Pengembangan Aplikasi Big Data pada Prediksi Lemak Tubuh dengan Metode Regresi Liniear berbasis DF Pandas

Hanifa Nur Halimah

Email: hanifanurhalimah@gmail.com

Lemak merupakan pembahasan yang penting sampai zaman saat ini karena berlebihan atau kekurangan lemak sangat berbahaya untuk tubuh sampai berujung kematian. Walaupun zaman semakin modern, tetapi untuk menetukan seberapa banyak lemak di dalam tubuh masih melakukan kesalahan terutama masyarakat awam. Diperlukan aplikasi big data untuk mengolah data lemak sehingga dapat mengetahui presentase lemak dengan mudah dan akurat. Pada penelitian ini menggunakan data real case dan data simulasi. Data real case diambil 252 pria diambil dari penelitian Dr. A. Garth Fisher sedangkan data simulasi diambil secara generate oleh sistem. Metode yang digunakan adalah regresi liniear yang mana mampu memprediksi hasil data yang diinginkan secara akurat. Dari 2 data tersebut akan dibandingkan menggunakan pyspark dan tidak. Data frame untuk mengolah data ini menggunakan library df pandas. Platform ini nantinya akan menggunakan aplikasi web sehingga semua masyarakat dapat menggunakannya. Frame work yang digunakan adalah flask dengan cloud pythonanywhere. Bahasa yang digunakan yaitu python. Nilai mape yang tertinggi yaitu 1.8 data simulasi non pyspark dan nilai terrendah yaitu -0.07 data real case non pyspark. Waktu komputasi yang tercepat 25.4 ms data simulasi nonpyspark dan terlama 2410 ms data real case pyspark.

Kata kunci : regresi linier, prediksi lemak tubuh, pyspark, flask, big data, pythonanywhere

**Pendahuluan**

Massa lemak tubuh adalah masa lemak yang berada dalam jaringan adipose dan jaringan lainnya dalam tubuh, menurut william (2002) yang memiliki fungsi :

Lemak esensial : lemak yang dibutuhkan untuk fungsi jaringan organ tubuh seperti pada jaringan otak, sistem syarat pusat, sumsum tulang, jantung, dan membran sel.

Lemak non esensial atau cadangan: lemak yang berasal dari energi yang berlebih dalam tubuh, lemak ini dapat ditemukan pada beberapa organ internal yang berfungsi sebagai pelindung. Sedangkan lebih dari 50% tersimpan dalam jaringan subkutan dan sebagian diantarnya tersimpan dalam rongga abdomen atau lemak viseral dan sering dihubungkan dengan risiko penyakit degeneratif.

Berdasarkan penjelasan diatas, lemak memang penting untuk tubuh manusia, namun jika kelebihan dapat menyebabkan bebagai penyakit yang berujung kematian. Di dunia medis, salah satu cara mengukur persen lemak tubuh dan lemak viseral adalah dengan menggunakan metode BIA (bioelektric impedance analysis) yang mengukur berdasarkan konduktifitas elektrik. Jaringan lemak tubuh memiliki konduktifitas elektri yang kecil, sedangkan otot, pembuluh darah, dan tulang memiliki konduktivitas elektrik yang besar. Menggunakan alat ini cukup mahal sehingga masyarakat lebih memilih mengukur secara gratis. Namun cara ini salah, seperti berbadan besar, berat badan, bmi, dan lain lain sehingga masyarakat merasa bahwa mereka obersitas. Karena takut obersitas, maka kadangkala masyarakat melakukan diet yang tidak wajar. Diet yang tidak wajar dapat membahayakan kesehatan manusia dalam waktu yang tidak dapat ditentukan. Kadangkala masyarakat memiliki lemak yang sedikit, tetapi karena merasa ingin kurus, maka takut makan yang berlemak, seperti penjelasan diatas kegunaan lemak penting untuk sistem kerja pada organ manusia. Jika kekurangan maka dapat dipastikan organ tidak dapat berfungsi dengan baik. Contoh korban diet yang tidak wajar yaitu rambut rontok, secara tampak saja rambut bisa rontok, bagaimana dengan organ di dalam tubuh manusia itu sendiri. Diet yang tidak wajar selain itu menggunakan obat diet secara herbal maupun kimia. Namun obat tersebut tidak dapat menghilangkan lemak secara otomatis, bahkan beberapa mengikat cairan di dalam tubuh sehingga merasa sudah berhasil diet padahal tidak berkurang sama sekali lemak di dalam tubuh. Kelebihan lemak tidak baik, kekurangan pun juga tidak baik. Sehingga diperlukan sebuah platform untuk memberikan nilai lemak tubuh manusia yang akurat.

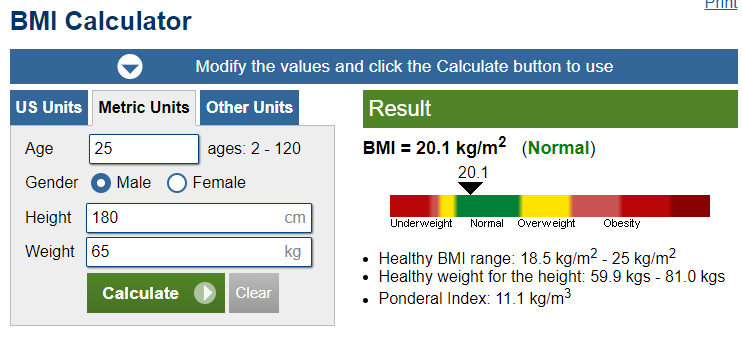
Pada penelitian ini, akan mengembangkan platform untuk memprediksi lemak pada tubuh. Dikarenakan tubuh manusia unik, perlu data yang besar untuk menghasilkan nilai akurasi yang tinggi. Metode prediksi yang digunakan menggunakan regresi Linear. Regresi Linear ini nantinya akan menggunakan python menggunakan platform colab yang mudah di jalankan. Menggunakan library pandas yang berlisensi bsd dan open source yang menyediakan struktur data dan analisis data yang mudah digunakan. Pandas dapat membaca file csv, txt, tsv, mengolah data dengan menggunakan join, distinct, group by, agreasi, dan teknik lainnya.

* **Dataset, Dasar Teori dan Metode Usulan**

1. **Tentang Dataset**

Dataset yang digunakan pada penelitian ini diambil dengan 2 cara yaitu: data real case dan data simulasi secara random. Nilai radom yang diambil melalui library numpy sebagai nilai variasi pada variabel input. Nilai batas atas dan bawah diambil dari nilai batas atas dan bawah pada data real case. Data akan diambil sebanyak 10.000.000. Data real case diambil dari penelitian Dr. A. Garth Fisher di website https://www.kaggle.com/fedesoriano/body-fat-prediction-datasetweb. Daftar perkiraan persentase lemak tubuh yang ditentukan berbagai pengukuran lingkar tubuh untuk 252 pria.

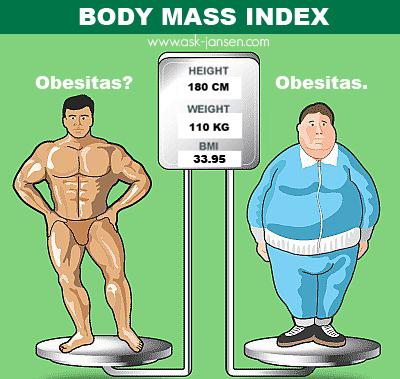
Pengukuran lemak yang paling mudah menggunakan perhitungan bmi yang hanya perlu memasukkan usia, berat badan, tinggi badan dapat dilihat pada gambar 1. Namun peprhitungan ini dapat diragukan karena berat badan seseorang bisa terjadi karena berat pada otot. Bisa kita liat perbandingan berat lemak 2kg dengan otot 2 kg pada gambar 2. Lemak lebih besar daripada otot namun berat sama. Pada gambar 3 memperlihatkan manusia dengan berat dan tinggi ya sama memiliki postur tubuh yang berbeda walaupun sama memiliki badan yang besar. Jika dilihat menggunakan rumus BMI 33.95 pasti sudah obersitas, namun apakah yang kiri bisa disebut obersitas? Perbedaan yang mudah di lihat yaitu berdasarkan ukuran badan anggota tubuh. Penelitian oleh Siri (1956) memprediksi kadar lemak dengan cara mengukur badan manusia itu sendiri antara lain lingkar leher, lingkar dada, lingkar perut, lingkar pinggul, lingkar paha, lingkar lutut, lingkar pergelangan kaki, lingkar bisep, lingkar lengan bawah, lingkar pergelangan tangan. Namun badan manusia tidak rata sehingga perlu cara mengukur secara standart yang tercatum dalam Benhke dan Wilmore (1974), hlm. 45-48. Contoh untuk mengukur lingkar perut dapat dilihat pada gambar 4. 2 penelitian tersebut digabung dan dimasukan dalam penelitian itu yaitu variabel masukan terdiri dari usia (tahun), Berat (lbs), Tinggi (inchi), Lingkar leher (cm), Lingkar dada (cm), Lingkar perut 2 (cm), Lingkar pinggul (cm), Lingkar paha (cm), Lingkar lutut (cm), Lingkar pergelangan kaki (cm), Lingkar bisep (diperpanjang) (cm), Lingkar lengan bawah (cm), Lingkar pergelangan tangan (cm). Variabel output pada peneitian ini adalah persentase lemak tubuh. Data ini digunakan untuk menghasilkan persamaan prediksi untuk berat badan tanpa lemak yang diberikan dalam abstrak "Persamaan prediksi komposisi tubuh umum untuk pria menggunakan teknik pengukuran sederhana", K.W. Penrose, A.G. Nelson, A.G. Fisher, FACSM, Human Performance Research Center, Brigham Young University, Provo, Utah 84602 as listed in Medicine and Science in Sports and Exercise, vol. 17, no. 2, April 1985, p. 189.



Gambar 1. kalkulator bmi



Gambar 2. Perbandingan lemak dan otot



Gambar 3 perbandingan manusia yang memiliki bmi sama

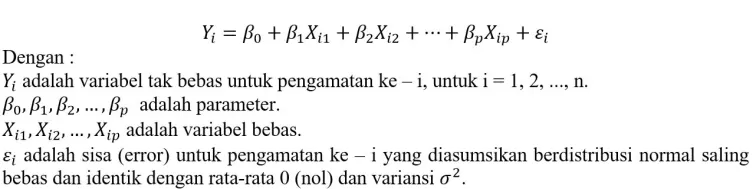


Gambar 4 mengukur lingkar perut

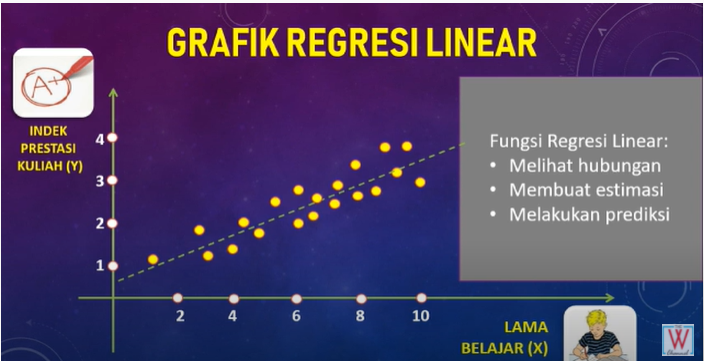
1. **Tentang Metode Regresi Liniear**

Analisis regresi merupakan salah satu teknik analisis data dalam statistika yang seringkali digunakan untuk mengkaji hubungan antara beberapa variabel dan meramal suatu variabel (Kutner, Nachtsheim dan Neter, 2004). Fungsi regresi linear adalah :

1. Membuat estimasi rata-rata dan nilai variabel tergantung dengan didasarkan pada nilai variabel bebas.
2. Membangun hipotesis karakteristik dependensi.
3. Untuk meramalkan nilai rata-rata variabel bebas dengan didasarkan pada nilai variabel bebas di luar jangkauan sampel.

Bentuk umum model regresi linear berganda dengan p variabel bebas seperti pada persamaan berikut. 

Contoh gambaran untuk linear regresi dapat dilihat pada gambar 5. Variabel x / variabel bebas/independen (yang mempengaruhi) yaitu lama belajar variabel, dan variabel y / variabel terikat/dependent (yang dipengaruhi) ialah indek pretasi kuliah. Mahasiswa yang ipk tinggi pasti memiliki waktu belajar lebih lama daripada ipk rendah.



Gambar 5. Regresi linier

1. **Pengembangan web**

* Regresi Linear

Tabel 1 regresi linear

| No | Regresi Linear |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | import numpy as np  import pandas as pd  from sklearn.linear\_model import LinearRegression  from sklearn.metrics import mean\_squared\_error  data = pd.read\_csv('dt.csv')  x = data.drop(["lemak"], axis=1)  y = data["lemak"]  datauji = pd.read\_csv('tu.csv')  x\_test = datauji.drop(["lemak"], axis=1)  y\_test = datauji["lemak"]  regr = LinearRegression()  regr.fit(x, y)  y\_pred = regr.predict(x\_test)  print("Prediksi regresi : ",y\_pred)  aktual, predict = y\_test, y\_pred  mape = np.sum(np.abs(((aktual - predict)/aktual)\*100))/len(predict)  print(mape) |

Penjelasan :

Baris 1 : memanggil library numpy

Baris 2 : memanggil library pandas

baris 3 : memanggil library sklearn.liniear\_model

baris 4 : memanggil library mean square eror

baris 5 : memasukkan file dt.csv (data latih)

Baris 6 :membuat variabel x

Baris 7 :membuat variabel y

Baris 8 :memasukkan file tu.csv (data uji)

Baris 9 :membuat variabel x

Baris 10 :membuat variabel y

baris 11 : memanggil fungsi regresi linear

baris 12 : memasukkan nilai x da y

Baris 13 :prediksi hasil variabel y\_pred

baris 14 : cetak hasil prediksi

baris 15 : variabel aktual dan prediksi

Baris 16 : rumus mape

baris 17 : cetak nilai Mape

* File Csv dengan Sqlite

Basis data SQLite adalah fitur bawaan Python dan sangat berguna. Memiliki semua fitur yang dibutuhkan untuk situs web berbasis data. Menggunakannya dengan Pandas sederhana. Dapat menyimpan kerangka data secara permanen dalam tabel dan membacanya langsung ke kerangka data baru saat membutuhkannya. Selain untuk penyimpanan, anda dapat memilih dan memfilter data menggunakan perintah SQL sederhana. Biasanya kita memperoleh data, atau menyimpan file berupa csv atau Comma Separated Values. Tata cara membuat sqlite dan memasukkan csv ke dalam sqlite dapat dilihat pada tabel 2.

Tabel 2 csv ke sqlite

| No | Csv to sqlite |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | !pip install db-sqlite3  import pandas as pd  import sqlite3  from datetime import date  #connecting with the database.  db = sqlite3.connect("hanifa.db")  db.execute("drop table if exists dataset") # next data Min. 30 dari case Anda  try:  db.execute("create table dataset(usia text, berat text, tinggi text, leher text,  dada text, perut text, pinggul text, paha text, lutut text, pk text, bisep text,  lb text, pt text, lemak text)")  except:  print("Already table existed !!")  usia = 24  berat = 300  tinggi = 50  leher = 33  dada = 111  perut = 99  pinggul = 120  paha = 66  lutut = 33  pk = 21  bisep = 39  lb = 22  pt= 17  lemak = 16  cmd = "insert into dataset(usia, berat, tinggi, leher, dada, perut, pinggul,  paha, lutut, pk, bisep, lb, pt, lemak) values('{}','{}','{}','{}','{}','{}', '{}','{}','{}',  '{}','{}','{}','{}','{}')".format(usia, berat, tinggi, leher, dada, perut, pinggul,  paha, lutut, pk, bisep, lb, pt, lemak)  db.execute(cmd)  db.commit()  db = sqlite3.connect("hanifa.db")  rs = db.execute('SELECT \* FROM dataset')  for row in rs:  print (row)  db = sqlite3.connect("hanifa.db")  qry = """  SELECT \* FROM dataset  """  df = pd.read\_sql\_query(qry, db)  df.head()  def pushCSVdatasetToDB(usia, berat, tinggi, leher, dada, perut, pinggul,  paha, lutut, pk, bisep, lb, pt, lemak):  #inserting values inside the created table  db = sqlite3.connect("hanifa.db")  cmd = "insert into dataset(usia, berat, tinggi, leher, dada, perut, pinggul,  paha, lutut, pk, bisep, lb, pt, lemak) values('{}','{}','{}','{}','{}','{}', '{}','{}','{}',  '{}','{}','{}','{}','{}')".format(usia, berat, tinggi, leher, dada, perut, pinggul,  paha, lutut, pk, bisep, lb, pt, lemak)  db.execute(cmd)  db.commit()  data = pd.read\_csv("bodyfat.csv")  data.shape  # data.head()  data.head(30)  pushCSVdatasetToDB("12","24","350","31","99","85","121", "56","31",  "23","31", "21","19","23")  db = sqlite3.connect("hanifa.db")  qry = "SELECT \* FROM dataset"  df = pd.read\_sql\_query(qry, db)  df.head()  # CSV\_to\_SQLite3 dari file dataset\_csv.csv  for i in range(0,len(data)):  pushCSVdatasetToDB(data.iloc[i][0],data.iloc[i][1],data.iloc[i][2],  data.iloc[i][3],data.iloc[i][4],data.iloc[i][5], data.iloc[i][6], data.iloc[i][7],  data.iloc[i][8], data.iloc[i][9], data.iloc[i][10], data.iloc[i][11],  data.iloc[i][12], data.iloc[i][13])  db = sqlite3.connect("hanifa.db")  qry = """  SELECT \* FROM dataset WHERE usia LIKE "24%"  """  df = pd.read\_sql\_query(qry, db)  df.head(30)  db = sqlite3.connect("hanifa.db")  qry = """  SELECT \* FROM dataset WHERE lemak ="16"  """  df = pd.read\_sql\_query(qry, db)  df.head()  db = sqlite3.connect("hanifa.db")  qry = """  SELECT \* FROM dataset  """  df = pd.read\_sql\_query(qry, db)  df.head(32) |

Penjelasan :

Baris 1 : mengisntall db sqlite

Baris 2 : memanggil library pandas

Baris 3 : memanggil library sqlite3

Baris 4 : memanggil library waktu

Baris 5 : membuat file hanifa.db

Baris 6 : membuat tabel jika file sudah tersedia

Baris 7 : mencoba

Baris 8 : membuat tabel dan diisi dengan nama variabel yang diinginkan

Baris 9 : jika sudah ada

Baris 10 : cetak tabel sudah tersedia

Baris 11 : membuat tabel yang isi terdiri dari variabel dan nilai

Baris 12 :memasukkan ke dalam file .db

Baris 13: menampilkan data secara baris

Baris 14: menampilkan data dengan tabel

Baris 15:memasukkan file csv ke db berdasarkan nama variabel

Baris 16 :membaca file bodyfat.csv

Baris 17 : menampilkan data

Baris 18 : memasukkan nilai ke db (beda cara dengan diatas)

Baris 19 : menampilkan file fga big data.db

Baris 20 : memasukkan file bodyfat.csv ke dalam hanifa.db

Baris 21 : menampilkan data yang mana usia memiliki nilai 24

Baris 22 :menampilkan data yang mana lemak memiliki nilai 16

Baris 23 : menampilkan semua terdiri dari 32 data, 30 data dari data.csv dan 2 data dari memasukkan secara manual ke dalam file db

* Flask

Flask adalah micro web framework yang ditulis dalam bahasa pemrograman Python dan berdasarkan Werkzeug toolkit dan template engine Jinja2. Berlisensi BSD. Flask disebut micro framework karena tidak membutuhkan alat-alat tertentu atau pustaka. Flask tidak memiliki database abstraction layer, validasi form, atau komponen lain di mana sudah ada pustaka pihak ketiga yang menyediakan fungsi umum. Namun, Flask mendukung ekstensi yang dapat menambahkan fitur aplikasi seolah-olah mereka diimplementasikan dalam Flask itu sendiri. Ekstensi yang ada untuk object-relational mapper, validasi form, penangananunggahan, berbagai teknologi otentikasi terbuka, dan beberapa alat-alat yang terkait kerangka umum. Ekstensi diperbarui jauh lebih teratur daripada inti program Flask.

