

ANALISIS DATA PERUMAHAN UNTUK MEMPREDIKSI HARGA RUMAH MENGGUNAKAN MACHINE LEARNING

SYAFRIAL FACHRI PANE, RAKASONA, RIFQI FATHURROHMAN



ANALISIS DATA PERUMAHAN UNTUK MEMPREDIKSI HARGA RUMAH MENGUNAKAN *MACHINE LEARNING*

**Syafrial Fachri Pane
Rakasona
Rifqi Fathurrohman**



ANALISIS DATA PERUMAHAN UNTUK MEMPREDIKSI HARGA RUMAH MENGUNAKAN MACHINE LEARNING

Penulis :

Rakasona
Rifqi Fathurrohman
Syafrial Fachri Pane

ISBN :

Editor :

Roni Andarsyah

Penyunting :

Roni Andarsyah

Desain sampul dan Tata letak :

Rifqi Fathurrohman

Penerbit :

Penerbit Buku Pedia

Redaksi :

Athena Residence Blok. E No. 1, Desa Ciwaruga,
Kec. Parongpong, Kab. Bandung Barat 40559
Tel. 628-775-2000-300
Email : penerbit@bukupedia.co.id

Distributor :

Informatics Research Center
Jl. Sariasih No. 54
Bandung 40151
Email : irc@ulbi.ac.id

Cetakan Pertama, 2023

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk dan
dengan cara apa pun tanpa ijin tertulis dari penerbit

KATA PENGANTAR

Sebagai salah satu bentuk upaya untuk memecahkan masalah yang ada dalam dunia properti, kami menghadirkan buku yang berjudul "Analisis Data Perumahan untuk Memprediksi Harga Rumah Menggunakan Machine Learning". Buku ini bertujuan untuk memberikan pemahaman dan solusi terhadap masalah yang ada dalam dunia properti, khususnya memprediksi harga rumah. Dengan adanya teknologi machine learning, kami yakin bahwa buku ini akan memberikan solusi dan pemahaman yang lebih baik bagi para pembaca. Pembahasan yang diberikan akan menunjukkan bagaimana teknologi machine learning dapat digunakan untuk memprediksi harga rumah dan membantu para pemain industri properti dalam membuat keputusan yang lebih baik.

Buku ini sendiri merupakan wujud dari kebutuhan akan pentingnya meningkatkan skill dengan tantangan yang akan dihadapi oleh perusahaan dan generasi milenial di Era Industri 4.0, terutama yang berkaitan dengan komitmen kerja/bisnis dan adversity quotient (AQ). Kami berharap bahwa buku ini dapat memberikan manfaat bagi para pembaca dan dapat menjadi salah satu sumber referensi yang berguna bagi para profesional dalam bidang properti.

Buku ini disajikan dalam setiap bab yang berisi langkah-langkah untuk pembaca pemula. Setiap bab dibuka dengan definisi dan kerangka kerja. Pada bagian selanjutnya diterangkan cara membuat kode program beserta penjelasan setiap baris kode program tersebut. Pada bagian akhir disertakan kode program untuk latihan dan evaluasi pembaca. Setiap chapter atau bab yang disajikan dalam buku ini disertakan juga kode program yang bisa diakses melalui link: <https://github.com/rifqifathurrohman/PREDIKSI-HARGA-RUMAH-MENGGUNAKAN-MACHINE-LEARNING>

Bandung, 12 Februari 2023

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR.....	v
DAFTAR TABEL	vii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
BAB 2 TUJUAN INTRUKSIONAL DAN CAPAIAN PEMBELAJARAN.....	4
2.1 Tujuan Intruksional	4
2.2 Capaian Pembelajaran	4
BAB 3 URAIAN MATERI	5
3.1 <i>State of the art</i>	5
3.2 Teori - teori yang digunakan	6
3.2.1 Data.....	6
3.2.2 Prediksi	7
3.2.3 Harga Rumah	7
3.2.4 <i>Machine Learning</i>	7
3.2.5 <i>Random Forest</i>	9
3.2.6 <i>Outlier</i>	11
3.2.7 Metode Evaluasi	13
3.2.8 CRISP-DM	17
3.2.9 UML (<i>Unified Modeling Language</i>).....	18
3.3 Alat dan aplikasi yang digunakan	19
3.3.1 HTML.....	19
3.3.2 CSS	20
3.3.3 XAMPP	20
3.3.4 Python.....	21

3.3.5	<i>Google Colab</i>	22
3.3.6	<i>VScode</i>	23
3.3.7	<i>Tableau</i>	24
3.3.8	<i>Django</i>	25
BAB 4	HASIL DAN PEMBAHASAN	27
4.1	Analisis	27
4.2	Business Understanding.....	27
4.3	Data Understanding	27
4.4	Preprocessing Data	29
4.4.1	Membaca Data (Pandas).....	29
4.4.2	Input Nilai Pada Kumpulan Data (<i>Simple Imputer</i>).....	30
4.4.3	Melakukan Encoding (<i>Ocean Proximity</i>).....	32
4.4.4	Mendapatkan Nilai <i>Outlier</i>	33
4.4.5	Standarisasi Data	37
4.4.6	Train Test Split	39
4.4.7	Pemodelan.....	39
4.5	Evaluasi Model	41
4.6	Analisis Sistem.....	42
4.6.1	<i>Flowmap</i> Prediksi Harga Rumah.....	42
4.7	Perancangan Sistem (UML)	43
4.7.1	Use Case Diagram	43
4.7.2	Class Diagram.....	51
4.7.3	Activity Diagram.....	52
4.7.4	Statechart Diagram.....	58
4.7.5	ERD (Entity Relationship Diagram)	63
4.7.6	Struktur Menu	64
BAB 5	IMPLEMENTASI APLIKASI	66
5.1	Implementasi Kode Aplikasi.....	66
5.1.1	Implementasi Package Yang Digunakan	66

