

# Men-deploy Model Machine Learning Sederhana Menggunakan Flask untuk Peramalan Harga Motor Piaggio Bekas

Laila Sirri Hayati

Program Studi Statistika, Fakultas MIPA, Universitas Islam Indonesia, Jl.  
Kaliurang Km. 14,5 Yogyakarta, Indonesia

Email: [19611050@students.uii.ac.id](mailto:19611050@students.uii.ac.id)

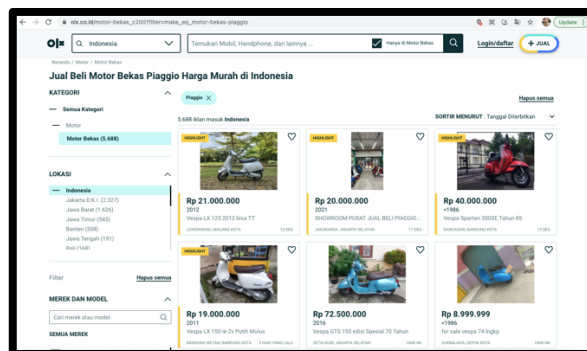
## PENDAHULUAN

Akan dilakukan pembuatan sebuah web dimana web tersebut bertujuan untuk mempermudah pembeli dalam mengetahui prediksi harga motor Piaggio bekas, sehingga dapat mempersiapkan perkiraan *budget* untuk membeli motor dengan type tahun keluaran sesuai dengan yang diinginkan. Langkah pertama yang dilakukan untuk membuat *website* ini adalah melakukan *scrapping* pada sebuah *website* yang difokuskan untuk membeli dan menjual produk serta jasa yaitu *olx.co.id*. Kemudian dari halaman *web* ini dilakukan *scrapping* dengan bantuan *tools extensions* di Google yaitu *Data Miner*. Data yang diambil adalah harga motor sebagai variabel dependen (Y) dan Tahun keluaran sebagai variabel independent (X). Setelah data sudah siap, dilakukan uji apakah data tersebut sesuai dengan model regresi linier. Setelah cocok, dilakukan *deploy* model *machine learning* sederhana menggunakan *flask*.

## LANGKAH-LANGKAH

### 1. Scrapping Data

Langkah pertama yang dilakukan adalah melakukan *scrapping* data dengan *Data Miner*. dari [https://www.olx.co.id/motor-bekas\\_c200?filter=make\\_eq\\_motor-bekas-piaggio](https://www.olx.co.id/motor-bekas_c200?filter=make_eq_motor-bekas-piaggio) yang memuat data motor bekas piaggio. Untuk menggunakan Data Miner harus mendownload *extensions* dari data miner.



Data yang diambil adalah harga motor dan tahun keluaran motor tersebut. *Scrapping* tersebut dilakukan dengan mengambil data dari 10 halaman sehingga didapatkan 309 data. Data hasil *scrapping* yang didapatkan adalah sebagai berikut.

	A	B
1	Y	X
2	31750000	2014
3	27750000	2015
4	69000000	2021
5	50000000	2018
6	35750000	2016
7	14500000	2013
8	37900000	2021
9	37900000	2021
10	37900000	2021
11	10000000	1986
12	31000000	2014

Data diatas merupakan data yang sudah dibersihkan, dimana Y merupakan harga jual motor piaggio dan X merupakan tahun dimana awal motor tersebut terbeli

## 2. Uji Model Terbaik

Selanjutnya dilakukan uji regresi linier untuk menentukan apakah model yang akan dipakai cocok dengan menggunakan R studio.

