False # set to True if running in Google Colab
ENABLE_JSON2CSV = False # set to True if you want to convert json dataset to csv, DOESNT NEEDED ANYMORE
```

```
In [ ]:
# if COLAB_MODE is True, then work around the repository
if COLAB_MODE:
    import os
    branch_name = 'dhupee-dev'
    cloned_repo_name = 'remote-clone'
    target_repo_dir = '/content/remote-clone/ML'
    repo_link = 'https://github.com/dhupee/Bangkit-C22CB-Company-Based-Capstone.git'
    # if current directory is not the cloned repo, clone it
    if not os.path.exists(target_repo_dir):
        !git clone --single-branch --branch $branch_name $repo_link $cloned_repo_name
        print('Repo successfully cloned!')
        %cd $target_repo_dir
        %pwd
    else:
        print('Repo already cloned')
```

```
In [114…
# check if transformers and tensorflow are installed, if not install them
# use transformers version 4.18.0 and tensorflow version 2.8.0
try:
    import transformers
    import tensorflow as tf
    print("transformers and tensorflow are installed")
except:
    print("transformers and tensorflow are not installed")
    print("installing transformers and tensorflow")
    # install transformers 4.18.0 and tensorflow 2.8.0
    %pip install transformers==4.18.0
    %pip install tensorflow==2.8.1
    # import transformers and tensorflow again
    import transformers
    import tensorflow as tf
```

```
transformers and tensorflow are installed
```

```
# loading base model

model_name = "cahya/bert-base-indonesian-522M"
batch_size = 16

from transformers import AutoTokenizer, TFAutoModel # make sure use tensorflow model
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = TFAutoModel.from_pretrained(model_name) # if not specified, it will use torch model
```

```
Some layers from the model checkpoint at cahya/bert-base-indonesian-522M were not used when initializing TFBertModel: ['mlm___cls']
- This IS expected if you are initializing TFBertModel from the checkpoint of a model trained on another task or with another architecture (e.g. initi
alizing a BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing TFBertModel from the checkpoint of a model that you expect to be exactly identical (initializing a Bert
ForSequenceClassification model from a BertForSequenceClassification model).
All the layers of TFBertModel were initialized from the model checkpoint at cahya/bert-base-indonesian-522M.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFBertModel for predictions without further traini
ng.
```

```
model.summary()
```

```
Model: "tf_bert_model_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bert (TFBertMainLayer)      multiple                  110617344

=================================================================
Total params: 110,617,344
Trainable params: 110,617,344
Non-trainable params: 0
_________________________________________________________________
```

```
# test tokenizer
tokenizer("Nama kamu siapa?")
```

```
{'input_ids': [3, 1769, 8343, 6186, 32, 1], 'token_type_ids': [0, 0, 0, 0, 0, 0], 'attention_mask': [1, 1, 1, 1, 1, 1]}
```

```
tokenizer("saya suka makan nasi goreng")
```

```
{'input_ids': [3, 3245, 5366, 2464, 6014, 11186, 1], 'token_type_ids': [0, 0, 0, 0, 0, 0, 0], 'attention_mask': [1, 1, 1, 1, 1, 1, 1]}
```

```
# see how base model works
unmasker = transformers.pipeline('fill-mask', model = model_name)
unmasker("mainan saya [MASK] di jalan")
```

```
[{'score': 0.08403640985488892,
  'sequence': 'mainan saya berada di jalan',
  'token': 2186,
  'token_str': 'berada'},
 {'score': 0.07038332521915436,
  'sequence': 'mainan saya ada di jalan',
  'token': 1821,
  'token_str': 'ada'},
 {'score': 0.04035765677690506,
  'sequence': 'mainan saya sendiri di jalan',
  'token': 1998,
  'token_str': 'sendiri'},
 {'score': 0.02904832921922207,
  'sequence': 'mainan saya lahir di jalan',
  'token': 2444,
  'token_str': 'lahir'},
 {'score': 0.02813726291060447,
  'sequence': 'mainan saya berdiri di jalan',
  'token': 3812,
  'token_str': 'berdiri'}]
```

