**MODEL PREDIKSI HARGA RUMAH**

Gilang Agung Saputra

*Artificial Intelligence* merupakan bidang ilmu yang mempelajari dan membuat mesin yang dapat berpikir layaknya manusia, *Artificial Intelligence* dapat diterapkan dalam berbagai hal seperti pengelompokan benda, rekomendasi lagu, film, memprediksi terjadinya kebakaran, memprediksi harga saham dan masih banyak lagi.

Prediksi harga rumah merupakan hal yang dapat dilakukan oleh *Artificial Intelligence,* dengan menggunakan data spesifikasi dan harga rumah serta menggunakan algoritma yang ada pada *Artificial Intelligence* kita dapat membuat mesin yang mampu memprediksi harga rumah.

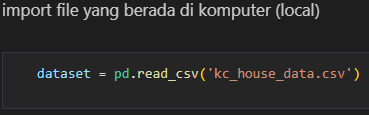
Percobaan kali ini kita akan menggunakan data harga rumah yang diperoleh dari “<https://github.com/adimasmudi/house_prediction>”, membandingkan beberapa algoritma kemudian menggunakan algoritma dengan nilai akurasi tertinggi. Berikut adalah langkah-langkah untuk membuat mesin prediksi harga rumah:

1. *Import* data yang telah diunduh

Import data yang ingin digunakan dengan *library* pandas

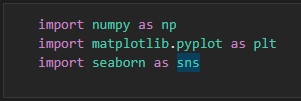


File yang digunakan memiliki format “csv”

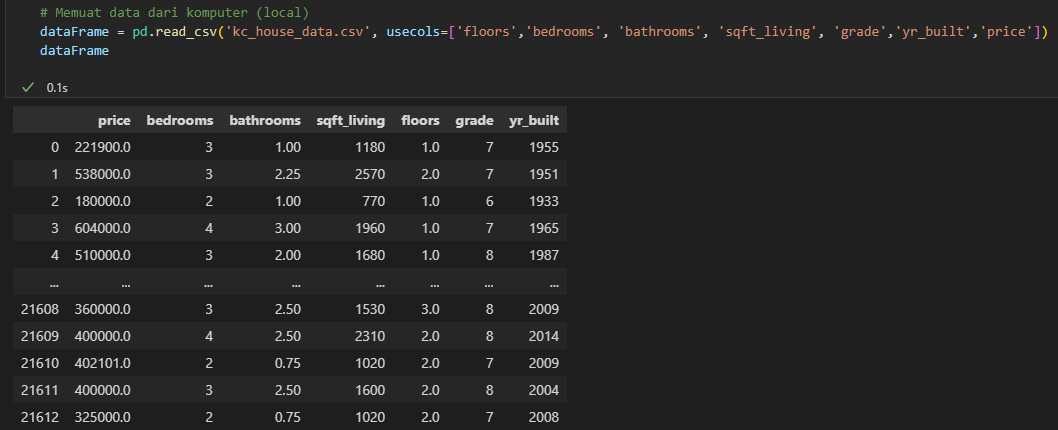


2. *Data Understanding*

Tahap ini merupakan proses untuk mengetahui definisi dari nama variabel yang ada pada data, memahami data dan variabel yang perlu digunakan, memeriksa data yang kita gunakan apakah memiliki kualitas yang baik dan sudah sesuai dengan mesin yang ingin kita buat. *Import* *library* yang kita butuhkan untuk melakukan proses *Data Understanding* dan proses-proses selanjutnya.



Variabel yang ingin akan digunakan adalah ‘floors’, ‘bedrooms’, ‘sqft\_living’, ‘grade’, ‘yr\_built’, dan ‘price’.



Penjelasan dari variabel:

- bedrooms : jumlah kamar tidur

- bathrooms : jumlah kamar mandi

- sqft\_living : luas ruangan dalam satuan sqft (square feet)

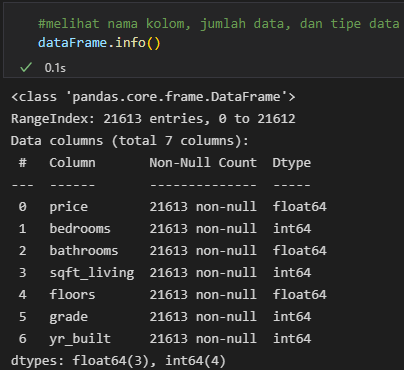
- floors : jumlah lantai

- grade : klasifikasi dari kelas rumah

- yr\_built : tahun bangunan

- price : harga rumah ($)

Melihat tipe data dari masing-masing variabel

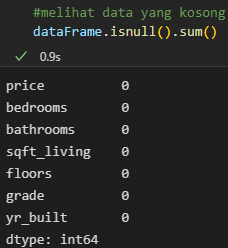


Ada tipe data yang kurang sesuai pada variabel ‘bathrooms’ dan ‘floors’, kita harus merubah menjadi *integer* agar menjadi bilangan bulat

3. *Data Preparation*

*Data Preparation* adalah proses pengumpulan, penggabungan, penataan, dan pengorganisasian data sehingga dapat digunakan dalam analitik, dan visualisasi data.

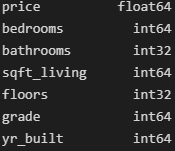
Melihat apakah ada data yang kosong pada masing-masing variabel



Dari hasil di atas tidak ada data yang kosong, sehingga tidak perlu manipulasi data kosong

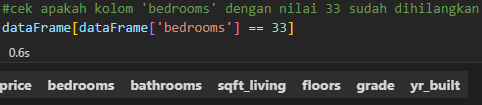
Mengubah tipe data *float* dari variabel ‘bathrooms’ dan ‘floors’ menjadi *integer*

Variabel ‘bathrooms’ dan ‘floors’ berhasil diubah menjadi *integer*

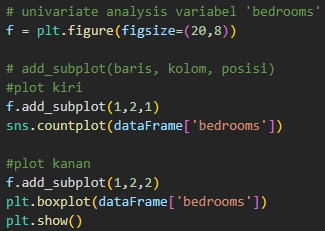
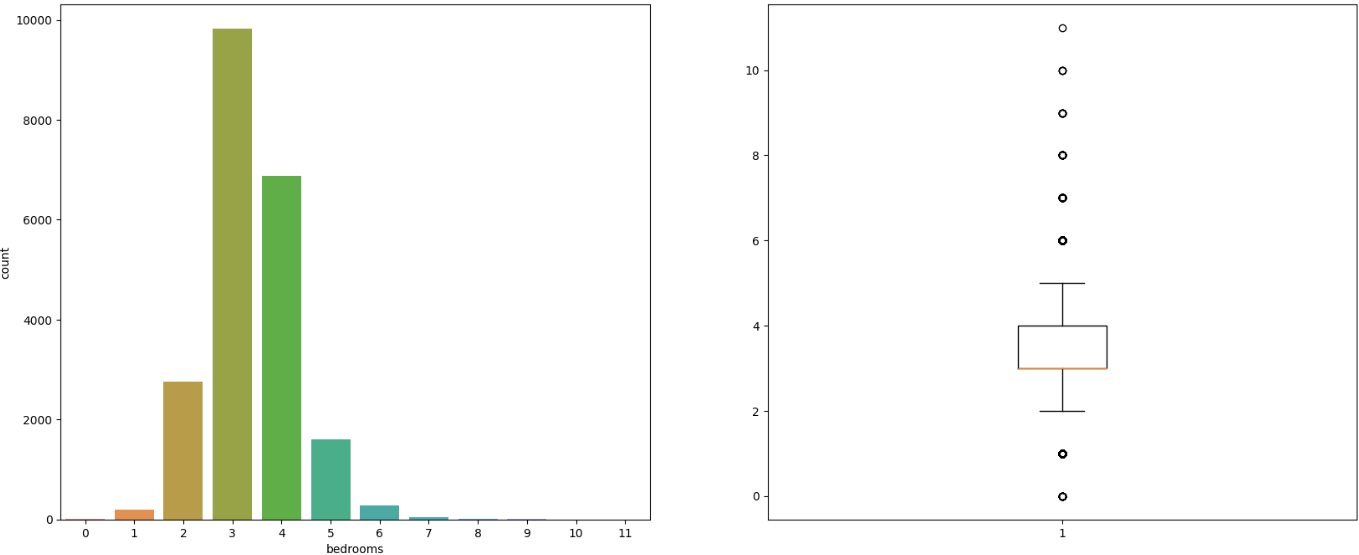
Menghapus variabel ‘bedrooms’ dengan nilai 33

Variabel ‘bedrooms’ dengan nilai 33 berhasil dihapus

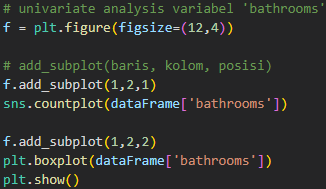
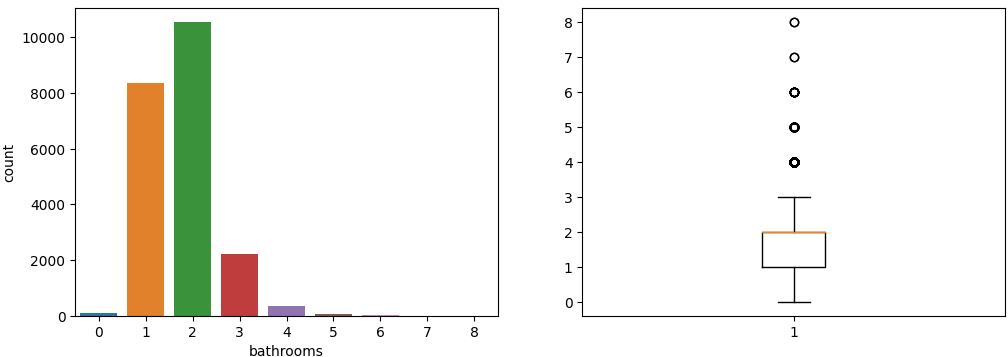
Proses selanjutnya adalah *Exploratory Data Analysis* (EDA), EDA merupakan proses untuk melihat pola-pola tertentu dalam data, EDA dapat dibagi menjadi *Univariate Analysis* yang menganalisa masing-masing variabel tanpa melihat hubungan dengan variabel lain dan *Bivariate Analysis* yang melihat hubungan satu variabel dengan variabel lain, dalam kasus ini *Bivariate Analysis* yang kita lakukan adalah melihat korelasi masing-masing variabel terhadap harga.

