



DIGITAL  
TALENT  
SCHOLARSHIP



# THEMATIC ACADEMY

Tema Pelatihan

Pertemuan #16- : Melakukan Deployment Model



KOMINFO



#JADIJAGOANDIGITAL

Badan Penelitian dan Pengembangan Sumber Daya Manusia

# Course Definition

- Melakukan Deployment Model
- Unit kompetensi:
  - J.62DMI00.016.1 - Membuat Rencana Deployment Model
    - Melakukan strategi deployment
    - Menyusun instruksi deployment
  - J.62DMI00.017.1 - Melakukan Deployment Model
    - Melakukan langkah deployment sesuai instruksi
    - Membuat laporan hasil deployment

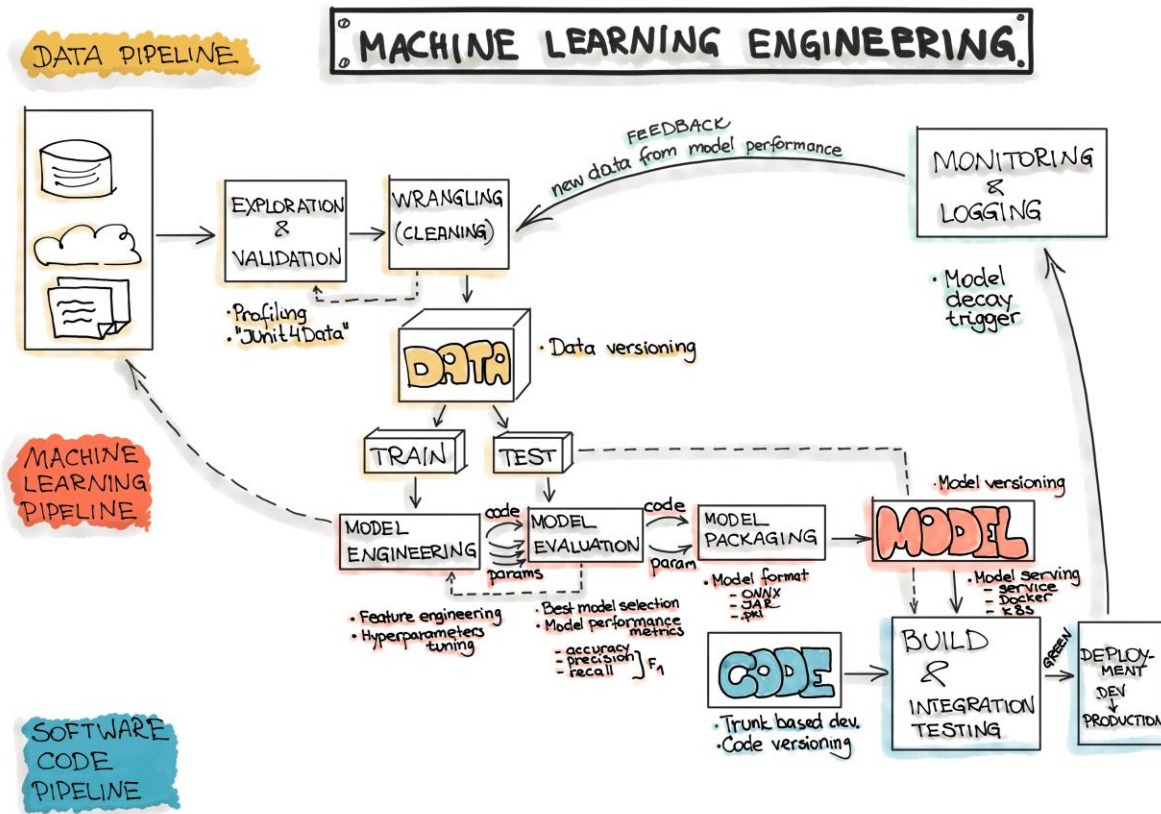
# Learning Objective

Peserta mampu melakukan deployment model menjadi sistem yang dapat dioperasikan (website)

## Course subtopic

- Pendahuluan
- Penerapan model berbasis pola
- Strategi deployment model
- Deployment model (Regresi Linier)
- Deployment model (ANN)

# Pendahuluan(1/6)

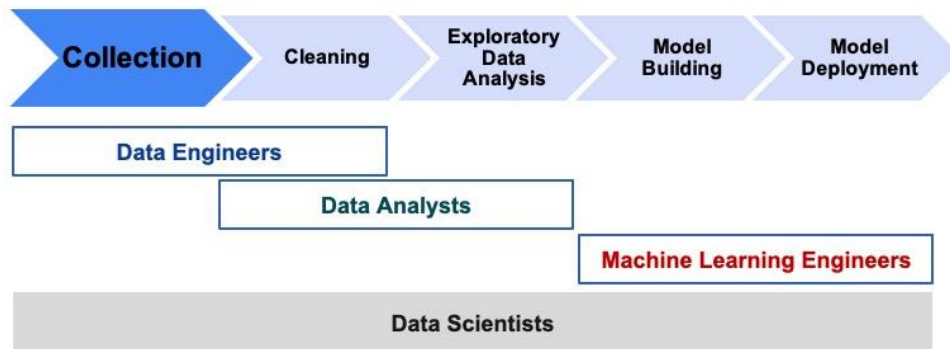
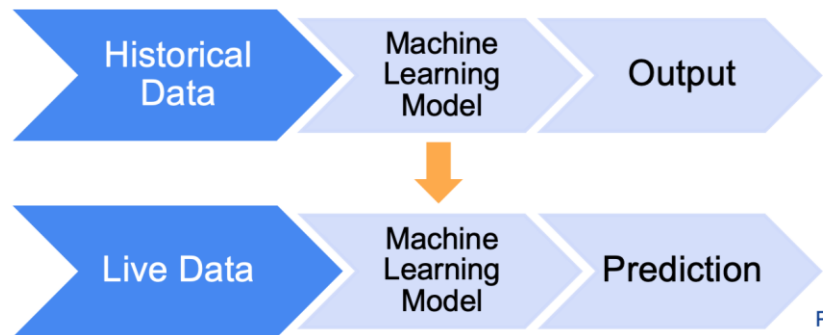
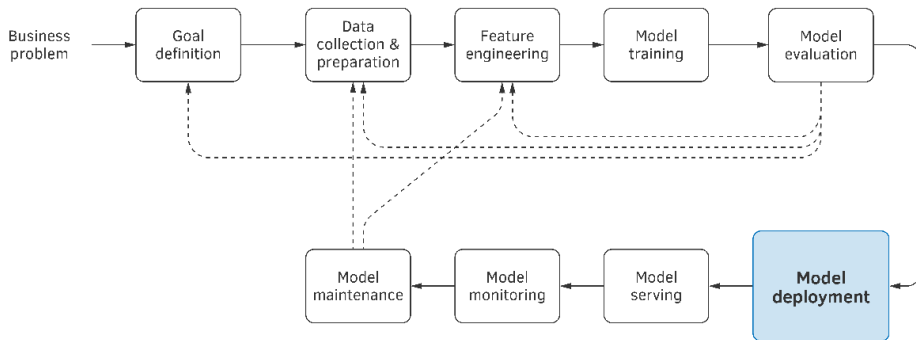


Source: Machine Learning Engineering

## Pendahuluan(2/6)

- **Deployment model** adalah suatu proses untuk membuat **model** (model machine learning) tersebut **tersedia** pada **lingkungan produksi**, dimana model tersebut **dapat memberikan prediksi ke sistem** perangkat lunak yang **lain**.
- Deployment model merupakan **tahapan terakhir** pada Machine Learning lifecycle dan merupakan tahapan yang paling menantang.

# Pendahuluan(3/6)



Source: Machine Learning Engineering

## Pendahuluan(4/6)

Deployment model **sangat penting:**

- Untuk menggunakan model Machine Learning, **penerapan secara efektif** model tersebut ke dalam lingkungan produksi sangat diperlukan. Hal ini ditujukan agar model tersebut dapat memberikan prediksi ke sistem perangkat lunak yang lain.
- Mengekstraksi prediksi dengan handal dan membagikan ke dalam sistem yang lain sangat penting dilakukan untuk **memaksimalkan nilai model** machine learning yang telah dibuat.

# Pendahuluan(5/6)

**Tantangan** pada implementasi Deployment model:

- Tantangan pada perangkat lunak biasa:
  - Reliability
  - Reusability
  - Maintainability
  - Flexibility
- Tantangan spesifik pada Machine Learning:
  - Reproducibility



# Pendahuluan(6/6)

**Tantangan** pada implementasi Deployment model:

- Memerlukan koordinasi antara Data Scientists, IT Teams, software developers, dan profesional bisnis:
  - Memastikan model bekerja secara reliable
  - Memastikan model mendeliver keluaran yang dimaksudkan.
- Potensi perbedaan antara bahasa pemrograman yang digunakan dalam mengembangkan model dengan bahasa yang digunakan pada sistem produksi.
  - Melakukan coding ulang akan memperpanjang durasi pengerjaan proyek dan mengurangi reproduktifitas.

# Penerapan model berbasis pola

Suatu model dapat dideploy berdasarkan beberapa pola:

- Secara statistik/static deployment
- Secara dinamis pada perangkat pengguna (Dynamic deployment on user's devices)
- Secara dinamis pada server (Dynamic deployment on a server)

# Penerapan model berbasis pola: Static Deployment(1/1)

Static Deployment mirip dengan deployment perangkat lunak seperti biasa.

- File binary yang dapat diinstall dari seluruh perangkat lunak.
- Model dikemas sebagai resources yang tersedia saat runtime.
- Dll files (windows), \*.so (linux), Java dan .Net.

Keuntungan static deployment:

- Eksekusi lebih cepat → perangkat lunak memiliki akses langsung ke model.
- Efisien waktu dan privacy → data pengguna tidak harus diupload di server.
- Model dapat dipanggil saat pengguna offline.
- Pengoperasian model menjadi tanggung jawab pengguna, bukan vendor perangkat lunak.

# Penerapan model berbasis pola: Dynamic Deployment(1/21)

## Dynamic deployment on user's devices

- Dynamic deployment **pada perangkat user** mirip dengan static deployment, perbedaannya adalah model bukan merupakan bagian dari binary code aplikasi. Pembaruan model dapat dilakukan tanpa harus memperbarui seluruh aplikasi yang berjalan pada perangkat pengguna.
- Dynamic deployment **pada perangkat user** dapat dilakukan dengan:
  - Deploying model parameters
  - Deploying a serialized object
  - Deploying ke web browser

# Penerapan model berbasis pola: Dynamic Deployment(2/21)

## Deploying model parameters

- Pada scenario ini, file model hanya berisi parameter yang dipelajari.
- Perangkat pengguna telah menginstall runtime environment untuk model tersebut.
- Dapat menggunakan beberapa versi ringan dari machine learning package sehingga dapat berjalan pada mobile devices seperti
  - TensorFlow,
  - Apple's Core ML
  - Scikit-learn, Keras, XGBoost

# Penerapan model berbasis pola: Dynamic Deployment(3/21)

## Deploying model parameters

- Contoh machine learning package untuk mobile devices:



# Penerapan model berbasis pola: Dynamic Deployment(4/21)

## Deploying of serialized object

- Pada scenario ini, file model adalah object serial yang akan di-deserialize oleh aplikasi.
- Keuntungan: tidak memerlukan runtime environment pada perangkat pengguna.
- Kelemahan: update system akan cukup berat → merupakan masalah jika perangkat lunak kita memiliki jutaan pelanggan.