```
#this for converting the file to huggingface format
#I suggest you to do it seperately

def convert_huggingface(dataset_path):
    import json
    with open(dataset_path, encoding="utf8") as f:
        content = json.load(f)

    hf_data = []
    for data in content["data"]:
        title = data["title"]
        for paragraph in data["paragraphs"]:
            context = paragraph["context"]
            for qa in paragraph["qas"]:
                fill = {
                    "id": qa["id"],
                    "title": title,
                    "context": context,
                    "question": qa["question"],
                    "answers": {"answer_start": [], "text": []}
                }
                if qa["is_impossible"]:
                    answers = qa["plausible_answers"]
                else:
                    answers = qa["answers"]
                for answer in answers:
                    fill["answers"]["answer_start"].append(answer["answer_start"])
                    fill["answers"]["text"].append(answer["text"])

                hf_data.append(fill)
    # Add "_hf" before .json extension
    hf_dataset_path = dataset_path.replace(".json", "_hf.json")
    with open(hf_dataset_path, "w") as f:
        json.dump({"data": hf_data}, f)

    return hf_data
```

```
# load dataset json file
import json

#colab pro + drive
train_json_dir = "/content/drive/MyDrive/Translated/hf_train-v2.0_indo.json"
valid_json_dir = "/content/drive/MyDrive/Translated/hf_dev-v2.0_indo.json"
tester_json_dir  = "/content/drive/MyDrive/Translated/tester_indo.json"

dataset_dirs = [train_json_dir, valid_json_dir, tester_json_dir]
# dataset_dirs = [tester_json_dir]
```

```
# importing dataset

try:
    from datasets import load_dataset
except:
    print("datasets module not found")
    print("installing dataset module")
    %pip install datasets
    from datasets import load_dataset

datasets = load_dataset(
    'json',
    data_files={'train': train_json_dir, 'validation': valid_json_dir},
    field='data'
    )
```

```
Using custom data configuration default-2599678fccd9e0e7
Reusing dataset json (/root/.cache/huggingface/datasets/json/default-2599678fccd9e0e7/0.0.0/ac0ca5f5289a6cf108e706efcf040422dbbfa8e658dee6a819f20d76bb84d26b)
  0%|          | 0/2 [00:00<?, ?it/s]
```

```
# sample of dataset randomly
import random
datasets["train"][4]
```

```
{'answers': {'answer_start': [304], 'text': ['akhir 1990-an']},
 'context': 'Beyoncé Giselle Knowles-Carter (/ biːˈjɒnseɪ / bee-YON-say) (lahir 4 September 1981) adalah seorang penyanyi, penulis lagu, produser rekaman dan aktris Amerika. Dilahirkan dan dibesarkan di Houston, Texas, ia tampil di berbagai kompetisi menyanyi dan menari sebagai seorang anak, dan mulai terkenal pada akhir 1990-an sebagai penyanyi utama dari grup wanita R&B Destiny\'s Child. Dikelola oleh ayahnya, Mathew Knowles, grup ini menjadi salah satu grup wanita terlaris di dunia sepanjang masa. Jeda mereka melihat perilisan album debut Beyoncé, Dangerously in Love (2003), yang menjadikannya sebagai artis solo di seluruh dunia, meraih lima Grammy Awards dan menampilkan single nomor satu Billboard Hot 100 "Crazy in Love" dan "Baby Boy".',
 'id': '56bf6b0f3aeaaa14008c9602',
 'question': 'Pada dekade berapa Beyonce menjadi terkenal?',
 'title': 'Beyoncé'}
```