5.1.2	Implementasi Proses <i>Login</i>	67
5.1.3	Implementasi Proses Register	68
5.1.4	Implementasi Proses Prediksi Harga Rumah	68
5.1.5	Implementasi Proses CRUD Data Rumah	70
5.1.6	Implementasi Proses Visualisasi	73
5.2	Implementasi Halaman Antar Muka.....	74
5.2.1	Implementasi Halaman Depan	74
5.2.2	Implementasi Halaman Login	74
5.2.3	Implementasi Halaman Register.....	75
5.2.4	Implementasi Halaman Prediksi Rumah.....	76
5.2.5	Implementasi Halaman Hasil Prediksi Rumah	77
5.2.6	Implementasi Halaman Menambahkan Data Rumah Baru	77
5.2.7	Implementasi Halaman Daftar data rumah baru	78
5.2.8	Implementasi Halaman Mengedit Data Rumah Baru	78
5.2.9	Implementasi Halaman Visualisasi	79
BAB 6	LATIHAN.....	81
6.1	LATIHAN	81
BAB 7	RANGKUMAN MATERI	80
7.1	RANGKUMAN MATERI.....	80
6.2	Saran	80
DAFTAR PUSTAKA	81
GLOSARIUM	88
TENTANG PENULIS	98

DAFTAR GAMBAR

Gambar 3.1 <i>Data Processing</i>	6
Gambar 3.2 Grafik Regresi dan Klasifikasi	8
Gambar 3.3 <i>Outlier box plot</i>	11
Gambar 3.4 Residual RMSE.....	14
Gambar 3.5 CRISP-DM	17
Gambar 3.6 HTML.....	19
Gambar 3.7 CSS.....	20
Gambar 3.8 XAMPP.....	20
Gambar 3.9 Python	21
Gambar 3.10 <i>Google Colab</i>	22
Gambar 3.11 VScode	23
Gambar 3.12 Tableau	24
Gambar 3.13 Django	25
Gambar 4.1 Sesudah <i>Encoding</i>	32
Gambar 4.2 Sebelum <i>Encoding</i>	32
Gambar 4.3 Plot sebelum <i>cleaning data</i>	34
Gambar 4.4 Plot setelah <i>cleaning data</i>	36
Gambar 4.5 Normalisasi Data	37
Gambar 4.6 <i>Heatmap</i> Normalisasi.....	38
Gambar 4.7 <i>Plot scatter</i> nilai prediksi dan aktual.....	40
Gambar 4.8 Visualisasi Model	41
Gambar 4.9 <i>Flowmap</i> Prediksi Harga Rumah	42
Gambar 4.10 <i>Use Case Diagram</i>	44
Gambar 4.11 <i>Class Diagram</i>	51
Gambar 4.12 <i>Activity Login</i>	52
Gambar 4.13 <i>Activity Registrasi</i>	53
Gambar 4.14 <i>Activity Pengolahan Data</i>	54
Gambar 4.15 <i>Activity</i> Prediksi Harga Rumah	55
Gambar 4.16 <i>Activity Diagram</i> Visualisasi	56
Gambar 4.17 <i>Activity Diagram Logout</i>	57
Gambar 4.18 <i>Statechart Diagram Login Admin</i>	58

Gambar 4.19 <i>Statechart</i> Register	59
Gambar 4.20 <i>Statechart</i> Prediksi harga Rumah	60
Gambar 4.21 <i>Statechart</i> pengolahan data	61
Gambar 4.22 <i>Statechart</i> Visualisasi Diagram.....	62
Gambar 4.23 <i>Entity Relationship Diagram</i>	64
Gambar 4.24 Struktur Menu.....	64
Gambar 5.1 Tampilan UI Halaman depan.....	74
Gambar 5.2 Tampilan UI Depan.....	74
Gambar 5.3 Tampilan UI Register	75
Gambar 5.4 Halaman UI Input nilai prediksi.....	76
Gambar 5.5 Halaman Hasil Prediksi Rumah	77
Gambar 5.6 Halaman Tambah data rumah baru	77
Gambar 5.7 Halaman Daftar data rumah baru	78
Gambar 5.8 Halaman Edit Data rumah baru	78
Gambar 5.9 Halaman Visualisasi dari Tableau.....	79

DAFTAR TABEL

Tabel 3.1 Residual RMSE.....	15
Tabel 4.1 Penjelasan Dataset.....	28
Tabel 4.2 Table Housing.head.....	29
Tabel 4.3 Table <i>Missing Value</i>	31
Tabel 4.4 Imputasi median	31
Tabel 4.5 Setelah Imputasi.....	32
Tabel 4.6 parameter pelatihan dan pengujian	39
Tabel 4.7 Hasil harga aktual dan prediksi	40
Tabel 4.8 Table Definisi Admin	44
Tabel 4.9 Table Definisi <i>Use Case</i>	46
Tabel 4.10 Skenario <i>Use Case</i> Pada <i>Machine learning</i>	47
Tabel 4.11 Skenario <i>Use Case</i> Melakukan Prediksi.....	47
Tabel 4.12 Skenario <i>Use Case</i> Melakukan Visualisasi	48
Tabel 4.13 Skenario <i>Use Case</i> Melakukan Pengolahan Data	49
Tabel 4.14 Skenario <i>Use Case</i> Melakukan <i>Logout</i>	50