### a) Mengaktifkan *packages*

Pertama, mengaktifkan *package* “tidyverse” untuk manipulasi dan visualisasi data dan “ggpubr” untuk membuat plot sebagai berikut.

```
library(tidyverse)
library(ggpubr)
theme_set(theme_pubr())
```

### b) Memasukkan data

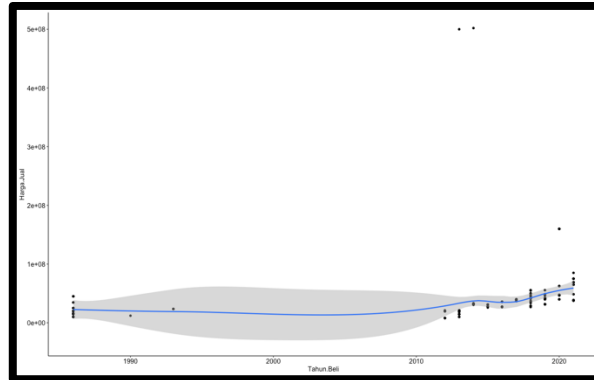
Selanjutnya input kemudian memanggil data dengan objek “piaggio” sebagai berikut.

```
> piaggio = read.table("/Users/yin/Documents/UAS/Piaggio.csv", header = TRUE, sep = ";")
> piaggio
  Harga.Jual Tahun.Beli
1  160000000      2020
2   31750000      2014
3   27750000      2015
4   69000000      2021
5   50000000      2018
6   35750000      2016
7   14500000      2013
8   37900000      2021
9   37900000      2021
10  37900000      2021
```

### c) Visualisasi Data

Selanjutnya membuat plot dengan kolom "Tahun.Beli" sebagai variabel x dan "Harga.Jual" sebagai variabel y seperti gambar berikut.

```
> ggplot(piaggio, aes(x = Tahun.Beli, y = Harga.Jual)) +  
+   geom_point() +  
+   stat_smooth()
```



Dari hasil visualisasi data tersebut, dapat dilihat bahwa terdapat korelasi positif ketika tahun pembelian motor semakin bertambah maka harga penjualan motor semakin tinggi. Hal ini juga menunjukkan bahwa satu asumsi regresi linier sudah terpenuhi karena terdapat pengaruh variabel x terhadap variabel y. Untuk memastikan hubungan kedua variabel, dilakukan uji korelasi.

```
> cor(piaggio$Harga.Jual, piaggio$Tahun.Beli)  
[1] 0.1930127
```

Didapatkan nilai korelasi antara kedua variabel nilainya 0.1930127. Nilai ini berarti bahwa terdapat korelasi positif yaitu ketika nilai x atau tahun pembelian semakin maju, maka nilai y atau harga motor akan semakin tinggi.

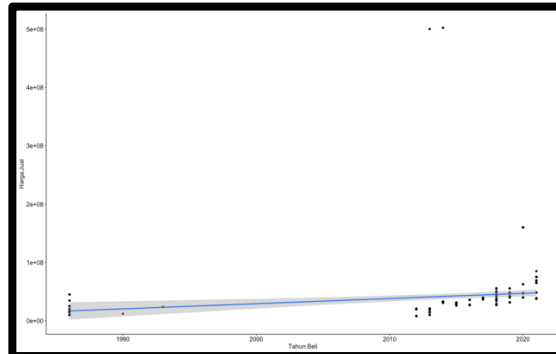
### d) Komputasi

Dibuat model regresi linier untuk melakukan prediksi harga motor Piaggio bekas. Fungsi yang digunakan pada RStudio adalah `lm()`. Model output yang didapatkan adalah sebagai berikut.

```
> model <- lm(Harga.Jual ~ Tahun.Beli, data = piaggio)  
> model  
  
Call:  
lm(formula = Harga.Jual ~ Tahun.Beli, data = piaggio)  
  
Coefficients:  
(Intercept)  Tahun.Beli  
-1.742e+09    8.856e+05
```

#### e) Regression line

```
> ggplot(piaggio, aes(Tahun.Beli, Harga.Jual)) +  
+   geom_point() +  
+   stat_smooth(method = lm)
```