```
# sample of dataset randomly
import random
datasets["validation"][4]
```

```
{'answers': {'answer_start': [671, 649, 671, 671],
  'text': ['abad ke-10',
   'paruh pertama abad ke-10',
   'tanggal 10',
   'tanggal 10']},
 'context': 'Bangsa Norman (Norman: Nourmands; Prancis: Normandia; Latin: Normanni) adalah orang-orang yang pada abad ke-10 dan ke-11 memberi nama kepada Normandia, sebuah wilayah di Prancis. Mereka adalah keturunan dari perampok dan bajak laut Norse ("Norman" berasal dari "Norseman") dan bajak laut dari Denmark, Islandia dan Norwegia yang, di bawah pemimpin mereka Rollo, setuju untuk bersumpah setia kepada Raja Charles III dari Francia Barat. Melalui generasi asimilasi dan pencampuran dengan penduduk asli Franka dan Romawi-Gaul, keturunan mereka secara bertahap akan bergabung dengan budaya berbasis Carolingian di Francia Barat. Identitas budaya dan etnis yang berbeda dari Normandia awalnya muncul pada paruh pertama abad ke-10, dan terus berkembang selama abad-abad berikutnya.',
 'id': '56ddde6b9a695914005b962c',
 'question': 'Abad berapa orang Normandia pertama kali mendapatkan identitas mereka yang terpisah?',
 'title': 'orang Normandia'}
```

```
from datasets import ClassLabel, Sequence
import random
import pandas as pd
from IPython.display import display, HTML


def show_random_elements(dataset, num_examples=10):
    '''display random elements from dataset

    Args:
        dataset (Dataset): dataset to show
        num_examples (int, optional): number of examples to show. Defaults to 10.
    '''
    assert num_examples <= len(
        dataset
    ), "Can't pick more elements than there are in the dataset."
    picks = []
    for _ in range(num_examples):
        pick = random.randint(0, len(dataset) - 1)
        while pick in picks:
            pick = random.randint(0, len(dataset) - 1)
        picks.append(pick)

    df = pd.DataFrame(dataset[picks])
    for column, typ in dataset.features.items():
        if isinstance(typ, ClassLabel):
            df[column] = df[column].transform(lambda i: typ.names[i])
        elif isinstance(typ, Sequence) and isinstance(typ.feature, ClassLabel):
            df[column] = df[column].transform(
                lambda x: [typ.feature.names[i] for i in x]
            )
    display(HTML(df.to_html()))
```

```
show_random_elements(datasets["train"]) # brace yourself, this is gonna be a long list
```

```
'''
    This function is for converting SQuAD json file to pandas dataframe, iteratively

    I dont want run this locally, better use colab

    doesn't needed anymore, use load_dataset instead
'''

if ENABLE_JSON2CSV:
    import utils
    for dir in dataset_dirs:
        with open(dir, encoding="utf-8") as json_file:
            file = json.load(json_file)
            dict_file = file
            data = dict_file['data']

        df = utils.json_to_df(data)
        df.to_csv(dir.replace(".json", ".csv"), index = False)
```

```
assert isinstance(tokenizer, transformers.PreTrainedTokenizerFast) # make sure tokenizer is pre-trained
```

```
max_length = 384  # The maximum length of a feature (question and context)
doc_stride = 128  # The allowed overlap between two part of the context when splitting is performed.
```

```
# check if there's dataset feature
# longer than max_length
for i, example in enumerate(datasets["train"]):
    if len(tokenizer(example["question"], example["context"])["input_ids"]) > 384:
        break
example = datasets["train"][i]
```

```
# check if there's dataset feature longer than max_length
len(tokenizer(example["question"], example["context"])["input_ids"])
```

```
402
```

```
# check truncate dataset length
len(
    tokenizer(
        example["question"],
        example["context"],
        max_length=max_length,
        truncation="only_second",
    )["input_ids"]
)
```

```
384
```

```
tokenized_example = tokenizer(
    example["question"],
    example["context"],
    max_length=max_length,
    truncation="only_second",
    return_overflowing_tokens=True,
    stride=doc_stride,
)
```

```
[len(x) for x in tokenized_example["input_ids"]]
```

```
[384, 156]
```

```
for x in tokenized_example["input_ids"][:2]:
    print(tokenizer.decode(x))
```