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Penulisan Seiring berjalannya tahun populasi manusia di muka bumi semakin meningkat dan juga kebutuhan primer dan sekunder pun ikut meningkat (Diadjeng S, 2021). Akibatnya, Kebutuhan rumah tinggalpun ikut meningkat. Namun, untuk mendapatkan tempat tinggal tidak mudah dikarenakan banyak faktor-faktor yang dijadikan indikator rumah ideal (Bhamani et al., 2020). Berdasarkan indikator tersebut tentunya tidak mudah untuk mendapatkan rumah ideal. Berdasarkan penelitian yang ada, Biasanya indeks harga rumah mewakili ringkasan perubahan harga rumah sedangkan untuk memprediksikan sebuah rumah satu keluarga di perlukan metode yang lebih akurat dengan di dasarkan lokasinya, tipe rumahnya, ukuran, total ruangan, total kamar tidur, dan populasi masyarakat yang tinggal di lokasi tersebut (Lu et al., 2018). Menurut penelitian yang ada mendokumentasikan respons dari harga keseluruhan lokasi perumahan dengan elastisitas 15-20 persen di seluruh siklus perumahan, disebutkan Peningkatan harga rumah disebabkan oleh tingkat kekayaan perumahan yang lebih tinggi yang membuat permintaan pemilik rumah menjadi lebih inelastis. (Maxence Valentin, 2022).

Dalam rangka memperoleh rumah yang ideal, penelitian telah menunjukkan bahwa diperlukan suatu strategi yang dikenal sebagai prediksi. Prediksi adalah suatu proses terstruktur di mana estimasi dibuat mengenai kemungkinan yang akan terjadi di masa depan berdasarkan informasi saat ini dan masa lalu sebagai dasar. (Moons et al., 2019), salah satunya ingin mencari jawaban sedekat apa prediksi harga rumah. Prediksi harga rumah untuk menentukan varians harga rumah dalam setiap daerah (Duca et al., 2021). Daerah yang digunakan merupakan dataset bersumber dari kaggle. Untuk memprediksi rumah ideal berdasarkan penelitian yang ada menggunakan metode HPI (House Price Index) dalam menentukan harga kisaran rumah (Adetunji et al., 2021). HPI adalah indeks repeatsale yang melacak perubahan harga dengan rata-rata dalam transaksi berulang atau pembiayaan kembali dengan aset yang sama (Adetunji et al., 2021). Berbeda halnya, Memprediksi harga rumah tidak bisa menggunakan satu faktor saja. Sedangkan HPI hanya fokus pada satu faktor untuk memprediksikan harga rumah (Duca et al., 2021). Akibatnya

pendekatan HPI dianggap kurang tepat untuk memprediksi harga rumah ideal. Sehingga dapat disimpulkan untuk memprediksi harga rumah ideal penelitian ini menggunakan banyak faktor-faktor yang digunakan sebagai penentu utama untuk memprediksi harga rumah.

Berdasarkan penelitian yang ada solusi untuk memprediksi harga rumah ideal berdasarkan banyak faktor-faktor diperlukan komputasi cerdas dalam hal ini disebut *Machine learning* (Tien Bui et al., 2018). *Machine learning* adalah disiplin ilmu yang mencakup proses pembuatan dan perbaikan algoritma yang memungkinkan suatu komputer untuk belajar dari data yang diperoleh secara langsung dari lingkungan atau sumber data lainnya. (Alzubi et al., 2018). Metode *Machine learning* yang digunakan adalah *Random Forest* Regresor (RFR), Pengelompokan dan regresi adalah metode yang mengandalkan proses agregasi pohon keputusan untuk mengelompokkan dan memprediksi nilai-nilai. (Zhang et al., 2021). *Random Forest* mampu untuk memprediksikan harga rumah berdasarkan banyak faktor (Zhang et al., 2021). *Random Forest* mampu memprediksi kasus berdasarkan data asli/aktual dengan akurasi yang baik. Namun diperlukan juga pendekatan regresi untuk menganalisis faktor-faktor apa saja yang signifikan terhadap menggunakan pendekatan atau metode *spearman*. *Spearman* adalah ukuran nonparametrik dari kekuatan dua arah dan hubungan yang ada untuk mengukur setidaknya pada skala ordinal (Cabras et al., 2021).

Langkah selanjutnya adalah melakukan evaluasi model untuk mendapatkan hasil pemodelan yang dapat diandalkan. Dalam penelitian yang ada, metode evaluasi yang digunakan untuk mengukur performa dalam melakukan prediksi adalah RSquared (Alwated et al., 2021). RSquared adalah sebuah angka yang mengindikasikan seberapa erat keterkaitan antara data yang diperoleh dan model yang digunakan. Semakin mendekati angka 1, maka semakin baik model tersebut dalam menjelaskan variasi dari data yang ada. Selain RSquared, terdapat juga dua metode evaluasi lainnya yaitu MAE (Mean Absolute Error) dan MSE (Mean Squared Error). MAE adalah rata-rata selisih absolut antara nilai sebenarnya dan nilai prediksi dari model. Sementara itu, MSE adalah hasil perbedaan kuadrat antara nilai sebenarnya dan nilai prediksi dari model. Selanjutnya, RMSE (Root Mean Square Error) dapat digunakan sebagai akar kuadrat dari MSE yang berguna untuk memberikan interpretasi yang lebih mudah terhadap nilai MSE.. (Alwated et al. 2021).