#### f) Summary model

```
> summary(model)  
  
Call:  
lm(formula = Harga.Jual ~ Tahun.Beli, data = piaggio)  
  
Residuals:  
    Min       1Q   Median       3Q      Max   
-31861452 -10632690  -7046404  -46404  460367310  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)      
(Intercept) -1.742e+09  5.118e+08  -3.404  0.000751 ***  
Tahun.Beli   8.856e+05  2.541e+05   3.486  0.000561 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 41910000 on 314 degrees of freedom  
Multiple R-squared:  0.03725,    Adjusted R-squared:  0.03419  
F-statistic: 12.15 on 1 and 314 DF,  p-value: 0.0005607
```

Dari summary model diatas dapat ditarik kesimpulan bahwa model regresi linier ini baik digunakan dibuktikan dengan nilai p-value 0.0005607 dimana nilainya lebih kecil dari alpha (0.05). Hal ini dapat dilihat juga dari tanda bintang 3 yang membuktikan bahwa nilainya tinggi. Hal ini mengartikan bahwa terdapat hubungan antara variabel x dan y.

#### g) Akurasi Model

Nilai akurasi kesalahan yang didapatkan juga dapat dilihat pada hasil summary model yaitu, RSE : 41910000, R Squared : 0.03725, Fstatistic : 12.15 dan Pvalue: 0.0005607

### 3. Men-Deploy Model

Selanjutnya setelah memastikan bahwa model regresi linier cocok digunakan pada peramalan ini, maka dilakukan deploy dengan persamaan regresi linier untuk melakukan prediksi harga motor Piaggio bekas berdasarkan tahun keluarannya. Dengan menyiapkan sebuah folder, dimana folder yang dibuat pada percobaan ini

adalah /Users/yin/Documents/UAS, pada folder ini, dimasukkan file excel data yang akan digunakan. Selanjutnya menuliskan skrip berikut dengan nama “model.py”.

```
1 #Library
2 import pandas as pd
3 import numpy as np
4 import joblib
5 from sklearn.linear_model import LinearRegression
6
7 #dataset
8 piaggio = pd.read_excel('piaggio.xlsx')
9
10 X = np.array(piaggio.iloc[:, 1]).reshape((-1, 1))
11 y = np.array(piaggio.iloc[:, 0])
12
13 from sklearn.linear_model import LinearRegression
14 regressor = LinearRegression()
15
16
17 #Import library pandas
18 import numpy as np
19 import matplotlib.pyplot as plt
20 import pandas as pd
21 import pickle
22
23
24 #Fitting model with training data
25 regressor.fit(X.values, y)
26
27
28 import warnings
29
30
31 def fxn():
32     warnings.warn("deprecated", DeprecationWarning)
33
34
35 with warnings.catch_warnings():
36     warnings.simplefilter("ignore")
37     fxn()
38
39
40 # Saving model to disk
41 pickle.dump(regressor, open('model.pkl', 'wb'))
42
43
44 # Loading model to compare the results
45 model = pickle.load(open('model.pkl', 'rb'))
46 print(model.predict([[99, 82]]))
47
48
49 # Loading model to compare the results
50 model = pickle.load(open('model.pkl', 'rb'))
51 print(model.predict([[99, 82]]))
```

Pada skrip diatas tersebut terlebih dahulu dilakukan “run” pada jupyter notebook dimana persamaan model regresi nantinya akan tersimpan dalam “model.pkl”. Ketika pengambilan data, diambil data dari file excel bernama piaggio.xlsx. Ketika menentukan variabel X dengan memanggil kolom ke 1 dan variabel Y dengan memanggil kolom ke 0 (pada python sedikit berbeda karena nilai dimulai pada angka 0). Pada data dimana urutan data excel dimulai pada kolom pertama yaitu variabel Y selanjutnya X.

```
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4
5 app = Flask(__name__)
6 model = pickle.load(open('model.pkl', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14     """
15     For rendering results on HTML GUI
16     """
17     int_features = [int(x) for x in request.form.values()]
18     final_features = [np.array(int_features)]
19     prediction = model.predict(final_features)
20
21     output = round(prediction[0], 2)
22
23     return render_template('index.html', prediction_text='Harga Motor Sebesar {}'.format(output))
24
25 if __name__ == "__main__":
26     app.run(debug=True)
```