```
[CLS] beyonce menikah pada 2008 dengan siapa? [SEP] pada 4 april 2008, beyonce menikahi jay z. dia secara terbuka mengungkapkan pernikahan mereka dala
m montase video di pesta mendengarkan untuk album studio ketiganya, i am... sasha fierce, di sony club manhattan pada 22 oktober 2008. i am... sasha f
ierce dirilis pada 18 november 2008 di amerika serikat. album ini secara resmi memperkenalkan alter ego beyonce sasha fierce, yang dibuat selama pembu
atan singel tahun 2003 " crazy in love ", terjual 482. 000 kopi di minggu pertama, memulai debutnya di atas billboard 200, dan memberikan beyonce albu
m nomor satu ketiganya berturut - turut di kami. album ini menampilkan lagu nomor satu " single ladies ( put a ring on it ) " dan lagu lima teratas "
if i were a boy " dan " halo ". mencapai pencapaian menjadi single hot 100 terlama dalam karirnya, kesuksesan " halo " di as membantu beyonce mencapai
lebih dari sepuluh single teratas dalam daftar daripada wanita lain selama tahun 2000 - an. ini juga termasuk " mimpi manis " yang sukses, dan single
" diva ", " ego ", " gadis patah hati " dan " telepon video ". video musik untuk " single ladies " telah diparodikan dan ditiru di seluruh dunia, memi
cu " kegilaan tari besar pertama " di era internet menurut toronto star. video ini telah memenangkan beberapa penghargaan, termasuk video terbaik di p
enghargaan musik eropa mtv 2009, penghargaan mobo skotlandia 2009, dan penghargaan bet 2009. pada mtv video music awards 2009, video tersebut dinomina
sikan untuk sembilan penghargaan, akhirnya memenangkan tiga penghargaan termasuk video of the year. kegagalannya untuk memenangkan kategori video wani
ta terbaik, yang jatuh ke lagu " you belong with me " dari penyanyi country amerika taylor swift, menyebabkan kanye west menyela upacara dan beyonce m
engimprovisasi presentasi ulang penghargaan swift selama pidato penerimaannya sendiri. pada bulan maret 2009, beyonce memulai i am... world tour, [SE
P]
[CLS] beyonce menikah pada 2008 dengan siapa? [SEP]an tari besar pertama " di era internet menurut toronto star. video ini telah memenangkan beberapa
penghargaan, termasuk video terbaik di penghargaan musik eropa mtv 2009, penghargaan mobo skotlandia 2009, dan penghargaan bet 2009. pada mtv video mu
sic awards 2009, video tersebut dinominasikan untuk sembilan penghargaan, akhirnya memenangkan tiga penghargaan termasuk video of the year. kegagalann
ya untuk memenangkan kategori video wanita terbaik, yang jatuh ke lagu " you belong with me " dari penyanyi country amerika taylor swift, menyebabkan
kanye west menyela upacara dan beyonce mengimprovisasi presentasi ulang penghargaan swift selama pidato penerimaannya sendiri. pada bulan maret 2009,
beyonce memulai i am... world tour, tur konser dunia keduanya, yang terdiri dari 108 pertunjukan, meraup $ 119, 5 juta. [SEP]
```

```
tokenized_example = tokenizer(
    example["question"],
    example["context"],
    max_length=max_length,
    truncation="only_second",
    return_overflowing_tokens=True,
    return_offsets_mapping=True,
    stride=doc_stride,
)
print(tokenized_example["offset_mapping"][0][:100])
```

```
[(0, 0), (0, 7), (8, 15), (16, 20), (21, 25), (26, 32), (33, 38), (38, 39), (0, 0), (0, 4), (5, 6), (7, 12), (13, 17), (17, 18), (19, 26), (27, 35),
(36, 39), (40, 41), (41, 42), (43, 46), (47, 53), (54, 61), (62, 75), (76, 86), (87, 93), (94, 99), (100, 104), (104, 107), (108, 113), (114, 116), (1
17, 122), (123, 135), (136, 141), (142, 147), (148, 154), (155, 164), (164, 165), (166, 167), (168, 170), (171, 172), (172, 173), (173, 174), (175, 17
8), (178, 180), (181, 183), (183, 186), (186, 187), (187, 188), (189, 191), (192, 196), (197, 201), (202, 211), (212, 216), (217, 219), (220, 227), (2
28, 232), (232, 233), (234, 235), (236, 238), (239, 240), (240, 241), (241, 242), (243, 246), (246, 248), (249, 251), (251, 254), (254, 255), (256, 26
3), (264, 268), (269, 271), (272, 280), (281, 285), (286, 288), (289, 296), (297, 304), (304, 305), (306, 311), (312, 315), (316, 322), (323, 328), (3
29, 343), (344, 349), (350, 353), (354, 361), (362, 365), (365, 367), (368, 370), (370, 373), (373, 374), (374, 375), (376, 380), (381, 387), (388, 39
4), (395, 404), (405, 411), (412, 417), (418, 422), (423, 424), (424, 429), (430, 432)]
```