Tata cara membuat flask login, regrister, user, logout

1. Pertama kita harus menginstall library yang dibutuhkan yaitu :

From flask import Flask,render\_template,flash, redirect,url\_for,session,logging,request

From flask\_sqlalchemy import sqlalchemy

From flask\_ngrok import run\_with\_ngrok

2. Membuat kelas user untuk menyimpan data user di dalam file db :

Class user(db.Model):

# \_\_tablename\_\_ = 'user'

Id = db.Column(db.Integer, primary\_key=True)

# Mail Password Name Level

Mail = db.Column(db.Text) # sbg Username

Password = db.Column(db.Text)

Name = db.Column(db.Text)

Level = db.Column(db.Text)

3. Untuk menampilkan halaman utama memanggil index.html :

@app.route("/")

Def index():

Return render\_template("index.html")

4. Ketika ingin keluar pada halaman bigdataapp.html ke halaman utama :

Def Logout():

Return render\_template("index.html")

5. Menampilkan halaman login, mencari data ketika user login, jika ditemukan maka akan masuk ke halaman bigdataapp.html dan menyimpan data user jika tidak ada maka tetap di halaman login dan menampilkan pesan bahwa masukkan email dan password dengan benar:

Def login():

Msg = ""

If request.method == "POST":

Mail = request.form["mail"]

Passw = request.form["passw"]

Login = user.query.filter\_by(Mail=mail, Password=passw).first()

Print(login)

If login is not None:

# return redirect(url\_for("index"))

Return render\_template('bigdataapps.html', login = login)

6. Menampikan halaman regrister, user akan memasukan nama, email dan password yang akan disimpan ke file db:

@app.route("/register", methods=["GET", "POST"])

Def register():

If request.method == "POST":

Mail = request.form['mail']

Uname = request.form['uname']

Passw = request.form['passw']

Register = user(Mail = mail, Name = uname, Password = passw)

# register = user(Mail = mail, Password = passw)

Db.session.add(register)

Db.session.commit()

Return redirect(url\_for("login"))

Return render\_template("register.html")

7. Menampilkan halaman bigdataapp :

@app.route("/bigdataapps", methods=["GET", "POST"])

Def bigdataapps():

If request.method == 'POST':

Import pandas as pd

Import numpy as np

Dataset = request.FILES['inputdataset']

# dataset = request.files['inputdataset']

#dataset = 'dataset\_dump.csv'

Persentase\_data\_training = 90

Banyak\_fitur = int(request.POST['banyakfitur'])

Banyak\_hidden\_neuron = int(request.POST['banyakhiddenneuron'])

Dataset = pd.read\_csv(dataset, delimiter=';', names = ['Tanggal', 'Harga'], usecols=['Harga'])

Minimum = int(dataset.min()-10000)

Maksimum = int(dataset.max()+10000)

New\_banyak\_fitur = banyak\_fitur + 1

Hasil\_fitur = []

For i in range((len(dataset)-new\_banyak\_fitur)+1):

Kolom = []

J = i

While j < (i+new\_banyak\_fitur):

Kolom.append(dataset.values[j][0])

J += 1

Hasil\_fitur.append(kolom)

Hasil\_fitur = np.array(hasil\_fitur)

Data\_normalisasi = (hasil\_fitur - minimum)/(maksimum - minimum)

Data\_training = data\_normalisasi[:int(persentase\_data\_training\*len(data\_normalisasi)/100)]

Data\_testing = data\_normalisasi[int(persentase\_data\_training\*len(data\_normalisasi)/100):]

#Training

Bobot = np.random.rand(banyak\_hidden\_neuron, banyak\_fitur)

Bias = np.random.rand(banyak\_hidden\_neuron)

H = 1/(1 + np.exp(-(np.dot(data\_training[:, :banyak\_fitur], np.transpose(bobot)) + bias)))

H\_plus = np.dot(np.linalg.inv(np.dot(np.transpose(h),h)),np.transpose(h))

Output\_weight = np.dot(h\_plus, data\_training[:, banyak\_fitur])

#Testing

H = 1/(1 + np.exp(-(np.dot(data\_testing[:, :banyak\_fitur], np.transpose(bobot)) + bias)))

Predict = np.dot(h, output\_weight)

Predict = predict \* (maksimum - minimum) + minimum

#MAPE

Aktual = np.array(hasil\_fitur[int(persentase\_data\_training\*len(data\_normalisasi)/100):, banyak\_fitur])

Mape = np.sum(np.abs(((aktual - predict)/aktual)\*100))/len(predict)

Return render\_template('bigdataapps.html', {

'Y\_aktual' : list(aktual),

'Y\_prediksi' : list(predict),

'Mape' : mape

})

Else:

Return render\_template('bigdataapps.html')

8. Menjalakan program ini:

If \_\_name\_\_ == '\_\_main\_\_':

Db.create\_all()

App.run()

9. Menampilkan header gambar pada halaman index, login, dan regrister :

<header class="container">

<div class="row">

<div class="col-12">

<img src="{{ url\_for('static', filename = 'img/headeroke.png') }}" />

</div>

</div>

</header>

10. Form dan tombol untuk halaman index, login, dan regrister berada ditengah menggunakan :

<div class="col-md-2">

</div>

<div class="col-md-8 ">

....

</div>

<div class="col-md-2">

</div>

11. Menampilkan footer dengan gambar big data pada halaman index dikarenakan banyak space yang kosong dan memberikan nilai jarak antara tombol dengan gambar sebesar 50 pixel :

<footer class="container">

<div class="row" style="margin-top: 50px">

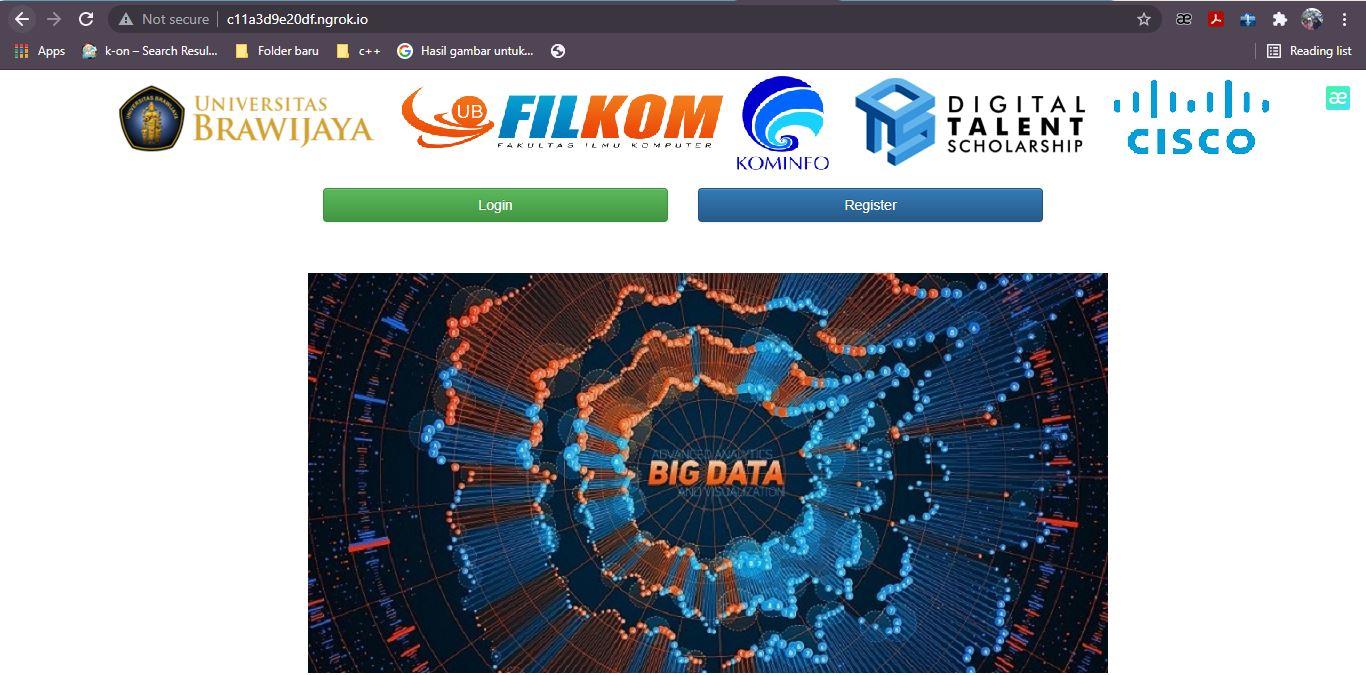
<div class="col-md-12">

</div>

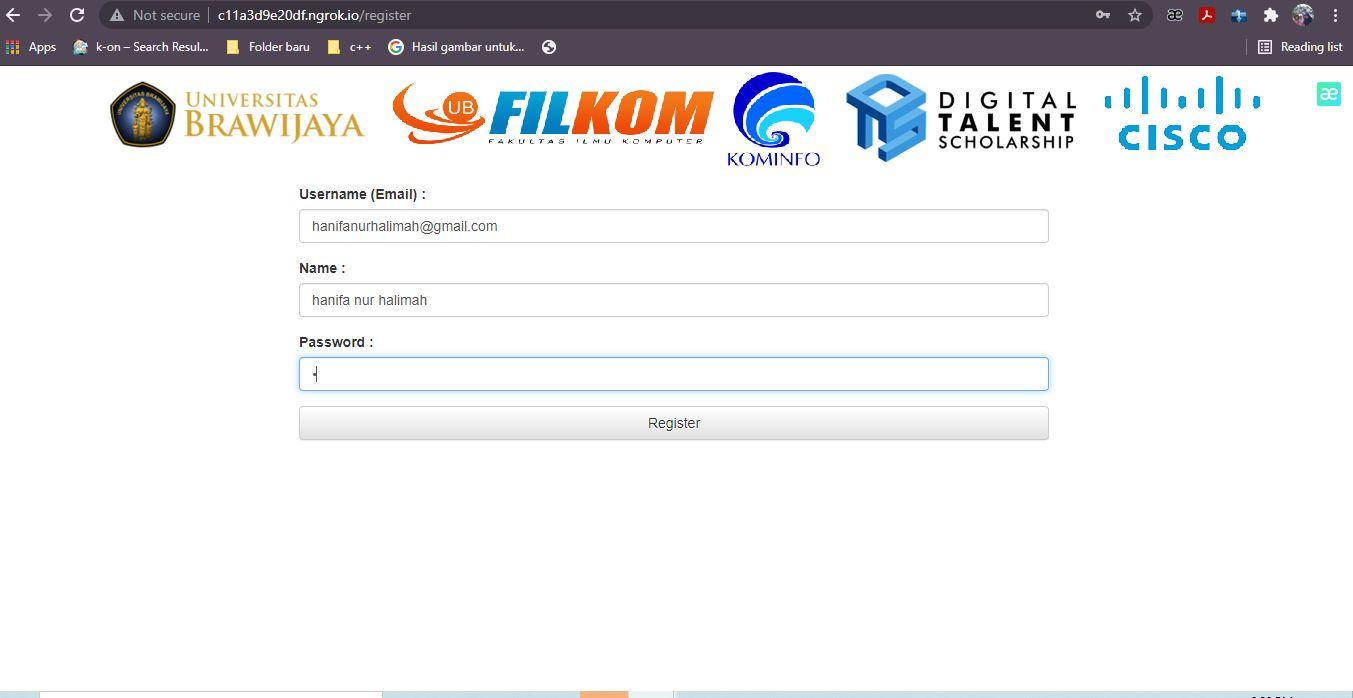
12. Untuk menampilkan nama user dan email :

<h4>Welcome,{{login.Name}} </h4>

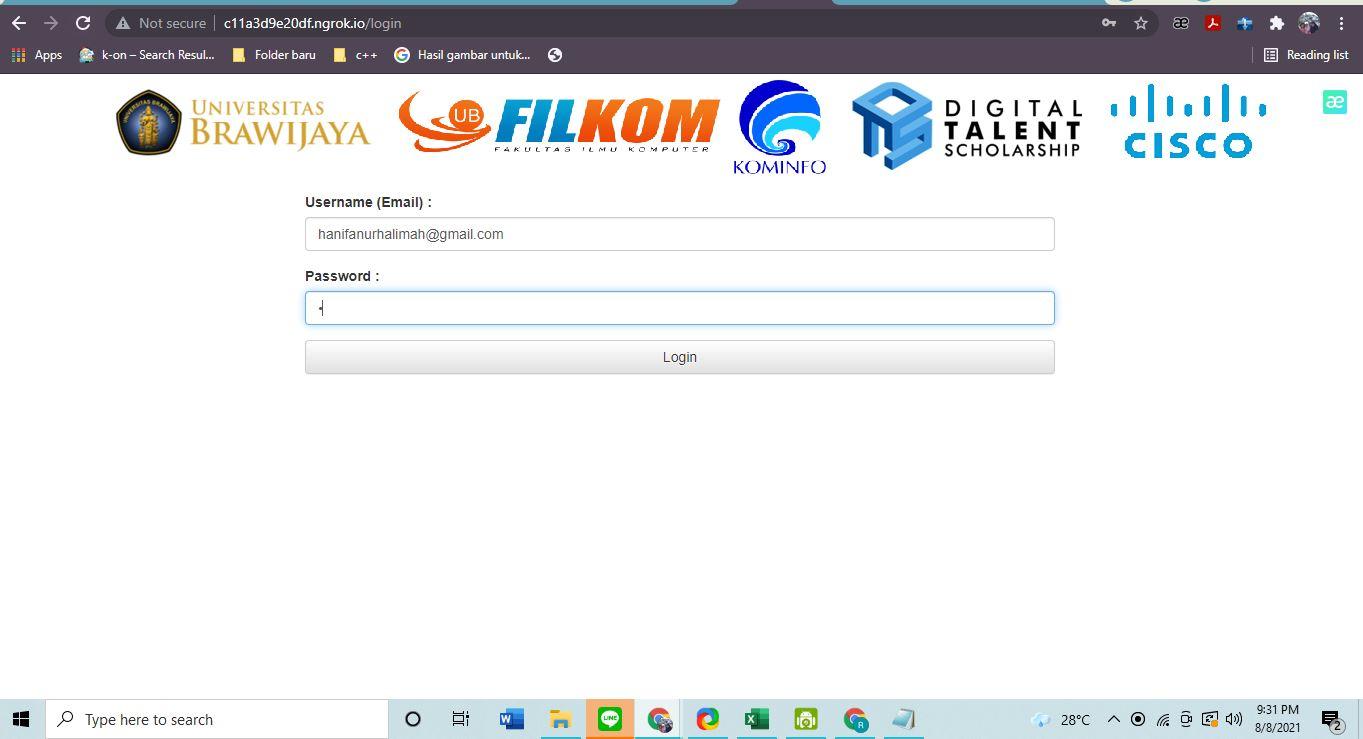
<p class="text-muted">{{login.Mail}}</p><a href="profile.html" class="btn btn-rounded btn-danger btn-sm">View Profile</a></div>



Gambar 6 menampilkan halaman index



Gambar 7 menampilkan halaman register



Gambar 8 menampilkan halaman login



Gambar 9 Menampilkan nama dan email pengguna



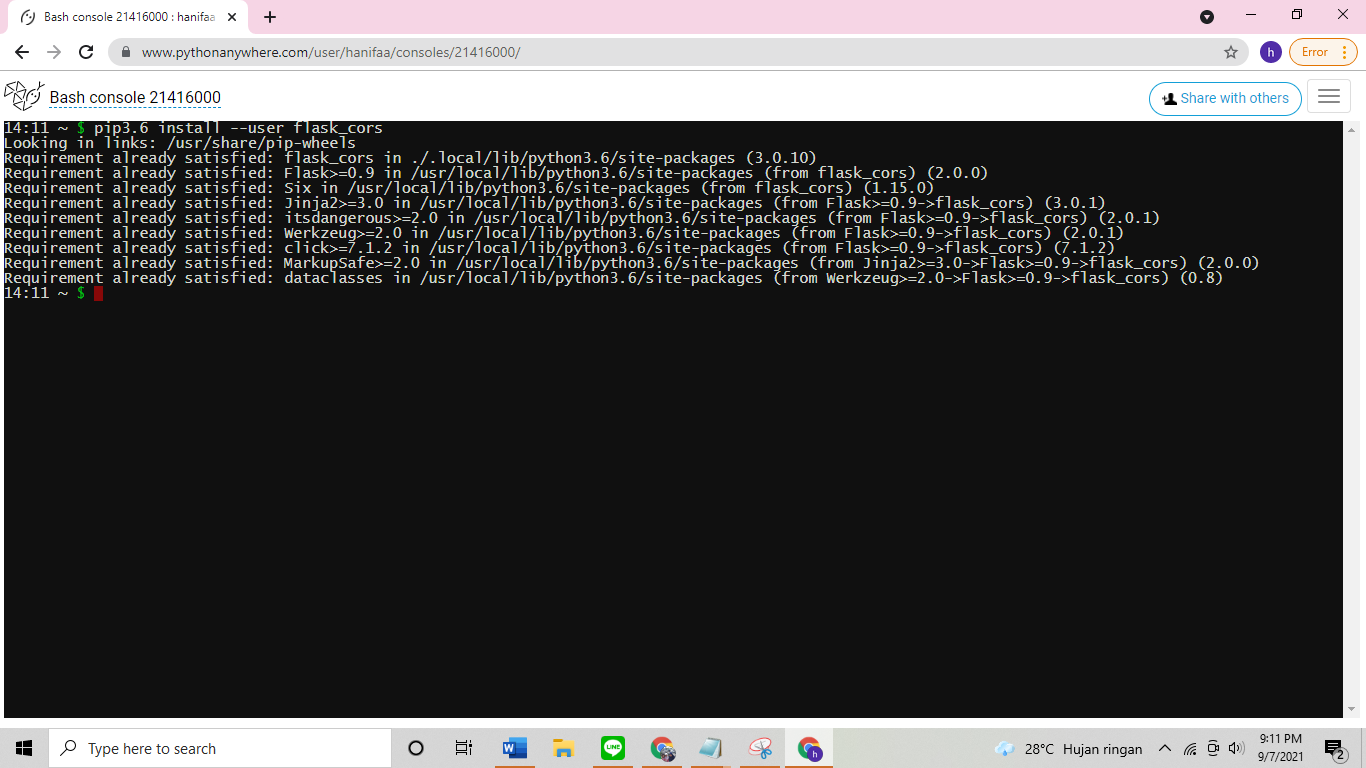
Gambar 10 ketika keluar

* Pythonanywhere

PythonAnywhere adalah lingkungan pengembangan terintegrasi online (IDE) dan layanan hosting web menggunakan bahasa pemrograman Python. Didirikan oleh Giles Thomas dan Robert Smithson pada 2012, ia menyediakan akses dalam browser ke antarmuka baris perintah Python dan Bash berbasis server, bersama dengan editor kode dengan penyorotan sintaks. File program dapat ditransfer ke dan dari layanan menggunakan browser pengguna. Aplikasi web yang dihosting oleh layanan dapat ditulis menggunakan kerangka aplikasi berbasis WSGI apa pun. Pythonanywhere memiliki fitur flask yang digunakan dalam penelitian ini dan support terhadap PySpark.