Selanjutnya, skrip di atas disimpan dengan nama “app.py”, Flask adalah framework atau kerangka yang menghubungkan Python dengan aplikasi web, selanjutnya dibuat skrip dengan nama “request.py” seperti berikut,

```
1 import requests
2
3 url = 'http://localhost:5000/predict_api'
4 r = requests.post(url,json={' Harga Motor Piaggio Bekas':99, 'Tahun Pembelian Motor':82})
5
6 print(r.json())
```

Selanjutnya dibuat skrip dengan nama “gitignore” seperti berikut,

```
1 venv/
2
3
4 *.pyc
5 __pycache__/
6
7
8 instance/
9
10
11 .pytest_cache/
12 .coverage
13 htmlcov/
14
15
16 dist/
17 build/
18 *.egg-info/
19
20
21 .DS_Store
```

Selanjutnya dibuat skrip dengan nama “procfile” seperti berikut,

```
web: gunicorn app:app
```

selanjutnya dibuat skrip dengan nama “requirements.txt” seperti berikut,

```
Flask==1.1.1
gunicorn==19.9.0
itsdangerous==1.1.0
Jinja2==2.10.1
MarkupSafe==1.1.1
Werkzeug==0.15.5
numpy>=1.9.2
scipy>=0.15.1
scikit-learn>=0.18
matplotlib>=1.4.3
pandas>=0.19
```

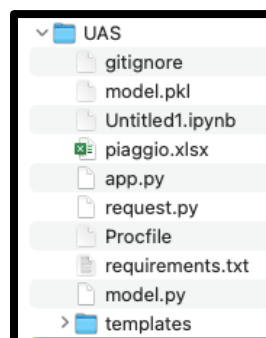
Selanjutnya dibuat skrip dengan nama “index.html” yaitu merupakan skrip halaman website seperti berikut,

```

<!DOCTYPE html>
<html>
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
<meta charset="UTF-8">
<title>Harga Motor</title>
<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
</head>
<center>
<br>
<br>
<br>
<body bgcolor="#DB7093">
<div style="border: 4px solid black;">
<table border="0" style="center">
<tr>
<td>
<div class="login">
<h1>Prediksi Harga Motor Bekas Piaggio</h1>
<!-- Main Input For Receiving Query to our ML -->
<form action="{{ url_for('predict')}}"method="post">
<p>Masukkan Tahun Beli<input type="text" name="X" placeholder="Tahun" required="required" />
<button type="submit" class="btn btn-primary btn-block btn-large">Cek Harga Motor</button></p>
</form>
<br>
<br>
<div>{{ prediction_text }}</div>
</div>
</td>
</tr>
</table>
</div>
</body>
</html>

```

Kemudian semua skrip diatas tadi disimpan didalam satu folder yang sama seperti berikut.



Kemudian masuk ke dalam command prompt dan masukkan perintah berikut

```
(base) yin@Lailas-MacBook-Air ~ % python3 app.py|
```