```
sequence_ids = tokenized_example.sequence_ids()
print(sequence_ids)
```

```
[None, 0, 0, 0, 0, 0, 0, 0, None, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, None]
```

```
from transformers import TFAutoModelForQuestionAnswering

model = TFAutoModelForQuestionAnswering.from_pretrained(model_name)
```

```
All model checkpoint layers were used when initializing TFBertForQuestionAnswering.

Some layers of TFBertForQuestionAnswering were not initialized from the model checkpoint at cahya/bert-base-indonesian-522M and are newly initialized:
['qa_outputs']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
answers = example["answers"]
start_char = answers["answer_start"][0]
end_char = start_char + len(answers["text"][0])

# Start token index of the current span in the text.
token_start_index = 0
while sequence_ids[token_start_index] != 1:
    token_start_index += 1

# End token index of the current span in the text.
token_end_index = len(tokenized_example["input_ids"][0]) - 1
while sequence_ids[token_end_index] != 1:
    token_end_index -= 1

# Detect if the answer is out of the span (in which case this feature is labeled with the CLS index).
offsets = tokenized_example["offset_mapping"][0]
if (
    offsets[token_start_index][0] <= start_char
    and offsets[token_end_index][1] >= end_char
):
    # Move the token_start_index and token_end_index to the two ends of the answer.
    # Note: we could go after the last offset if the answer is the last word (edge case).
    while (
        token_start_index < len(offsets) and offsets[token_start_index][0] <= start_char
    ):
        token_start_index += 1
    start_position = token_start_index - 1
    while offsets[token_end_index][1] >= end_char:
        token_end_index -= 1
    end_position = token_end_index + 1
    print(start_position, end_position)
else:
    print("The answer is not in this feature.")
```

```
16 17
```

```
print(
    tokenizer.decode(
        tokenized_example["input_ids"][0][start_position : end_position + 1]
    )
)
print(answers["text"][0])
```

```
jay z
Jay Z
```

```
pad_on_right = tokenizer.padding_side == "right"
```

```python
def prepare_train_features(examples):
    # Tokenize our examples with truncation and padding, but keep the overflows using a stride. This results
    # in one example possible giving several features when a context is long, each of those features having a
    # context that overlaps a bit the context of the previous feature.
    tokenized_examples = tokenizer(
        examples["question" if pad_on_right else "context"],
        examples["context" if pad_on_right else "question"],
        truncation="only_second" if pad_on_right else "only_first",
        max_length=max_length,
        stride=doc_stride,
        return_overflowing_tokens=True,
        return_offsets_mapping=True,
        padding="max_length",
    )

    # Since one example might give us several features if it has a long context, we need a map from a feature to
    # its corresponding example. This key gives us just that.
    sample_mapping = tokenized_examples.pop("overflow_to_sample_mapping")
    # The offset mappings will give us a map from token to character position in the original context. This will
    # help us compute the start_positions and end_positions.
    offset_mapping = tokenized_examples.pop("offset_mapping")

    # Let's label those examples!
    tokenized_examples["start_positions"] = []
    tokenized_examples["end_positions"] = []

    for i, offsets in enumerate(offset_mapping):
        # We will label impossible answers with the index of the CLS token.
        input_ids = tokenized_examples["input_ids"][i]
        cls_index = input_ids.index(tokenizer.cls_token_id)

        # Grab the sequence corresponding to that example (to know what is the context and what is the question).
        sequence_ids = tokenized_examples.sequence_ids(i)

        # One example can give several spans, this is the index of the example containing this span of text.
        sample_index = sample_mapping[i]
        answers = examples["answers"][sample_index]
        # If no answers are given, set the cls_index as answer.
        if len(answers["answer_start"]) == 0:
            tokenized_examples["start_positions"].append(cls_index)
            tokenized_examples["end_positions"].append(cls_index)
        else:
            # Start/end character index of the answer in the text.
            start_char = answers["answer_start"][0]
            end_char = start_char + len(answers["text"][0])