# Penerapan model berbasis pola: Dynamic Deployment(5/21)

## Deploying to web browser

- Pada skenario ini, akses ke browser banyak digunakan oleh aplikasi desktop maupun seluler.
- Deploying to web browser berarti model dapat dilatih dan berjalan pada browser. Contoh: framework TensorFlow.js
- Skenario lain: Model TensorFlow ditraining menggunakan python kemudian dideploy dan dijalankan di browser dengan runtime environment JavaScript.
- GPU (Graphic Processing Unit) dapat digunakan juga oleh TensorFlow.js



# Penerapan model berbasis pola: Dynamic Deployment(6/21)

## Kelebihan dan kekurangan

- Kelebihan dynamic deployment pada user devices:
  - Proses model lebih cepat, komputasi sebagian di perangkat pengguna
  - Jika diimplementasikan ke browser, pengorganisasian infrastruktur hanya perlu menyajikan halaman web yang menyertakan parameter model
  - Model sangat mudah tersedia untuk analisis pihak ketiga
- Kekurangan dynamic deployment pada user devices:
  - Biaya bandwidth meningkat (jika berbasis browser)
  - Updating model (jika serial)
  - Performansi model susah dimonitor

# Penerapan model berbasis pola: Dynamic Deployment(7/21)

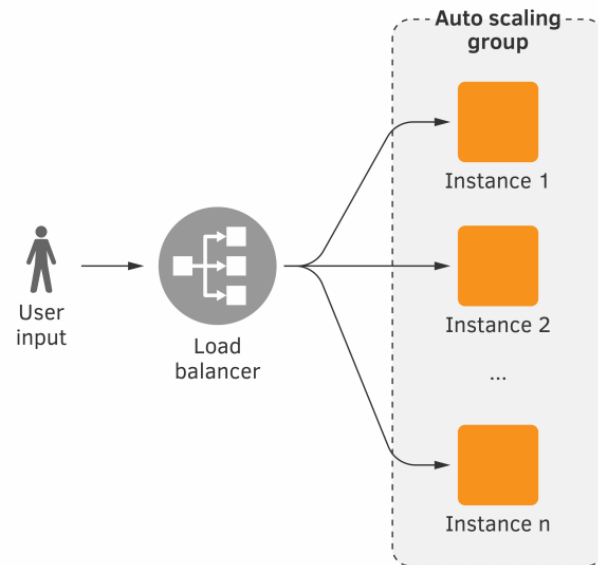
## Dynamic deployment on a server

- Dynamic deployment **pada server** merupakan salah satu **solusi** untuk mengatasi kekurangan pada dynamic deployment **pada perangkat user**.
- Model ditempatkan pada server, tersedia untuk antarmuka REST API atau gRPC Google.
- Dynamic deployment **pada server** dapat dilakukan dengan:
  - Deployment pada virtual machine
  - Deployment dalam container
  - Serverless deployment
  - Model streaming

# Penerapan model berbasis pola: Dynamic Deployment(8/21)

## Deployment pada virtual machine

- Pada domain web di cloud, prediksi disajikan sebagai respons HTTP.
- Ilustrasi:
  - Pengguna → Layanan website (pada Virtual Machine/VM) → Koneksi ke machine learning → Mengubah output ke dalam bentuk JavaScript Object Notation (JSON) atau XML.
  - VM berjalan secara paralel untuk mengatasi beban komputasi yang tinggi.



- Setiap instance berisi codes untuk menjalankan ekstraktor fitur dan model
- Layanan web yang memiliki akses code tersebut
- Dalam python, layanan web REST API biasanya diimplementasikan menggunakan Flask atau FastAPI

# Penerapan model berbasis pola: Dynamic Deployment(9/21)

## Deployment pada virtual machine

- Tensorflow → TensorFlow Serving (layanan gRPC bawaan).
- Keuntungan implementasi pada VM:
  - Arsitektur sistem perangkat lunak secara konseptual sederhana.
- Kekurangan implementasi pada VM:
  - Pemeliharaan server (fisik maupun virtual).
  - Latensi jaringan
  - Biaya relative tinggi dibanding dengan menggunakan container atau serverless.

# Penerapan model berbasis pola: Dynamic Deployment(10/21)

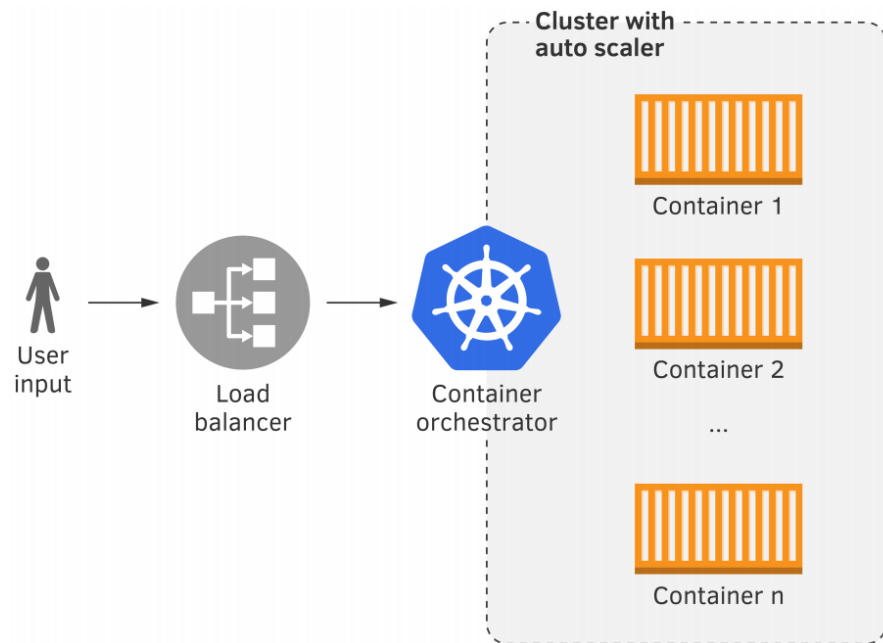
## Deployment pada container

- Container mirip dengan VM
  - Memiliki runtime yang terisolasi dengan sistem file, CPU, Memory, dan ruang prosesnya sendiri.
- Container merupakan suatu alternatif modern dibanding dengan deployment pada VM.
  - Semua container berjalan pada VM atau fisik yang sama dan berbagi sistem operasi.
  - VM Menjalankan sistem operasi sendiri.
- **Keuntungan** deployment pada container:
  - Lebih hemat resources dan fleksibel dibanding menggunakan VM
- **Kelemahan** deployment pada container:
  - Deployment pada container sering dianggap kompleks dan membutuhkan expert/tenaga ahli

# Penerapan model berbasis pola: Dynamic Deployment(11/21)

## Deployment pada container

- Sistem machine learning dan web services diinstall pada container (**Docker Container**).
- **Container orchestrator** digunakan untuk menjalankan container pada sekelompok server fisik/virtual.
- **Contoh container orchestrator:** Kubernetes, AWS fargate, Google Kubernetes Engine.



Deploying suatu model sebagai web service pada container yang berjalan pada cluster

# Penerapan model berbasis pola: Dynamic Deployment(12/21)

## Deployment pada container

- **Docker** adalah aplikasi open source untuk menyatukan file-file yang dibutuhkan suatu perangkat lunak sehingga menjadi satu kesatuan yang lengkap dan berfungsi.
- Data pengaturan dan file pendukung disebut sebagai **image**.
- Kumpulan image digabung menjadi suatu wadah yang disebut **Container**.



# Penerapan model berbasis pola: Dynamic Deployment(13/21)

## Deployment pada container

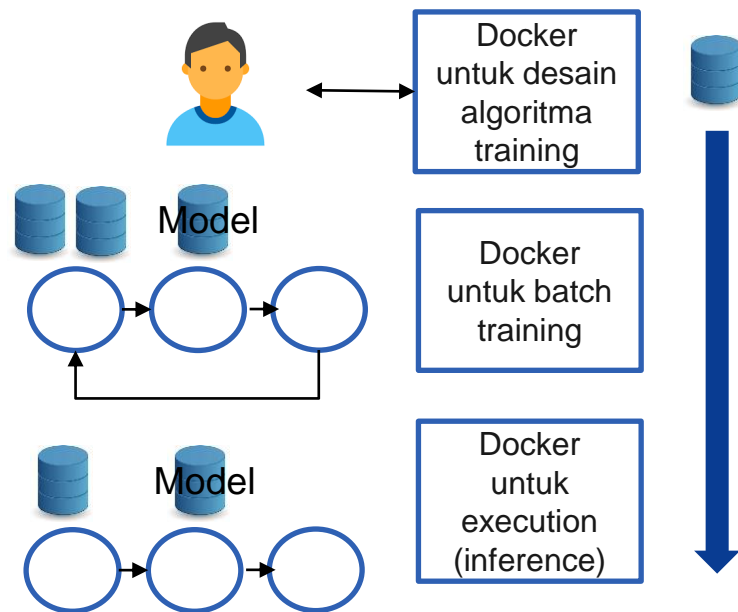
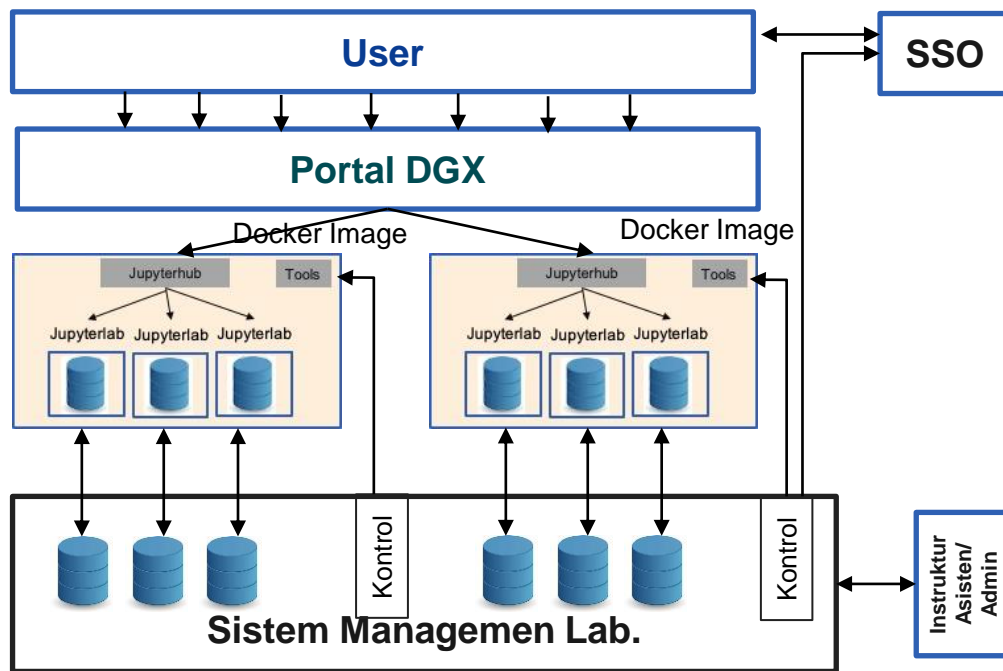
- **Fitur Docker:**
  - **Docker Engine**, digunakan untuk membangun Docker Images dan membuat Container Docker
  - **Docker Hub**, registry yang digunakan untuk berbagai macam Docker Image.
  - **Docker Compose**, digunakan untuk mendefinisikan aplikasi menggunakan banyak Container Docker.
  - **Docker Mac, Docker Linux, Docker Windows**





# Penerapan model berbasis pola: Dynamic Deployment(14/21)

## Deployment pada container



# Penerapan model berbasis pola: Dynamic Deployment(15/21)

## Serverless deployment

- Serverless deployment merupakan deployment yang memanfaatkan serverless computing dari cloud services providers
  - Amazon (lambda functions pada AWS),
  - Google (Google cloud platform) dan
  - Microsoft (functions pada Microsoft Azure).
- Serverless deployment terdiri dari:
  - Arsip ZIP yang berisi codes untuk menjalankan machine learning (model, feature extractor, dan scoring code).
  - File arsip ZIP berisi: nama tertentu yang berisi fungsi tertentu, atau class-method dengan spesifik signature.
  - File arsip ZIP diupload pada cloud platform dan dilakukan registrasi dengan nama yang unik.