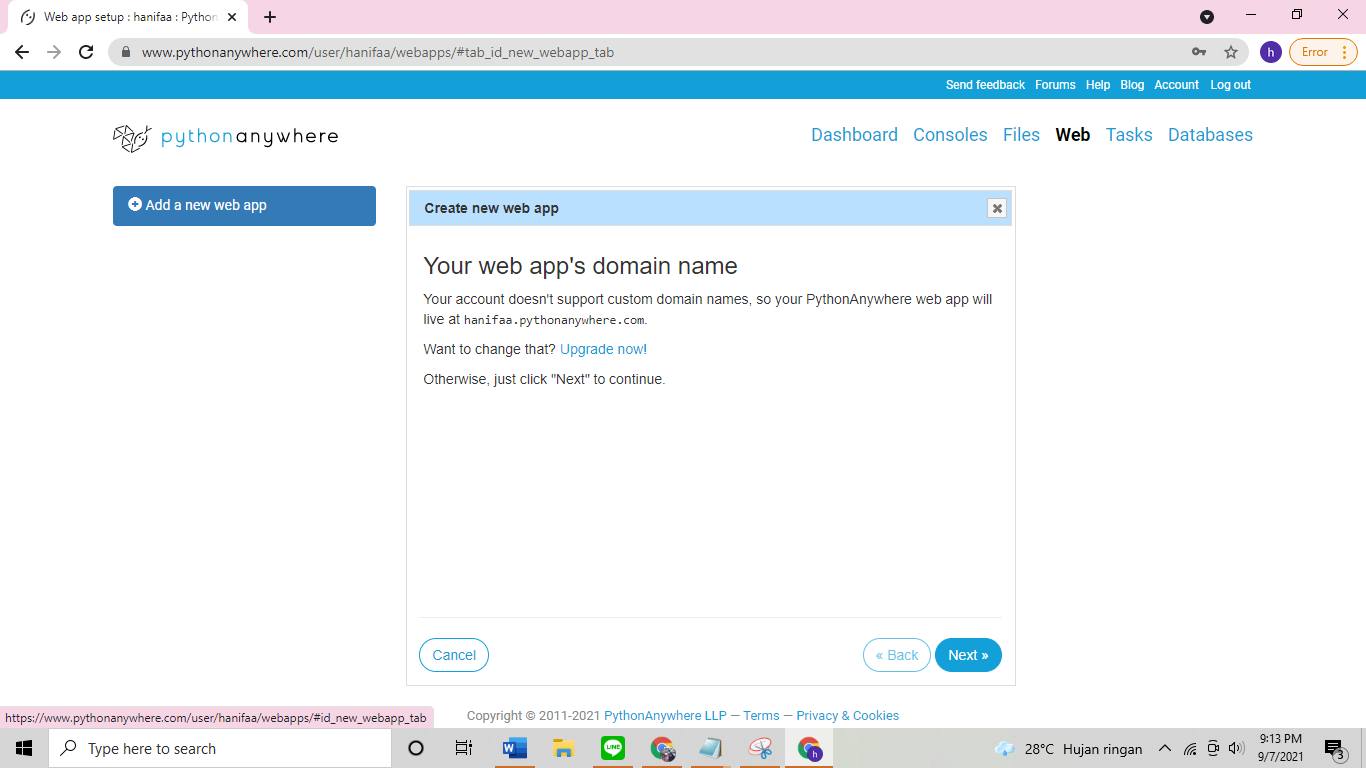
Tata cara memakai pythonanywhere :

1. Install flaskcors terlebih dahulu versi 3.6



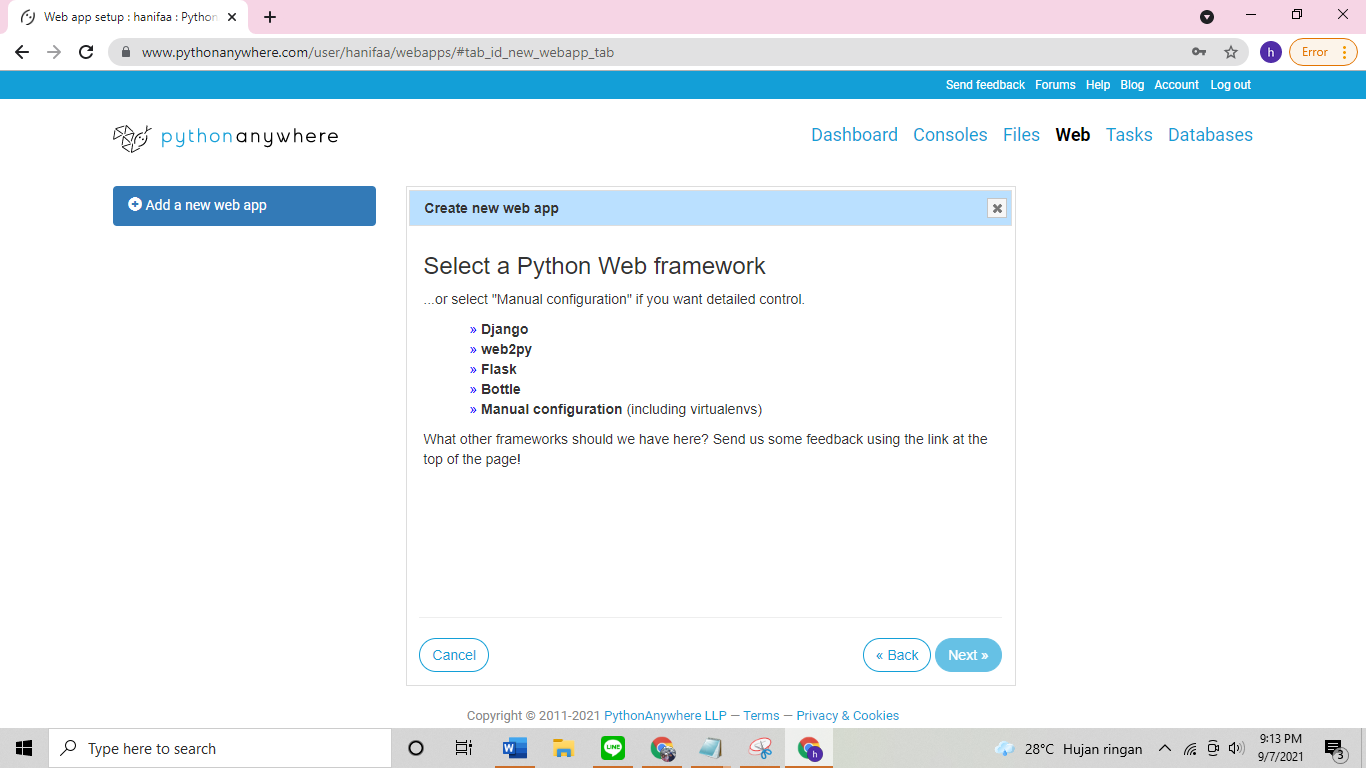
Gambar 11 Install flask cors

1. Menuju ke halaman web



Gambar 12 Membuat website

1. Memilih flask



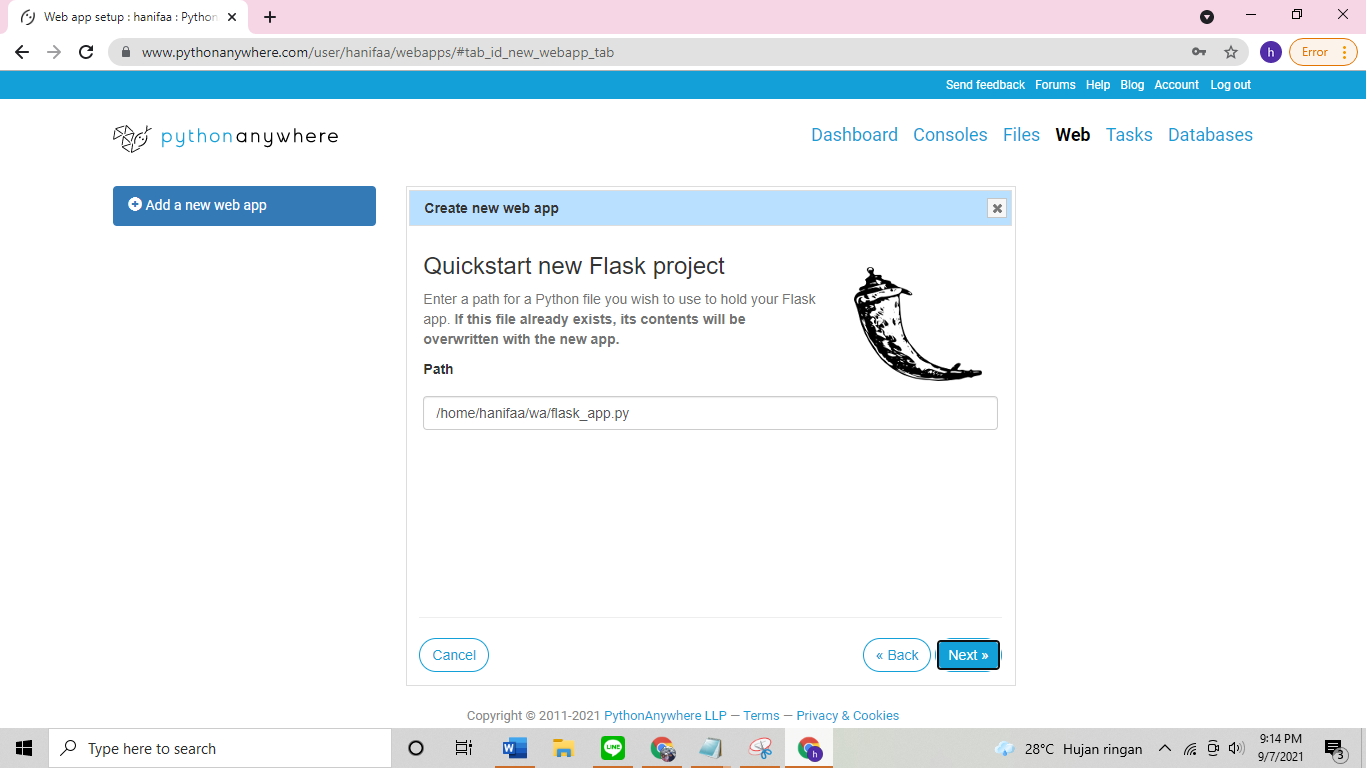
Gambar 13 web framework : flask

1. Memilih versi 3.6



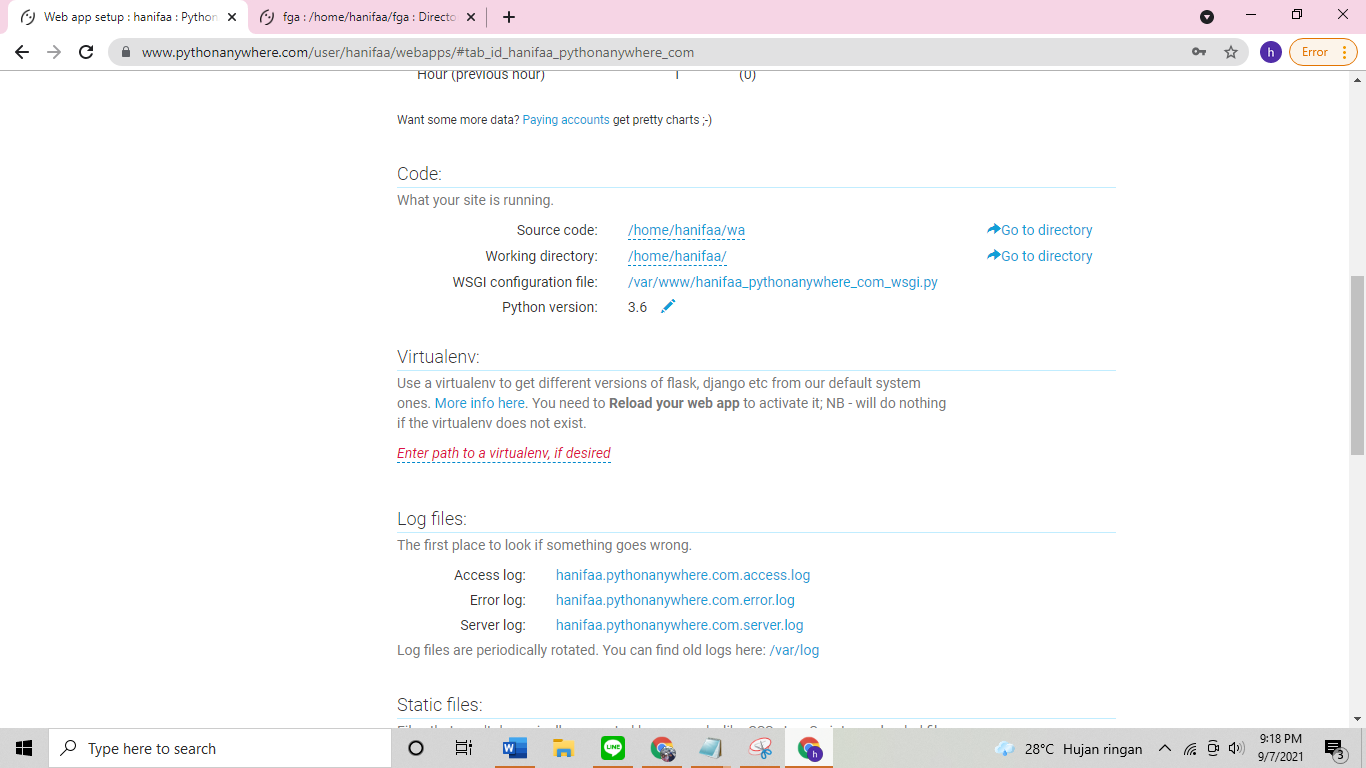
Gambar 14 versi python 3.6

1. Inisialisasi lokasi penyimpanan file



Gambar 15 path kodingan

1. Lalu tekan wsgyi



Gambar 16 WSGYI

Tulis sesuai dengan yang diinginkan yaitu nama file lokasi dan nama aplikasi. Selesai website bisa dijalankan.



Gambar 17 kodingan WSGI

* PySpark

Spark adalah engine analitik umum (general engine) yang cepat dalam pemrosesan large-scale Big Data. Spark merupakan general purpose cluster engine yang mendukung konsep sistem terdistribusi dengan application programming interface (APIs). Penelitian ini menggunakan bahasa python. Spark disebut juga dengan “Lightning Fast Cluster Computing”. Spark 100x lebih cepat dari Hadoop MapReduce pada memory, dan 10x lebih cepat pada disk

Install pyspark

1. Download library pyspark di ttps://github.com/H4nifa/pyspark
2. Upload di file pythonanywhere
3. Lalu di ektrak menggunakan sintax tar -xf /home/hanifasplit1.tar -C ./
4. Setelah selesai, masuk ke dalam file tersebut dan buat file jars
5. Lalu masuk ke file dengan sintax cd ./jars/
6. 3 file jars di ekstrak ke dalam file jars.
7. Di file bashrsc ubah sintax ini sesuai dengan kebutuhan anda: export SPARK\_HOME=/home/hanifa/spark-3.1.2-bin-hadoop3.2

export PYSPARK\_PYTHON=python3.6

1. Di file findspark ubah sintax ini sesuai dengan kebutuhan anda:

os.environ["SPARK\_HOME"] ="/home/hanifa/spark-3.1.2-bin-hadoop3.2"

os.environ["PYSPARK\_PYTHON"] = "python3.6"

1. Install pip 3.6 install –user findspark
2. tulis pyspark, jika berhasil terinstall maka akan muncul seperti ini :



Gambar 18. Install PySpark

Tabel 3 regresi linear menggunakan PySpark

| No | Regresi linier, spark |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | @app.route("/PRC", methods=["GET", "POST"])  def PRC():  import numpy as np  import matplotlib.pyplot as plt  import pandas as pd  import os  BASE\_DIR = os.path.dirname(os.path.abspath(\_\_file\_\_))  url = os.path.join(BASE\_DIR, "reall.csv")  import findspark  findspark.init()  from pyspark.sql import SparkSession  spark = SparkSession.builder.appName("Linear Regression Model").getOrCreate()  from pyspark.ml.regression import LinearRegression  from pyspark.ml.linalg import Vectors  from pyspark.ml.feature import VectorAssembler  from pyspark.ml.feature import IndexToString, StringIndexer  from pyspark import SQLContext, SparkConf, SparkContext  from pyspark.sql import SparkSession  sc = SparkContext.getOrCreate()  if (sc is None):  sc = SparkContext(master="local[\*]", appName="Linear Regression")  spark = SparkSession(sparkContext=sc)  sqlcontext = SQLContext(sc)  data = sqlcontext.read.csv(url, header = True, inferSchema = True)  from pyspark.ml.feature import VectorAssembler  dataset = data.select(data.usia , data.berat, data.tinggi, data.leher, data.dada, data.perut, data.pinggul, data.paha, data.lutut, data.pk, data.bisep, data.lb, data.pt, data.lemak.alias('label'))  training, test = dataset.randomSplit([0.7, 0.3], seed = 100)  assembler = VectorAssembler().setInputCols(['usia','berat','tinggi','dada', 'perut', 'pinggul', 'paha', 'lutut','pk','bisep','lb','pt',]).setOutputCol('features')  trainingSet = assembler.transform(training)  trainingSet = trainingSet.select("features","label")  from pyspark.ml.regression import LinearRegression  lr = LinearRegression()  lr\_Model = lr.fit(trainingSet)  testSet = assembler.transform(test)  testSet = testSet.select("features", "label")  testSet = lr\_Model.transform(testSet)  from pyspark.ml.evaluation import RegressionEvaluator  evaluator = RegressionEvaluator()  y\_pred2 = testSet.select("prediction")  return render\_template('PRC.html', y\_aktual = y\_pred2.rdd.flatMap(lambda x: x).collect(), y\_prediksi = y\_pred2.rdd.flatMap(lambda x: x).collect(), mape = evaluator.evaluate(testSet, {evaluator.metricName: "r2"})) |

Tabel 4 penjelasan kodingan PySpark

| No | Penjelasan Pyspark |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | membuat route prc (pyspark real case)  membuat fungsi prc  import library numpy  import library matplotib  import library pandas  import library os  inisialisasi lokasi penyimpanan  url = file reall.csv di lokasi penyimpanan  import library findspark  memanggil fungsi findspark.init()  import library sparksession  memberi nama pada aplikasi spark : linear regresi  import library linear regresi versi pyspark  import library vectors  import library vector assembler  import library indexto string, stringindexer  import library sqlcontext, sparkconf, sparkcontext  import library sparksession  variabel sc memanggil fungsi spark context  jika sc tidak ada  Meberi nilai sc sehingga aplikasi Apache Spark untuk dapat mengakses cluster dengan bantuan dari Cluster Manager  Titik masuk ke pemrograman Spark dengan Dataset dan DataFrame API.  Titik masuk untuk bekerja dengan data terstruktur (baris dan kolom) di Spark 1.x.  Spark akan secara otomatis menelusuri file csv dan menyimpulkan skema setiap kolom.  import library vectorassembler  mendefinisikan variabel x dan y  data latih sebanyak 70% dan data uji 30%  mengubah fitur menjadi vektor  memilih kolom yang akan di vektorisasi  import library linearregression  inisialisasi fungsi linear regresi  memasukkan data latih ke regresi  assembler : fitur menjadi vektor  memilih kolom fitur (x) dan label (y)  memasukkan data latih ke linear regression  import library regressi evaluasi  memanggil fungsi evaluasi regresi  variabel y=hasil prediksi  menuju ke front end prc.html dengan data y aktual, y prediksi, nilai mape |

* Non Pyspark

Tabel 5 Kodingan Regresi Linear

| No | Regresi linier, nonspark, flask |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | @app.route("/NRC", methods=["GET", "POST"])  def NRC():  import numpy as np  import pandas as pd  import os.path  import joblib  BASE\_DIR = os.path.dirname(os.path.abspath(\_\_file\_\_))  url = os.path.join(BASE\_DIR, "reall.csv")  dataset = pd.read\_csv(url)  X = dataset.iloc[:, :-1].values  y = dataset.iloc[:, 1].values  from sklearn.model\_selection import train\_test\_split  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 1/4, random\_state = 0)  from sklearn.linear\_model import LinearRegression  regressor = LinearRegression()  regressor.fit(X\_train, y\_train)  myModelReg = regressor.fit(X\_train, y\_train)  with open (os.path.join(BASE\_DIR, "real.joblib.pkl"), 'wb') as f: joblib.dump(myModelReg, f, compress=9)  with open (os.path.join(BASE\_DIR, "real.joblib.pkl"), 'rb') as f : myModelReg\_load = joblib.load(f)  y\_pred = myModelReg\_load.predict(X\_test)  y\_pred2 = myModelReg\_load.predict(X\_train)  aktual, predict = y\_train, y\_pred2  mape = np.sum(np.abs(((aktual - predict)/aktual)\*100))/len(predict)  return render\_template('NRC.html', y\_aktual = list(y\_train), y\_prediksi = list(y\_pred2), mape = mape) |

Tabel 6 Penjelasan Kodingan Regresi Linear

| No | Non Pyspark |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | membuat route NRC (Nonpyspark real case)  membuat fungsi NRC  import library numpy  import library pandas  import library os.path  import library joblib  inisialisasi lokasi penyimpanan  url = file reall.csv di lokasi penyimpanan  membaca isi file  menentukan nilai x dengan mengurangi 1 kolom  menentukan nilai y dengan mengambil 1 kolom  import library train\_test\_split  data latih 75% dan data uji 25%  import library linear regresi  inisialisasi fungsi linear regresi  memasukkan variabel x dan y data latih  memproses linear regresi  membuka file real.joblib.pkl  memasukkan file real.joblib.pkl dan memasukan nilai nya  variabel y = prediksi dari variabel x uji  variabel y2 = prediksi dari variabel x latih  membuat rumus mape  menuju front end nrc.html dan mengirim data y aktual, y prediksi, dan mape |

* RESTFUL API

RESTful API adalah sebuah web service yang diimplementasikan dengan menggunakan HTTP dan prinsip REST (Representational State Transfer), dan sumber dayanya tersimpan di penyimpanan data. Web service ini digunakan sebagai media pertukaran data antara sisi klien dengan sisi server yang menyimpan basis data.