            # Start token index of the current span in the text.
            token_start_index = 0
            while sequence_ids[token_start_index] != (1 if pad_on_right else 0):
                token_start_index += 1

            # End token index of the current span in the text.
            token_end_index = len(input_ids) - 1
            while sequence_ids[token_end_index] != (1 if pad_on_right else 0):
                token_end_index -= 1

            # Detect if the answer is out of the span (in which case this feature is labeled with the CLS index).
            if not (
                offsets[token_start_index][0] <= start_char
                and offsets[token_end_index][1] >= end_char
            ):
                tokenized_examples["start_positions"].append(cls_index)
                tokenized_examples["end_positions"].append(cls_index)
            else:
                # Otherwise move the token_start_index and token_end_index to the two ends of the answer.
                # Note: we could go after the last offset if the answer is the last word (edge case).
                while (
                    token_start_index < len(offsets)
                    and offsets[token_start_index][0] <= start_char
                ):
                    token_start_index += 1
                tokenized_examples["start_positions"].append(token_start_index - 1)
                while offsets[token_end_index][1] >= end_char:
                    token_end_index -= 1
                tokenized_examples["end_positions"].append(token_end_index + 1)

    return tokenized_examples
```

```python
features = prepare_train_features(datasets["train"][:5])
```

```python
tokenized_datasets = datasets.map(
    prepare_train_features, batched=True, remove_columns=datasets["train"].column_names
)
```

```
  0%|          | 0/115 [00:00<?, ?ba/s]
  0%|          | 0/12 [00:00<?, ?ba/s]
```

```python
from transformers import TFAutoModelForQuestionAnswering

model = TFAutoModelForQuestionAnswering.from_pretrained(model_name)
```

```
All model checkpoint layers were used when initializing TFBertForQuestionAnswering.

Some layers of TFBertForQuestionAnswering were not initialized from the model checkpoint at cahya/bert-base-indonesian-522M and are newly initialized:
['qa_outputs']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```python
push_to_hub_model_id = "chatbot-trvlk-finetuned-squad"
learning_rate = 2e-5
num_train_epochs = 2
weight_decay = 0.01
```

```
push_to_hub_model_id = "chatbot-trvlk-finetuned-squad"
learning_rate = 2e-5
num_train_epochs = 2
weight_decay = 0.01
```

```
from transformers import DefaultDataCollator

data_collator = DefaultDataCollator(return_tensors="tf")
```

```
train_set = tokenized_datasets["train"].to_tf_dataset(
    columns=["attention_mask", "input_ids", "start_positions", "end_positions"],
    shuffle=True,
    batch_size=batch_size,
    collate_fn=data_collator,
)
validation_set = tokenized_datasets["validation"].to_tf_dataset(
    columns=["attention_mask", "input_ids", "start_positions", "end_positions"],
    shuffle=False,
    batch_size=batch_size,
    collate_fn=data_collator,
)
```

```
from transformers import create_optimizer

total_train_steps = (len(tokenized_datasets["train"]) // batch_size) * num_train_epochs

optimizer, schedule = create_optimizer(
    init_lr=learning_rate, num_warmup_steps=0, num_train_steps=total_train_steps
)
```

```
import tensorflow as tf

model.compile(optimizer=optimizer)
```

```
No loss specified in compile() - the model's internal loss computation will be used as the loss. Don't panic - this is a common way to train TensorFlo
w models in Transformers! To disable this behaviour, please pass a loss argument, or explicitly pass `loss=None` if you do not want your model to comp
ute a loss.
```

```
from huggingface_hub import notebook_login

notebook_login()
```

```
Login successful
Your token has been saved to /root/.huggingface/token
Authenticated through git-credential store but this isn't the helper defined on your machine.
You might have to re-authenticate when pushing to the Hugging Face Hub. Run the following command in your terminal in case you want to set this creden
tial helper as the default

git config --global credential.helper store
```

```
!apt install git-lfs
!git config --global user.email "your-email"
!git config --global user.name "your-username"
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
git-lfs is already the newest version (2.3.4-1).
The following package was automatically installed and is no longer required:
  libnvidia-common-460
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 42 not upgraded.
```