*Univariate Analysis* variabel ‘bedrooms’

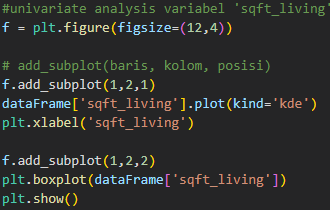
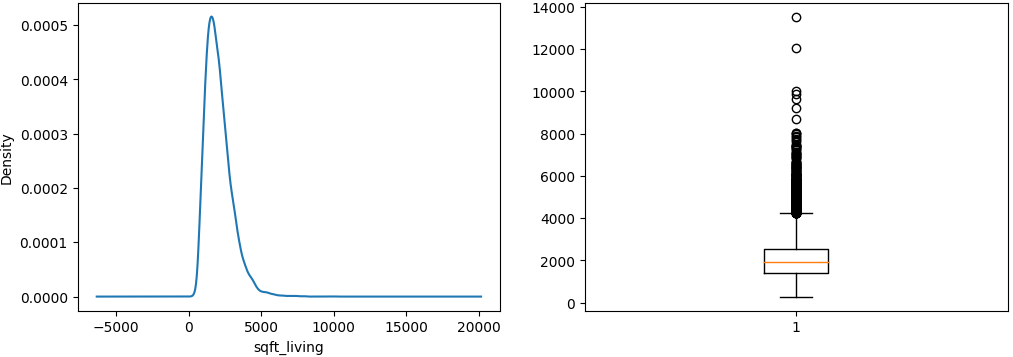
  


Grafik menunjukkan variabel ‘bedrooms’ memiliki nilai terbanyak di angka 3 dan 4, variabel ‘bedrooms’ memiliki *outliers*.

*Univariate Analysis* variabel ‘bathrooms’

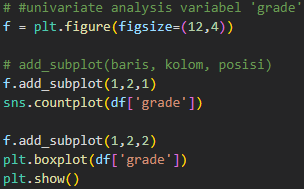
  
  
Nilai variabel ‘bathrooms’ terbanyak adalah 2 dan 1, dan memiliki *outlier* di atas garis maksimum.

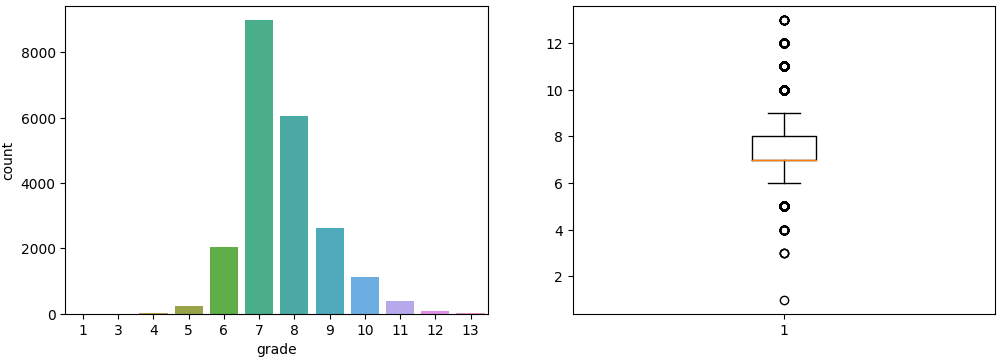
*Univariate Analysis* variabel ‘sqft\_living’

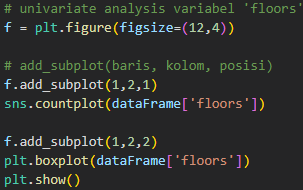
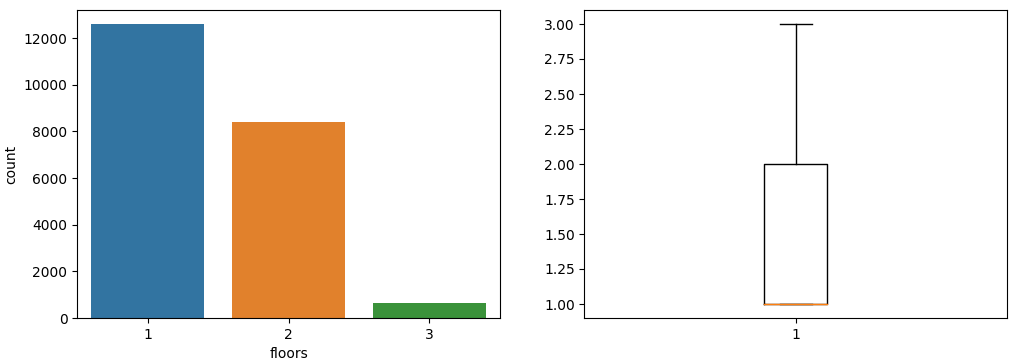
Kepadatan variabel ‘sqft\_living’ ada di angka 2000, dan memiliki banyak outliers di atas garik maksimum.

*Univariate Analysis* variabel ‘grade’



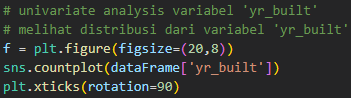
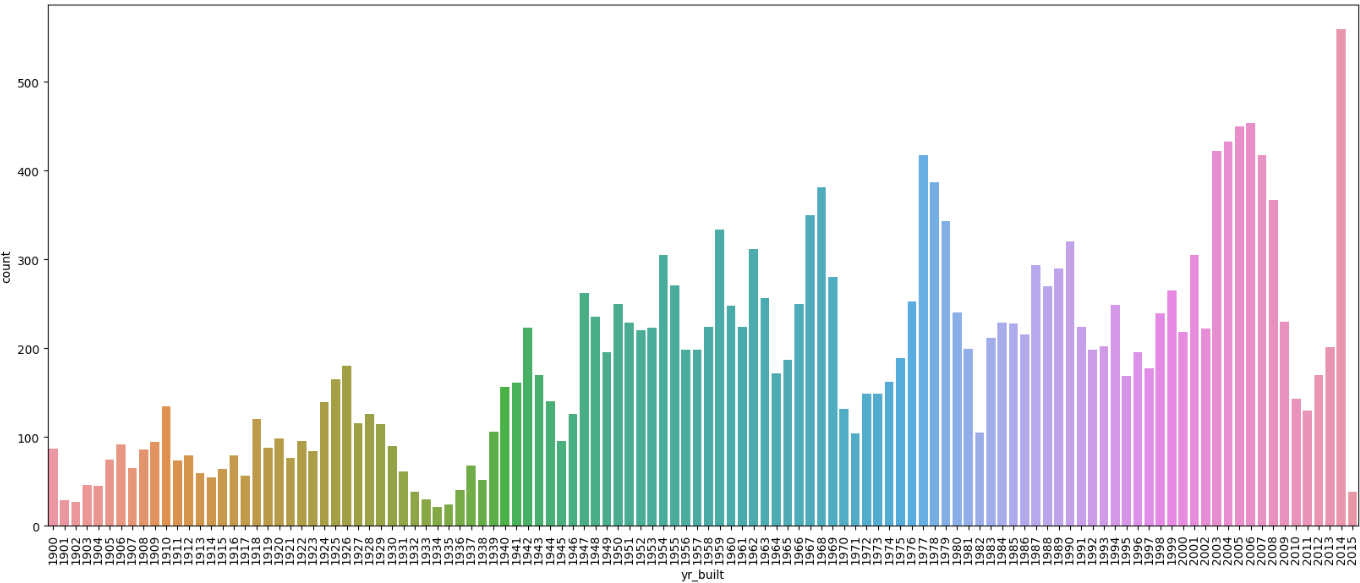
  
Data ini memiliki kekurangan dimana tidak ada ‘grade’ yang bernilai 2, kebanyakan rumah memiliki ‘grade’ di angka 7 dan 8, data memiliki *outlier* di bawah 6 dan di atas 9.