Penggunaan restful api :

@app.route("/api/fp/3.0/", methods=["GET", "POST"])

def api():

from flask import jsonify

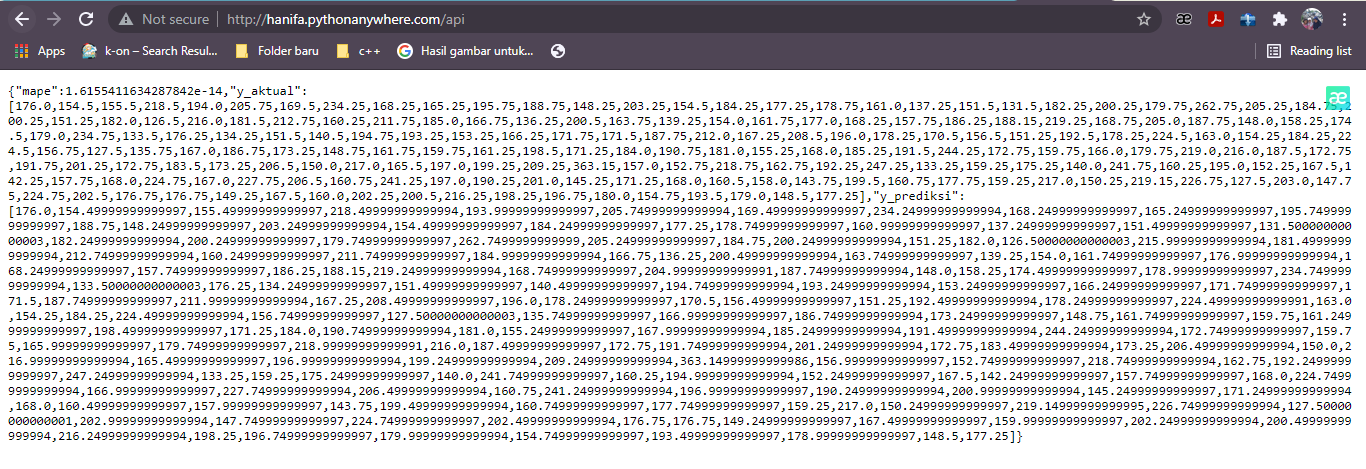
from flask import Response

……..(sama seperti kodingan pyspark/non)…..

response = jsonify({'y\_aktual': list(y\_train), 'y\_prediksi': list(y\_pred2), 'mape': mape})

response.headers.add("Access-Control-Allow-Origin", "\*")

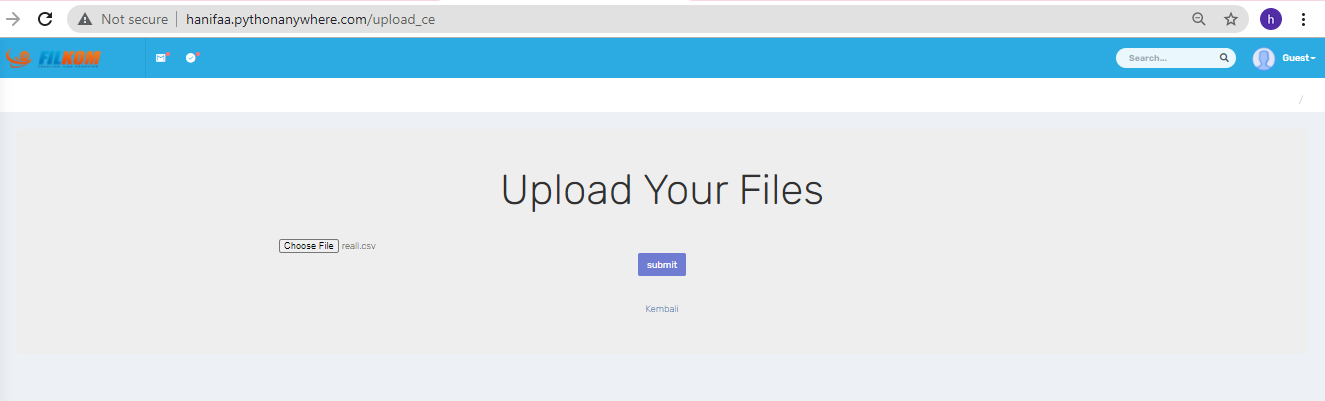
return response



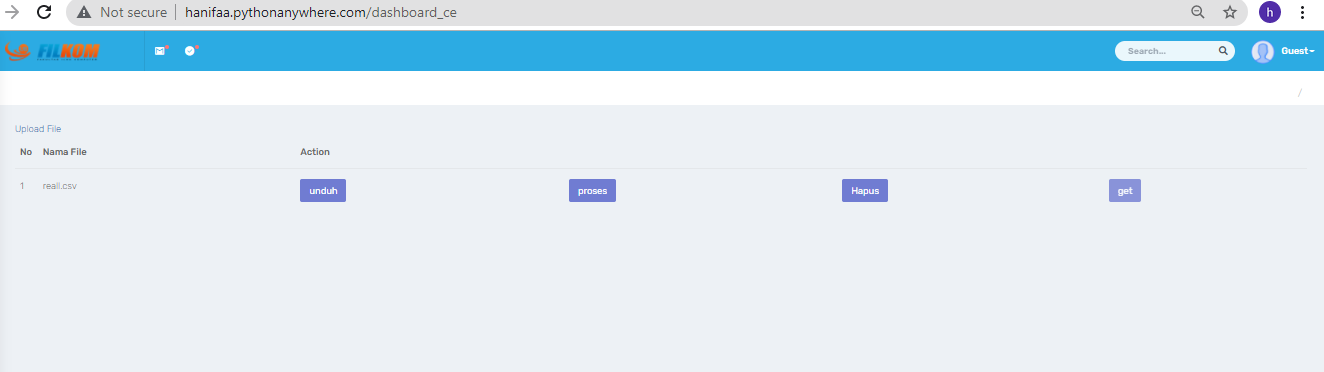
Gambar 19. Tampilan restful api

* Compute Engine

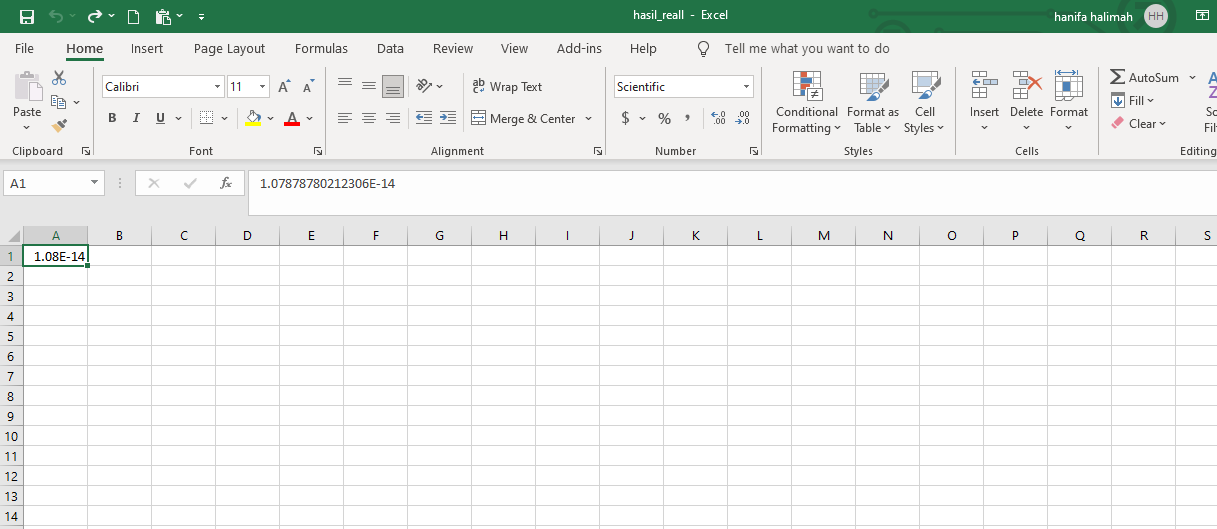
Compute Engine adalah layanan yang menyediakan virtual machine, penelitian ini memakai aplikasi web dengan menggunakan microframework flask yang cocok digunakan pada embedded system. Pada penelitian ini penyimpanan lokal menggunakan database SQLite yang cocok digunakan pada perangkat dengan spesifikasi rendah.



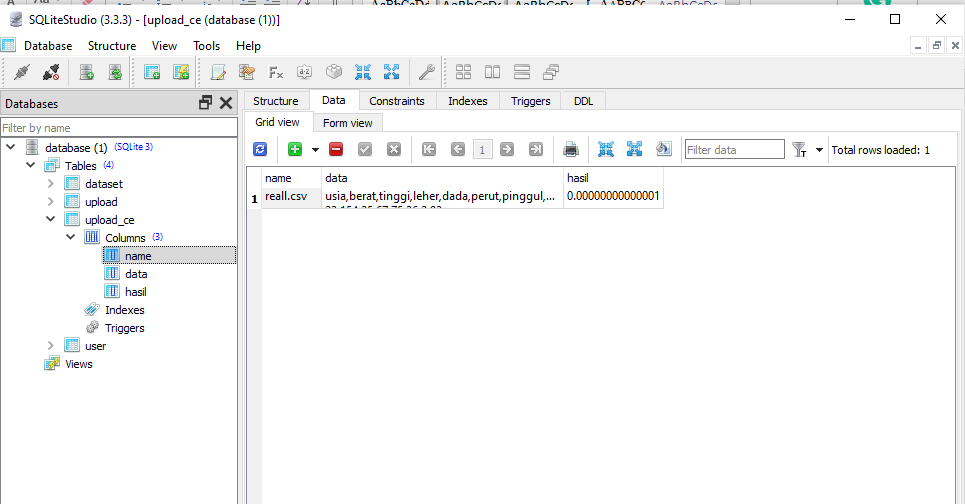
Gambar 20 halaman upload\_ce



Gambar 21 halaman dashboard



Gambar 22 tampilan di hasil.csv



Gambar 23 tampilan di database

Tabel 7 kodingan aplikasi compute engine

| No. | App.py |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | def database\_ce(name, data, hasil):  conn = connect\_db()  db = conn.cursor()  db.execute(“””CREATE TABLE IF NOT EXISTS upload\_ce (name TEXT,data BLOP, hasil BLOP) “””)  db.execute(“””INSERT INTO upload\_ce (name, data, hasil) VALUES (?,?,?) “””,(name,data,hasil))  conn.commit()  db.close()  conn.close()  @app.route(‘/dashboard\_ce’)  def dashboard\_ce():  conn = connect\_db()  db = conn.cursor()  print(“IN DATABASE FUNCTION “)  c = db.execute(“”” SELECT \* FROM upload\_ce “””)  mydata = c.fetchall()  for x in c.fetchall():  name\_v=x[0]  data\_v=x[1]  hasil\_v=x[2]  break  hasil = []  for v\_login in c:  hasil.append(v\_login)  conn.commit()  db.close()  conn.close()  return render\_template(‘dashboard\_ce.html’, header = mydata)  @app.route(‘/upload\_ce’, methods=[“GET”, “POST”])  def upload\_ce():  form = UploadForm()  if request.method == “POST”:  if form.validate\_on\_submit():  file\_name = form.file.data  database\_ce(name=file\_name.filename, data=file\_name.read(), hasil=None )  return redirect(url\_for(“dashboard\_ce”))  return render\_template(“upload\_ce.html”, form=form)  @app.route(‘/unduh\_ce/file/’, methods=[“GET”])  def unduh\_ce():  name = request.args.get(‘name’)  conn = connect\_db()  db = conn.cursor()  c = db.execute(“SELECT \* FROM upload\_ce WHERE name =\’”+ name +”\’”)  for x in c.fetchall():  name\_v=x[0]  data\_v=x[1]  break  conn.commit()  db.close()  conn.close()  return send\_file(BytesIO(data\_v), attachment\_filename=name\_v, as\_attachment=True)  @app.route(‘/hapus\_ce/file/’, methods=[“GET”])  def hapus\_ce():  name = request.args.get(‘name’)  conn = connect\_db()  db = conn.cursor()  db.execute(“DELETE FROM upload\_ce WHERE name =\’”+ name +”\’”)  conn.commit()  db.close()  conn.close()  return redirect(url\_for(“dashboard\_ce”))  def read\_ce(name):  conn = connect\_db()  db = conn.cursor()  c = db.execute(“SELECT \* FROM upload\_ce WHERE name =\’”+ name +”\’”)  for x in c.fetchall():  name\_v=x[0]  data\_v=x[1]  break  conn.commit()  db.close()  conn.close()  return BytesIO(data\_v)  @app.route(‘/proses\_ce/file/’, methods=[“GET”, “POST”])  def proses\_ce():  name = request.args.get(‘name’)  import numpy as np  import pandas as pd  dataset = pd.read\_csv(read\_ce(name))  X = dataset.iloc[:, :-1].values  y = dataset.iloc[:, 1].values  from sklearn.model\_selection import train\_test\_split  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 1/3, random\_state = 0)  from sklearn.linear\_model import LinearRegression  regressor = LinearRegression()  regressor.fit(X\_train, y\_train)  # Predicting the Test set results  y\_pred = regressor.predict(X\_test)  y\_pred2 = regressor.predict(X\_train)  ctual, predict = y\_train, y\_pred2  mape = np.sum(np.abs(((ctual – predict)/ctual)\*100))/len(predict)  conn = connect\_db()  db = conn.cursor()  db.execute(“UPDATE upload\_ce SET hasil = ? WHERE name = ?”,(mape,name))  conn.commit()  db.close()  conn.close()  return redirect(url\_for(“dashboard\_ce”))  @app.route(‘/get\_ce/file/’, methods=[“GET”])  def get\_ce():  name = request.args.get(‘name’)  conn = connect\_db()  db = conn.cursor()  c = db.execute(“SELECT \* FROM upload\_ce WHERE name =\’”+ name +”\’”)  for x in c.fetchall():  name\_v=”hasil\_”+x[0]  hasil\_v=x[2]  break  conn.commit()  db.close()  conn.close()  rubah\_ke\_string = str(hasil\_v)  rubah\_ke\_bytes = str.encode(rubah\_ke\_string)  return send\_file(BytesIO(rubah\_ke\_bytes), attachment\_filename=name\_v, as\_attachment=True) |

Tabel 8 penjelasan kodingan aplikasi compute engine

|  | Penjelasan app.py |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | Membuat fungsi database  Membuat Tabel Upload\_ce jika tidak ada  Memasukkan data ke dalam tabel Upload\_ce terdiri dari nama : file nama, data : file isi data, hasil: nilai mape  Membuat route dashboard\_ce  Menampilkan tabel upload\_ce dari database  menampilkan front end dashboard\_ce.html dan data header=mydata  membuat route upload\_ce  mengaktifkan uploadform  jika ditekan  jika file berhasil terupload maka disimpan ke dalam database  menuju ke route dashboard\_ce  menampilkan front end upload\_ce.html dan data form=form  membuat route unduh\_ce  menyimpan nama file yang diinginkan  menampilkan file dari database  mengirim data file dan terdownload.  membuat route hapus  menyimpan nama file yang diinginkan  menghapus nama file di database  menuju ke route dashboard  membuat fungsi read  menampilkan file dari database  kembali ke route sebelumnya dengan mengirim data file tersebut  membuat route proses\_ce  menyimpan nama file yang diinginkan  proses regresi linear untuk mendapatkan nilai mape  menyimpan hasil mape di database  menuju ke route dashboard\_ce  membuat route get\_ce  menyimpan nama file yang diinginkan  menampilkan filed ari database  nilai hasil diubah ke string  nilai hasil diubah ke byte  mengirim data file dan terdownload |

Tabel 9 kodingan front end Dashboard\_ce.html

|  | Dashboard\_ce |
| --- | --- |
| 1  2  3  4  5  6  7 | <a href="/upload\_ce">  <span style="display: block;">  Upload File  </span>  </a>  <table class="table table-hover" data-height="10" table-bordered">  <thead>  <tr>  <th style="width: 10px">No</th>  <th>Nama File</th>  <th colspan="2" style="width: 40px">Action</th>  </tr>  </thead>  <tbody>  {% for item in header %}  <tr>  <td>{{ loop.index }}</td>  <td>{{item[0]}}</td>    <td>  <a href="/unduh\_ce/file/?name={{item[0]}}">  <button type="submit" class="btn btn-primary moss-3">unduh</button>  </a>  </td>  <td>  <a href="/proses\_ce/file/?name={{item[0]}}">  <button type="submit" class="btn btn-primary moss-3">proses</button>  </a>  </td>  <td>  <a href="/hapus\_ce/file/?name={{item[0]}}">  <button type="submit" class="btn btn-primary moss-3">Hapus</button>  </a>  </td>  <td>  <a href="/get\_ce/file/?name={{item[0]}}">  <button type="submit" class="btn btn-primary moss-3">get</button>  </a>  </td> |

Tabel 10 penjelasan kodingan front end Dashboard\_ce.html

| No | Penjelasan Dashboard |
| --- | --- |
| 1  2  3  4  5  6  7 | tombol upload file menuju ke route upload\_ce  membuat tabel sesuai ukuran diinginkan  menampilkan data file dari database  tombol unduh menuju ke route unduh  tombol proses menuju ke route proses  tombol hapus menuju ke route hapus  tombol unduh get ke route get |

Tabel 11 kodingan front end Upload\_ce.html

|  | Upload\_ce |
| --- | --- |
| 1  2  3 | <div class="container">  <h1 class="display-4"> Upload Your Files </h1>  <br>  <br>  <div class="">  <form class="" action="{{ url\_for('upload\_ce') }}" method="POST", enctype="multipart/form-data">  {{ form.hidden\_tag() }}  {{ form.file() }}  {{ form.submit( class="btn btn-primary" ) }}  </form>  </div>  <br>  <br>  <a href="/dashboard\_ce">  <span style="display: block;">  Kembali  </span>  </a>  </div> |

Tabel 12 penjelasan kodingan front end Upload\_ce.html

| No | Penjelasan Upload\_ce |
| --- | --- |
| 1  2  3 | Membuat tombol upload file  Ketika udah keupload maka memanggil route upload\_ce  Tombol kembali menuju route dashboard\_ce |

* Membuat Toko Online sederhana

Toko online pet lovers yang berisi barang yang diperlukan oleh hewan peliharaan, dokter hewan, hewan peliharaan, dan donasi. Donasi diperlukan untuk hewan peliharaan yang berada di luar rumah, seperti menangani populasi dengan cara kb, memberi makanan, dan menolong hewan peliharaan yang sakit.