Buku ini membahas kontribusi penelitian dalam memprediksi harga rumah, yaitu membuat model prediksi ideal menggunakan algoritma *Random Forest Regressor*, menganalisis variabel-variabel yang memiliki korelasi signifikan dengan menggunakan metode *Spearman*, dan melakukan evaluasi model untuk

menghasilkan kinerja performance terbaik berdasarkan beberapa indeks seperti RSquared, MAE, MSE, dan RMSE.

Berdasarkan latar belakang masalah, buku ini membahas beberapa permasalahan utama dalam memprediksi harga rumah menggunakan teknologi machine learning, yaitu: bagaimana membuat pemodelan yang efektif, menganalisis variabel-variabel yang signifikan, dan mengevaluasi model prediksi untuk menghasilkan model yang baik.

Pada penelitian ini memiliki beberapa manfaat, antara lain dalam membantu menentukan harga rumah yang sesuai dengan harga pasaran dan kriteria rumah yang ada, memahami variabel yang memiliki korelasi sebagai penentu harga rumah, dan mengetahui evaluasi model prediksi yang dibangun untuk menghasilkan model yang baik.

Penulis pun memberikan informasi bahwa data yang digunakan bersumber dari sebuah kota di California, periode pengerjaan adalah 6 bulan, bahasa pemrograman yang digunakan adalah Python, dan software yang digunakan adalah Google Colab dan Visual Studio Code.

BAB 2

TUJUAN INTRUKSIONAL DAN CAPAIAN PEMBELAJARAN

2.1 Tujuan Intruksional

Buku "Analisis Data Perumahan untuk Memprediksi Harga Rumah Menggunakan Machine Learning" bertujuan untuk memberikan pemahaman dan solusi terhadap masalah yang ada dalam dunia properti, khususnya memprediksi harga rumah. Buku ini ditujukan untuk para profesional dan pemain industri properti yang ingin memperoleh wawasan dan solusi yang berkaitan dengan teknologi machine learning dalam memprediksi harga rumah.

2.2 Capaian Pembelajaran

Setelah membaca buku ini, para pembaca diharapkan dapat:

- 1) Memahami bagaimana data perumahan dapat dianalisis menggunakan teknologi machine learning.
- 2) Mengidentifikasi dan mengatasi masalah yang sering dijumpai dalam analisis data perumahan.
- 3) Membangun model machine learning untuk memprediksi harga rumah.
- 4) Menguji dan memvalidasi model machine learning yang dibangun.
- 5) Menggunakan model machine learning secara efektif dalam dunia nyata.
- 6) Membuat keputusan bisnis yang lebih baik dalam bidang properti dengan memanfaatkan informasi yang diperoleh dari analisis data perumahan.

Dengan demikian, buku ini diharapkan dapat memberikan wawasan dan solusi yang berguna bagi para pemain industri properti dalam memprediksi harga rumah dan membuat keputusan bisnis yang lebih baik.

BAB 3

URAIAN MATERI

3.1 *State of the art*

Sebuah industri pasti selalu berkembang begitupun dengan industri 4.0 yang dimana ilmu dan teknologi pada revolusi ini sangat berkembang pesat dimana intelegensia dan internet of things sebagai jalur penghubung antara mesin dan manusia (Prasetyo et al. 2018). Dalam revolusi industri 4.0, Perlu kita ketahui bahwa pada dasarnya *Machine learning* adalah bagian dari *artificial intelligence*.

Artificial Intelligence (AI) adalah istilah umum yang menyiratkan penggunaan komputer untuk memodelkan perilaku cerdas dengan intervensi yang mirip seperti manusia (Paschen et al., 2019). Perkembangan AI atau kecerdasan buatan sebagai platform terkuat, hingga Cloud Computing, atau yang disebut dengan "komputasi awan" dalam bahasa Indonesia, merupakan kombinasi dari teknologi komputer dan pengembangan yang berbasis internet. Salah satu tugas dari komputasi awan adalah menyediakan layanan penyimpanan dan akses data program melalui internet dari berbagai lokasi. (Rashid et al. 2019).

Dalam kasus perkembangan industri di *Artificial Intelligence* (AI) terdapat cabang dari *Artificial Intelligence* (AI) ini yaitu pembelajaran mesin atau *Machine learning* adalah mesin pembelajar yang dirancang untuk belajar secara otomatis dengan menggunakan studi yang didasarkan pada data empiris, seperti data sensor atau data dari *database*. (Paschen et al., 2019).