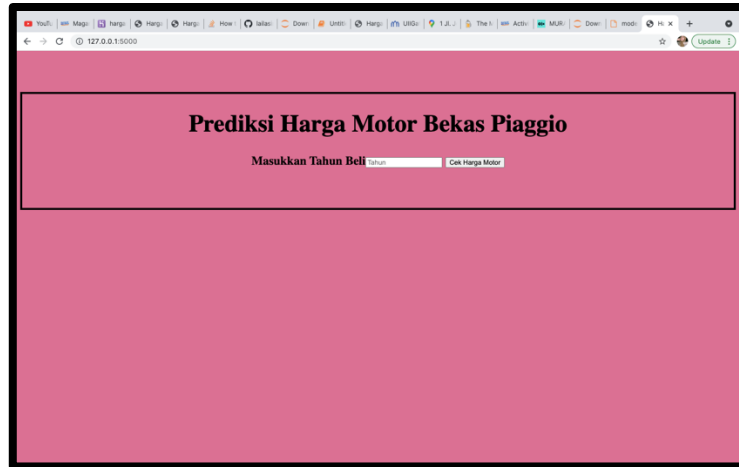
Kemudian akan muncul proses berikut yang artinya bahwa deploy web sudah berhasil.

```

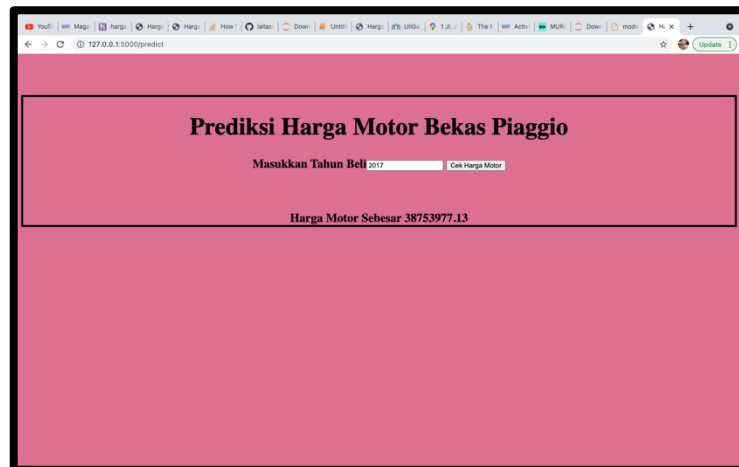
(base) yin@Lailas-MacBook-Air UAS % python3 app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with watchdog (fsevents)
* Debugger is active!
* Debugger PIN: 143-484-563

```

Lalu mengetikkan `http://127.0.0.1:5000/` hingga muncul jendela website seperti berikut,



Kemudian akan dilakukan prediksi harga motor Piaggio bekas dengan tahun pembelian 2017. Hasil prediksi pada web yang ditampilkan adalah sebagai berikut.

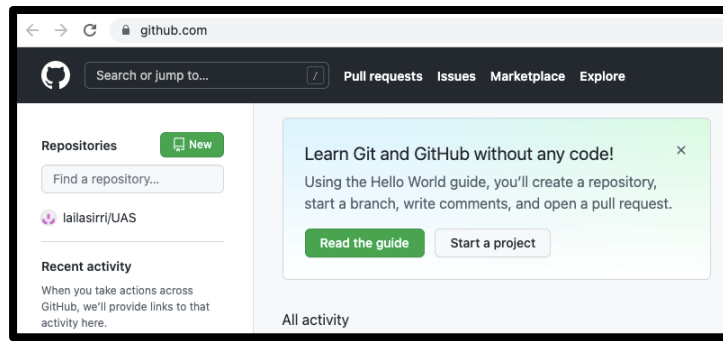


Dari hasil yang ditampilkan, didapatkan bahwa prediksi harga motor Piaggio bekas dengan tahun pembelian tahun 2017 adalah Rp 38.753.977,13