*Univariate Analysis* variabel ‘floors’

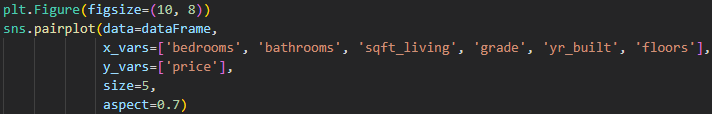
Data menunjukkan kebanyakan rumah hanya memiliki 1 lantai

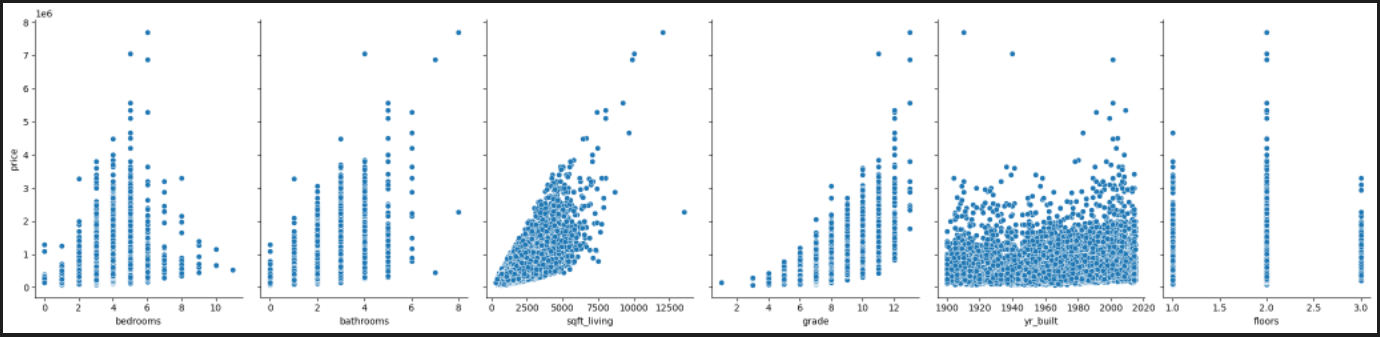
*Univariate Analysis* variabel ‘yr\_built’

Rumah yang dijual dengan ‘yr\_built’ 1901 hingga 1938 memiliki jumlah yang lebih sedikit, mulai tahun 1939 hingga 2014 memiliki jumlah yang lebih banyak

*Bivariate Analysis* pada masing-masing variabel terhadap harga rumah

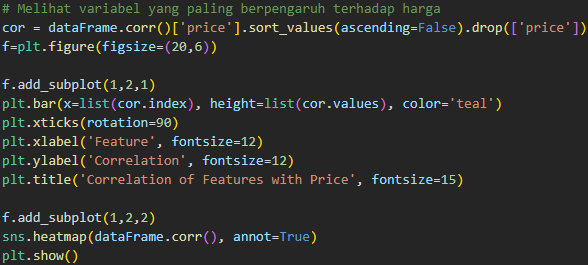
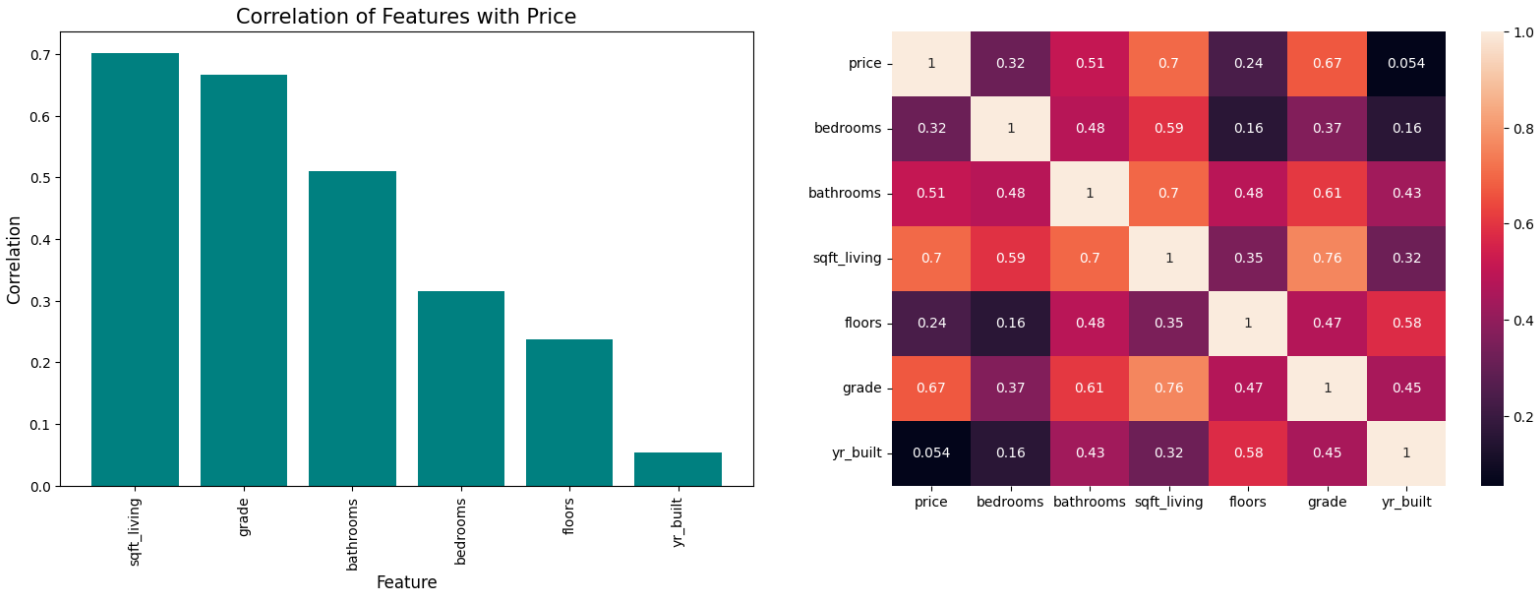




Dari plot yang dihasilkan didapatkan informasi yaitu:

* Rumah dengan jumlah ‘bedrooms’ 5 dan 6 memiliki harga lebih bervariasi
* Semakin banyak jumlah ‘bathroom’ semakin bervariasi harganya dan cenderung lebih mahal
* ‘sqft\_living’ hingga 6000 memiliki kepadatan yang tinggi, semakin besar nilai ‘sqft\_living’ maka semakin bervariasi harganya, namun cenderung semakin mahal
* 'yr\_built' persebaran yang merata, rumah dengan 'yr\_built' tua bisa memiliki harga yang murah maupun mahal
* 'floors' dengan angka 2 memiliki harga yang lebih bervariasi dan cenderung lebih mahal dibanding 1 dan 3

Melihat variabel yang paling berpengaruh terhadap harga

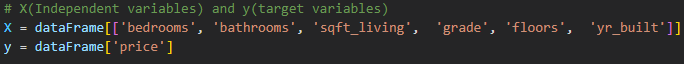
  


Dari variabel yang digunakan, ‘sqft\_living’ memiliki korelasi tertinggi terhadap harga, sedangkan ‘yr\_built’ memiliki pengaruh paling rendah terhadap harga.

3. *Modelling*

Pada proses *Modelling* kita akan memisahkan antara variabel independen (yang mempengaruhi) dan variabel dependen/target (yang dipengaruhi), membagi *data train* dan *data test*, membandingkan beberapa algoritma regresi, dan memilih algoritma dengan nilai akurasi tertinggi.

Membuat variabel ‘X’ yang berisi variabel independen dan variabel ‘y’ yang berisi variabel dependen/target.



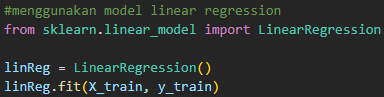
*Import* fungsi train\_test\_split dari sklearn.

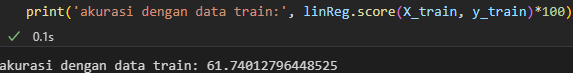


Membagi seluruh data menjadi data *train* sebanyak 70% dan data *test* sebanyak 30%, pembagian data tidak ada ketentuan khusus namun kebanyakan menggunakan pembagian 70% 30% atau 80% 20%, random\_state berguna untuk menentukan nilai acak dari data, jika random\_state ditentukan maka keacakan data akan tetap sama dan hasil training akan tetap sama ketika kode dijalankan.



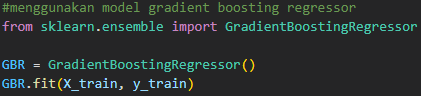
Percobaan dengan algoritma *Linear Regression*

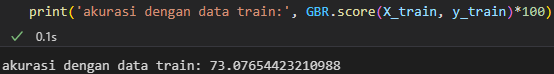




Hasil *training* mendapatkan nilai sebesar 61.74%

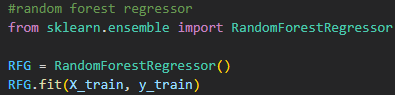
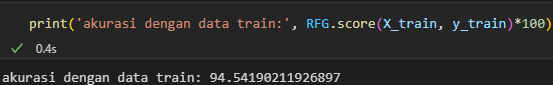
Percobaan dengan algoritma *Gradient Boosting Regressor*





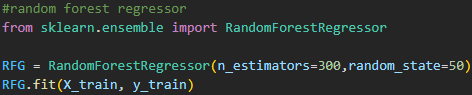
Hasil *training* mendapatkan nilai sebesar 73.07%

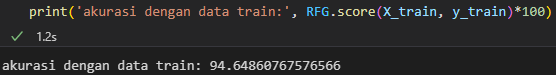
Percobaan dengan algoritma *Random Forest Regressor*

Hasil *training* mendapat nilai sebesar 94.54%

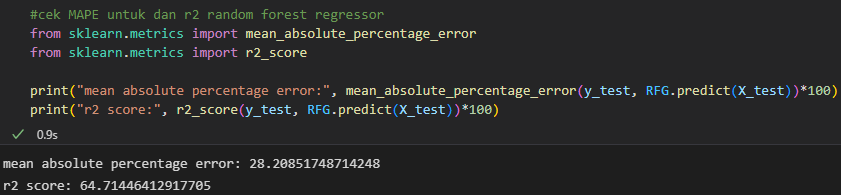
*Random Forest Regressor* menghasilkan nilai terbesar, kita akan coba untuk memberi parameter n\_estimator dan random\_state, kemudian melihat apakah dapat menghasilkan nilai lebih tinggi





Penggunakan parameter n\_estimator dapat meningkatkan nilai akurasi, besar n\_estimator dapat diubah untuk melihat apakah nilai akurasi dapat lebih besar atau bahkan menurunkan nilai akurasi.