Pembelajaran mesin diterapkan pada berbagai bidang yaitu: Robotik, Asisten pribadi virtual (Seperti google), Game Pada komputer, pengenalan pola, penambahan data, prediksi harga. Hasil mesin pencari juga bergantung pada ketersediaan jenis dan kategori data yang di latih *Machine learning* juga terbagi oleh banyak teknik *Machine learning* seperti teknik *supervised learning*, *unsupervised learning*, *semi supervised learning*, dan *reinforcement learning* (Sen et al., 2020). pemakain teknik ini juga bergantung pada algoritma yang di gunakan. Dalam kasus ini penelitian mengambil cara untuk memprediksikan

harga rumah menggunakan *Machine learning* ditambah sebuah rumah tidak memiliki ketetapan harga (Adetunji et al., 2021). Rumah selalu berubah-ubah harga setiap tahunnya maka di dalam menentukan harga rumah para peneliti mencoba untuk melakukan pendekatan (AI) terkhusus *Machine learning* agar lebih akurat untuk menentukan harga rumah dengan data yang nantinya akan diolah oleh *Machine learning* untuk menetapkan harga rumah sesuai data yang sudah ditentukan (Varma et al., 2018). Para peneliti sebelumnya menggunakan pendekatan *Machine learning* dengan algoritma *Random Forest* yang dimana algoritma tersebut memiliki nilai keakurasian RSquared: 0.90, MAE: 1.90, MSE: 6.70, RMSE: 2.58 (Adetunji et al., 2021).

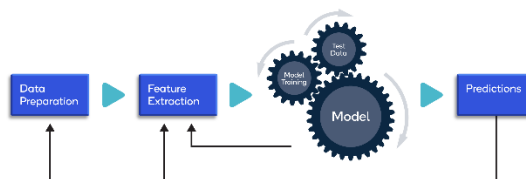
Berdasarkan kumpulan referensi *state of the art* yang digunakan untuk membangun model prediksi menggunakan *Random Forest Regressor* saat membuat prediksi. Django berdasarkan nilai akurasi algoritma *Random Forest* akan jauh lebih baik dari penelitian sebelumnya.

3.2 Teori yang dimplementasikan

Dasar dari teori ini biasanya adalah argumen yang disusun secara sistematis dengan variabel yang kuat dan terbukti. Landasan teori ini meliputi definisi, konsep, dan juga yang disusun secara sistematis sesuai variabel penelitian. Adapun teori yang digunakan dalam penelitian ini sebagai berikut.

3.2.1 Data

Data awal terdiri dari karakter-karakter seperti alfabet, angka, atau simbol khusus yang belum diolah dan memerlukan suatu model pengolahan agar dapat menghasilkan informasi yang bermanfaat. (Meng et al., 2020).



Gambar 3.1 Data Processing

data adalah cara umum untuk menangani data mentah yang tidak sempurna untuk mendapatkan informasi yang andal, berharga, dan akurat. Oleh karena itu, sangat bermanfaat untuk meninjau dan meringkas kecanggihan untuk mendapatkan wawasan mendalam tentang bagaimana

pembelajaran mesin dapat bermanfaat dan mengoptimalkan fusi data (Meng et al., 2020). Data kemudian disiapkan untuk diolah menjadi prediksi sebuah harga perumahan.

3.2.2 Prediksi

Prediksi merupakan suatu proses yang dilakukan secara terstruktur untuk meramalkan kemungkinan terjadinya suatu peristiwa di masa depan dengan memanfaatkan informasi saat ini dan masa lalu. Hal ini dilakukan agar dapat mengurangi kesalahan dalam estimasi yang dibuat. (Kafil, 2019).

Pada penelitian ini, memprediksi harga jual perumahan yang akurat dalam pengoperasian pasar perumahan. Penjual dan pembeli rumah ingin mengetahui nilai wajar untuk rumah mereka khususnya pada saat transaksi penjualan. Perkiraan yang tepat dari harga jual sebuah rumah sangat penting bagi investor yang menghadapi pilihan antara sekuritas perumahan dan peluang investasi lainnya (Heidari et al., 2021). Prediksi pada penelitian ini berada pada sektor rumah.

3.2.3 Harga Rumah

Harga rumah yang tinggi tidak sebanding dengan tingkat apresiasi harga rumah yang tinggi. Meskipun variabel yang sama digunakan, dampaknya terhadap harga rumah dan tingkat apresiasi bisa sangat berbeda. Oleh karena itu, penting dan menjanjikan untuk menguji pengaruh berbagai variabel terhadap apresiasi harga rumah. Sebuah rumah adalah bangunan yang digunakan sebagai tempat tinggal dalam jangka waktu tertentu. (Kang et al., 2021).

Berdasarkan penelitian yang ada solusi untuk memprediksi harga rumah ideal berdasarkan banyak faktor-faktor diperlukan komputasi cerdas dalam hal ini disebut *Machine learning*.

3.2.4 Machine Learning

Machine Learning (ML) merupakan salah satu bentuk implementasi dari *Artificial Intelligence* (AI) yang difokuskan pada pengembangan sistem yang mampu belajar sendiri tanpa perlu diprogram ulang secara berulang. Dengan menggunakan Machine Learning, komputer mampu meningkatkan kinerjanya dalam menyelesaikan tugas-tugas tertentu dengan cara menganalisis dan menyesuaikan diri terhadap data input yang diberikan. (Retnoningsih et al. 2020). ML membutuhkan sebuah data (data training) sebagai proses learning sebelum menghasilkan sebuah hasil (Chazar et al. 2020). Menggambarkan hubungan antar variabel dalam sebuah dataset dapat dilakukan dengan menggunakan persamaan linier di beberapa