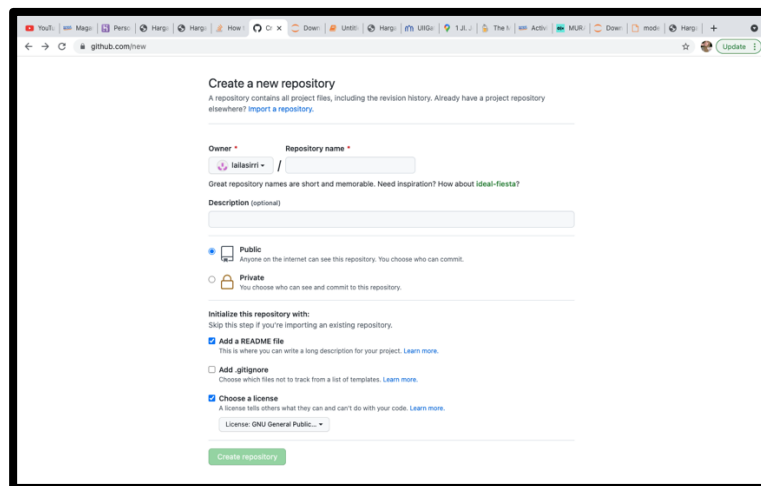
#### 4. Melakukan Hosting Web

Setelah berhasil melakukan deploy pada localhost, selanjutnya dilakukan hosting agar web bisa diakses. Hosting dilakukan dengan menggunakan Heroku untuk deploy web dan Github untuk memasukkan *file-file* lalu dikoneksikan ke Heroku. Langkah pertama adalah *install* Heroku dan Github. Setelah berhasil melakukan instalasi selanjutnya masuk atau membuat akun baru pada Heroku dan Github. Setelah berhasil masuk dengan akun, selanjutnya membuka Github dan membuat repository baru untuk memasukkan *file* yang sudah disiapkan sebelumnya.

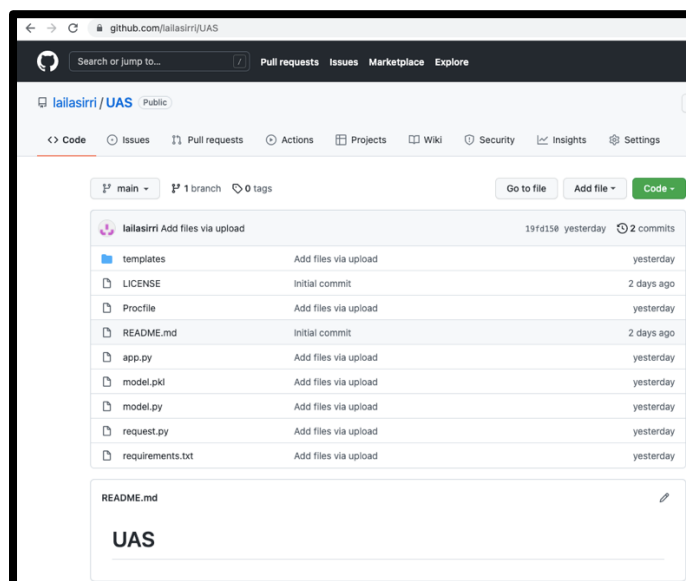




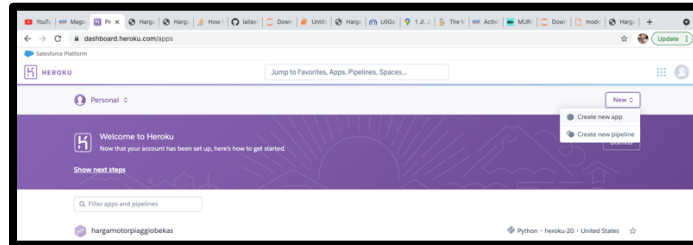
Selanjutnya tampilan jendela baru lalu menentukan nama *respository*, serta pilihan lainnya, dimana penulis mengisi nama repitory dengan “UAS” dan ketentuan lain sebagai berikut lalu klik tombol “Create respitory”.