```
from transformers.keras_callbacks import PushToHubCallback
from tensorflow.keras.callbacks import TensorBoard

push_to_hub_callback = PushToHubCallback(
    output_dir="./qa_model_save",
    tokenizer=tokenizer,
    hub_model_id=push_to_hub_model_id,
)

tensorboard_callback = TensorBoard(log_dir="./qa_model_save/logs")

callbacks = [tensorboard_callback, push_to_hub_callback]

model.fit(
    train_set,
    validation_data=validation_set,
    epochs=num_train_epochs,
    callbacks=callbacks,
)
```

```
/content/qa_model_save is already a clone of https://huggingface.co/Andaf/chatbot-trvlk-finetuned-squad. Make sure you pull the latest changes with `r
epo.git_pull()`.
Epoch 1/2
7222/7222 [==============================] - 6080s 840ms/step - loss: 2.1851 - val_loss: 6.1907
Epoch 2/2
7222/7222 [==============================] - 6058s 839ms/step - loss: 1.5335 - val_loss: 6.4566
Several commits (2) will be pushed upstream.
The progress bars may be unreliable.
Upload file tf_model.h5:   0%|          | 3.34k/420M [00:00<?, ?B/s]
Upload file logs/validation/events.out.tfevents.1654148844.8b5c72b9c6a7.83.12.v2: 100%|##########| 350/350 [00…
Upload file logs/train/events.out.tfevents.1654142985.8b5c72b9c6a7.83.11.v2:   0%|          | 3.34k/2.55M [00:…
```

```
model.save_pretrained("/content/drive/MyDrive/Translated/Model and Weight/model_trvlk")
```

```
!zip -r qa_model_save.zip qa_model_save/
```

```
#dont use it
model.save('/content/drive/MyDrive/Translated/new_model')
# assign location
path='/content/drive/MyDrive/Translated/weight_new/Weights'

# saveweights
model.save_weights(path)
```

```
WARNING:absl:Found untraced functions such as embeddings_layer_call_fn, embeddings_layer_call_and_return_conditional_losses, encoder_layer
coder_layer_call_and_return_conditional_losses, LayerNorm_layer_call_fn while saving (showing 5 of 416). These functions will not be direc
after loading.
INFO:tensorflow:Assets written to: /content/drive/MyDrive/Translated/Model_trainindo/assets
INFO:tensorflow:Assets written to: /content/drive/MyDrive/Translated/Model_trainindo/assets
```

```
batch = next(iter(validation_set))
output = model.predict_on_batch(batch)
output.keys()
```

```
odict_keys(['loss', 'start_logits', 'end_logits'])
```

```
output.start_logits.shape, output.end_logits.shape
```

```
((16, 384), (16, 384))
```

```
import numpy as np

np.argmax(output.start_logits, -1), np.argmax(output.end_logits, -1)
```

```
(array([ 19,  33,  78,  87, 154,  20,   7,  11, 146, 181, 108,  38,  18,
         30,  29, 109]),
 array([ 19,  40,  82,  88, 157,  20,   7,  11, 161, 162, 111,  46,  22,
         30,  29, 110]))
```

```
n_best_size = 20
```

```
import numpy as np

start_logits = output.start_logits[0]
end_logits = output.end_logits[0]
# Gather the indices the best start/end logits:
start_indexes = np.argsort(start_logits)[-1 : -n_best_size - 1 : -1].tolist()
end_indexes = np.argsort(end_logits)[-1 : -n_best_size - 1 : -1].tolist()
valid_answers = []
for start_index in start_indexes:
    for end_index in end_indexes:
        if (
            start_index <= end_index
        ):  # We need to refine that test to check the answer is inside the context
            valid_answers.append(
                {
                    "score": start_logits[start_index] + end_logits[end_index],
                    "text": "",  # We need to find a way to get back the original substring corresponding to the answer in the context
                }
            )
```