# Penerapan model berbasis pola: Dynamic Deployment(16/21)

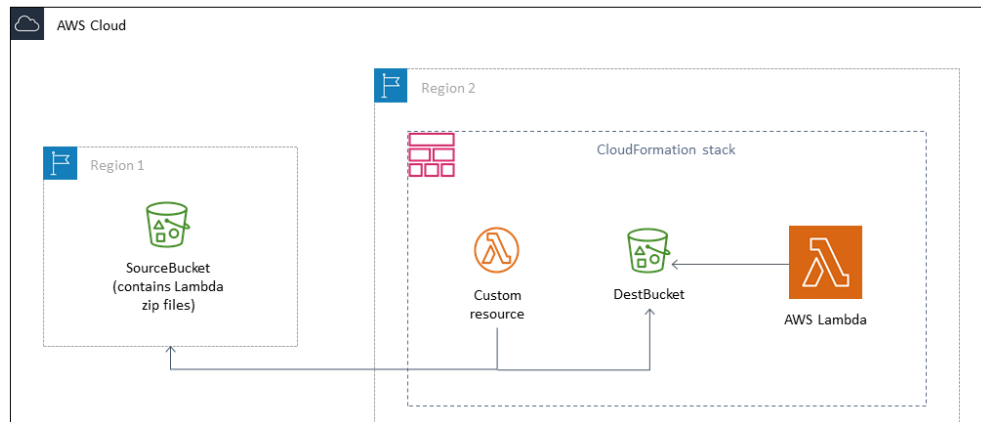
## Serverless deployment

- Cloud services:
  - Menyediakan API untuk mensubmit masukan ke Serverless Function.
  - Menangani deploying codes dan model agar memadai pada sumber daya yang dimiliki.
  - Mengeksekusi codes dan mengarahkan keluaran kembali ke client.
  - Limitasi pada waktu eksekusi fungsi, ukuran file ZIP, dan jumlah RAM
    - Harus menyertakan Python Library agar model dapat dieksekusi dengan benar: Numpy, SClpy, Scikit-Learn.
    - Bergantung pada platform cloud: Java, Go, PowerShell, Node.js, C#, Ruby

# Penerapan model berbasis pola: Dynamic Deployment(17/21)

## Serverless deployment

### AWS Lambda (Amazon)



Upload your code to AWS Lambda or write code in Lambda's code editor



Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity



**AWS Lambda**

Lambda runs your code only when triggered, using only the compute resources needed

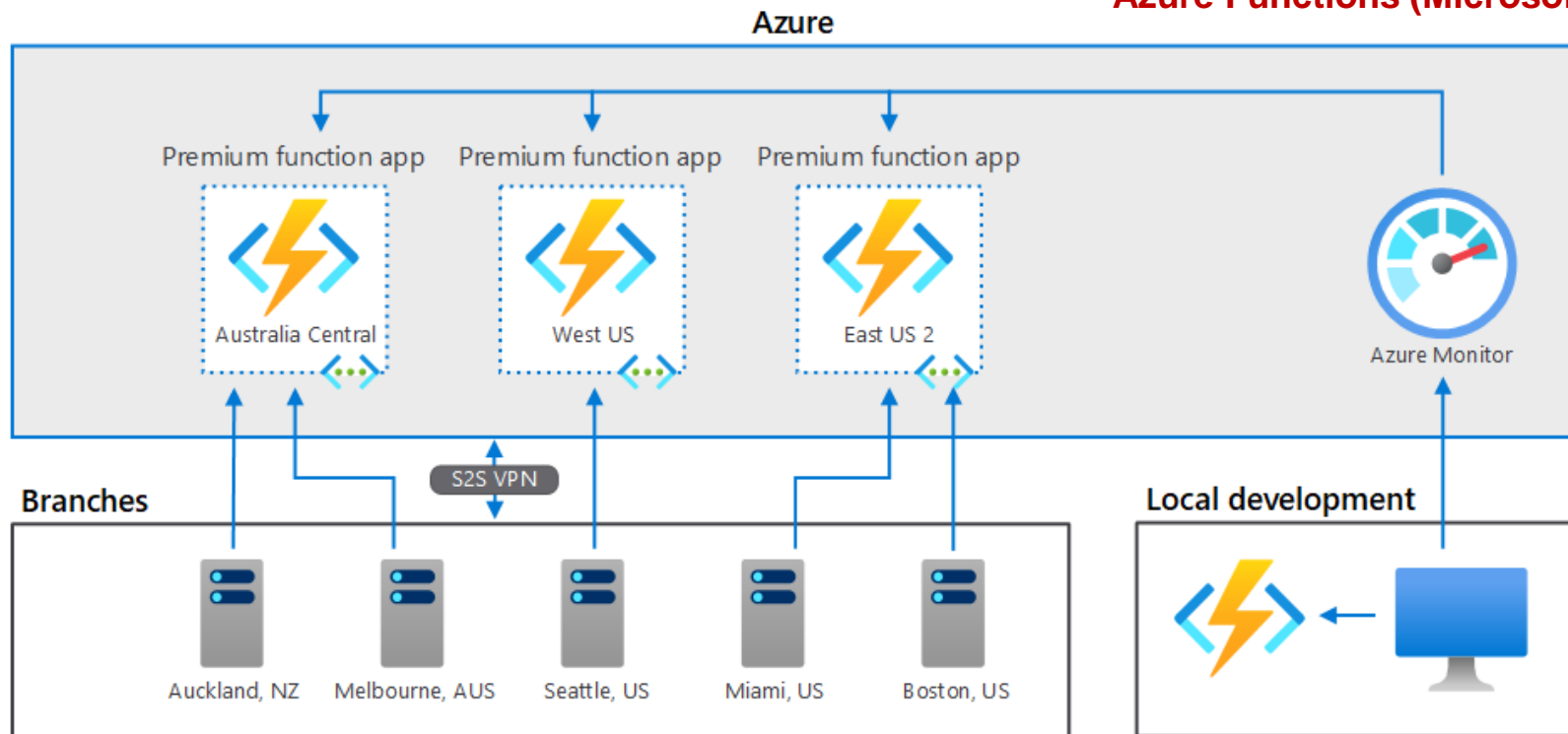


Just pay for the compute time you use

# Penerapan model berbasis pola: Dynamic Deployment(18/21)

## Serverless deployment

Azure Functions (Microsoft)

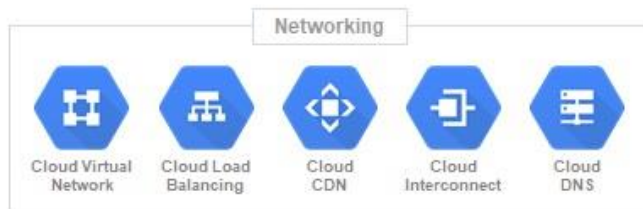


# Penerapan model berbasis pola: Dynamic Deployment(19/21)

## Serverless deployment

Google Cloud Platform  
(Google)

Google Cloud Platform Icons



# Penerapan model berbasis pola: Dynamic Deployment(20/21)

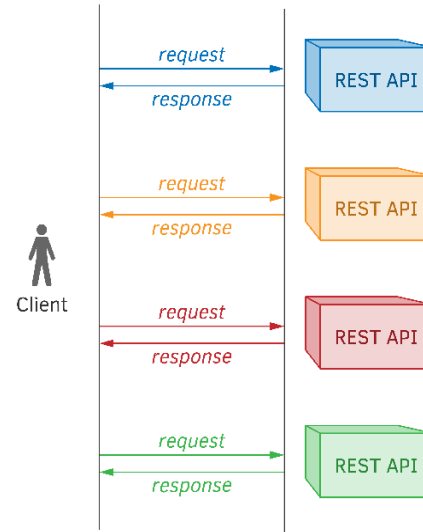
## Serverless deployment

- Kelebihan serverless deployment:
  - Tidak perlu menyediakan server maupun VM.
  - Tidak perlu install depedensi, maintenance atau update system.
  - Bersifat scalable.
  - Hemat biaya: hanya membayar waktu komputasi.
  - Mendukung operasi sinkron dan asinkron
  - Rollback yang mudah untuk kembali ke versi sebelumnya.
- Kekurangan serverless deployment:
  - Limitasi pada waktu eksekusi fungsi, ukuran file ZIP, dan jumlah RAM
  - Tidak tersedianya akses GPU: terbatas jika ingin mendeploy Deep Model.

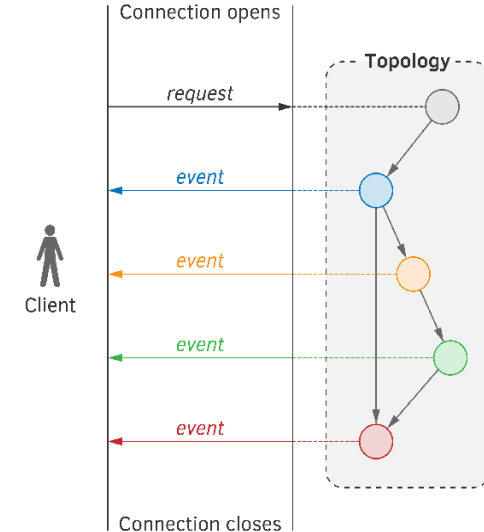
# Penerapan model berbasis pola: Dynamic Deployment(21/21)

## Model streaming

- Merupakan kebalikan dari REST API.
- Semua model dan codes didaftarkan pada mesin pemrosesan aliran (SPE)
  - Apache Storm, Apache Spark, Apache Flink
  - Aplikasi berbasis stream-processing library (SPL): Apache Samza, Apache Kafka Streams, Akka Streams.



<REST API>



<Streaming>

REST API: Memproses elemen data, client menggunakan REST API mengirimkan permintaan satu-per-satu dan menerima respon secara sinkronus

STREAMING: Memproses elemen data, client menggunakan STREAMING dengan membuka koneksi, mengirimkan permintaan, dan mendapatkan update event ketika hal tersebut terjadi



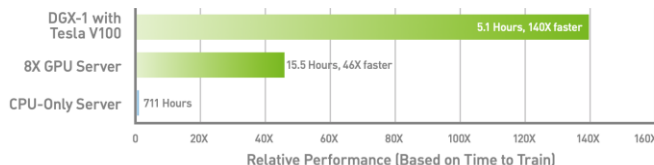
# Penerapan model berbasis pola: AI infrastructure, platform (1/5)



## ITERATE AND INNOVATE FASTER UNPARALLELED DEEP LEARNING TRAINING PERFORMANCE

(NVIDIA)

### NVIDIA DGX-1 Delivers 140X Faster Deep Learning Training



Workload: ResNet-50, 90 epochs to solution | CPU Server: Dual Xeon E5-2699v4, 2.6GHz

### SYSTEM SPECIFICATIONS

	NVIDIA DGX A100 640GB	NVIDIA DGX A100 320GB
GPUs	8x NVIDIA A100 80 GB GPUs	8x NVIDIA A100 40 GB GPUs
GPU Memory	640 GB total	320 GB total
Performance	5 petaFLOPS AI 10 petaOPS INT8	
NVIDIA NVSwitches	6	
System Power Usage	6.5 kW max	
CPU	Dual AMD Rome 7742, 128 cores total, 2.25 GHz (base), 3.4 GHz (max boost)	