Cek hasil *Mean Absolute Percent Error* (MAPE) dan *R squared*

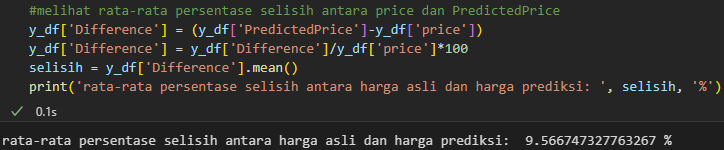


MAPE merupakan acuan untuk mengetahui seberapa besar kesalahan nilai prediksi dibanding nilai aslinya, nilai MAPE dapat dibagi menjadi 4 kategori yaitu: <10% = sangat akurat, 10-20% = baik, 20-50% = wajar (). Nilai MAPE yang kita dapatkan sebesar 28.20% yang masuk dalam kategori wajar, sebagai referensi dapat dilihat di pada website berikut: <https://www.aindhae.com/2019/12/cara-menghitung-mean-absolute.html>

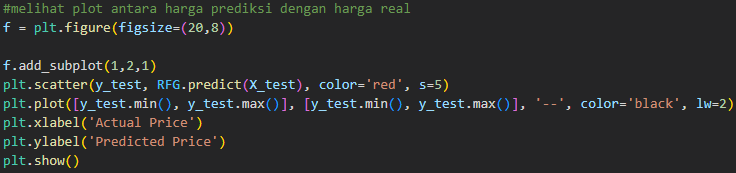
R squared (r2) merupakan angka yang berkisar antara 0 sampai 1 (0 sampai 100 jika dalam persen) yang mengindikasikan besarnya kombinasi variabel independen secara bersama – sama mempengaruhi nilai variabel dependen/target semakin mendekati angka satu (100 dalam persen) maka model yang digunakan dalam regresi semakin baik. nilai r2 dikategorikan kuat jika lebih dari 0,67, moderat jika 0,33 < r2 < 0,67, dan lemah jika 0,19 < r2 < 0,33. Nilai r2 yang kita dapatkan sebesar 64.71% dan termasuk dalam moderat, sebagai referensi dapat dilihat pada website berikut:

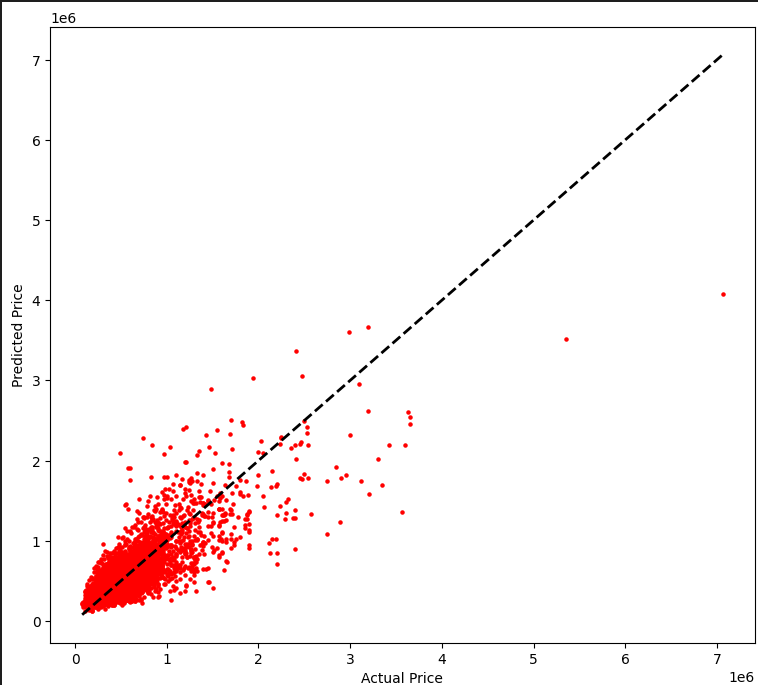
https://accounting.binus.ac.id/2021/08/12/memahami-koefisien-determinasi-dalam-regresi-linear/

Melihat rata-rata selisih antara harga asli dan harga prediksi dari *data test*



Melihat persebaran dari harga asli terhadap harga prediksi

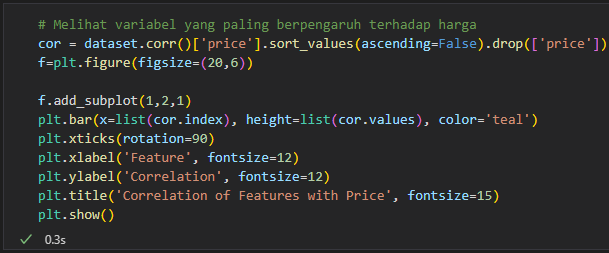


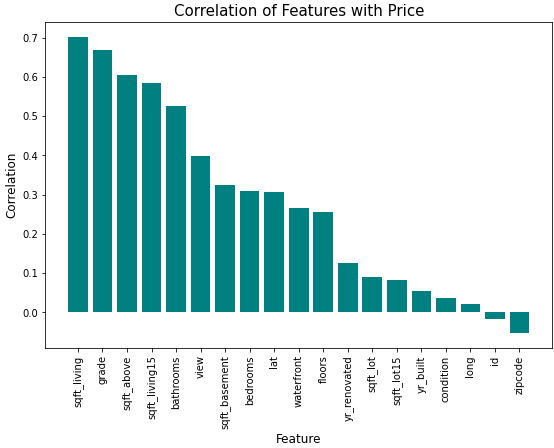


Mencoba Menambahkan Variabel Data

Kita akan melihat apakah dengan penambahan variabel mampu mendapatkan akurasi, nilai MAPE, r2, dan persebaran harga asli terhadap harga prediksi menjadi lebih baik.

Melihat korelasi seluruh variabel terharap harga



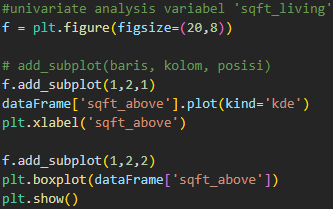


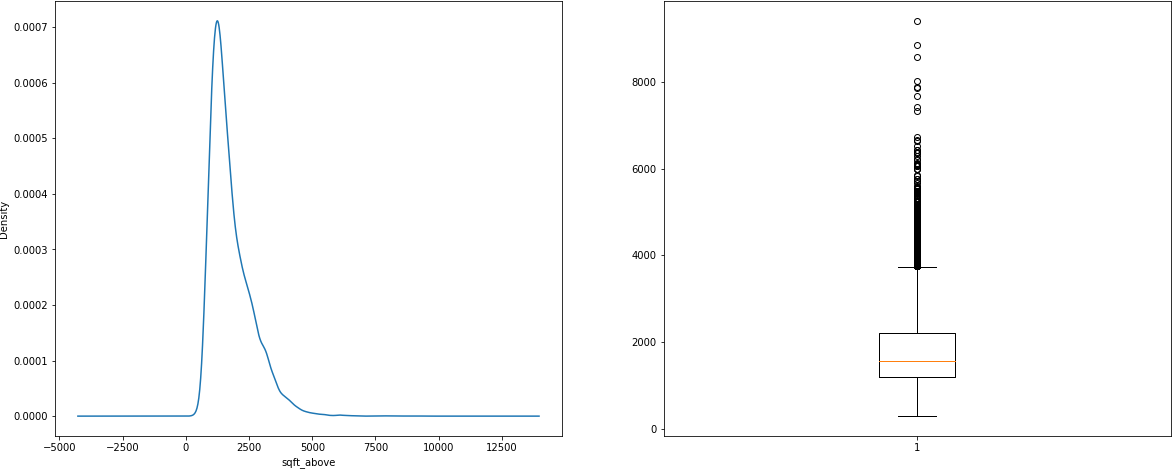
Jika kita lihat dari *bar chart* di atas, terdapat variabel dengan korelasi tinggi namun tidak digunakan di model sebelumnya, pada percobaan kedua kita akan tambahkan variabel ‘sqft\_above’ dan ‘sqft\_living15’.

- ‘sqft\_above’ : luas ruangan yang berada di permukaan tanah

- ‘sqft\_living15’ : rata-rata luas tanah dari 15 tetangga terdekat

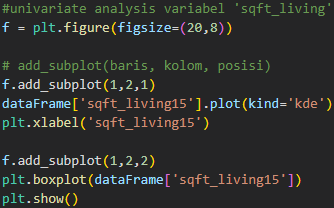
*Univariate Analysis* variabel *‘*sqft\_above*’*

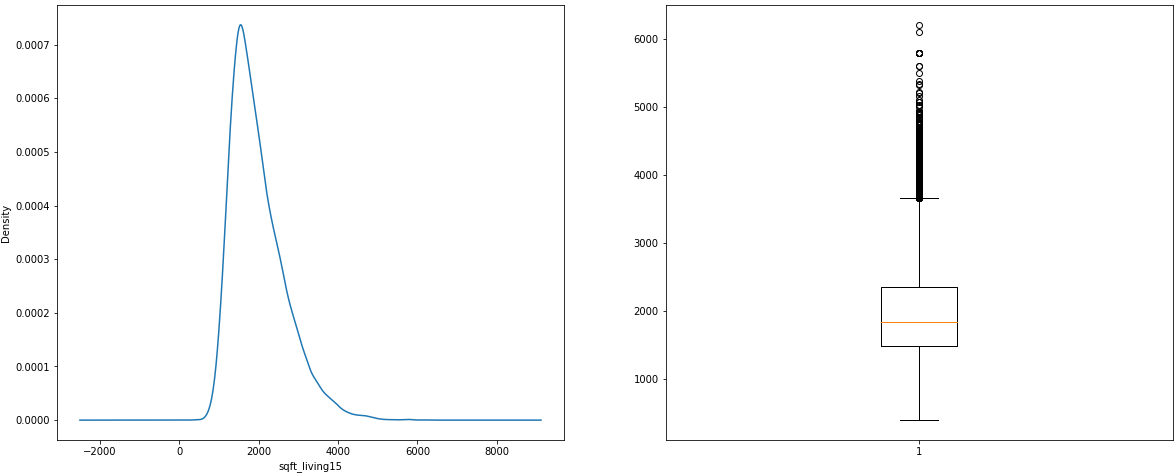




Kepadatan variabel ‘sqft\_above’ berada di angka 1500, terdapat *outlier* di atas 4000.