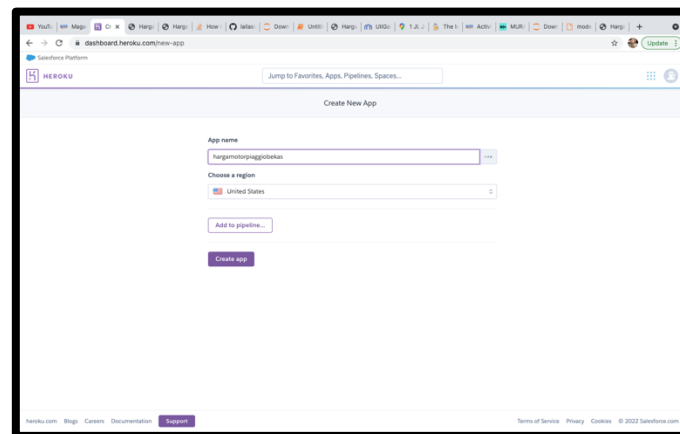
Selanjutnya memasukkan *file* ke dalam respitory UAS dan berikut tampilan yang sudah dimasukkan.



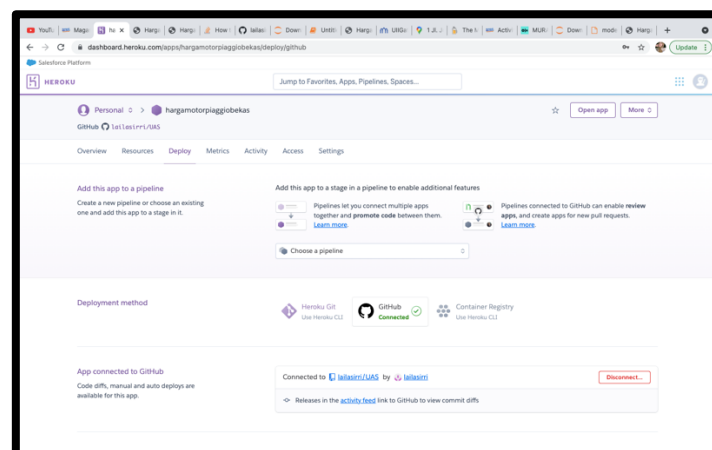
Jika sudah membuat repository di Github, langkah selanjutnya masuk ke Heroku. Setelah berhasil masuk dengan akun, langkah pertama yang dilakukan adalah membuat app baru.



Selanjutnya membuat nama web dengan nama “hargamotorpiaggiobekas” lalu klik “Create app”

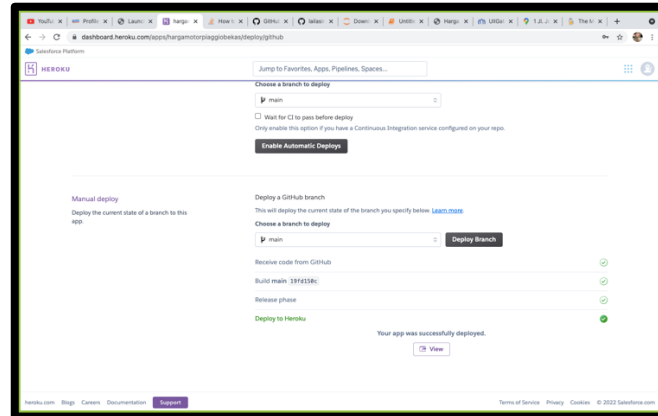


Selanjutnya pada Deployment method memilih untuk konek dengan Github lalu masukkan nama repository yang sudah disediakan sebelumnya yaitu “UAS”. Kemudian dikoneksikan.



Kemudian klik “Deploy Branch”. Lalu heroku akan melakukan *download file* dan menjalankan perintah pada *file respiratory* UAS yang sudah di koneksikan. Jika sudah

berhasil maka akan muncul seperti berikut dan web sudah dapat dibuka dengan klik “View” pada halaman web heroku atau membuka alamat link berikut <https://hargamotorpiaggiobekas.herokuapp.com/>



Berikut tampilan halaman web yang sudah dihosting melalui Heroku dan Github.