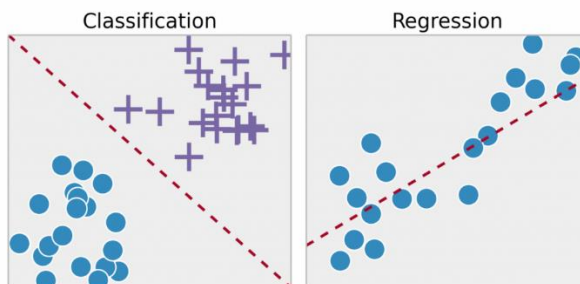
dataset. Misalnya, model regresi linier bertujuan untuk menyesuaikan garis linier yang menghubungkan variabel masukan dengan variabel keluaran. Namun, ketika ada sejumlah besar variabel, atau hubungannya dengan variabel keluaran terlalu kompleks, atau tidak linier untuk dijelaskan oleh garis linier, atau bahkan tidak ada variabel keluaran yang jelas untuk diprediksi dalam kumpulan data, maka model regresi linier klasik gagal (Fachri et al. 2022). Dalam kasus seperti itu, diperlukan pendekatan yang berbeda. Strategi alternatif ini disebut *Machine learning*.

Model *Machine learning* pada umumnya dibagi menjadi empat kategori (Muhammad et al., 2021), yaitu :

- a) Teknik *Supervised Learning*
- b) Teknik *Unsupervised Learning*
- c) Teknik *Reinforcement Learning*

Pada buku ini, model *Machine learning* yang digunakan adalah Supervised learning merupakan metode *Machine learning* yang melibatkan pembelajaran sebuah fungsi yang sesuai dengan pasangan input nilai dan output. Tujuan dari *supervised learning* adalah membuat model yang dapat menghasilkan output yang diinginkan dengan mempelajari fungsi yang sesuai. Prosesnya melibatkan memberikan input dan output yang telah diketahui kepada model, sehingga model dapat memahami bagaimana menghasilkan output yang sesuai dengan input yang diberikan. (Sen et al., 2020).

Dataset pelatihan mengandung input dan output yang benar, sehingga memungkinkan model Machine Learning untuk terus belajar dari waktu ke waktu. Algoritma Machine Learning akan menilai tingkat akurasi model dengan menggunakan fungsi kerugian (Loss Function), dan menyesuaikan model agar kesalahan yang terjadi bisa diminimalkan. Dengan demikian, model akan terus belajar dan meningkatkan performanya dengan menyesuaikan diri terhadap data yang ada. (Siami-Namini et al., 2019).



Gambar 3.2 Grafik Regresi dan Klasifikasi

Supervised Learning dipisahkan menjadi dua jenis masalah terkait, yaitu:

1. **Klasifikasi (*Classification*)** : Menggunakan algoritme untuk mengkategorikan data uji secara akurat adalah salah satu cara penerapan *Machine learning*. Jenis ini mencoba memeriksa entitas yang relevan dalam pengumpulan data dan menghasilkan beberapa hipotesis tentang bagaimana entitas yang relevan harus diberi label atau didefinisikan. Hal ini berguna untuk mengelompokkan data ke dalam kelompok yang sesuai atau mengidentifikasi tipe tertentu dari entitas dalam data. (R. Saravanan et al. 2018).
2. **Regresi (*Regression*)** : *Machine learning* menggunakan teknik regresi untuk memahami hubungan antara variabel terikat dan bebas. Regresi sering digunakan untuk membuat proyek seperti menentukan tingkat pembayaran untuk bisnis tertentu. Algoritma regresi yang paling populer saat ini adalah regresi linier, regresi logistik, dan regresi polinomial. Masing-masing algoritma ini digunakan untuk memprediksi nilai numerik atau probabilitas dengan menggunakan data yang telah ada. (R. Saravanan et al. 2018). Adapun pemodelan yang digunakan menggunakan *Random Forest Regressor*.

3.2.5 *Random Forest*

Algoritma Machine learning supervised yang disebut *Random Forest* diperkenalkan oleh Breiman pada tahun 2001 dan dapat digunakan untuk klasifikasi dan regresi. Metode ini melibatkan pembuatan beberapa pohon keputusan dari dataset pelatihan yang terdiri dari input dan output, dan kemudian menggabungkan hasil dari masing-masing pohon untuk menghasilkan prediksi yang lebih akurat (Mohammady et al., 2019). Algoritma *Random Forest* digunakan untuk menangani masalah seperti klasifikasi dan regresi, dan sebagainya. Ada dua aspek pada algoritma ini yang menjadikannya "random", yaitu:

1. Satu dari dua hal yang membuat algoritma ini random adalah setiap pohon di dalam sampel bootstrap diambil secara acak dari data pelatihan.
2. Dalam *Random Forest*, setiap pohon dalam bootstrap sample diambil secara acak dari data pelatihan dan setiap node split pada pembuatan decision tree menggunakan sampel acak dari variabel data asli.

Algoritma *Random Forest* terdiri dari beberapa pohon prediktor atau decision tree yang digabungkan, dimana setiap pohon bergantung pada vektor acak yang diambil secara acak dan seragam dari semua pohon dalam hutan. Prediksi hasil dari *Random Forest* didapatkan dengan menggunakan

mayoritas hasil dari setiap pohon keputusan individu (untuk klasifikasi) dan rata-rata hasil (untuk regresi)(Chen et al., 2020).

Sebuah RF yang terdiri dari N pohon diformulasikan sebagai berikut:

$$l(y) = \underset{c}{\operatorname{argmax}} \left(\sum_{n=1}^n I_{hn}(y) \right) \quad (1)$$

$$= c$$

Pada rumus (1) I adalah fungsi indikator dan hn adalah tree ke- n dari RF.