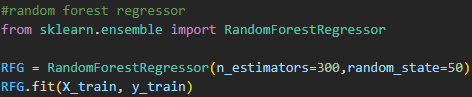
*Univariate Analysis* variabel *‘*sqft\_living15*’*

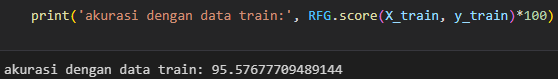




Kepadatan variabel ‘sqft\_above’ berada di angka 1500, terdapat *outlier* di atas 3600.\

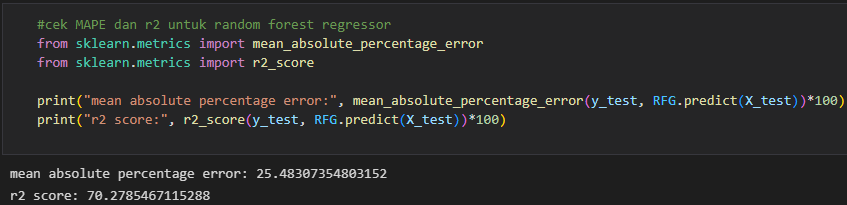
*Training* model dengan algoritma *Random Forest Regression*





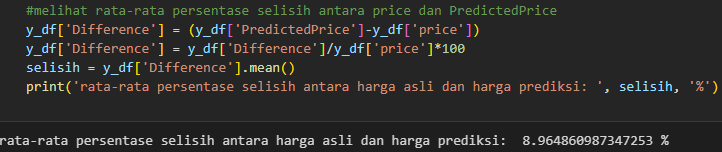
Dengan pembagian dan parameter yang sama, penambahan variabel ‘sqft\_above’ dan ‘sqft\_living15’ meningkatkan nilai akurasi hingga 95.57%

Cek hasil MAPE dan *R Square*

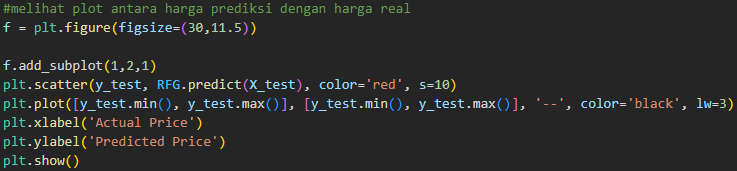
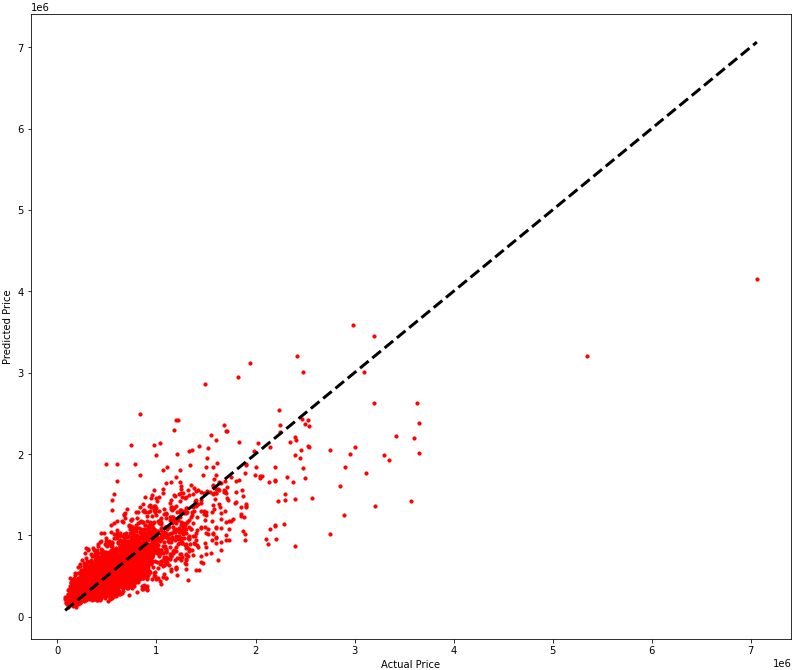


MAPE mendapatkan nilai sebesar 25.48% yang termasuk dalam kategori wajar, *R Square* mendapatkan nilai sebesar 70.27% yang termasuk dalam kategori kuat

Melihat rata-rata selisih antara harga asli dan harga prediksi dari *data test*



Melihat plot antara harga prediksi dengan harga asli

Tabel Perbandingan

| Nilai akurasi | |
| --- | --- |
| Tanpa penambahan variabel | Penambahan 2 variabel |
| 94.64% | 95.57% |

Penambahan variabel ‘sqft\_above’ dan ‘sqft\_living15’ meningkatkan nilai akurasi dari *Random Forest Regression* sebesar 0.93%.

| Nilai MAPE | |
| --- | --- |
| Tanpa penambahan variabel | Penambahan 2 variabel |
| 28.20% | 25.48% |

Penambahan variabel ‘sqft\_above’ dan ‘sqft\_living15’ dapat menurunkan nilai MAPE hingga 2.72%, semakin menurun nilai MAPE menunjukkan semakin kecil kesalahan dalam prediksi.

| Nilai *R Square* | |
| --- | --- |
| Tanpa penambahan variabel | Penambahan 2 variabel |
| 64.71% | 70.27% |

Penambahan variabel ‘sqft\_above’ dan ‘sqft\_living15’ dapat meningkatkan nilai *R Square* hingga 5.56%, dari kategori moderat menjadi kategori kuat.

| Rata-rata Kesalahan Prediksi | |
| --- | --- |
| Tanpa penambahan variabel | Penambahan 2 variabel |
| 9.56% | 8.96% |

Penambahan variabel ‘sqft\_above’ dan ‘sqft\_living15’ dapat menurunkan rata-rata nilai kesalahan prediksi hingga 0.6%

| Plot antar Harga Asli dengan Harga Prediksi | |
| --- | --- |
| Tanpa penambahan variabel | Penambahan 2 variabel |
|  |  |

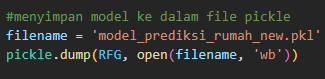
Penambahan variabel ‘sqft\_above’ dan ‘sqft\_living15’ membuat plot semakin rapat dengan garis acuan, menunjukkan bahwa harga prediksi dengan harga asli memiliki selisih yang lebih kecil.