Tabel 13 Aplikasi toko PetLovers

|  | PetLovers app.py |
| --- | --- |
| 1  2  3  4  5  6  7 | @app.route("/PetLovers", methods=["GET", "POST"])  def PetLovers():  conn = connect\_db()  db = conn.cursor()  return render\_template("bli.html")  @app.route("/makanan", methods=["GET", "POST"])  def makanan():  conn = connect\_db()  db = conn.cursor()  return render\_template("makanan.html")  @app.route("/obat", methods=["GET", "POST"])  def obat():  conn = connect\_db()  db = conn.cursor()  return render\_template("obat.html")  @app.route("/peralatan", methods=["GET", "POST"])  def peralatan():  conn = connect\_db()  db = conn.cursor()  return render\_template("peralatan.html")  @app.route("/donasi", methods=["GET", "POST"])  def donasi():  conn = connect\_db()  db = conn.cursor()  return render\_template("donasi.html")  @app.route("/dokter", methods=["GET", "POST"])  def dokter():  conn = connect\_db()  db = conn.cursor()  return render\_template("dokter.html")  @app.route("/hewan", methods=["GET", "POST"])  def dokter():  conn = connect\_db()  db = conn.cursor()  return render\_template("dokter.html") |

Tabel 14 Penjelasan aplikasi toko PetLovers

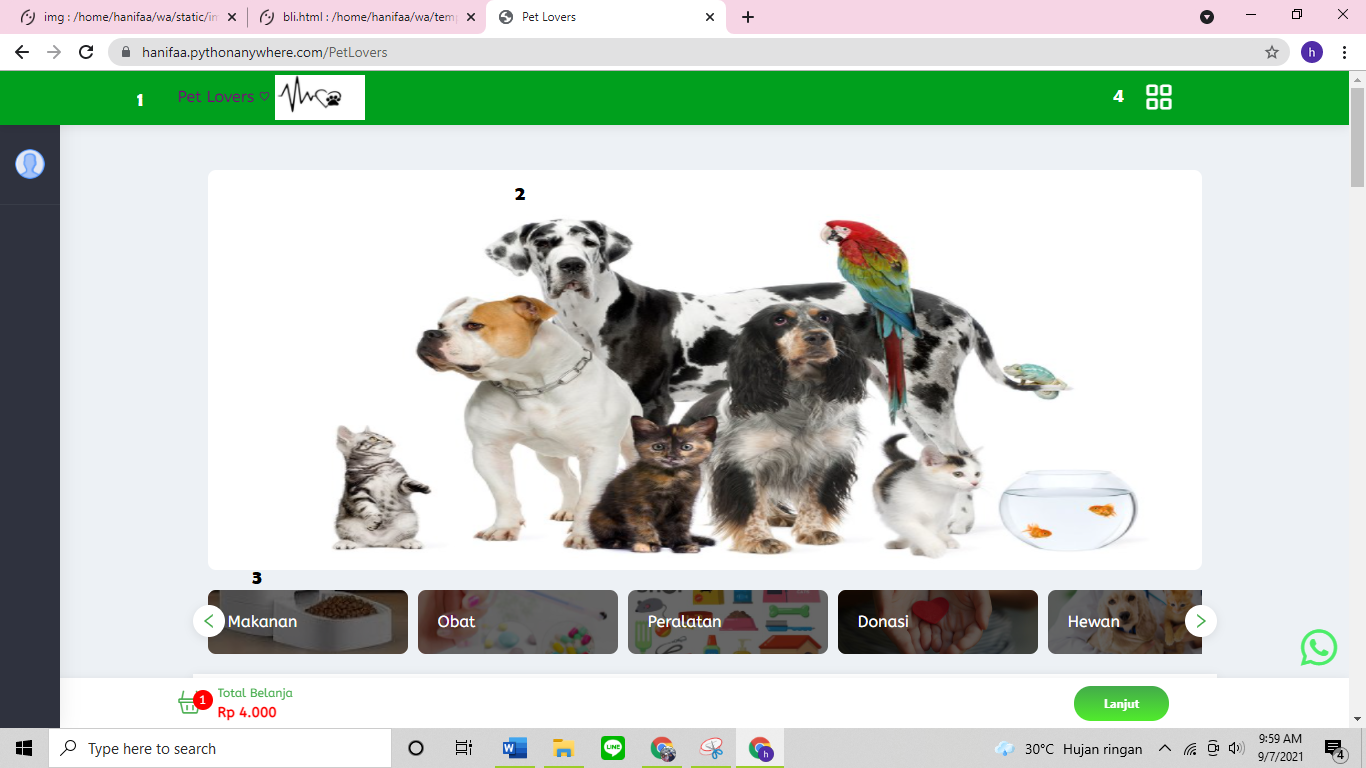
| No | Penjelasan app.py |
| --- | --- |
| 1  2  3  4  5  6  7 | Route halaman utama PetLovers  Route makanan  Route obat  Route Peralatan  Route Donasi  Route Dokter  Route Hewan |

Bli.html :

Menulis variabel yang konstan tidak dapat dirubah seperti nama toko, nomer wa penjual, ongkir, kalimat awal pesan wa dapat dilihat pada tabel 15 .

Tabel 15 nilai tetap

| const main = {  "site\_url":" http://hanifaa.pythonanywhere.com/PetLovers ",  "ajax\_url":"http://hanifaa.pythonanywhere.com/PetLovers/wp-admin/admin-ajax.php",  "currency": new Intl.NumberFormat('id-ID', {  style: 'currency',  currency: 'IDR',  minimumFractionDigits: 0  }),  "nonce": "f935bd1fd8",  "font\_uri" : "//fonts.googleapis.com/css?family=ABeeZee%3Aregular%2Citalic%26subset%3Dlatin%2C",  "store\_id" : "999999",  "store\_name" : "PetLovers",  "store\_link" : "https://hanifaa.pythonanywhere.com/PetLovers",  "store\_ongkir" : "5000",  "store\_ongkir\_name" : "Flat Ongkir",  "store\_ongkir\_enable" : true,  "store\_ongkir\_provider" : "flatongkir",  "store\_admin\_phone" : "6285745149829",  "store\_opened\_message" : "Halo Admin!",  }; |
| --- |



Gambar 24 halaman utama PetLovers

1. Membuat kotak untuk header, didalam header terdapat tulisan petlovers♡, gambar logo yang mana jika ditekan akan menuju halaman utama dan tombol untuk navigasi bar :

Tabel 16 Header

| <div class="headerbox">  <div class="logo-img">  <a href="/PetLovers" style="text-decoration: none;">  Pet Lovers ♡  <img class="lazy loaded" data-src="{{ url\_for('static', filename = 'img/cat.png') }}" src="{{ url\_for('static', filename = 'img/cat.png') }}" data-was-processed="true" width="auto" height="auto"> |
| --- |

1. Untuk menampikan gambar halaman utama:

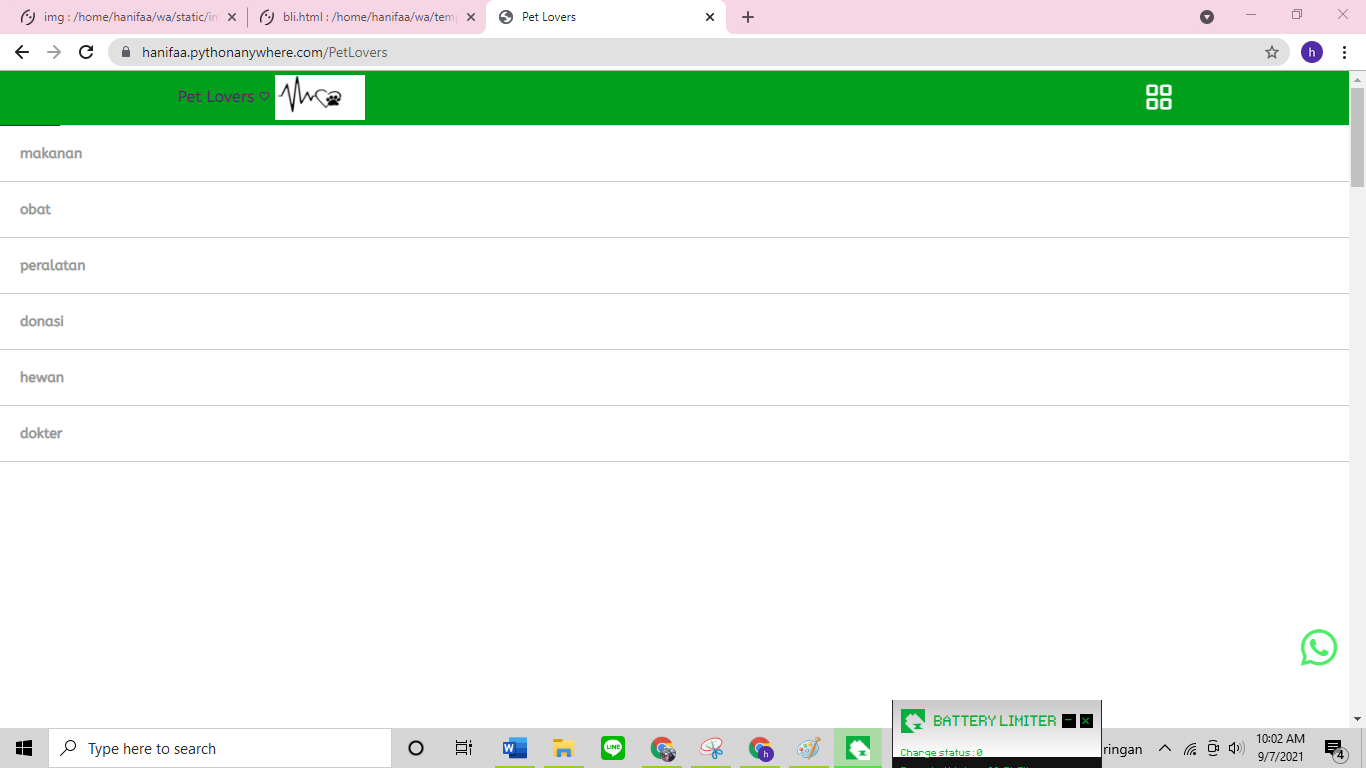
Tabel 17 Gambar utama

| <img class="tns-lazy-img loaded tns-complete" data-src="{{ url\_for('static', filename = 'img/foto.jpg') }}" src="{{ url\_for('static', filename = 'img/foto.jpg') }}"> |
| --- |

1. Menampilkan slide kategori box yang berisi gambar berserta tulisan perkategori yang mana Ketika ditekan akan menuju ke halaman kategori tersebut.

Tabel 18 kotak perkategori

| <div class="index">  <div class="wrapper">  <div id="category-slide" class="slide-categorybox">  <div class="arrow arrow-left" style="display: block;">  <svg viewBox="64 64 896 896" focusable="false" class="" data-icon="left" width="1em" height="1em" fill="currentColor" aria-hidden="true"><path d="M724 218.3V141c0-6.7-7.7-10.4-12.9-6.3L260.3 486.8a31.86 31.86 0 0 0 0 50.3l450.8 352.1c5.3 4.1 12.9.4 12.9-6.3v-77.3c0-4.9-2.3-9.6-6.1-12.6l-360-281 360-281.1c3.8-3 6.1-7.7 6.1-12.6z"></path></svg>  </div>  <div class="slide-category">  <div class="categorybox lazy" data-bg="{{ url\_for('static', filename = 'img/m.JPG') }}" style="background-image: url(&quot;{{ url\_for('static', filename = 'img/m.JPG') }}&quot;);" data-was-processed="true">  <a href="/makanan">  Makanan </a>  </div>  <div class="categorybox lazy" data-bg="{{ url\_for('static', filename = 'img/o.JPG') }}" style="background-image: url(&quot;{{ url\_for('static', filename = 'img/o.JPG') }}&quot;);" data-was-processed="true">  <a href="/obat">  Obat </a>  </div>  <div class="categorybox lazy" data-bg="{{ url\_for('static', filename = 'img/peralatan.JPG') }}" style="background-image: url(&quot;{{ url\_for('static', filename = 'img/peralatan.JPG') }}&quot;);" data-was-processed="true">  <a href="/peralatan">  Peralatan </a>  </div>  <div class="categorybox lazy" data-bg="{{ url\_for('static', filename = 'img/donasi.jpg') }}" style="background-image: url(&quot;{{ url\_for('static', filename = 'img/donasi.jpg') }}&quot;);" data-was-processed="true">  <a href="/donasi">  Donasi </a>  </div>  <div class="categorybox lazy" data-bg="{{ url\_for('static', filename = 'img/hewan.jpg') }}" style="background-image: url(&quot;{{ url\_for('static', filename = 'img/hewan.jpg') }}&quot;);" data-was-processed="true">  <a href="/hewan">  Hewan </a>  </div>  <div class="categorybox lazy" data-bg="{{ url\_for('static', filename = 'img/dokter.jpg') }}" style="background-image: url(&quot;{{ url\_for('static', filename = 'img/dokter.jpg') }}&quot;);" data-was-processed="true">  <a href="/dokter">  Dokter </a>  </div>  </div>  <div class="arrow arrow-right" style="display: block;">  <svg viewBox="64 64 896 896" focusable="false" class="" data-icon="right" width="1em" height="1em" fill="currentColor" aria-hidden="true"><path d="M765.7 486.8L314.9 134.7A7.97 7.97 0 0 0 302 141v77.3c0 4.9 2.3 9.6 6.1 12.6l360 281.1-360 281.1c-3.9 3-6.1 7.7-6.1 12.6V883c0 6.7 7.7 10.4 12.9 6.3l450.8-352.1a31.96 31.96 0 0 0 0-50.4z"></path></svg>  </div>  </div>  </div>  </div> |
| --- |

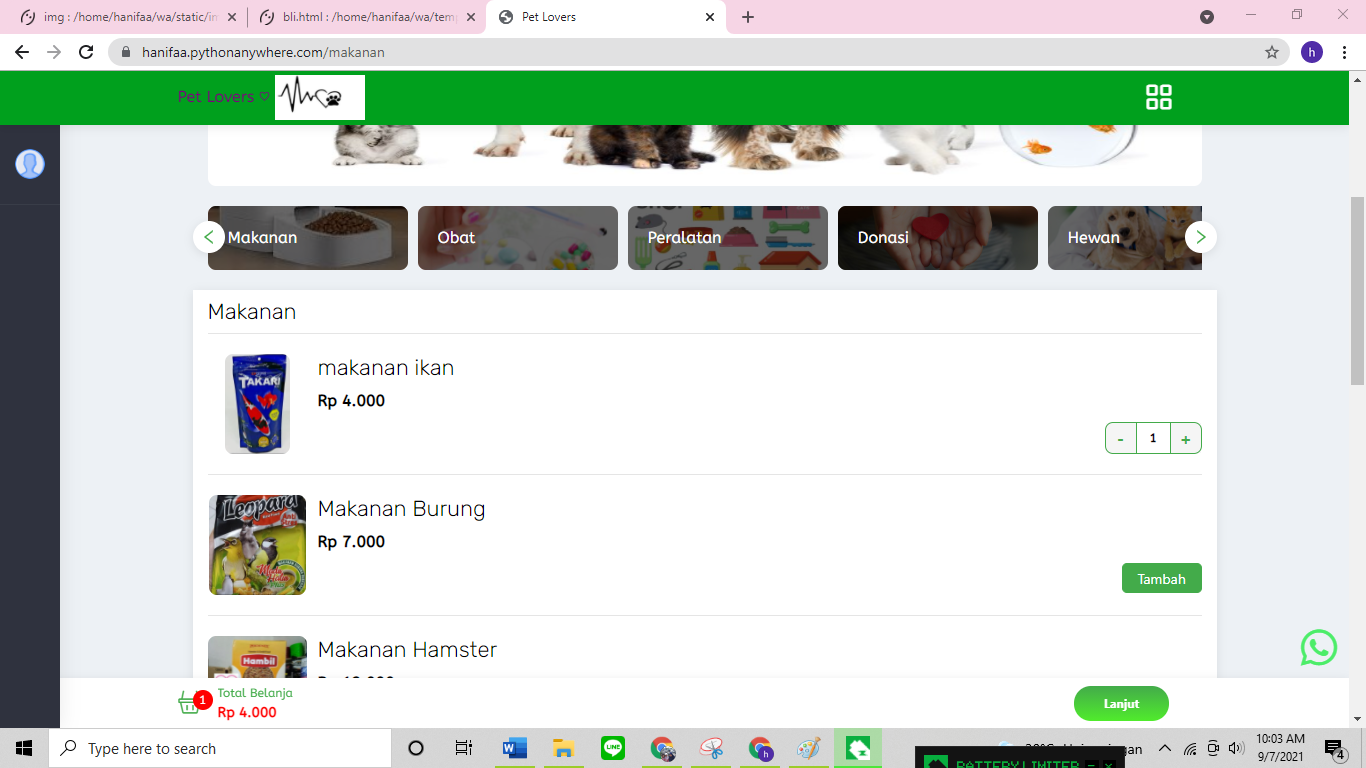


Gambar 25 Navigasi bar

1. Menampilkan navigasi bar yang berisi kategori di toko petlovers, dan ketika di klik akan menuju ke halaman sesuai kategori yang diinginkan

Tabel 19 Navigasi Bar

| <nav id="navigation" class="navigation">  <i id="nav-menu-toggle" class="lni lni-grid-alt"></i>  <div id="nav-menu" class="nav-menu" style="visibility: hidden; opacity: 0; display: none;"><ul id="menu-header" class="menu-header">  <li id="menu-item-257" class="menu-item menu-item-type-custom menu-item-object-custom menu-item-257"><a href="/makanan">makanan</a></li>  <li id="menu-item-258" class="menu-item menu-item-type-custom menu-item-object-custom menu-item-258"><a href="/obat">obat</a></li>  <li id="menu-item-258" class="menu-item menu-item-type-custom menu-item-object-custom menu-item-258"><a href="/peralatan">peralatan</a></li>  <li id="menu-item-257" class="menu-item menu-item-type-custom menu-item-object-custom menu-item-257"><a href="/donasi">donasi</a></li>  <li id="menu-item-257" class="menu-item menu-item-type-custom menu-item-object-custom menu-item-257"><a href="/hewan">hewan</a></li>  <li id="menu-item-257" class="menu-item menu-item-type-custom menu-item-object-custom menu-item-257"><a href="/dokter">dokter</a></li>  </ul></div> </nav> |
| --- |



Gambar 26 kategori makanan

Membuat 1 tabel untuk menampilkan perkategori, diawali dengan nama perkategori <h2>