```python
def prepare_validation_features(examples):
    # Tokenize our examples with truncation and maybe padding, but keep the overflows using a stride. This results
    # in one example possible giving several features when a context is long, each of those features having a
    # context that overlaps a bit the context of the previous feature.
    tokenized_examples = tokenizer(
        examples["question" if pad_on_right else "context"],
        examples["context" if pad_on_right else "question"],
        truncation="only_second" if pad_on_right else "only_first",
        max_length=max_length,
        stride=doc_stride,
        return_overflowing_tokens=True,
        return_offsets_mapping=True,
        padding="max_length",
    )

    # Since one example might give us several features if it has a long context, we need a map from a feature to
    # its corresponding example. This key gives us just that.
    sample_mapping = tokenized_examples.pop("overflow_to_sample_mapping")

    # We keep the example_id that gave us this feature and we will store the offset mappings.
    tokenized_examples["example_id"] = []

    for i in range(len(tokenized_examples["input_ids"])):
        # Grab the sequence corresponding to that example (to know what is the context and what is the question).
        sequence_ids = tokenized_examples.sequence_ids(i)
        context_index = 1 if pad_on_right else 0

        # One example can give several spans, this is the index of the example containing this span of text.
        sample_index = sample_mapping[i]
        tokenized_examples["example_id"].append(examples["id"][sample_index])

        # Set to None the offset_mapping that are not part of the context so it's easy to determine if a token
        # position is part of the context or not.
        tokenized_examples["offset_mapping"][i] = [
            (o if sequence_ids[k] == context_index else None)
            for k, o in enumerate(tokenized_examples["offset_mapping"][i])
        ]

    return tokenized_examples
```

```python
validation_features = datasets["validation"].map(
    prepare_validation_features,
    batched=True,
    remove_columns=datasets["validation"].column_names,
)
```

```
  0%|          | 0/12 [00:00<?, ?ba/s]
```

```python
validation_dataset = validation_features.to_tf_dataset(
    columns=["attention_mask", "input_ids"],
    shuffle=False,
    batch_size=batch_size,
    collate_fn=data_collator,
)
```

```python
raw_predictions = model.predict(validation_dataset)
```

```python
max_answer_length = 30
```

```python
start_logits = output.start_logits[0]
end_logits = output.end_logits[0]
offset_mapping = validation_features[0]["offset_mapping"]
# The first feature comes from the first example. For the more general case, we will need to be match the example_id to
# an example index
context = datasets["validation"][0]["context"]

# Gather the indices the best start/end logits:
start_indexes = np.argsort(start_logits)[-1 : -n_best_size - 1 : -1].tolist()
end_indexes = np.argsort(end_logits)[-1 : -n_best_size - 1 : -1].tolist()
valid_answers = []
for start_index in start_indexes:
    for end_index in end_indexes:
        # Don't consider out-of-scope answers, either because the indices are out of bounds or correspond
        # to part of the input_ids that are not in the context.
        if (
            start_index >= len(offset_mapping)
            or end_index >= len(offset_mapping)
            or offset_mapping[start_index] is None
            or offset_mapping[end_index] is None
        ):
            continue
        # Don't consider answers with a length that is either < 0 or > max_answer_length.
        if end_index < start_index or end_index - start_index + 1 > max_answer_length:
            continue
        if (
```

```
datasets["validation"][0]["answers"]
```

```
{'answer_start': [159, 159, 159, 159],
 'text': ['Perancis', 'Perancis', 'Perancis', 'Perancis']}
```

```
model.save('/content/drive/MyDrive/Translated/latest_model')

# assign location
path='/content/drive/MyDrive/Translated/Weight_latest/Weights'

# saveweights
model.save_weights(path)
```

```
WARNING:absl:Found untraced functions such as embeddings_layer_call_fn, embeddings_layer_call_and_return_conditional_losses, encoder_layer_call_fn, encoder_layer_call_and_return_conditional_losses, LayerNorm_layer_call_fn while saving (showing 5 of 416). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: /content/drive/MyDrive/Translated/latest_model/assets
INFO:tensorflow:Assets written to: /content/drive/MyDrive/Translated/latest_model/assets
```

```
tf.keras.models.load_model('')
```

```
# python
from transformers import TFAutoModel
# bert = TFAutoModel.from_pretrained("bert-base-uncased")
bert = TFAutoModel.from_pretrained("/content/drive/MyDrive/Translated/Model and Weight/latest_model")
```

Lampiran D. Interim Transcript