	2 TB	1 TB
System Memory		
Networking	8x Single-Port Mellanox ConnectX-6 VPI 200Gb/s HDR InfiniBand 2x Dual-Port Mellanox ConnectX-6 VPI 10/25/50/100/200 Gb/s Ethernet	8x Single-Port Mellanox ConnectX-6 VPI 200Gb/s HDR InfiniBand 1x Dual-Port Mellanox ConnectX-6 VPI 10/25/50/100/200 Gb/s Ethernet
Storage	OS: 2x 1.92 TB M.2 NVMe drives Internal Storage: 30 TB (8x 3.84 TB) U.2 NVMe drives	OS: 2x 1.92 TB M.2 NVMe drives Internal Storage: 15 TB (4x 3.84 TB) U.2 NVMe drives

# Penerapan model berbasis pola: AI infrastructure, platform (2/5)

## Atlas AI Computing Platform Portfolio

(Huawei)



Superior computing power



All-scenario deployment



Cloud-edge-device collaboration



Atlas 900 AI cluster



Atlas 800 AI server  
Model: 9000/9010



Atlas 800 AI server  
Model: 3000/3010



Atlas 300 AI accelerator card  
Model: 3000/9000



Atlas 500 AI  
edge station



Atlas 200 AI  
accelerator module

-----Cloud-----Edge-----Device-----



Ascend 310  
AI processor

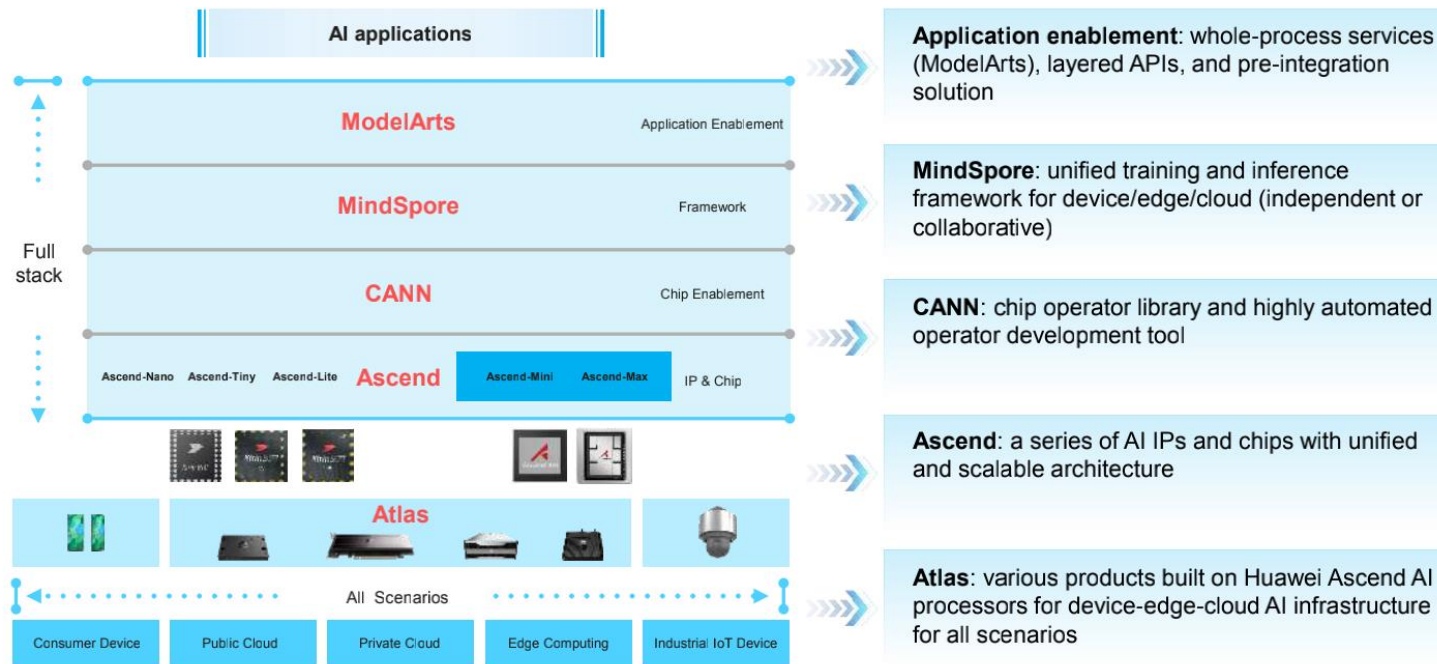


Ascend 910  
AI processor

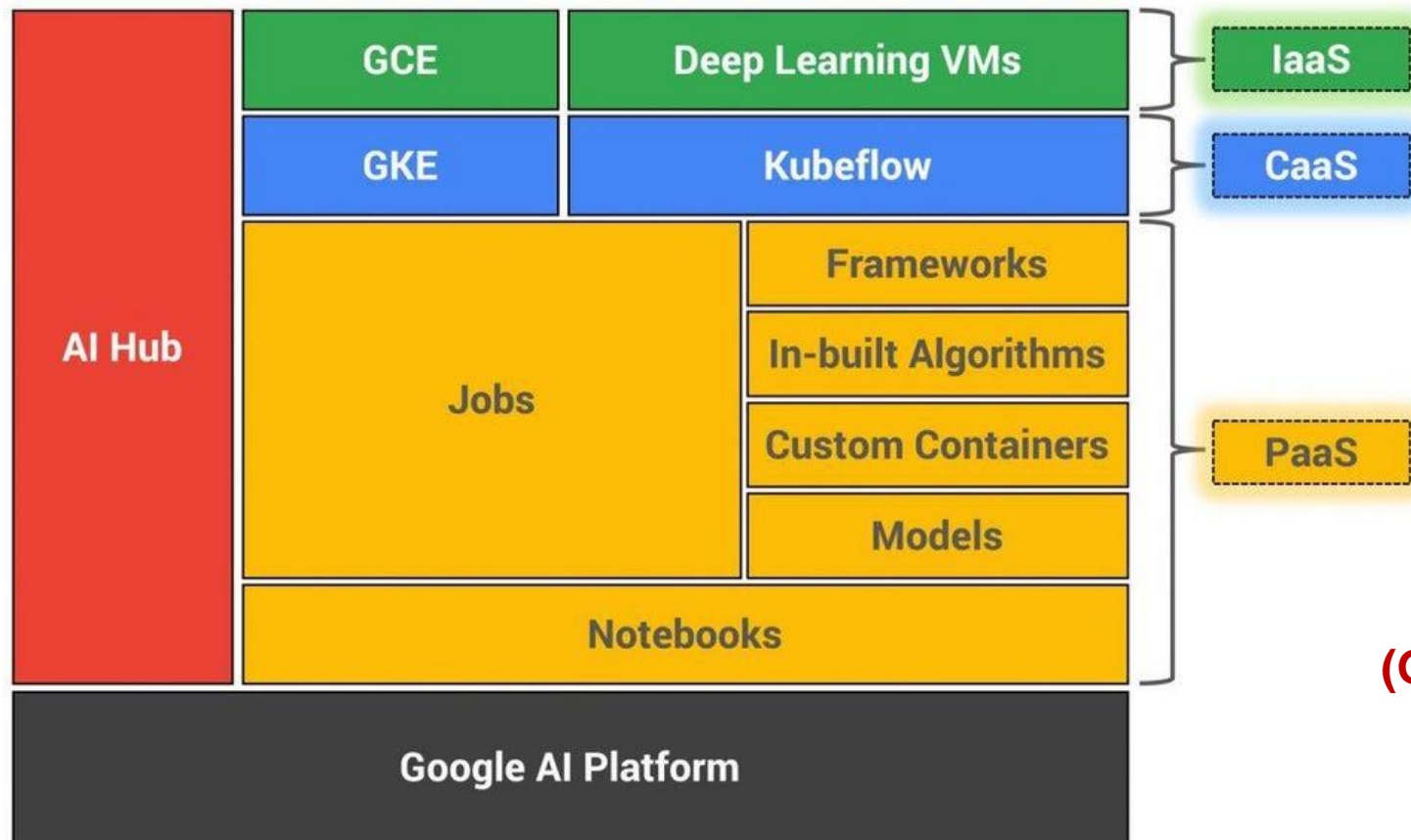
# Penerapan model berbasis pola: AI infrastructure, platform (3/5)

## Huawei's Full-Stack, All-Scenario AI Solution

(Huawei)

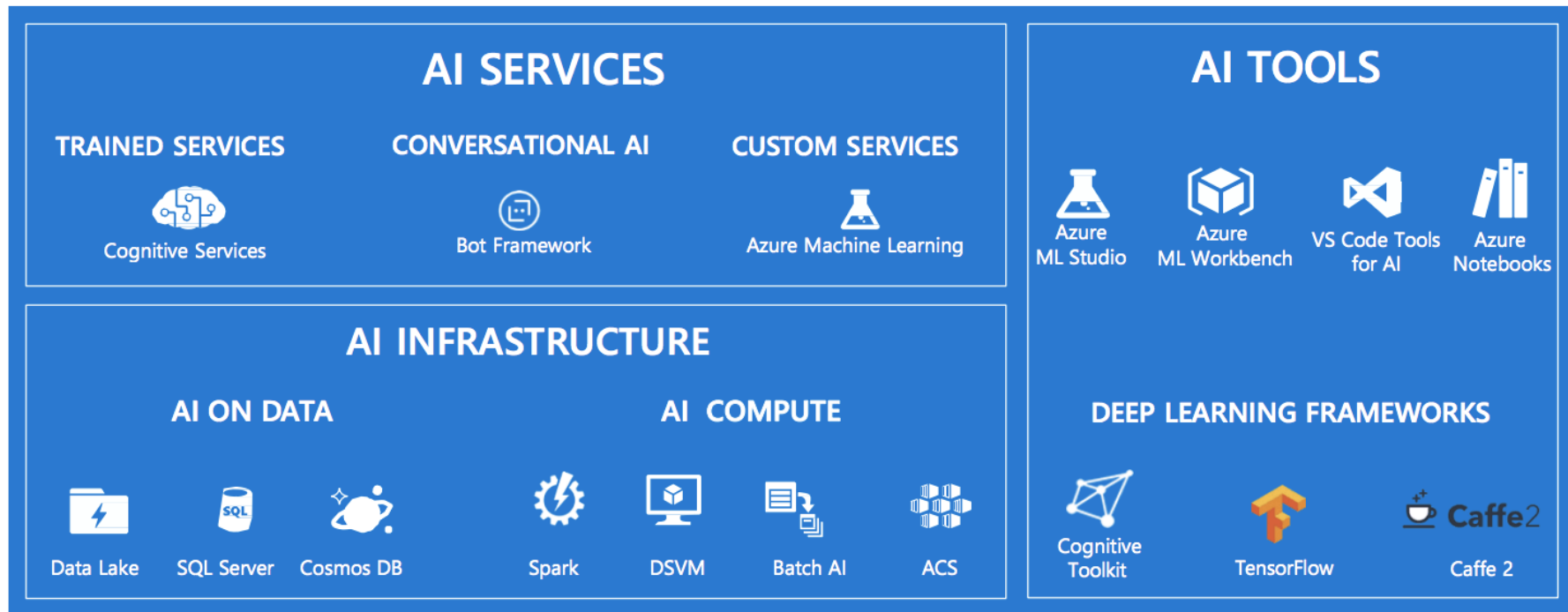


## Penerapan model berbasis pola: AI infrastructure, platform (4/5)



(Google)