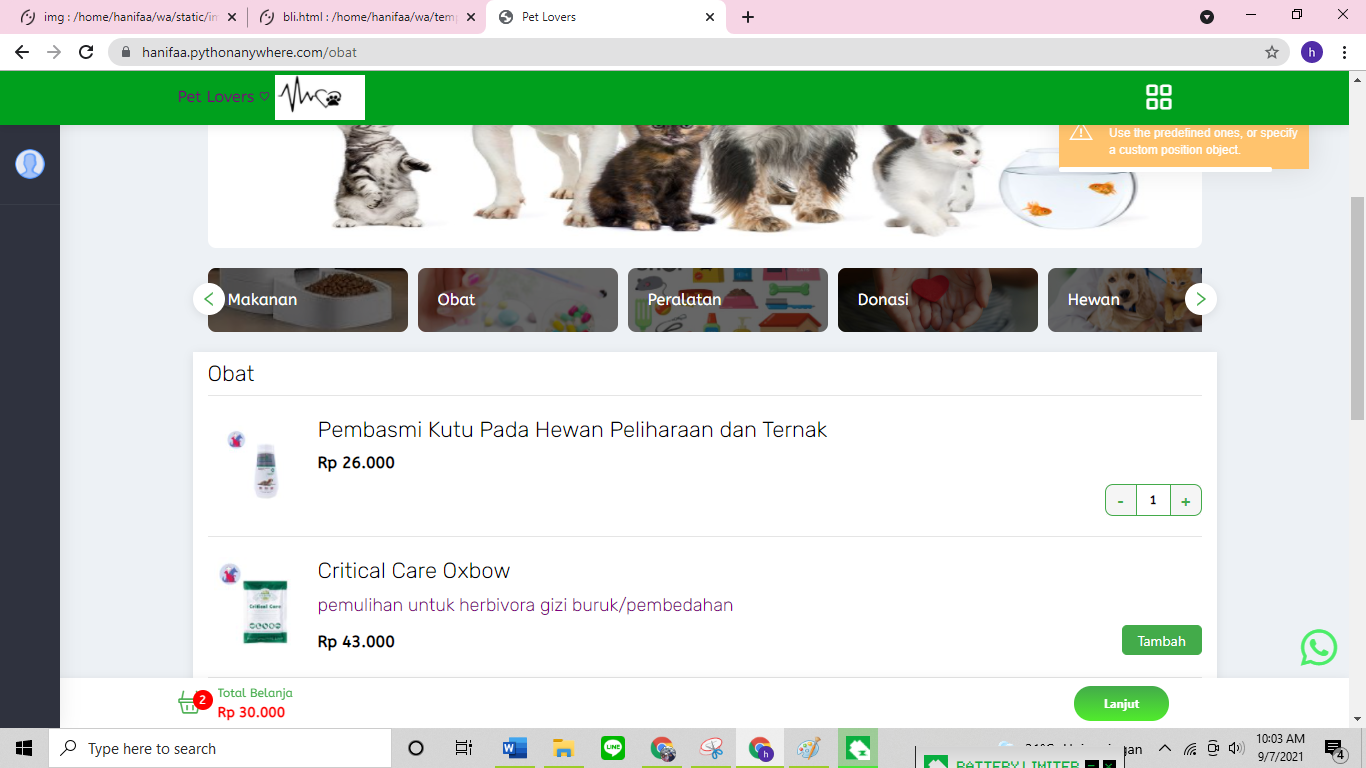
Tabel 20. Judul per kategori dalam satu kotak

| <main id="main" class="query">  <div class="wrapper products-list">  <div class="products-list-title">  <h2>Makanan</h2>  </div>  <div class="products clear"> |
| --- |

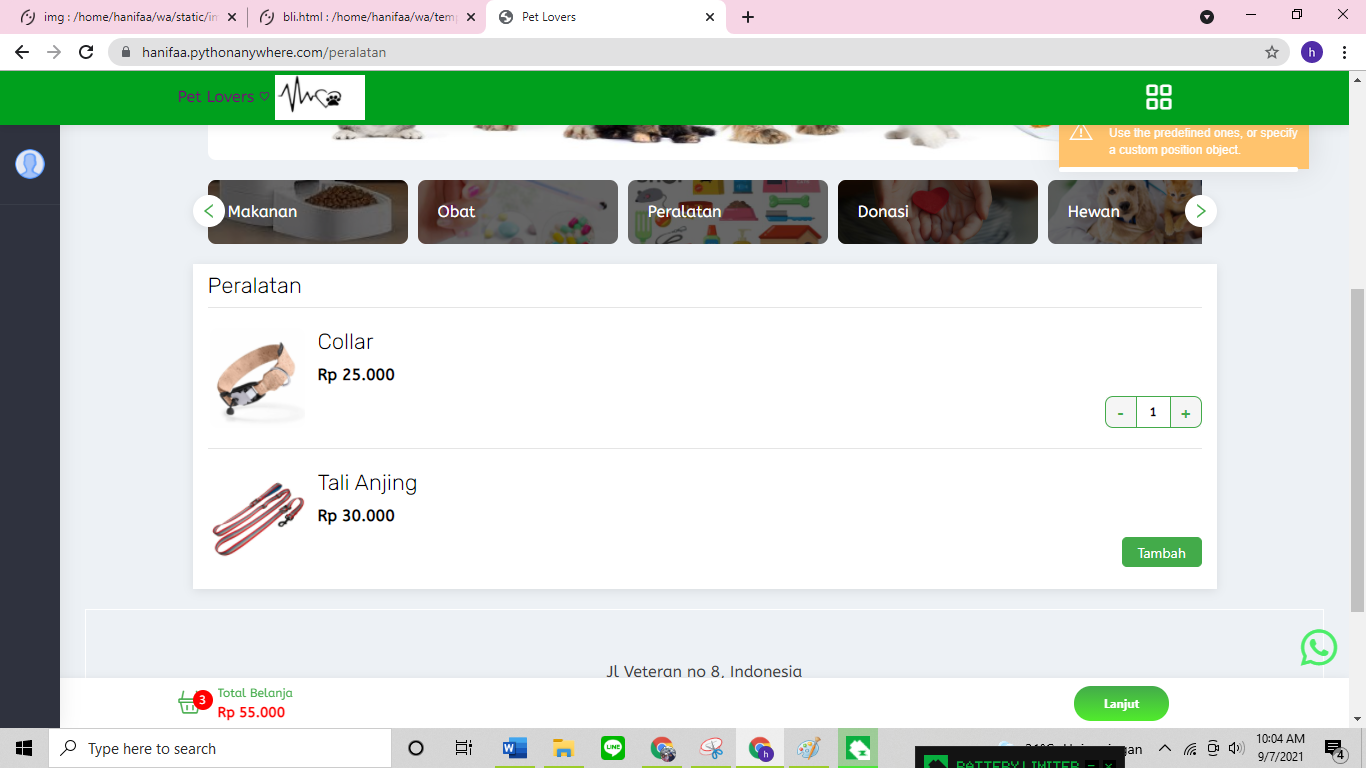
Menampilkan produk yang akan dijual

Tabel 21 Produk

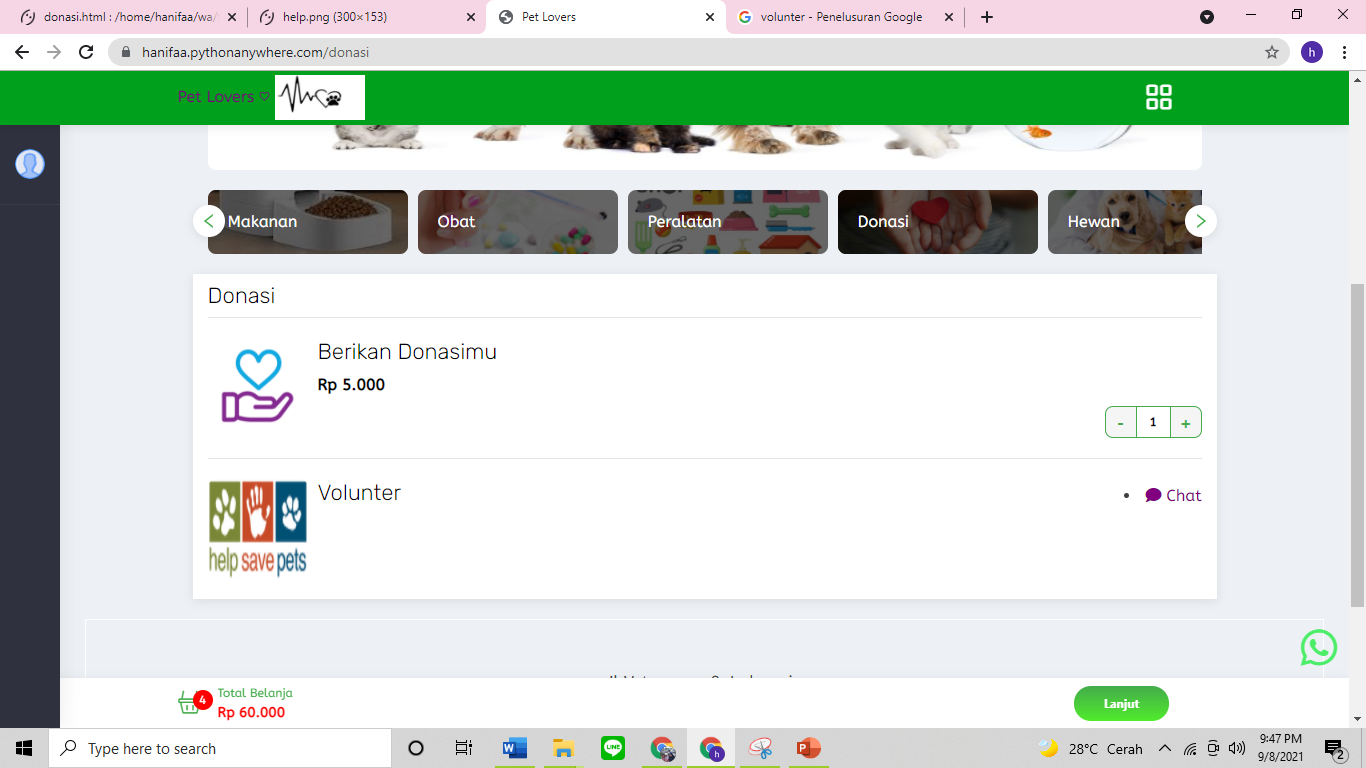
| <article id="product-1" class="productbox">  <div class="content">  <div class="image">  <img class="lazy loaded" data-src="{{ url\_for('static', filename = 'img/m1.JPG') }}" src="{{ url\_for('static', filename = 'img/m1.JPG') }}">  </div>  <div class="detail">  <h3><a href="/makanan">makanan ikan</a></h3>  <div class="desc">  </div>  <div class="pricing">  <span class="price">Rp 4.000</span> </div>  <div class="atc atc-item-1 clear">  <input type="hidden" name="store\_id" value="999999">  <input type="hidden" name="item\_id" value="1">  <input type="hidden" name="item\_image" value="{{ url\_for('static', filename = 'img/m1.JPG') }}">  <input type="hidden" name="item\_name" value="makanan ikan">  <input type="hidden" name="item\_price" value="4000">  <input type="hidden" name="item\_weight" value="500">  <input type="hidden" name="item\_price\_slik" value="4000">  <input type="hidden" name="note" value="">  <button class="add button-add" style="display: block;">Tambah</button>  <div class="qty qty-selector clear" style="display: none;">  <button type="button" class="minus button-qty" data-qty-action="minus">-</button>  <input min="0" type="number" value="0" name="qty">  <button type="button" class="plus button-qty" data-qty-action="plus">+</button>  </div>  </div>  </div>  </div>  </article> |
| --- |



Gambar 27 kategori obat

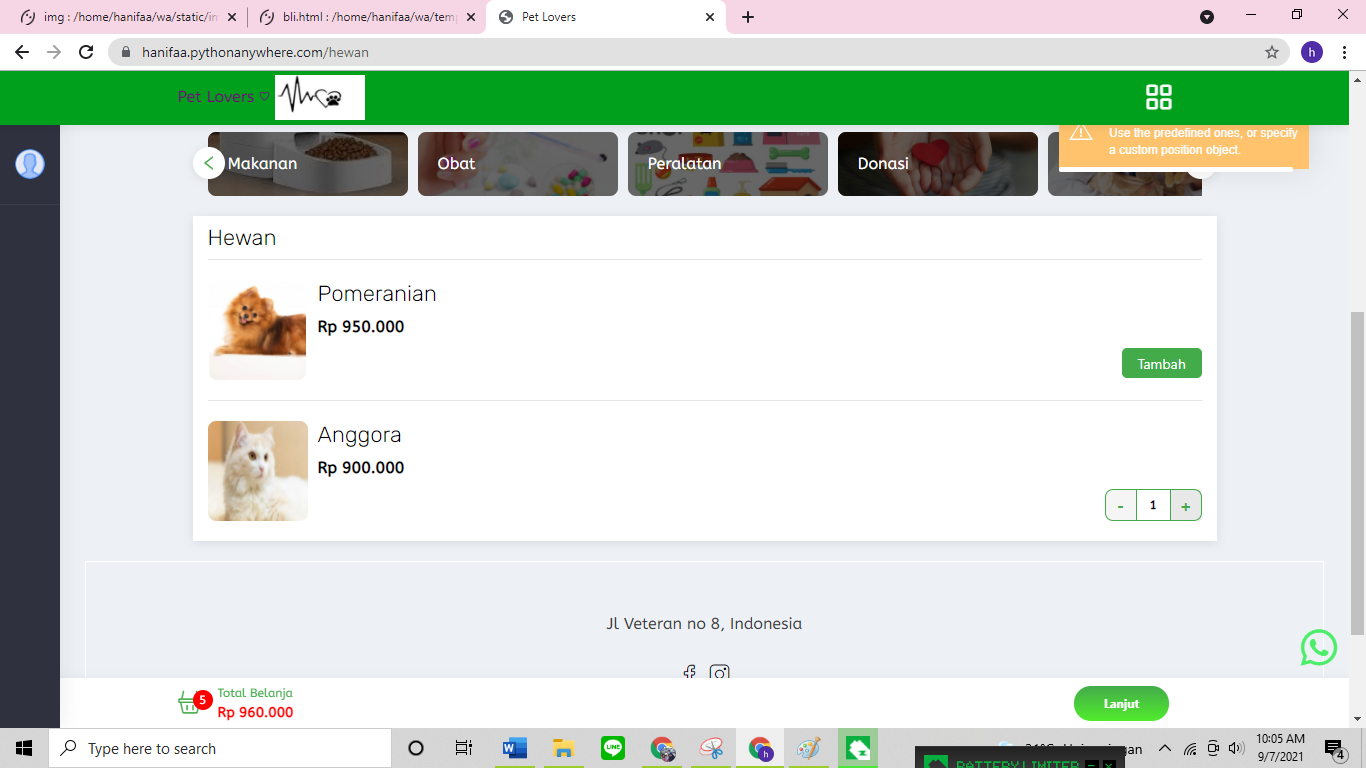


Gambar 28 kategori peralatan



Gambar 29 kategori donasi

Kategori donasi terdapat 2 jenis yaitu berupa dana dan tenaga. Uang dapat bersamaan dengan ketika ingin memberi produk, sedangkan volunter dapat menghubungi tombol chat untuk proses selanjutnya.

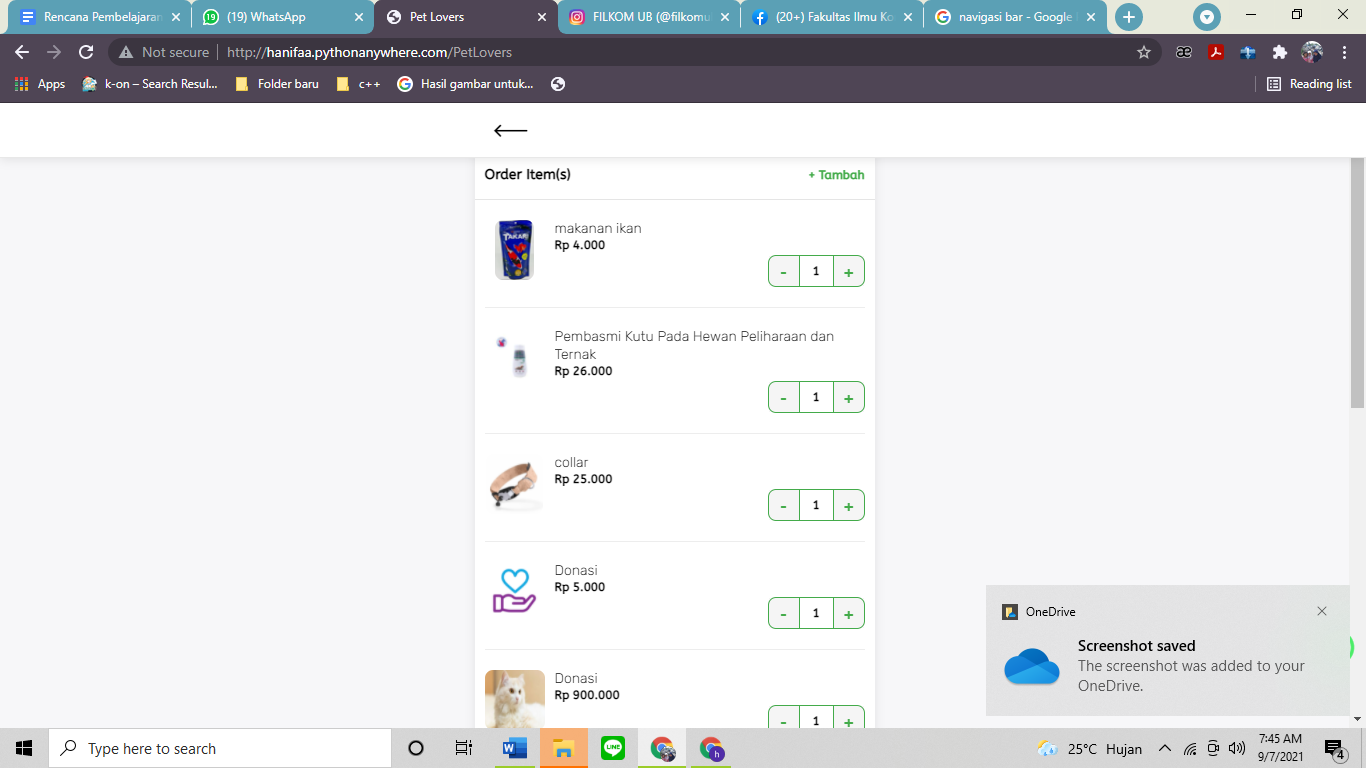


Gambar 30 Kategori hewan

Setelah memilih barang yang diinginkan maka akan menampilkan total belanja yang terletak di footer, jika sudah selesai maka tekan lanjut.

Tabel 22. Total belanja dan lanjut ke tahap selanjutnya

| <div class=”basket”>  <div class=”basket-cart”>  <I class=”lni lni-shopping-basket”></i>  <div class=”basket-cart-item-count”>2</div>  </div>  <div class=”basket-detail”>  <div class=”basket-items-label”>  Total Belanja  </div>  <div class=”basket-items-total”>Rp&nbsp;14.000</div>  </div>  <div class=”basket-next”>  <button type=”button”>Lanjut</button>  </div>  </div> |
| --- |



Gambar 31 Ketika mau pesan

Sebelum terkirim ke penjual maka akan menampilkan produk yang ingin dibeli , bisa ditambah atau dikurangi, atau ingin menambah produk lain.