4. Menyimpan Model ke Dalam *File* Pickle

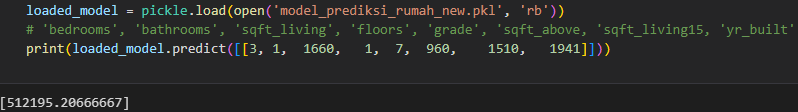
*Import* pickle



Membuat *file* untuk menyimpan model, dan memasukkan model ke *file* tersebut

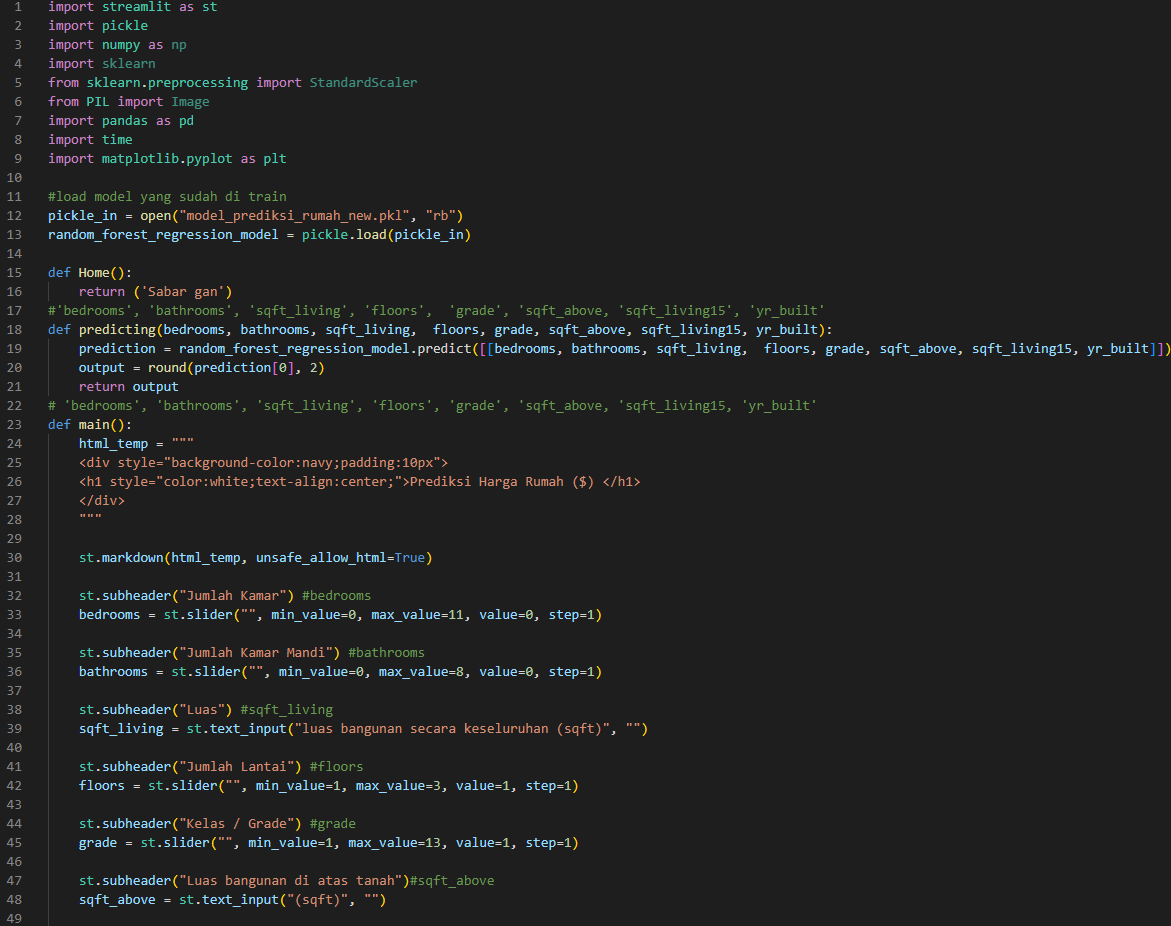


Mencoba prediksi harga dengan *file* pickle yang telah dibuat



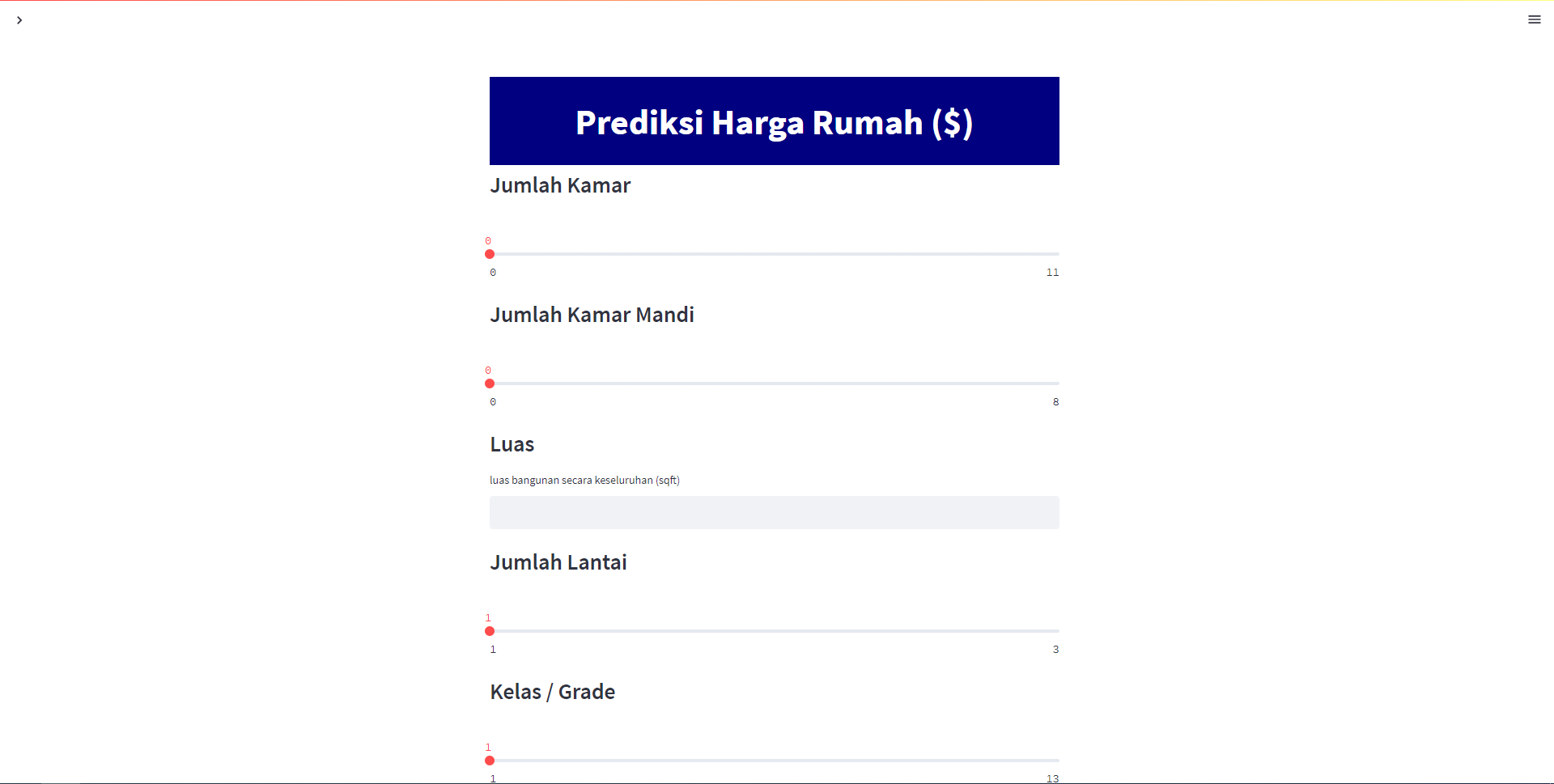
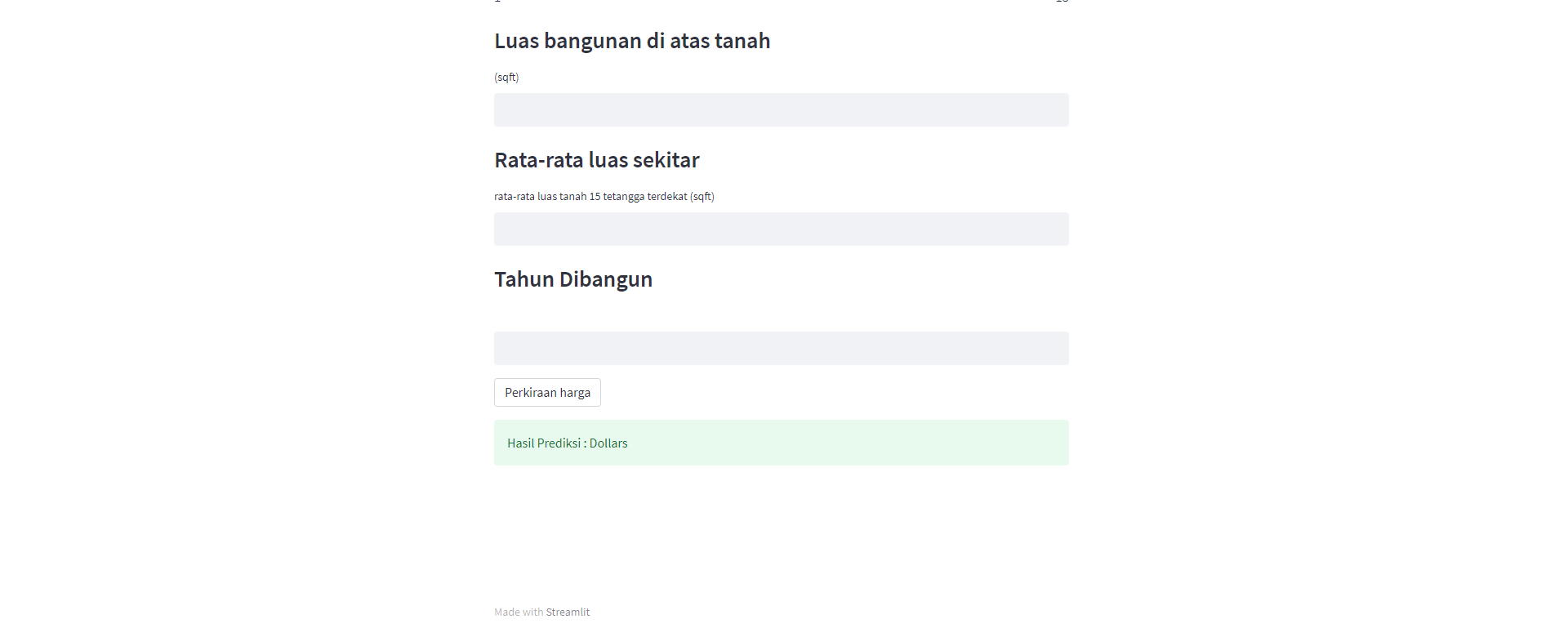
*File* pickle yang telah dibuat dapat berjalan dengan baik

Membuat tampilan *website* agar *user* dapat melakukan prediksi dengan mudah, *website* dibuat dengan library streamlit dari python agar lebih mudah.