# Penerapan model berbasis pola: AI infrastructure, platform (5/5)



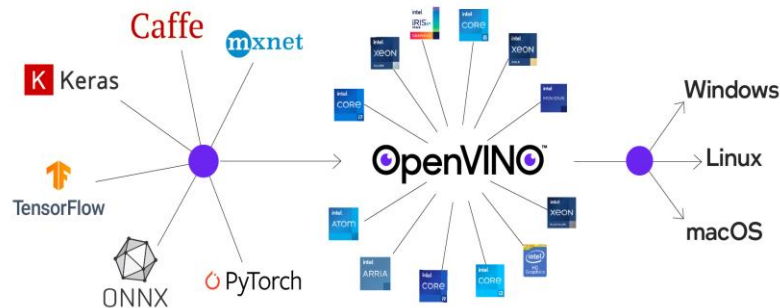
(Microsoft)

# Penerapan model berbasis pola



# Deployment strategy(1/6)

- Deployment strategy
  - Computational resources: melakukan prediksi (inference) lebih baik dibandingkan melakukan training.
  - Inferensi pada perangkat (berbasis Apps):
    - Performansi
    - Data dengan volume yang besar → bandwidth resources
    - User tidak memiliki akses internet D
  - Dapat menggunakan OPENVINO, Jetson NANO, dll (berbasis Desktop) → Robotics, Surveillance System, dll
  - Dapat menggunakan microplatform FLASK dan HEROKU (berbasis Web)
- Deployment strategy secara tipikal dapat dibagi menjadi:
  - Single Deployment
  - Silent Deployment
  - Canary Deployment





# Deployment strategy(2/6)

## Single Deployment

- Single deployment merupakan strategi yang paling sederhana.
  - Model dibuat serial menjadi file, file lama bisa direplace dengan yang baru, termasuk ekstraktor fitur jika diperlukan.
  - Keunggulan: sederhana
  - Kelemahan: Jika ada bug, maka semua pengguna akan terpengaruh
- Deployment di cloud environment:
    - VM atau container baru harus disiapkan
    - VM image atau container lama direplace
    - Autoscaler memulai dengan yang baru
  - Deployment di server:
    - Upload new model file pada server
    - Replace yang lama dengan yang baru
    - Lakukan restart web service



# Deployment strategy(3/6)

## Silent Deployment

- Merupakan strategi dengan menjalankan versi model yang baru dan lama secara paralel.
- Keunggulan:
  - Menyediakan waktu yang cukup untuk memastikan model baru dapat digunakan sesuai kriteria
- Kelemahan:
  - Menghabiskan banyak resource karena menjalankan dua model secara bersama-sama

# Deployment strategy(4/6)

## Canary Deployment

- Merupakan strategi deployment dengan cara mendorong versi dan kode model baru ke sebagian kecil pengguna, dengan tetap menjalankan versi lama untuk sebagian besar pengguna
- Keunggulan:
  - Memungkinkan untuk memvalidasi model baru dan efek prediksinya.
  - Deployment tidak mempengaruhi banyak user jika ada bug
- Kelemahan:
  - Lebih rumit dan kompleks

## Deployment strategy(5/6)

- **Reproducibility** adalah akuntabilitas yang diperlukan dalam bisnis untuk lebih memahami dan mempercayai penerapan machine learning pada kehidupan sehari-hari.
- **Reproducibility pada machine learning** yaitu suatu kemampuan untuk menduplikasi model secara tepat sehingga ketika model dilewati dengan data input yang sama, model yang direproduksi akan mengembalikan output yang sama.
- **Tantangan pada proses Reproducibility ketika Deployment Model:**
  - Feature tidak terdapat pada Live Environment
  - Perbedaan bahasa pemrograman
  - Perbedaan perangkat lunak
  - Kondisi real tidak cocok dengan data dan kondisi pada saat training model

## Deployment strategy(6/6)

### **Solusi** pada proses Reproducibility ketika Deployment Model:

- Versi perangkat lunak harus sama persis dan aplikasi harus mencantumkan semua dependensi library pihak ketiga beserta versinya
- Menggunakan container dan perangkat lunak untuk melacak spesifikasi
- Research, develop dan deploy menggunakan bahasa yang sama (contoh: Python)
- Sebelum membangun model, harus memahami terlebih dahulu bagaimana model akan diintegrasikan dengan sistem lain.

# Deployment Model: Cara menyimpan model

```
H=model.fit(trainX, trainY,validation_data=(testX, testY),  
batch_size=b, epochs=e, shuffle=True )
```

```
regressor.fit(X,y)
```

```
from keras.models import load_model
```

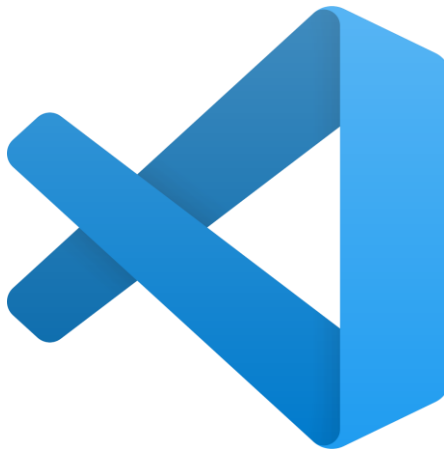
```
model.save('model.h5')
```

```
import pickle
```

```
pickle.dump(regressor, open('model.pkl','wb'))
```

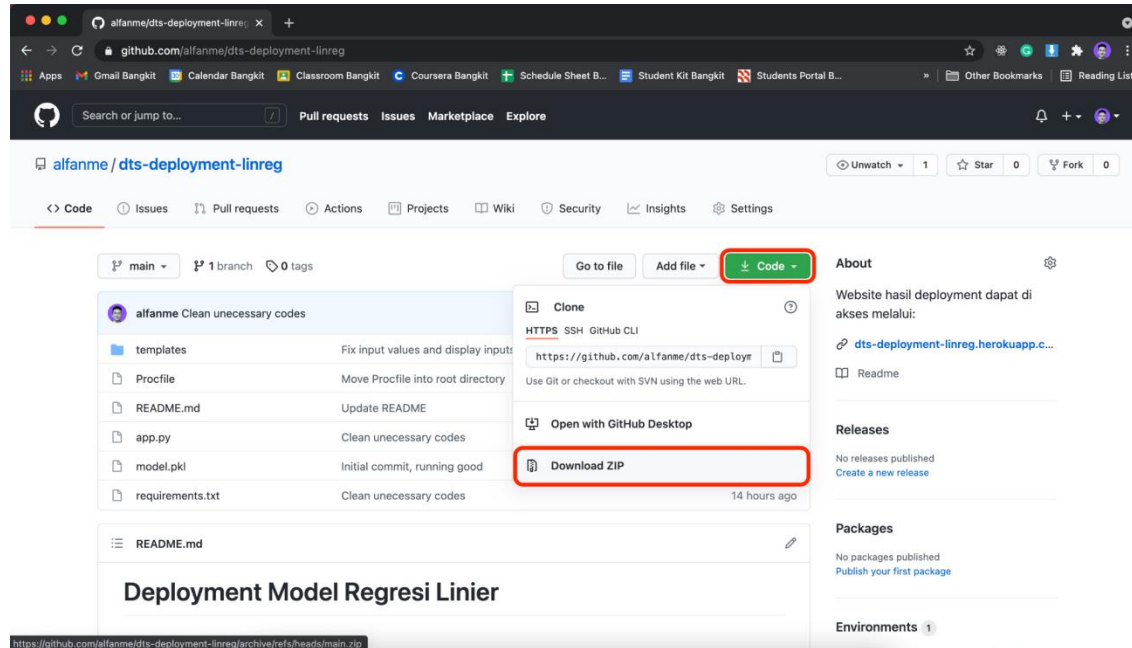
# Deployment Model: Langkah-Langkah model deployment (inferencing)

0. Download dan lakukan instalasi perangkat lunak di bawah ini



# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (1/15)

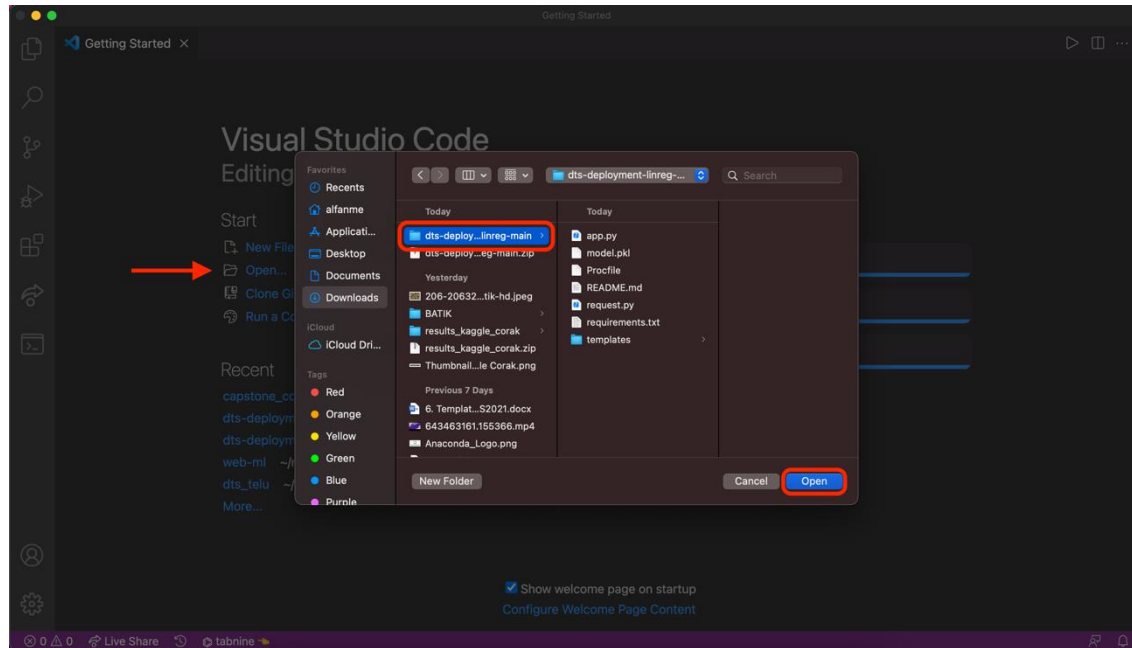
## 1. Download source code dari repository Github



# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (2/15)

2. Buka dan extract file ZIP yang sudah di-download.

3. Buka folder source code yang sudah di-extract menggunakan VSCode dengan cara klik Open... pada halaman awal VSCode, pilih folder, lalu tekan tombol Open.