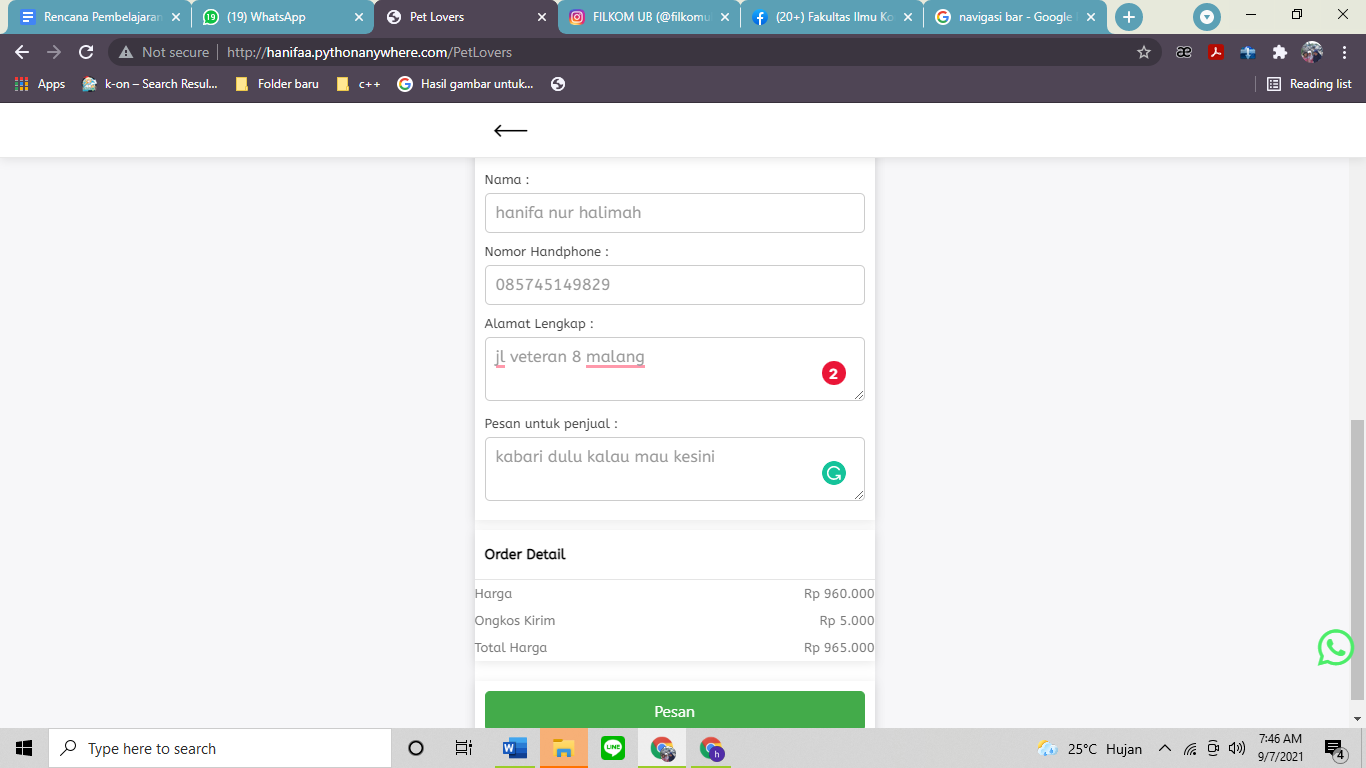
Tabel 23. Header List yang ingin dibeli

| <div class=”cart-content”>  <div class=”wrapper-cart”>  <div class=”cart-content-items”>  <div class=”cart-content-items-head”>  <div class=”cart-content-items-head-title”>  Order Item(s)  </div>  <div class=”cart-content-items-head-add”><a href=”/PetLovers”>+ Tambah</a></div>  </div>  <div class=”cart-content-items-list”> |
| --- |

Menampilkan semua produk yang ingin dibeli yang terdiri dari gambar, nama, harga, bisa ditambah dan dikurangi

Tabel 24 List yang ingin dibeli

| <div class=”productcart”>  <div class=”content”>  <div class=”image”>  <img src=”{item\_image}”>  </div>  <div class=”detail”>  <h3>{item\_name}</h3>  <div class=”pricing”>{item\_price}</div>  <div class=”atc clear”>  <div class=”qty qty-selector clear” data-item-id=”{item\_id}”>  <button type=”button” class=”minus cart-button-qty” data-qty-action=”minus”>-</button>  <input min=”0” type=”number” value=”{item\_qty}” name=”qty”>  <button type=”button” class=”plus cart-button-qty” data-qty-action=”plus”>+</button> |
| --- |



Gambar 32 menulis keterangan penerima

Memberi keterangan kepada penjual seperti nama, nomor handphone, alamat lengkap, pesan untuk penjual.

Tabel 25 Keterangan lengkap pembeli

| <div class=”cart-content-form”>  <div class=”cart-content-form-head”>  <div class=”cart-content-form-head-title”>  Data Pemesan </div>  </div>  <div class=”cart-content-form-list”>  <div class=”customer”>  <div class=”customer-field”>  <label>Nama :</label>  <input type=”text” name=”name”>  </div>  <div class=”customer-field”>  <label>Nomor Handphone :</label>  <input type=”number” name=”phone”>  </div>  <div class=”customer-field”>  <label>Alamat Lengkap :</label>  <textarea name=”address”></textarea>  </div>  <div class=”customer-field”>  <label>Pesan untuk penjual :</label>  <textarea name=”note”></textarea>  </div>  </div>  </div>  </div> |
| --- |

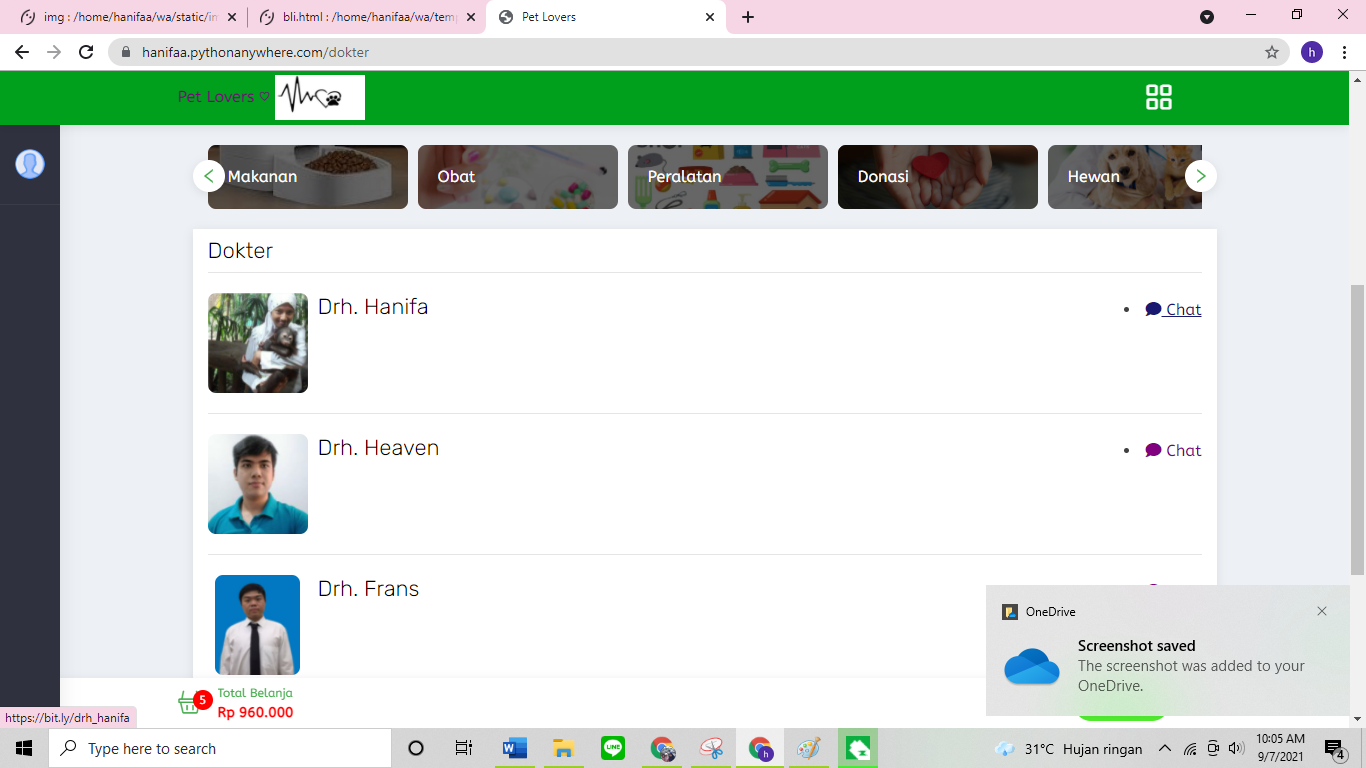
Informasi ongkos kirim serta total belanjaan.

Tabel 26 Harga total dan ongkos kirim

| <div class=”cart-content-detail”>  <div class=”cart-content-detail-head”>  <div class=”cart-content-detail-head-title”>  Order Detail  </div>  </div>  <div class=”cart-content-detail-list”>  <table>  <tbody><tr>  <td>Harga</td>  <td class=”value”>Rp&nbsp;14.000</td>  </tr>  <tr>  <td>Ongkos Kirim</td>  <td class=”value”>  Rp&nbsp;5.000  </td>  </tr>  <tr>  <td>Total Harga</td>  <td class=”value”>Rp&nbsp;19.000</td>  </tr>  </tbody></table>  </div>  </div>  <div class=”cart-content-submit”>  <button type=”button” id=”submit”>Pesan</button>  </div>  </div> |
| --- |



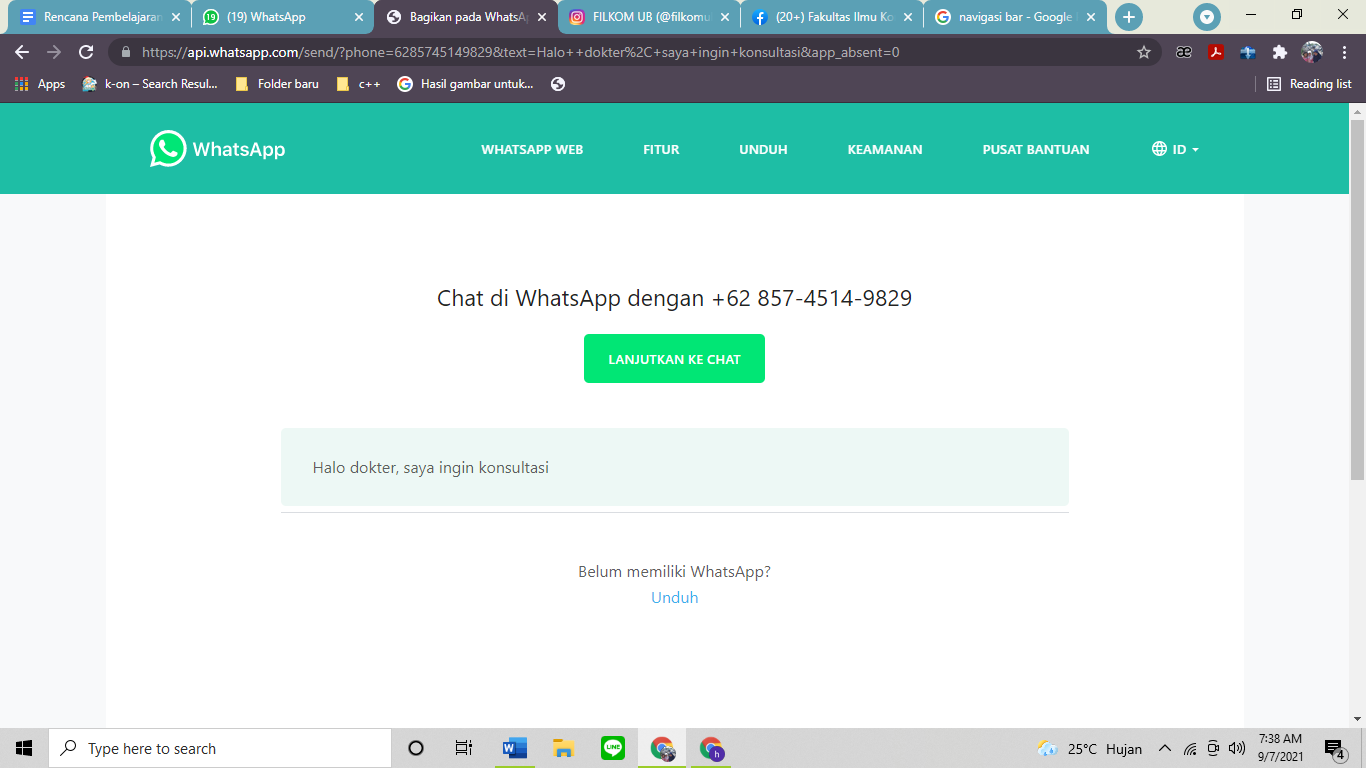
Gambar 33 terkirim ke penjual



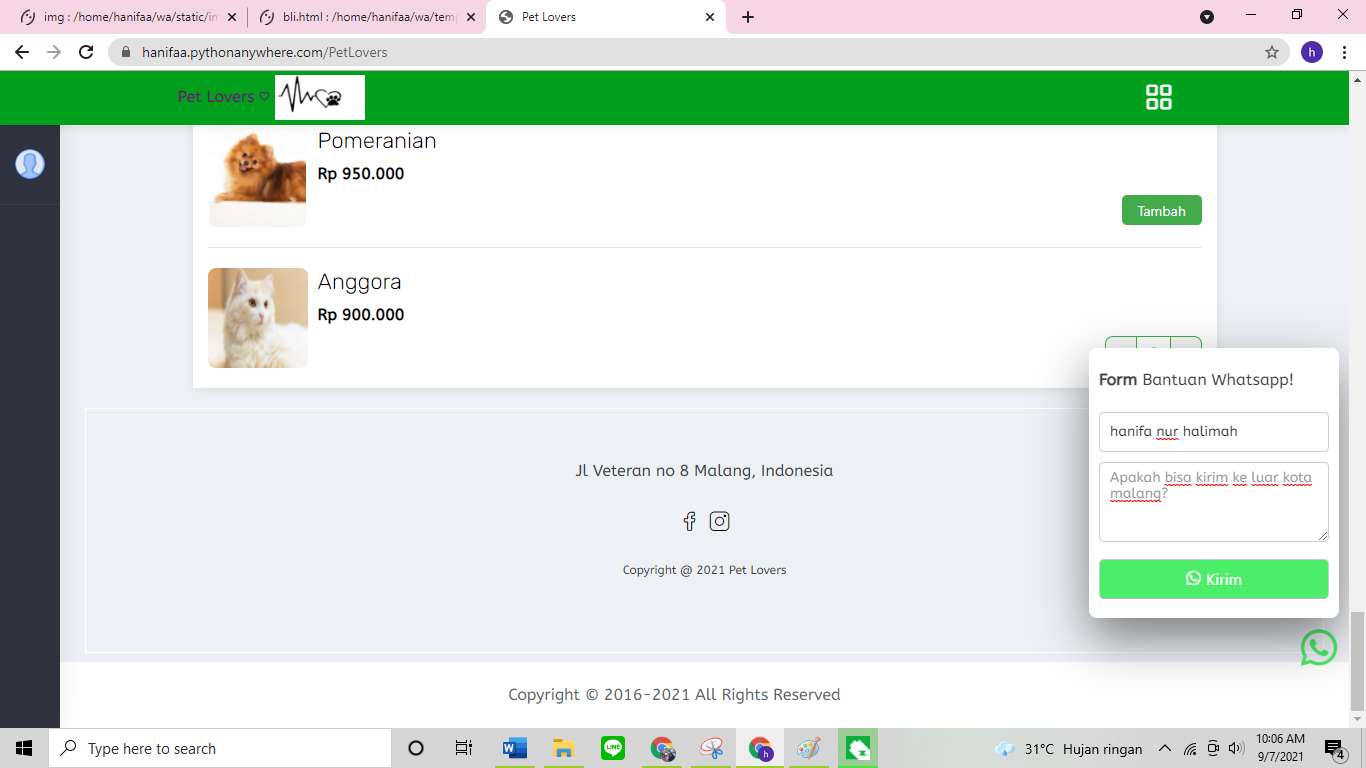
Gambar 34 kategori dokter hewan

Pengguna dapat chat Bersama dokter hewan dengan tekan tombol chat yang berisi pesan sesuai konsultasi pengguna ke wa dokter hewan tersebut :

<https://bit>.ly/drh\_hanifa



Gambar 35 terhubung ke wa untuk konsultasi



Gambar 36 tampilan halaman bagian footer

Halaman footer terdapat :

1. form bantuan whatsapp akan terhubung ke penjual seperti pada gambar , kodingan dapat dilihat pada tabel 28.

2. terdapat tombol ig dan fb masing masing terhubung ke url <https://www.instagram.com/filkomub/> dan <https://www.facebook.com/filkombrawijaya>

3. Alamat toko di jl veteran no 8 malang

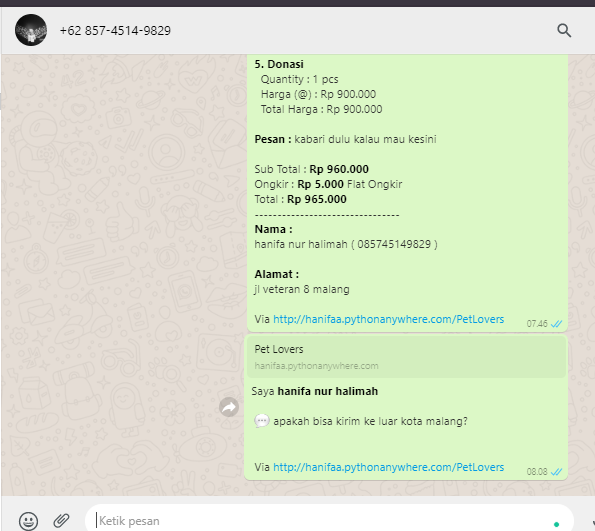
4. copyright yang membuat website adalah petlovers

Tabel 27 Footer halaman

| <div class=”site-footer”>  <div class=”wrapper”>  <div class=”address”>  Jl Veteran no 8, Indonesia </div>  <div class=”social”>  <a href=”https://www.facebook.com/filkombrawijaya”target=”\_blank”>  <I class=”lni lni-facebook”></i>  </a>  <a href=”https://www.instagram.com/filkomub/” target=”\_blank”>  <I class=”lni lni-instagram”></i>  </a>  </div>  <div class=”copyright”>  Copyright @ 2021 Pet Lovers  </div>  </div>  </div> |
| --- |

Tabel 28 Balon WA

| <div id="support-wa" class="wa-support-box">  <div class="wa-support-body">  <div class="wa-support-heading clear">  <b>Form</b> Bantuan Whatsapp!  </div>  <form class="wa-support-form" method="post" enctype="multipart/form-data">  <div class="input">  <i class="icon ion-md-person"></i>  <input type="text" name="name" placeholder="Nama Lengkap" required="" oninvalid="this.setCustomValidity('Input Nama Lengkap Anda')" oninput="this.setCustomValidity('')" value="">  </div>  <div class="input">  <textarea name="message" placeholder="Pesan Anda"></textarea>  </div>  <button type="submit">  <i class="lni lni-whatsapp"></i> Kirim  </button>  </form>  </div>  </div> |
| --- |

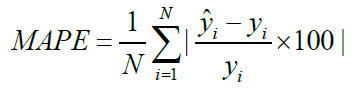


Gambar 37 chat form bantuan wa terkirim ke penjual

1. **Hasil Pengujian dan Pembahasan**

Pada bab ini akan membahas hasil uji coba yang telah dilaksanakan. Terdapat percobaan sebanyak 4 kali yaitu: data real case dengan spark dan non spark, data simulasi dengan spark dan non spark. Gambar 38 menunjukkan data training yang real case, dapat dilihat terdapat variabel x yang memiliki 13 input yang meiliki nilai rentang sebelah kiri, dan variabel y sebelah kanan memiliki 1 output. Gambar 39 hasil dari proses regresi linear Non Pyspark data realcase memperoleh nilai mape sebesar 1.6 waktu 27 ms. Gambar 40 hasil dari proses regresi linear Non Pyspark data simulasi memperoleh nilai mape sebesar 1.8 waktu 25.4 ms. Gambar 41 hasil dari proses regresi linear Pyspark data realcase memperoleh nilai mape sebesar -0.07 waktu 2160 ms. Gambar 42 hasil dari proses regresi linear Pyspark data simulasi memperoleh nilai mape sebesar 0.7 waktu 2410 ms. Tabel 29 rangkuman dari hasil penelitian.