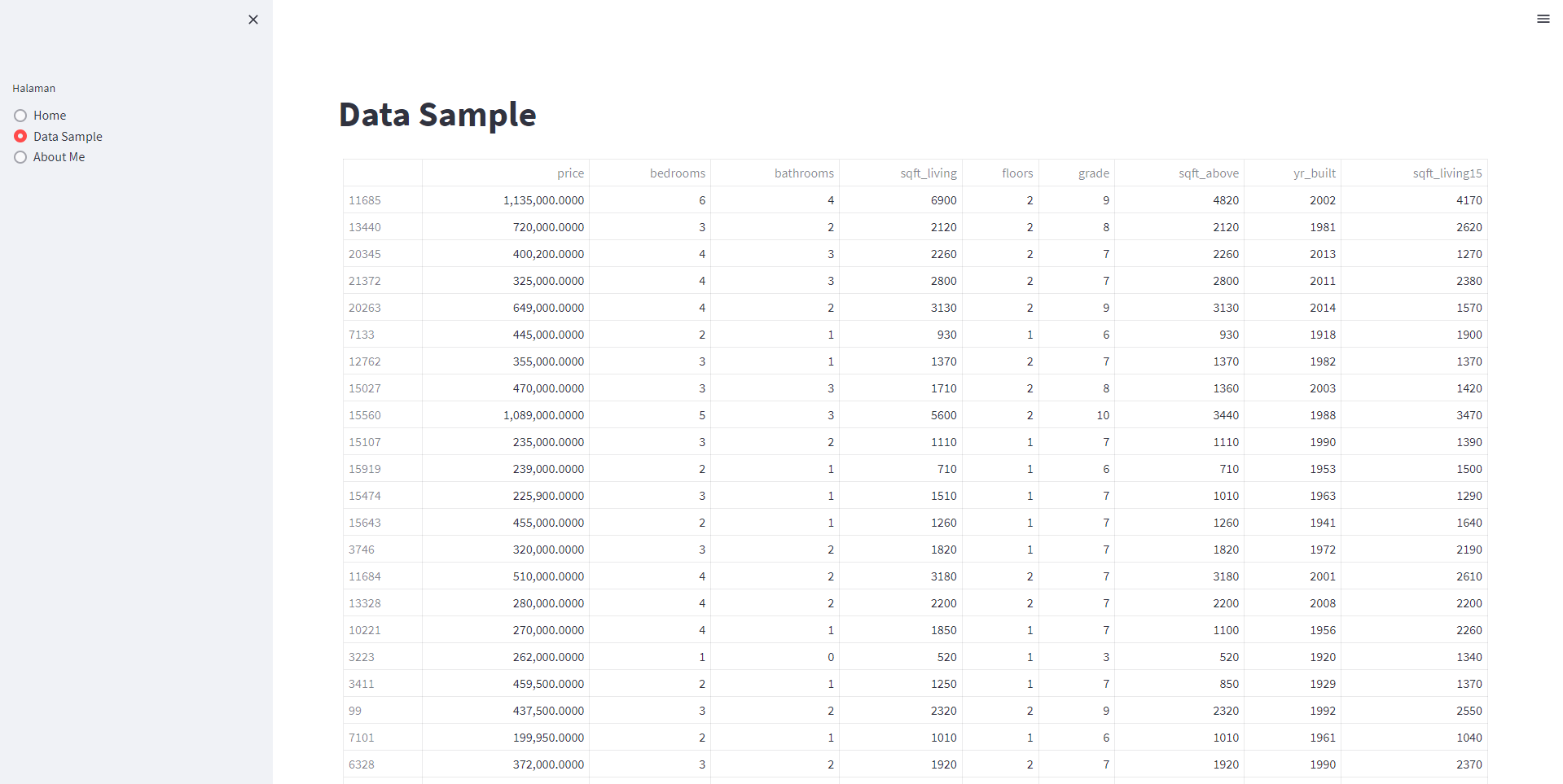
  


Berikut adalah tampilan dari *website* yang telah dibuat

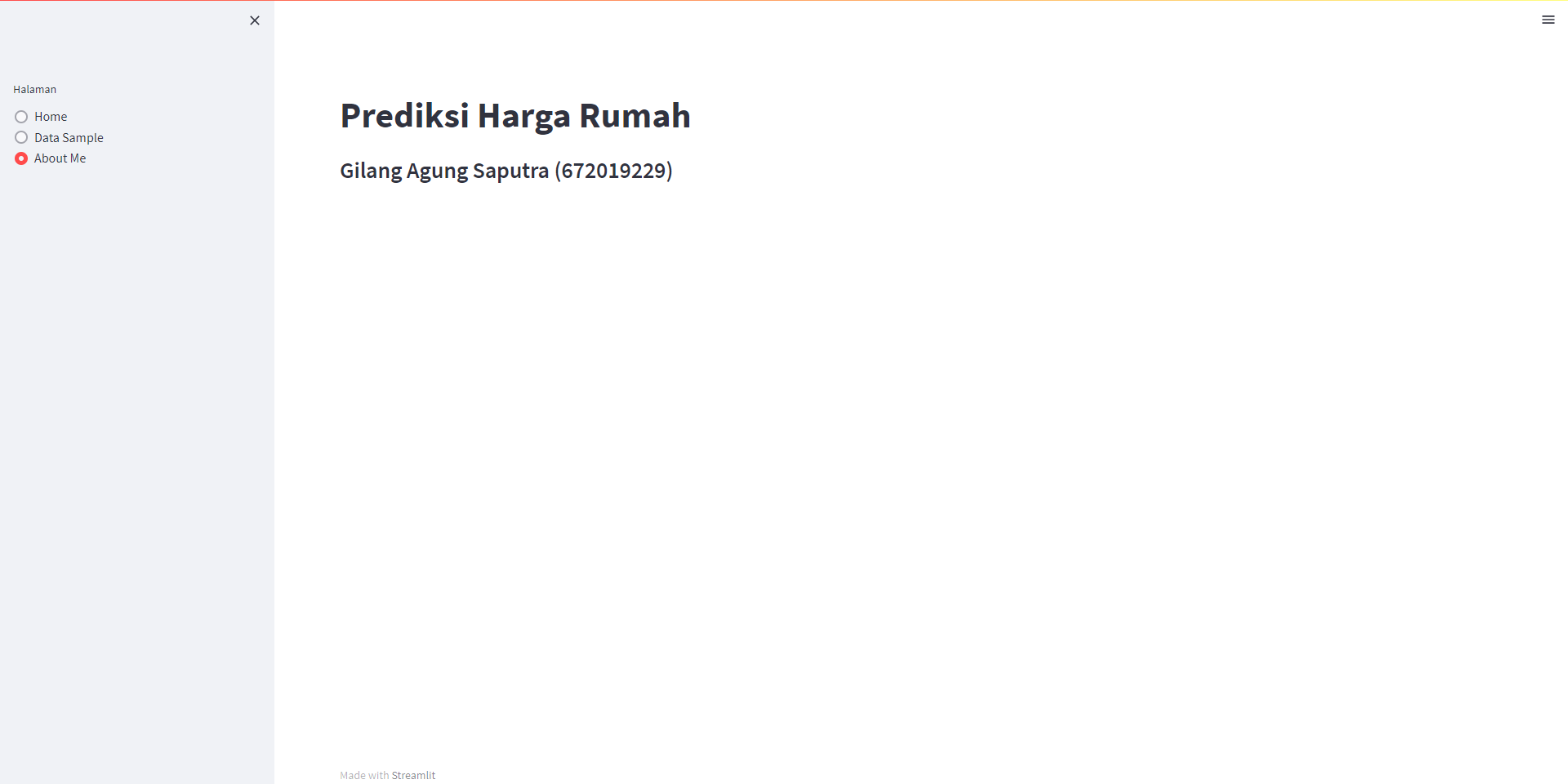
Halaman pertama:

Halaman kedua:



Halaman ketiga:

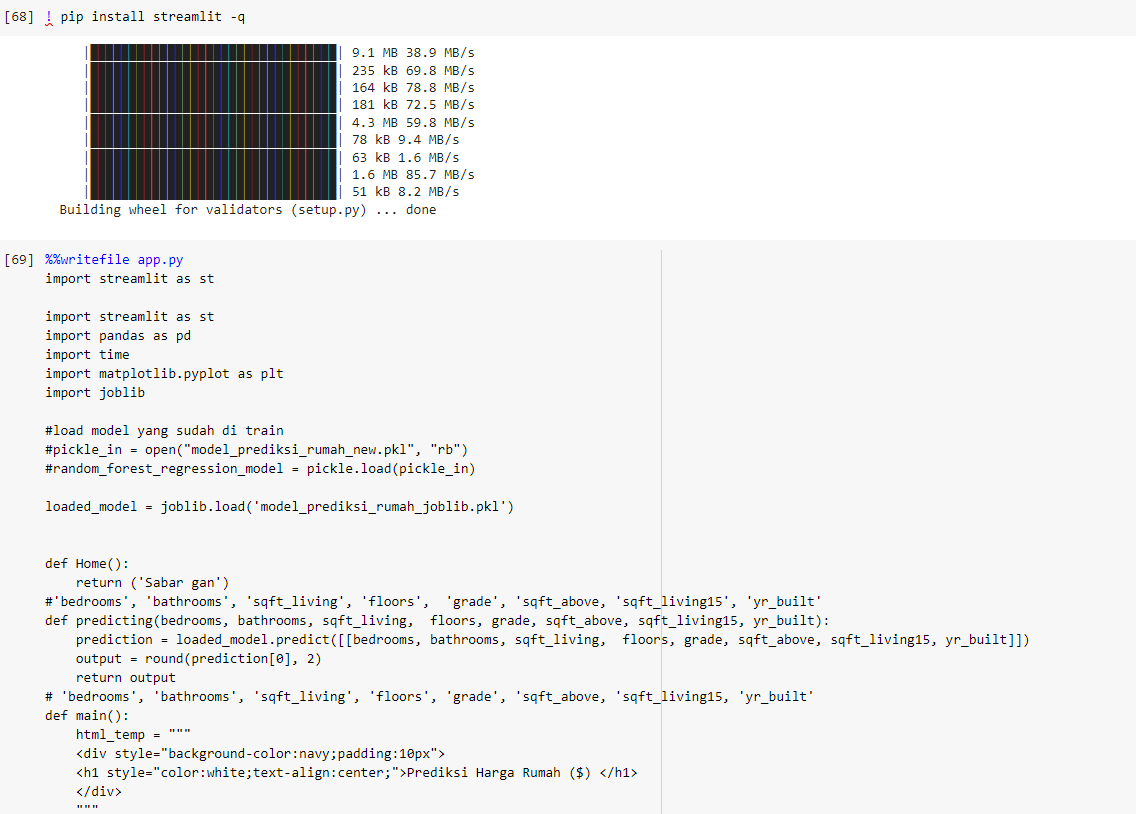


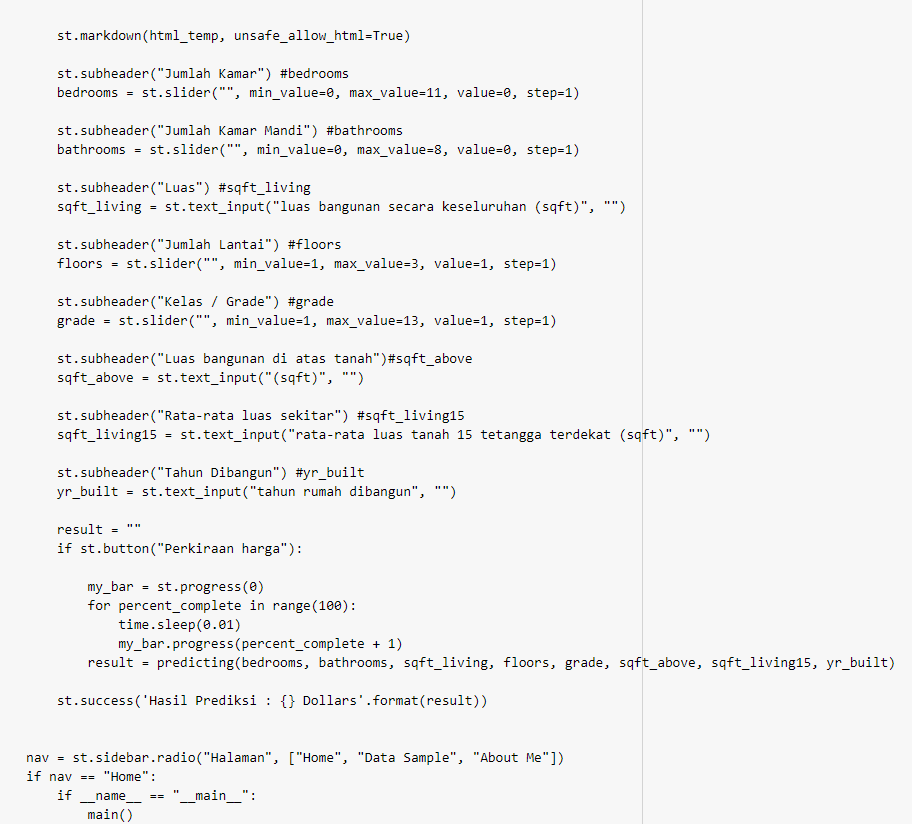
Kita coba di localhost dengan membuka CommandPrompt ke path folder kita, lalu ketik “streamlit run \*nama\_file.py\*”

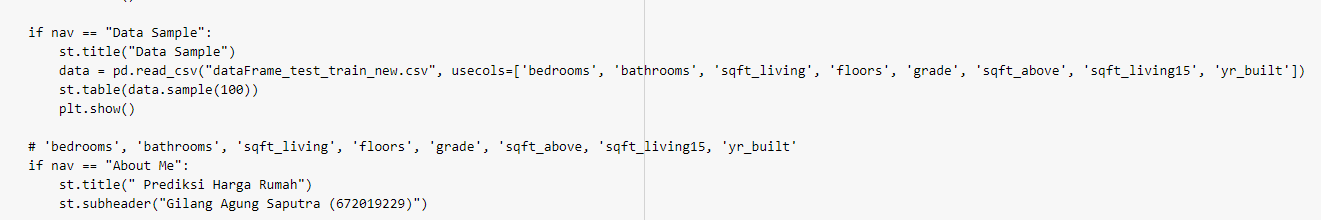
Seluruh kode dan data bisa di akses di “https://github.com/GilangAgungS/house-prediction-fix1”

Hosting Website dari Google Colaboratory dengan LocalTunnel

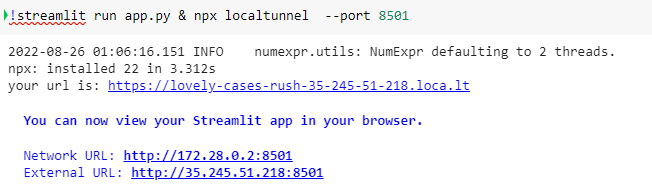
Unggah file ipynb dan data yang digunakan untuk *train test* (bisa upload ke Google Colab atau ke Google Drive), install streamlit pada Colab dan buat file baru bernama “app.py”(atau bisa nama lain berformat .py) dengan kode sebagai berikut:



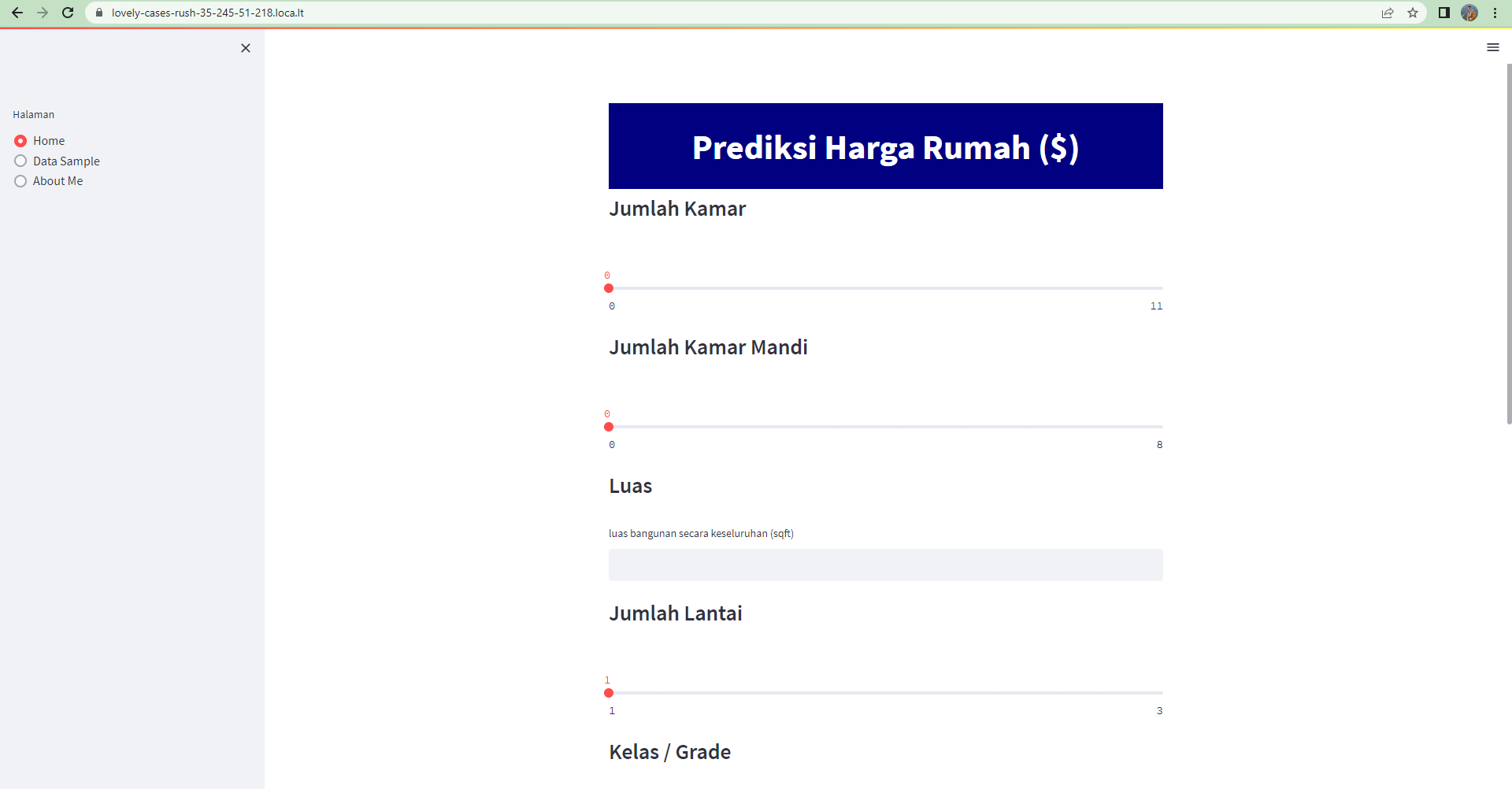




Atau dapat dengan mengetik “%%writefile app.py” kemudian salin kode untuk GUI yang telah dibuat, jalankan *cell* kemudian jalankan kode dibawah ini:



Tunggu hingga muncul url, *Network* URL dan *External* URL, klik url tersebut



Website berhasil dijalankan dan dapat diakses oleh banyak orang, namun localtunnel memiliki kekurangan, jika Google Colab berhenti bekerja maka website tidak akan bisa diakses, dan jika *cell* Colab dijalankan kembali maka url akan berubah. Tutorial menjalankan localtunnel melalui Google Colab dapat dilihat di video berikut:

<https://www.youtube.com/watch?v=NEhrkeF2o_M>

Link github: <https://github.com/GilangAgungS/house-prediction-fix1>