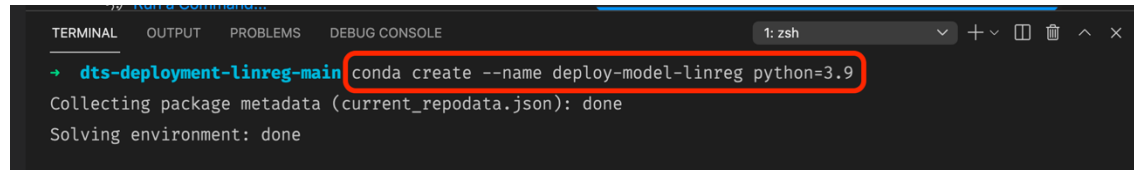




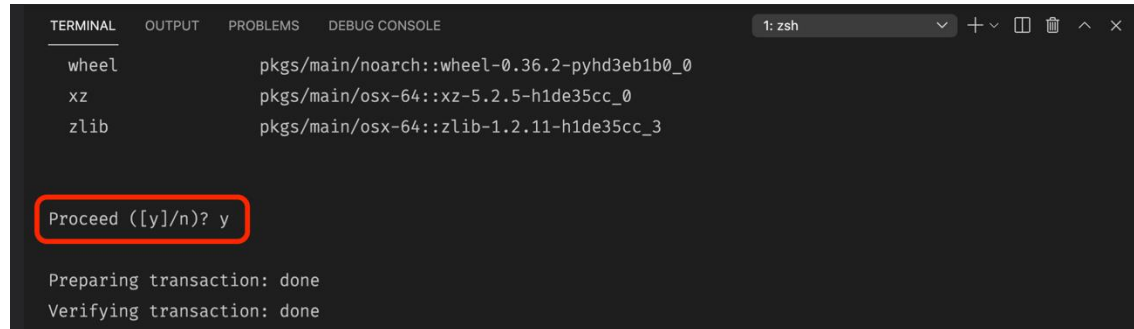
# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (3/15)

4. Buka terminal pada VSCode menggunakan Ctrl+J.

5. Pada terminal, ketikkan ***conda create --name deploy-model-linreg python=3.9*** lalu tekan Enter, Anda bebas memberi nama apa saja selain “***deploy-model-linreg***”, ketikkan ‘y’ lalu Enter jika diberikan pertanyaan, tunggu proses pembuatan virtual environment selesai.



```
→ dts-deployment-linreg-main conda create --name deploy-model-linreg python=3.9
Collecting package metadata (current_repodata.json): done
Solving environment: done
```



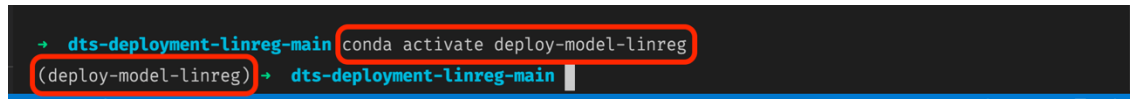
```
TERMINAL OUTPUT PROBLEMS DEBUG CONSOLE 1: zsh
wheel pkgs/main/noarch::wheel-0.36.2-pyhd3eb1b0_0
xz pkgs/main/osx-64::xz-5.2.5-h1de35cc_0
zlib pkgs/main/osx-64::zlib-1.2.11-h1de35cc_3

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
```

## Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (4/15)

6. Aktifkan virtual environment menggunakan perintah **conda activate deploy-model-linreg** lalu tekan Enter, tunggu sampai virtual environment aktif, sekarang pada terminal akan terlihat nama virtual environment yang aktif yaitu (deploy-model-linreg).



```
→ dts-deployment-linreg-main conda activate deploy-model-linreg  
(deploy-model-linreg) → dts-deployment-linreg-main
```

7. Install semua dependencies/library yang dibutuhkan dengan perintah **pip install -r requirements.txt**, tunggu sampai selesai.

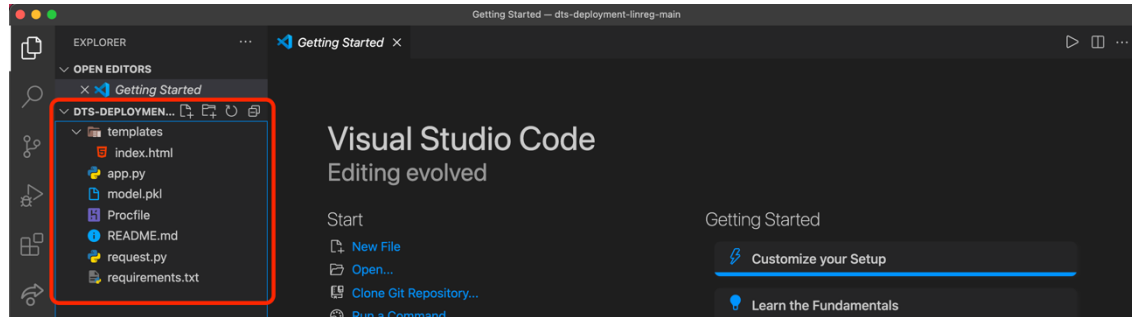


```
TERMINAL OUTPUT PROBLEMS DEBUG CONSOLE 1: zsh  
(deploy-model-linreg) → dts-deployment-linreg-main pip install -r requirements.txt  
Collecting flask  
Using cached Flask-2.0.1-py3-none-any.whl (94 kB)  
Collecting gunicorn
```

# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (5/15)

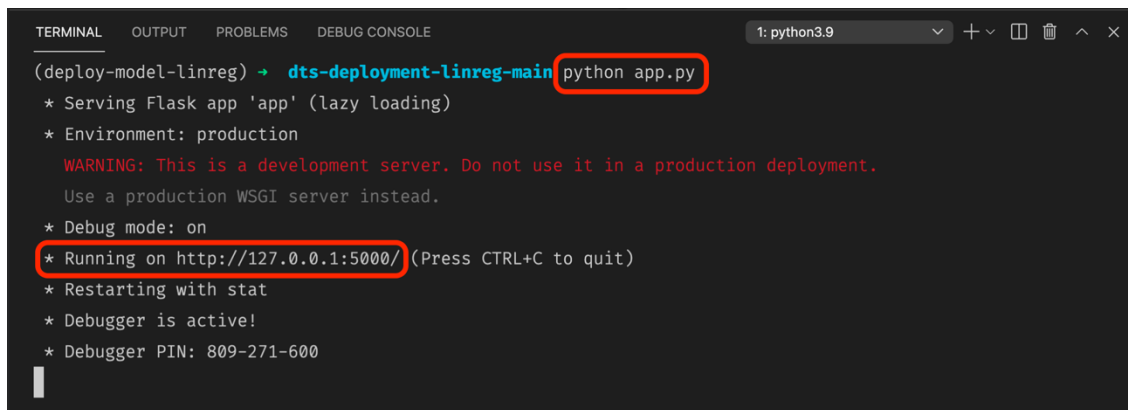
8. Setelah semua install selesai, Anda bisa menyembunyikan terminal dengan menekan Ctrl+J atau klik icon 'x' pada sisi kanan atas terminal.

9. Pada sisi kiri VSCode, terdapat struktur direktori dari folder yang dibuka sebelumnya.



## Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (6/15)

10. Sebelum melakukan deployment ke platform Heroku, Anda dapat mencoba menjalankan server websitenya secara local pada PC/laptop dengan cara membuka terminal (Ctrl+J), ketikkan perintah **python app.py** lalu tekan Enter. Anda akan diberikan URL berupa **http://localhost:5000/** atau **http://127.0.0.1:5000/**.



```
TERMINAL  OUTPUT  PROBLEMS  DEBUG CONSOLE  1: python3.9  +  -  [ ]  [ ]  ^  x
(deploy-model-linreg) → dts-deployment-linreg-main python app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 809-271-600
```

# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (7/15)

11. Buka URL tersebut menggunakan browser, maka websitenya akan terbuka. Lakukan pengetesan dengan cara memasukkan data Usia, Jenis Kelamin, dan Status Perokok. Klik tombol Prediksi Sekarang dan lihat hasil prediksi biaya asuransinya seperti berikut.

The screenshot shows a web browser window with the URL `127.0.0.1:5000/predict` in the address bar. The page title is "Apikasi Prediksi Biaya Asuransi". The main content area is divided into two columns. The left column, titled "Prediksi Biaya Asuransi Anda:", contains three input fields: "Usia" (with a text input), "Jenis Kelamin:" (with a dropdown menu showing "Laki-laki"), and "Perokok:" (with a dropdown menu showing "Ya"). Below these fields is a green button labeled "PREDIKSI SEKARANG →". The right column displays the result: "Estimasi biaya asuransi Anda adalah" followed by "USD 3146.79" in large bold text. Below the result, it lists the input data: "Usia: 20 tahun", "Jenis kelamin: Laki-laki", and "Status perokok: Tidak". At the bottom of the page, there are two logos: "DIGITAL TALENT SCHOLARSHIP" and "KOMINFO".

## Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (8/15)

12. Jika sudah selesai menjalankan dan mencoba website secara local pada PC/laptop Anda, buka kembali VSCode. Tekan tombol Ctrl+C untuk menghentikan server.

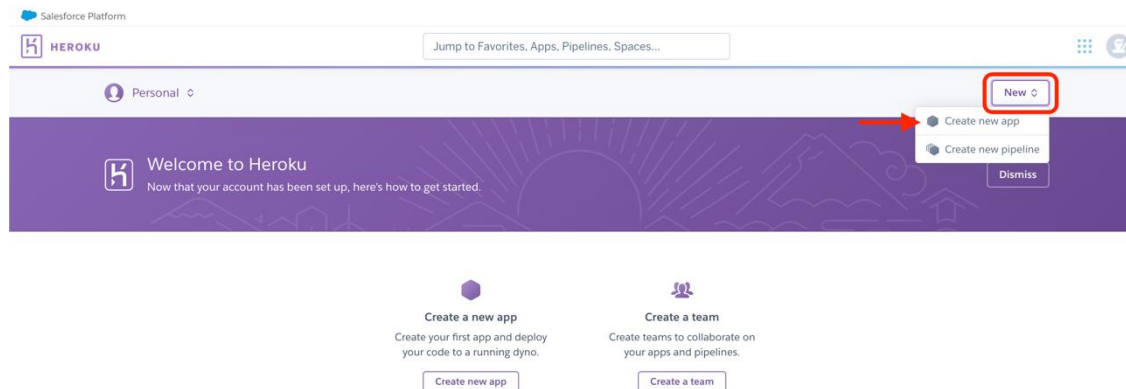


```
127.0.0.1 - - [26/May/2021 21:19:41] "GET / HTTP/1.1" 200 -  
127.0.0.1 - - [26/May/2021 21:27:51] "GET / HTTP/1.1" 200 -  
^C  
(deploy-model-linreg) → dts-deployment-linreg-main
```

The screenshot shows a terminal window with a black background and white text. It displays two successful HTTP GET requests from 127.0.0.1. Below these, a red square highlights the '^C' symbol, which represents the Ctrl+C command used to stop the server. The prompt '(deploy-model-linreg) →' is followed by the text 'dts-deployment-linreg-main'. At the bottom of the terminal, there is a blue status bar with the text 'Live Share', 'tabnine', and '-- NORMAL --'.

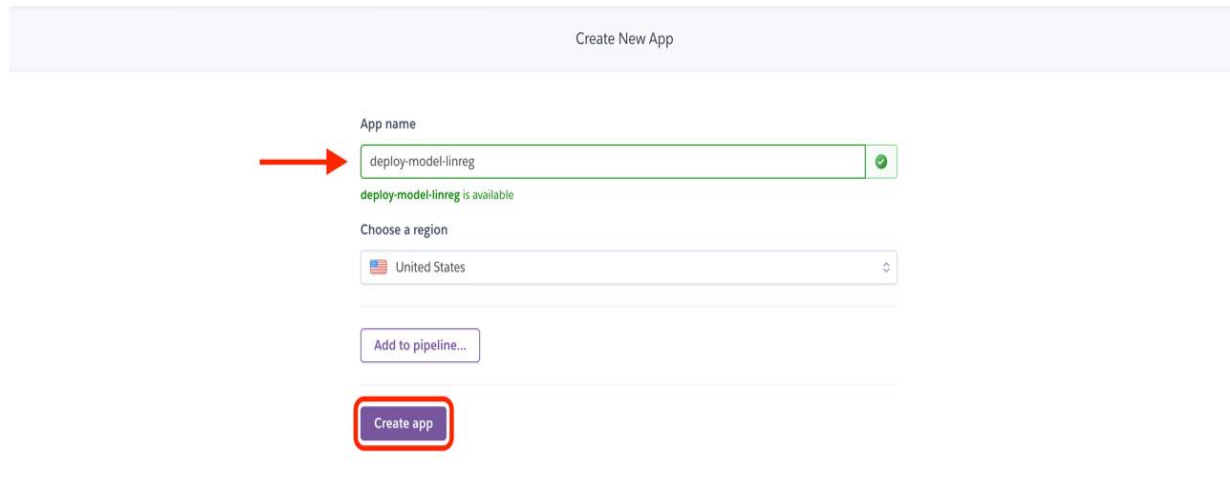
# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (9/15)

13. Sekarang saatnya men-deploy website model Linear Regression ini menggunakan Heroku. Pastikan Anda sudah memiliki akun dan login ke Heroku serta sudah menginstall Heroku CLI. Buka dashboard Heroku. Klik tombol New kemudian pilih Create new app.



# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (10/15)

14. Masukkan nama aplikasi sesuai keinginan, lalu klik tombol Create. Setelah itu, Anda akan diarahkan ke halaman awal bagian Deploy.




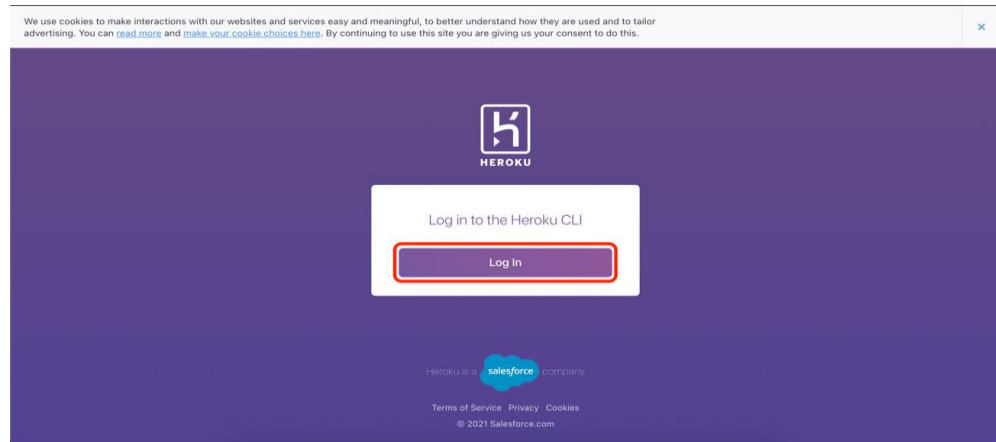
The screenshot shows a 'Create New App' form. At the top is a light blue header with the text 'Create New App'. Below this, the form has several sections. The first section is labeled 'App name' and contains a text input field with the value 'deploy-model-linreg'. A red arrow points to this input field. To the right of the input field is a green checkmark icon. Below the input field, a green message states 'deploy-model-linreg is available'. The second section is labeled 'Choose a region' and contains a dropdown menu with the selected option 'United States' and a small flag icon. Below this section is a button labeled 'Add to pipeline...'. At the bottom of the form is a button labeled 'Create app', which is highlighted with a red rectangular border.



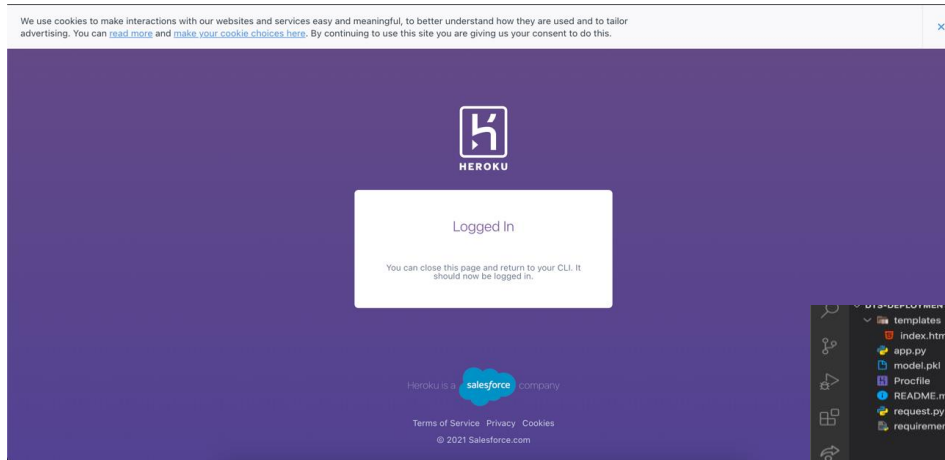
# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (11/15)

15. Login pada Heroku CLI dengan mengetikkan perintah heroku login, tekan Enter, lalu tekan sembarang kunci/tombol keyboard untuk membuka Browser. Anda akan diarahkan untuk login melalui Browser. Setelah selesai, buka kembali ke VSCode.

```
(deploy-model-linreg) → dts-deployment-linreg-main heroku login  
heroku: Press any key to open up the browser to login or q to exit:   
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/48c6b57f-d7ba-4707-919c-b632661fd460?requestor=SFMynTY.g2gDbQAAAAszNi43Ny45NS43OG4GAIZ2N6lSAWIAAVGA.MpZjaH_NLuVAo_pmM0xPiee13oq2IX0JWLUK62HVywA
```



# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (12/15)



```
TERMINAL OUTPUT PROBLEMS DEBUG CONSOLE
1: zsh
(deploy-model-linreg) + dts-deployment-linreg-main heroku --version
heroku/7.53.0 darwin-x64 node-v15.6.0
(deploy-model-linreg) + dts-deployment-linreg-main heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/48c6b57f-d7ba-4707-919c-b632661fd460?requestor=SFMyNTY. g2gDbQAAAszNi43Ny4SNs43OGAIZ2N6I5AWIAAVGA. MpZjaH_NLUVao_pmM0xPiee13oq2IX0JWLUK62HVywA
Logging in... done
Logged in as ahmdalfan@student.telkomuniversity.ac.id
(deploy-model-linreg) + dts-deployment-linreg-main
```

# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (13/15)

16. Buat git repository dan upload semua file yang ada ke Heroku dengan menggunakan beberapa perintah berikut secara berurutan satu persatu.

- git init

```
(deploy-model-linreg) → dts-deployment-linreg-main git init  
Initialized empty Git repository in /Users/alfanme/Downloads/dts-deployment-linreg-main/.git/
```

- heroku git:remote -a deploy-model-linreg

```
(deploy-model-linreg) → dts-deployment-linreg-main git:(master) x heroku git:remote -a deploy-model-linreg  
set git remote heroku to https://git.heroku.com/deploy-model-linreg.git
```

- git add .

```
(deploy-model-linreg) → dts-deployment-linreg-main git:(master) x git add .
```

- git commit -am "deployment pertama"

```
(deploy-model-linreg) → dts-deployment-linreg-main git:(master) x git commit -am "deployment pertama"  
[master (root-commit) 15235aa] deployment pertama  
7 files changed, 270 insertions(+)  
create mode 100644 Procfile  
create mode 100644 README.md  
create mode 100644 app.py  
create mode 100644 model.pkl  
create mode 100644 request.py  
create mode 100644 requirements.txt  
create mode 100644 templates/index.html
```

# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (14/15)

- git init

17. Setelah selesai, Anda akan diberikan URL untuk mengakses website yang sudah di-deploy pada Heroku. Sekarang Anda dapat membagikan URL tersebut untuk diakses oleh banyak orang.

```
(deploy-model-linreg) → dts-deployment-linreg-main git:(master) git push heroku master
```

```
Enumerating objects: 10, done.  
Counting objects: 100% (10/10), done.  
Delta compression using up to 4 threads  
Compressing objects: 100% (7/7), done.  
Writing objects: 100% (10/10), 4.11 KiB | 701.00 KiB/s, done.  
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0  
remote: Compressing source files... done.  
remote: Building source:  
remote:
```

```
remote: Installing collected packages: click, itsdangerous, Werkzeug, MarkupSafe, Jinja2, flask, gunicorn, joblib, scipy, scikit-learn, idna, chardet, certifi, urllib3, requests  
remote: Successfully installed Jinja2-3.0.1 MarkupSafe-2.0.1 Werkzeug-2.0.1 certifi-2020.12.5 chardet-2.0.1 gunicorn-20.1.0 idna-2.10 itsdangerous-2.0.1 joblib-1.0.1 numpy-1.20.3 requests-2.25.1 scikit-learn-0.24.0 poolctl-2.1.0 urllib3-1.26.4  
remote: -----> Discovering process types  
remote: Procfile declares types -> web  
remote:  
remote: -----> Compressing...  
remote: Done: 124.6M  
remote: -----> Launching...  
remote: Released v3  
remote: https://deploy-model-linreg.herokuapp.com/ deployed to Heroku  
remote:  
remote: Verifying deploy... done.  
To https://git.heroku.com/deploy-model-linreg.git  
* [new branch] master -> master  
(deploy-model-linreg) → dts-deployment-linreg-main git:(master)
```

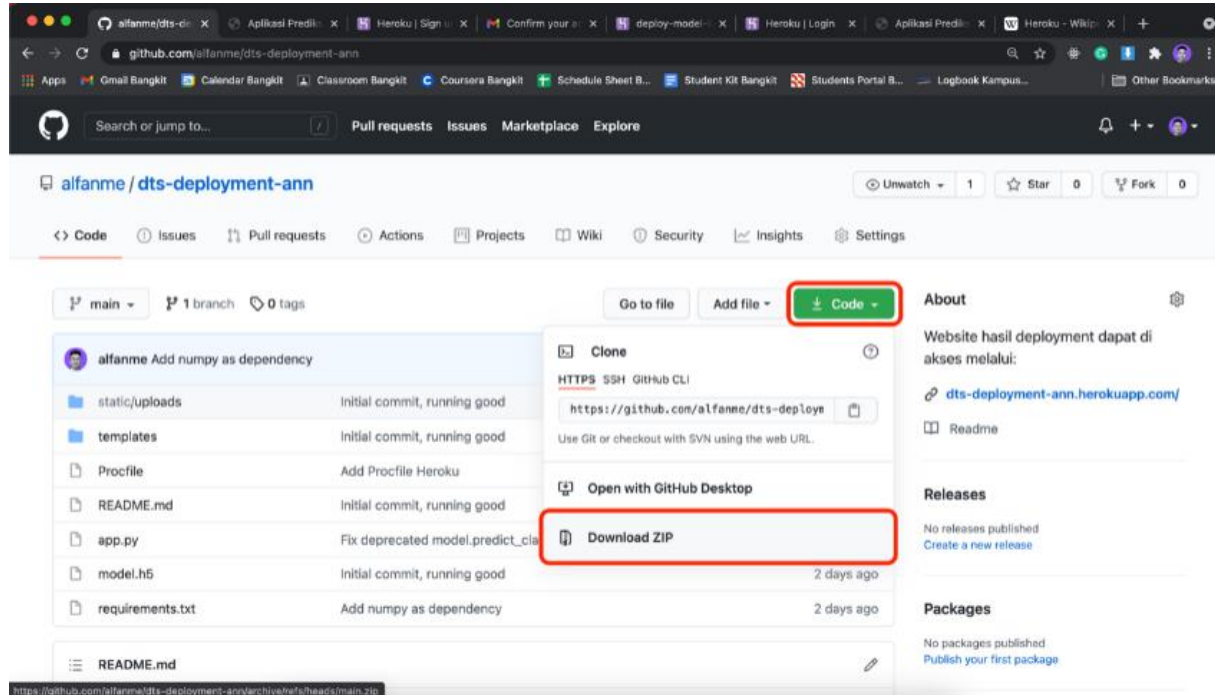
# Deployment Model: Langkah-Langkah model deployment untuk regresi linier (inferencing) (15/15)

18. Selamat, Anda telah berhasil melakukan *deployment* model *Linear Regression* dalam bentuk *website* menggunakan *Heroku*!