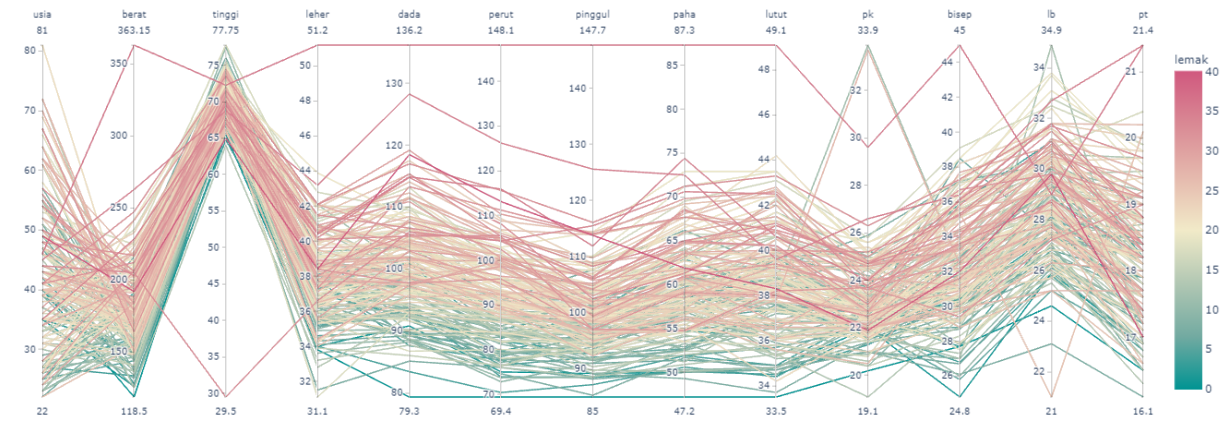
Nilai Mape diperoleh dari persamaan :



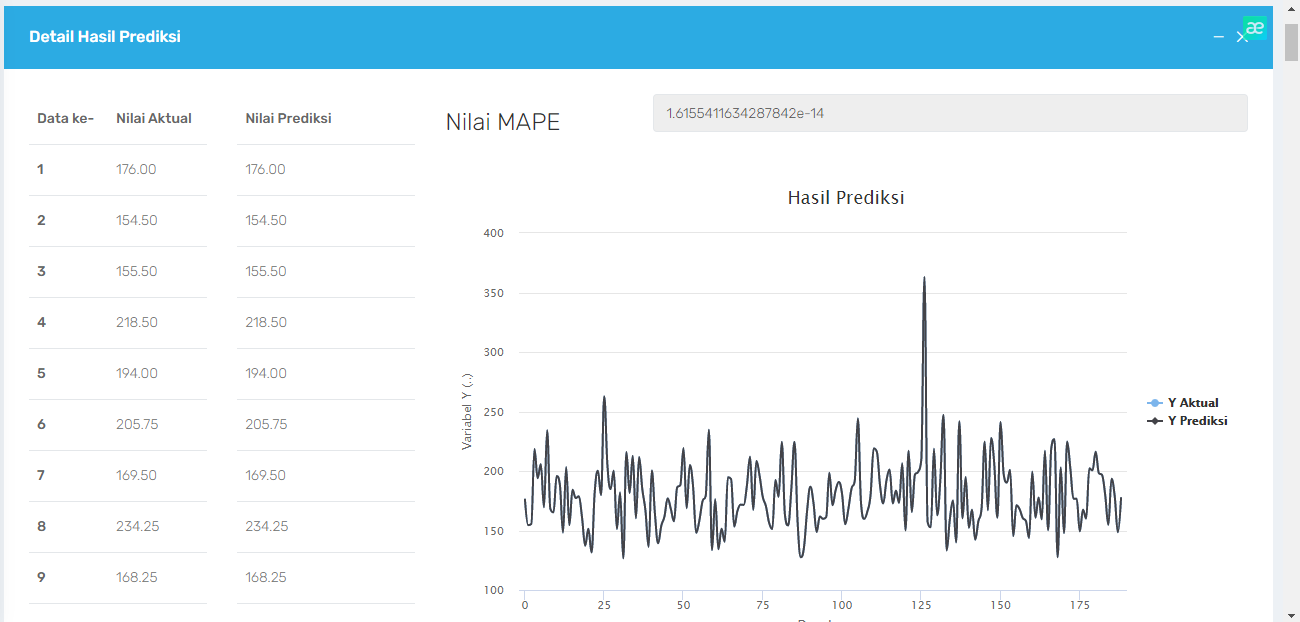
Waktu komputasi diperoleh dari kodingan :

!pip install ipython-autotime

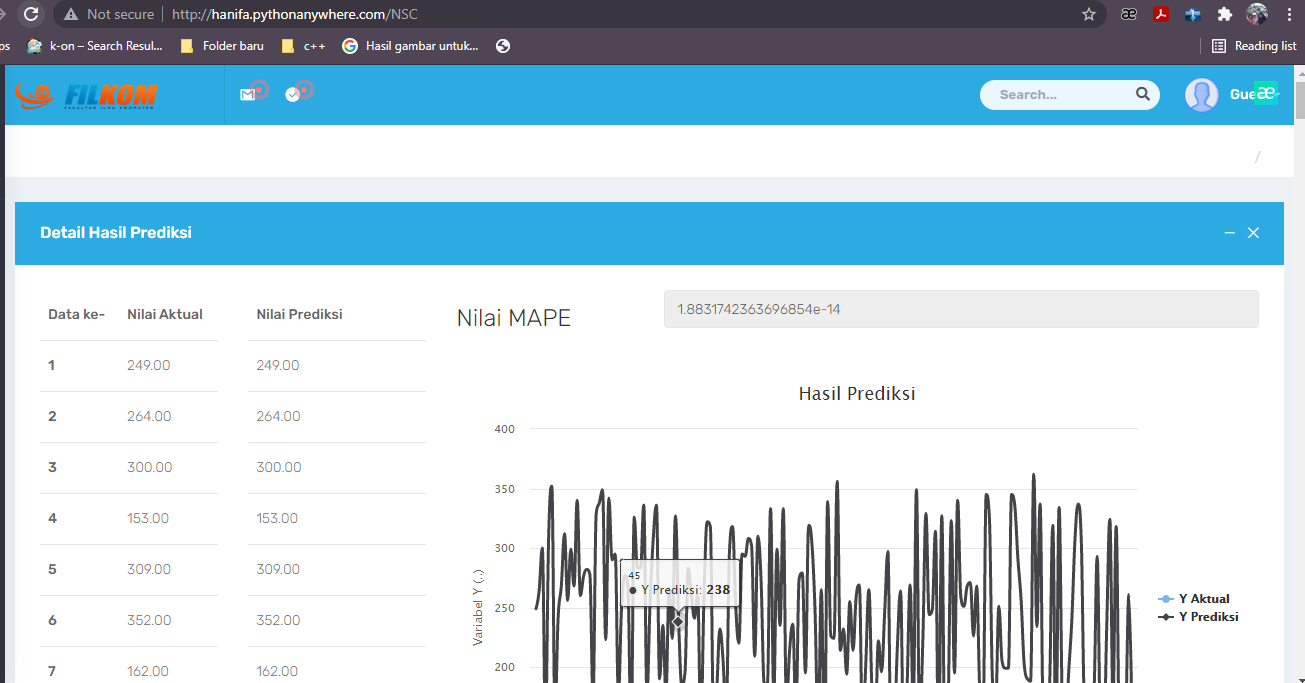
%load\_ext autotime



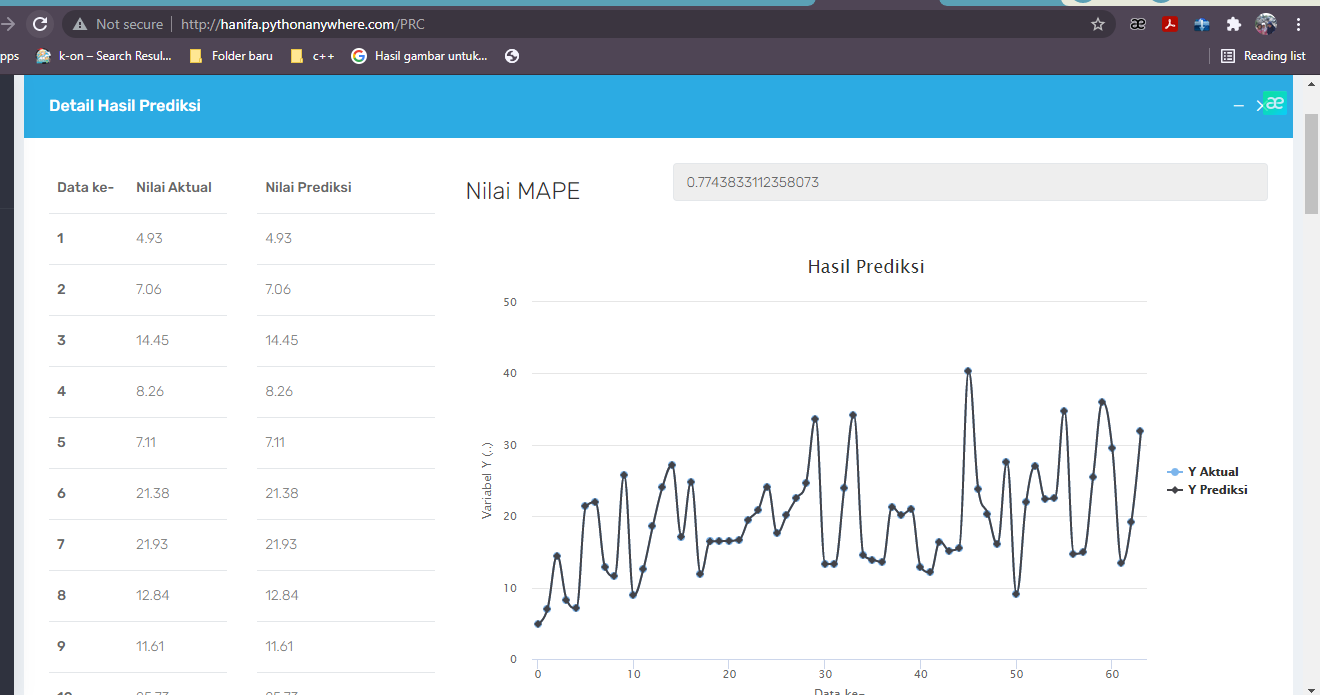
Gambar 38 . Data visual (real case)



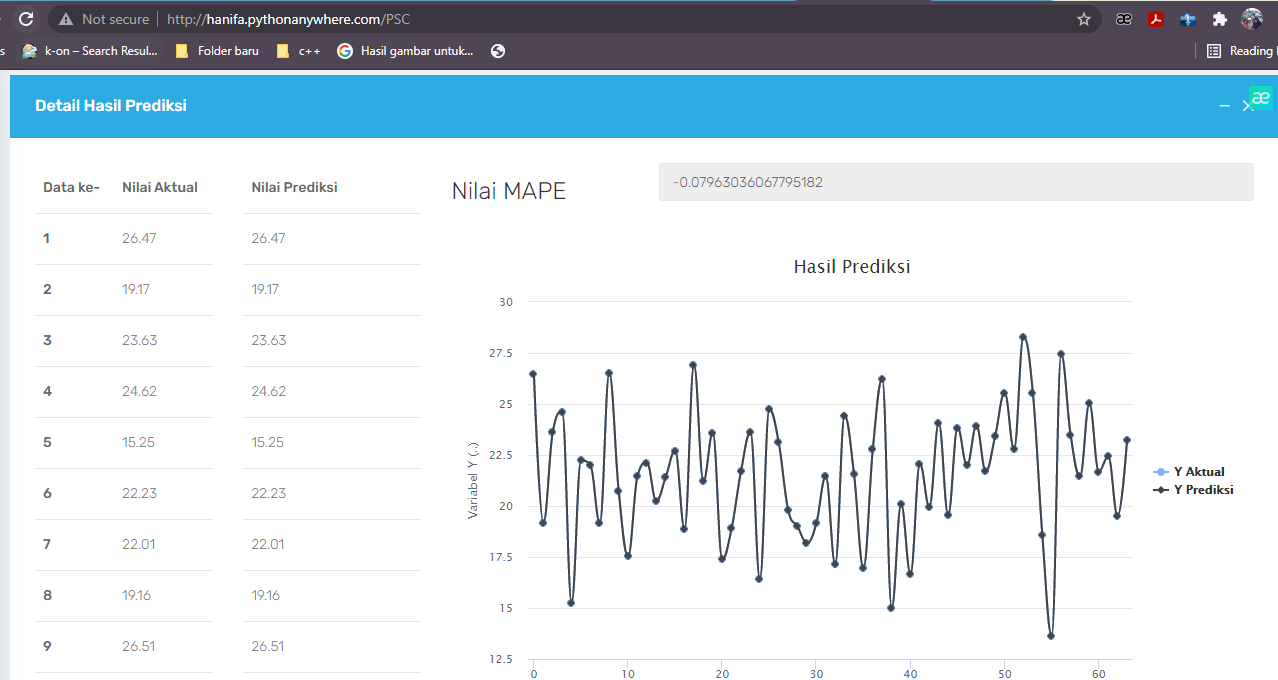
Gambar 39. Perbandingan data sebenarnya dan prediksi (real case)



Gambar 40. Perbandingan data sebenarnya dan prediksi (simulasi)



Gambar 41. Perbandingan data aktual dan prediksi (real case) menggunakan pyspark



Gambar 42. Perbandingan data aktual dan prediksi (simulasi case) menggunakan pyspark

Tabel 29. Hasil keluaran penelitian.

| Jenis Data | Uji ke (data 75%, test 25%) | Machine Learning (Regresi Linier) | | | |
| --- | --- | --- | --- | --- | --- |
| Nilai Performa (Mape) | | Waktu Komputasi (ms) | |
| Non | PySpark | Non | PySpark |
| Data Simulasi | 1 | 1.8 | - | 25.4 ms | - |
| 2 | - | -0.07 | - | 2160 ms |
| Data Real Case | 1 | 1.6 | - | 27 ms | - |
| 2 | - | 0.7 | - | 2410 ms |

**Kesimpulan**

Pengembangan aplikasi big data pada prediksi lemak tubuh memiliki nilai mape yang rendah sehingga nilai dari proses regresi linear dapat memprediksi nilai mendekati nilai akurat. Menggunakan flask dengan pythonanywhere dan sqlite terbukti berkeja dengan baik dan cepat walaupun ringan penggunaanya. Hasil nilai mape yang terbaik diperoleh oleh data simulasi pyspark sedangkan waktu tercepat diperoleh oleh data simulasi non pyspark, kemungkinan terjadi kenapa pyspark lebih lambat dikarenakan data yang diolah sedikit sehingga terkesan lebih lama dan tidak terlihat proses untuk mempercepat data. Sehingga proses untuk mempercepat data terkesan membuat lebih lama. Sedangkan non Pyspark hanya proses ke regresi linear. Walaupun proses lama, pyspark memiliki nilai mape terbaik daripada tidak menggunakan PySpark.

**References ()**

1. Anwar, A. (2017). Digital LIbrary. Retrieved from eprints.akakom.ac.id: IMPLEMENTASI BASIS DATA NOSQL CASSANDRA PADA SISTEM PENGARSIPAN DOKUMEN SKRIPSI DAN TUGAS AKHIR DI PERPUSTAKAAN STMIK AKAKOM BERBASIS WEB DALAM PRIVATE CLOUD
2. Baltschun , Y. (2020). I Hate Diet.
3. Calculator.net. (2021). BMI Calculator. Retrieved september 6, 2021, from https://www.calculator.net/bmi-calculator.html
4. Carberry, C. (2021). Cara Akurat Menghitung Persentase Lemak Tubuh. Retrieved september 6, 2021, from https://id.wikihow.com/Akurat-Menghitung-Persentase-Lemak-Tubuh#/Berkas:Calculate-Body-Fat-Percentage-Accurately-Step-1-Version-4.jpg
5. Channel, T. W. (2019). Mudah memahami Regresi Linear (Marchine Learning dengan Python. Retrieved september 6, 2021, from https://www.youtube.com/watch?v=YKv4sOEHhYo&t=135s
6. Cholissodin, I. (2021). Biz. Retrieved september 6, 2021, from https://drive.google.com/drive/folders/1JDOSB3xIlHjbOcjCaH4TfU4cksBum\_j3
7. Cholissodin, I. (2021). FGA Big Data Using Python Filkom x Mipa UB 2021. Retrieved september 6, 2021, from https://github.com/imamcs19/FGA-Big-Data-Using-Python-Filkom-x-Mipa-UB-2021
8. Cholissodin, I. (2021). Generate. Retrieved september 2021, 2021, from https://colab.research.google.com/drive/1bfMBO5uuV6HeJxPL26HHt1Qe5zEEOXV4?usp=sharing
9. Cholissodin, I. (2021). Sesi 17 flask x google colab part 1 of 2 csv to sqlite3. Retrieved september 6, 2021, from https://drive.google.com/file/d/1frwcD-goleT8BmpFJIWQu-qm6Pk3-o-6/view?usp=sharing
10. Cholissodin, I. (2021). Sesi 17 flask x google colab part 2 of 2. Retrieved september 6, 2021, from https://colab.research.google.com/drive/1vJLGEaaioUsr9lPFSXgTiRY52H1slVOq?usp=sharing
11. Cholissodin, I., & Riyandani, E. (2018). ANALISIS BIG DATA (Teori & Aplikasi). In I. Cholissodin, & E. Riyandani, ANALISIS BIG DATA (Teori & Aplikasi).
12. Fauzi, M. A. (2019). Spark. Retrieved from http://malifauzi.lecture.ub.ac.id: http://malifauzi.lecture.ub.ac.id/files/2019/02/Spark.pdf
13. Fedesoriano. (2021). Body fat prediction dataset. Retrieved from Kaggle: https://www.kaggle.com/fedesoriano/body-fat-prediction-dataset
14. GGL. (2021). Perbedaan Lemak dan Otot pada Berat Badan. Retrieved september 6, 2021, from https://ggl.life/detail-blog?p=perbedaan-lemak-dan-otot-pada-berat-badan
15. Jhony, H. (2015). Perancangan Sistem Mobile Pos (Point Of Sale) dengan Menggunakan Restful Web Services. Retrieved from digital repository universitas internasional batam: http://repository.uib.ac.id/443/
16. Jones, A. (2020, September 17). Python Pandas and SQLite. Retrieved from towardsdatascience: https://towardsdatascience.com/python-pandas-and-sqlite-a0e2c052456f
17. MUBAROK, M. I. (2018). regresi liniear. Retrieved from muhammadilhammubarok: https://muhammadilhammubarok.wordpress.com/2018/08/09/regresi-linier-berganda-dasar-teori/
18. Muhardian, A. (2021). Tutorial Bootstrap: Memahami Sistem Grid untuk membuat Layout. Retrieved september 6, 2021, from https://www.petanikode.com/bootstrap-grid/
19. Nuridhuha, D., Ichsan, M. H., & Maulana, R. (2020). Sistem MonitoringLingkungan Rumah Cerdasberbasis Fog Computingdan nRF24l01. JPTIIK, 622-631.
20. Ongko, J. (2016). Ketahui Kadar Lemak Tubuh Anda. Retrieved september 6, 2021, from http://www.ask-jansen.com/kadar-lemak-tubuh/
21. Putri, L. N. (2019). PENGARUH KOMBINASI HIGH INTENSITY INTERVAL TRAINING DAN RESISTANCE TRAINING TERHADAP PENURUNAN PERSENTASE LEMAK TUBUH PADA MAHASISWI DI GONTOR PUTRI KAMPUS LIMA KEDIRI. Retrieved september 6, 2021, from http://eprints.umm.ac.id/46551/
22. PythonAnywhere. (2021). Retrieved from Wikipedia: https://translate.google.com/translate?u=https://en.wikipedia.org/wiki/PythonAnywhere&hl=id&sl=en&tl=id&client=srp&prev=search.