The screenshot shows a web browser window with the URL `dts-deployment-linreg.herokuapp.com/predict` highlighted in the address bar. The page has a teal and blue color scheme. On the left, there is a form titled "Prediksi Biaya Asuransi Anda:" with three input fields: "Usia" (with a value of 20), "Jenis Kelamin:" (with a value of Laki-laki), and "Perokok:" (with a value of Ya). Below these fields is a green button labeled "PREDIKSI SEKARANG" with a right arrow. On the right, a white box displays the result: "Estimasi biaya asuransi Anda adalah" followed by "USD 3146.79" in large bold text. Below the result, it lists the input values: "Usia: 20 tahun", "Jenis kelamin: Laki-laki", and "Status perokok: Tidak". At the bottom, there are two logos: "DIGITAL TALENT SCHOLARSHIP" and "KOMINFO".

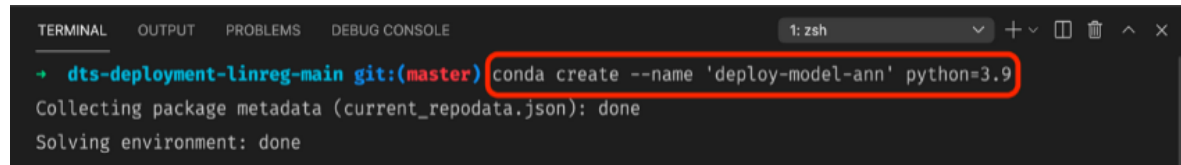
# Deployment Model: Langkah-Langkah model deployment untuk ANN (inferencing) (1/5)

## 1. Download source code dari repository Github

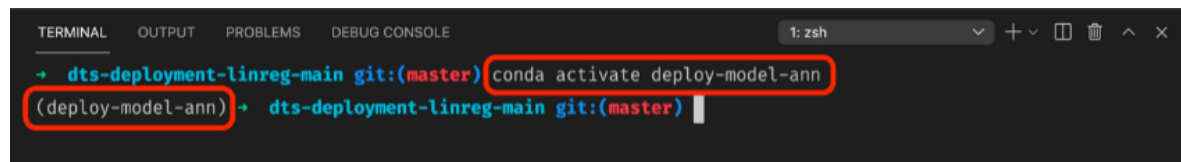


# Deployment Model: Langkah-Langkah model deployment untuk ANN (inferencing) (2/5)

2. Pastikan Anda menggunakan *virtual environment* **Anaconda** yang berbeda. Nama *virtual environment* sebelumnya adalah “**deploy-model-linreg**”, Anda dapat membuat *virtual environment* baru dengan nama “**deploy-model-ANN**” atau nama lainnya sesuai keinginan.



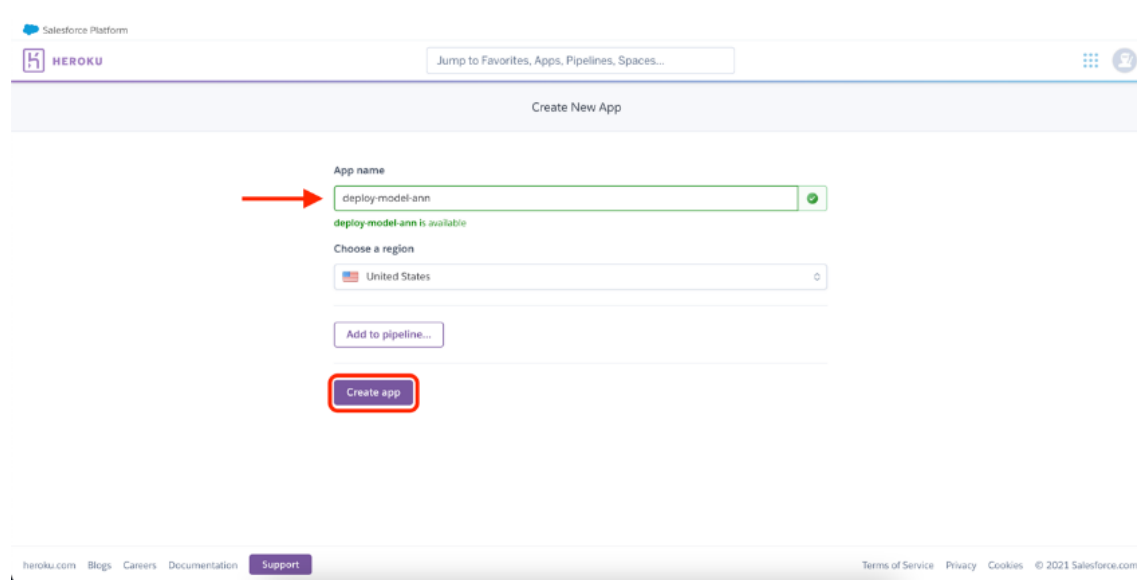
```
TERMINAL OUTPUT PROBLEMS DEBUG CONSOLE 1: zsh
+ dts-deployment-linreg-main git:(master) conda create --name 'deploy-model-ann' python=3.9
Collecting package metadata (current_repodata.json): done
Solving environment: done
```



```
TERMINAL OUTPUT PROBLEMS DEBUG CONSOLE 1: zsh
+ dts-deployment-linreg-main git:(master) conda activate deploy-model-ann
(deploy-model-ann) + dts-deployment-linreg-main git:(master)
```

# Deployment Model: Langkah-Langkah model deployment untuk ANN (inferencing) (3/5)

3. Pastikan Anda membuat project/app baru pada dashboard **Heroku**. Anda dapat memberi nama app-nya dengan “**deploy-model-ANN**” atau “**cat-vs-dog**” atau nama lainnya sesuai keinginan.



Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Create New App

App name

deploy-model-ann

deploy-model-ann is available

Choose a region

United States

Add to pipeline...

Create app

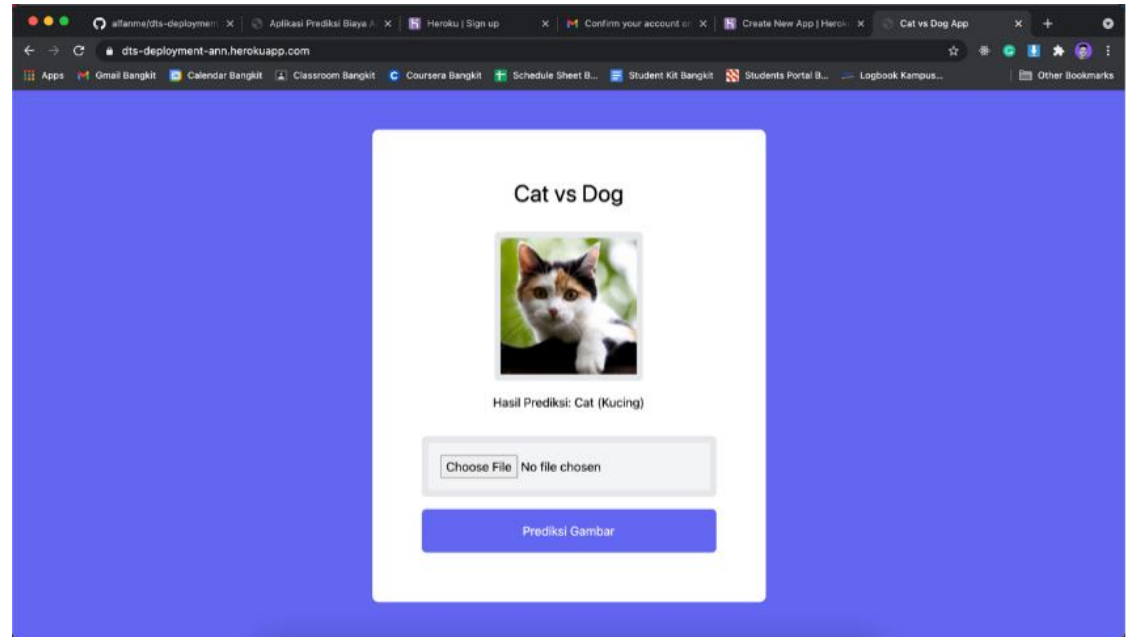
heroku.com Blogs Careers Documentation Support

Terms of Service Privacy Cookies © 2021 Salesforce.com



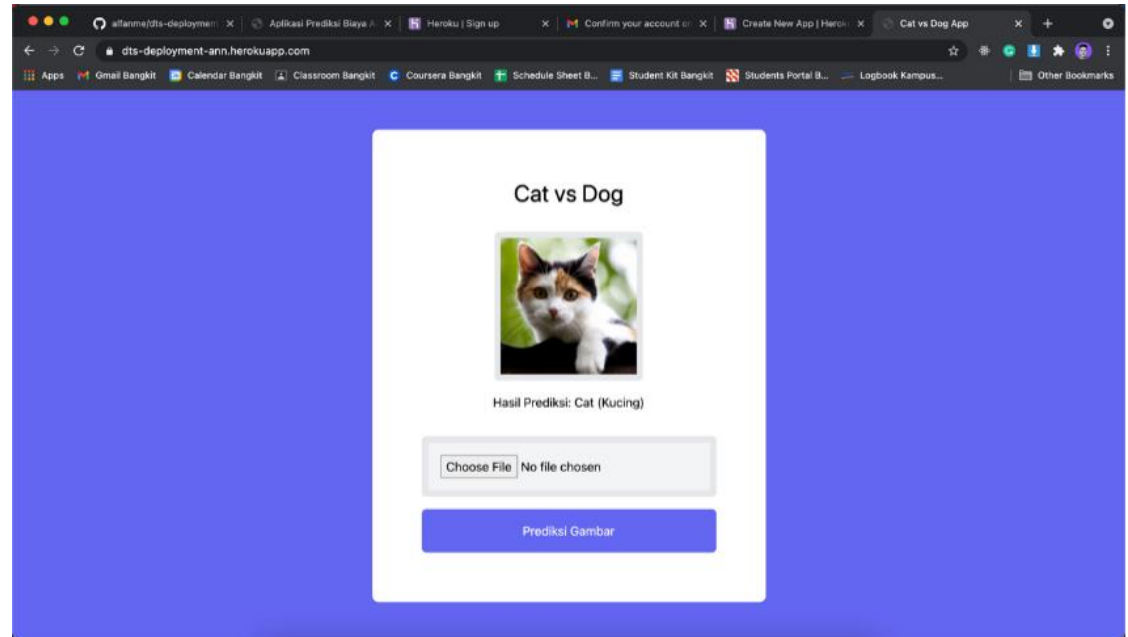
# Deployment Model: Langkah-Langkah model deployment untuk ANN (inferencing) (4/5)

4. Jika *website* model *Linear Regression* menerima input data **usia, jenis kelamin, dan status perokok** untuk memprediksi biaya asuransi. Pada *website* model ANN ini akan menerima input berupa gambar kucing atau anjing yang di-upload dan memprediksi apakah gambar tersebut merupakan gambar kucing atau anjing seperti berikut ini.



# Deployment Model: Langkah-Langkah model deployment untuk ANN (inferencing) (5/5)

4. Setelah memahami perbedaan antara *deployment* model *Linear Regression* dan ANN, maka Anda dapat memulai langkah-langkah yang sama seperti *deployment Linear Regression*. Selamat mencoba! Jika berhasil, buka *URL website* yang diberikan **Heroku** menggunakan *Browser*. Masukkan gambar kucing atau anjing, lalu lihat hasil prediksinya. Sekarang Anda dapat membagikan *URL* tersebut kepada orang lain.



# Tugas

- Implementasikan deployment model dari contoh di atas dan pastikan dapat bekerja dengan baik.

# Tools / Lab Online

- Anaconda
- Visual Studio Code
- Heroku
- <https://code.visualstudio.com/download>
- <https://www.heroku.com>
- <https://www.anaconda.com/products/individual>
- <https://github.com/alfanme/dts-deployment-linreg>
- <https://github.com/alfanme/dts-deployment-ann>

#JADIJAGOANDIGITAL  
**TERIMA KASIH**



digitalent.kominfo



DTS\_kominfo



digitalent.kominfo



digital talent scholarship