Random Forest memiliki mekanisme internal yang memberikan estimasi kesalahan generalisasinya sendiri, yaitu dengan menggunakan estimasi kesalahan *out-of-bag* (OOB). OOB merupakan cara untuk mengestimasi kinerja model dengan menggunakan data yang tidak terpakai dalam proses pelatihan model. Fitur ini berguna untuk mengevaluasi kinerja model tanpa harus mengeluarkan dataset baru untuk uji coba.(Dimitriadis et al., 2018).

Dalam proses pembuatan model *Random Forest*, hanya $2/3$ dari data asli yang digunakan untuk bootstrap saat membangun pohon. Sisanya digunakan untuk menguji kinerja model yang dibuat. Estimasi kesalahan OOB (Out-of-Bag) adalah rata-rata kesalahan prediksi untuk kasus-kasus pelatihan yang tidak digunakan dalam sampel bootstrap. Setelah model dibuat, semua kasus pelatihan akan diuji dengan setiap pohon dan matriks ketetanggaan dari setiap kasus dihitung berdasarkan pasangan kasus yang tiba di node terminal yang sama. Dengan demikian, model *Random Forest* dapat mengestimasi tingkat kesalahan generalisasinya sendiri dengan menggunakan estimasi kesalahan OOB.(Dimitriadis et al., 2018).

Beberapa studi telah menunjukkan bahwa model *Random Forest* menunjukkan kinerja prediksi yang baik dalam berbagai bidang, termasuk regresi, klasifikasi, prediksi finansial, remote sensing, analisis genetik, dan biomedis. Dalam perbandingannya dengan metode lain seperti partial least squares regression, support vector machine, dan neural network, model ini juga menunjukkan performa yang lebih baik. Oleh karena itu, hal ini menunjukkan bahwa model *Random Forest* dapat digunakan untuk

menyelesaikan berbagai jenis masalah prediksi dan klasifikasi dengan hasil yang akurat. (Zolfaghari et al. 2021).

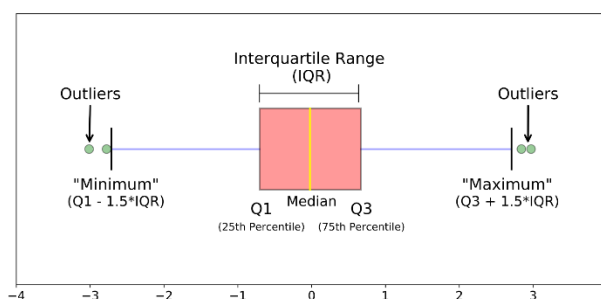
3.2.6 Outlier

Outlier atau yang biasa dikenal dengan *outlier* memiliki nilai ekstrim dibandingkan dengan data rata-rata, dimana penyimpangan dapat muncul berupa angka yang terlalu tinggi atau terlalu rendah. Berikut adalah jenis-jenis *outlier*: *Global Outlier* adalah tipe data *outlier* global yang nilainya jauh melampaui keseluruhan tempat data berada *Outlier* kontekstual yang menyimpang dari data lain Penyimpangan kolektif dalam kumpulan data tertentu, beberapa poin signifikan dari kumpulan data lainnya (Hong et al., 2020). data, secara keseluruhan, menyimpang secara. Objek yang dianggap sebagai *outlier* dengan metode ini adalah ketika

$$X < Q1 - 1.5 \times IQR \text{ dengan } IQR = Q3 - Q1 \quad (2)$$

Pada rumus (2) IQR (Interquartile Range) adalah perbedaan antara persentil 75 dan persentil 25 dari suatu distribusi data. Dapat diformulasikan sebagai $IQR = Q3 - Q1$, dengan Q3 merupakan kuartil ketiga dan Q1 merupakan kuartil pertama (Grech, 2018).

Gambar 2.3 yang diilustrasikan di bawah ini menunjukkan bahwa.



Gambar 3.3 Outlier box plot

Dari nilai yang melebihi/menyimpang secara signifikan dari nilai rata-rata lainnya, sehingga diperlukan beberapa metode evaluasi.

3.2.7 Metode Evaluasi

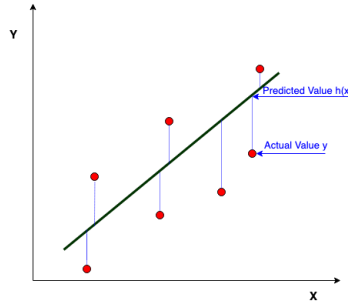
Evaluasi Model Pembelajaran Mesin digunakan untuk mengevaluasi kualitas prediksi suatu model. Model dinilai berdasarkan nilai kesalahannya dalam melakukan prediksi. Beberapa parameter perhitungan digunakan untuk mengevaluasi model yang dihasilkan, seperti Root Mean Square Error (RMSE) yang mengukur rata-rata selisih kuadrat antara hasil prediksi dan hasil aktual yang diharapkan. Mean Absolute Percent Error (MAPE) yang mengukur rata-rata selisih persentase antara hasil prediksi dan hasil aktual yang diharapkan. R-Squared (R²) yang mengukur korelasi antara hasil prediksi dan hasil aktual. Semakin tinggi nilai R², semakin akurat model dalam melakukan prediksi. Ketiga ukuran ini sering digunakan dalam mengevaluasi model Machine learning.(Alwated et al. 2021). Adapun metode evaluasi:

1. *Root Mean Square Error (RMSE)*

RMSE adalah suatu metrik yang mengukur rata-rata perbedaan kuadrat antara nilai prediksi dan nilai aktual. Semakin kecil nilai RMSE, semakin baik model dalam memprediksi data. Biasanya, RMSE digunakan sebagai ukuran evaluasi dalam model Machine Learning, khususnya pada regresi. Cara menghitung RMSE adalah dengan mengurangi nilai prediksi dari nilai aktual, mengkuadratkan selisih tersebut, menjumlahkan hasil total, dan mengambil akar kuadrat dari rata-rata jumlah tersebut. (Botchkarev, 2018). Kemudian hitung kembali hasil perhitungan tersebut dan temukan nilai akar kuadratnya. contoh rumus:

$$RMSE = \sqrt{\frac{\sum (Y' - Y)^2}{n}} \quad (3)$$

Pada rumus (3), Y' merupakan nilai dari variabel dependen yang diprediksi. Kemudian Y merupakan nilai sebenarnya, dan n adalah banyaknya jumlah data



Gambar 3.4 Residual RMSE

Untuk mendapatkan RMSE, yang digunakan rumus Standar deviasi bukannya akar kuadrat dari varians yang akan menghitung akar kuadrat dari rata-rata residu kuadrat (Ahrens et al., 2020).

2. Mean Absolute Error (MAE)

MAE dapat digunakan jika *outlier* mewakili bagian data yang rusak. Faktanya, MAE tidak menghukum terlalu banyak *outlier* pelatihan (norma L1 entah bagaimana memuluskan semua kesalahan *outlier* yang mungkin terjadi), sehingga memberikan ukuran kinerja yang umum dan terbatas untuk model tersebut. Di sisi lain, jika kumpulan pengujian juga memiliki banyak *outlier*, kinerja model akan biasa-biasa saja (Chicco et al., 2021).

$$MAE = \frac{1}{m} \sum_{i=1}^m |X_i - Y_i| \quad (4)$$

Rumus (4) menunjukkan bahwa Mean Absolute Error (MAE) adalah metrik evaluasi yang menghitung rata-rata perbedaan absolut antara nilai prediksi dan nilai aktual. Penggunaan tanda absolut $| \cdot |$ menjamin bahwa nilai MAE selalu positif. Dalam rumus MAE, $|X_i - Y_i|$ menunjukkan besarnya kesalahan atau perbedaan antara nilai aktual dan nilai prediksi. MAE sering digunakan sebagai metrik evaluasi dalam model Machine Learning. Tujuan dari MAE adalah untuk mencari perbedaan yang terkecil antara nilai sebenarnya dan nilai peramalan. MAE biasanya digunakan dalam evaluasi model Machine learning, terutama dalam regresi. (Chicco et al., 2021).

Mean Absolute Error (MAE) sering digunakan untuk membandingkan performa dua model Machine learning dalam melakukan peramalan. Semakin kecil nilai MAE, semakin baik model tersebut dalam memprediksi nilai sebenarnya. Oleh karena itu, saat membandingkan dua atau lebih model, model terbaik adalah model yang memiliki nilai MAE paling kecil. MAE merupakan ukuran kinerja yang penting untuk dipertimbangkan saat memilih model Machine learning yang sesuai untuk digunakan dalam suatu masalah tertentu.

3. Mean Absolute Percentage Error (MAPE)

MAPE (Mean Absolute Percentage Error) adalah sebuah metrik untuk mengukur rata-rata proporsi kesalahan absolut. Untuk menghitung MAPE, nilai prediksi dikurangi dengan nilai aktual, kemudian hasilnya dikalikan dengan 100 dan dibagi dengan jumlah data. MAPE digunakan untuk mengevaluasi akurasi prediksi dari suatu metode peramalan. Baik nilai MAPE dapat bernilai positif atau negatif, tergantung seberapa dekat nilai prediksi dengan nilai sebenarnya. Semakin kecil nilai MAPE, semakin baik kinerja model dalam memprediksi nilai sebenarnya. (Hyndman & Koehler, 2006).

$$MAPE = \frac{\sum_{t=1}^n |(\frac{A_t - F_t}{A_t})|}{n} \quad (5)$$

Pada rumus (5) MAPE, awalnya menghitung selisih absolut antara Nilai Aktual (A) dan nilai Estimasi/Perkiraan (F). Selanjutnya, menerapkan fungsi rata-rata pada hasil untuk mendapatkan nilai MAPE (Shahid et al., 2021).

Adanya simbol absolut dalam rumus MAPE menandakan bahwa hasil perhitungan yang memiliki nilai negatif akan selalu dianggap sebagai nilai positif.

Sesuai dengan penelitian Lewis (1982), nilai MAPE dapat dikategorikan ke dalam 4 kategori, yaitu:

Tabel 3.1 Residual RMSE

<i>Range MAPE</i>	<i>Arti Nilai</i>
<10%	model prediksi yang luar biasa
10 - 20%	model prediksi yang memadai
20 - 50%	model prediksi yang dapat diterima

Ketika nilai MAPE rendah, maka kesalahan dalam hasil perkiraan juga rendah. Namun, ketika nilai MAPE tinggi, maka kesalahan dalam hasil perkiraan juga tinggi. Metode perkiraan dengan kemampuan yang sangat baik akan memiliki nilai MAPE kurang dari 10%, sedangkan metode yang memiliki kemampuan yang baik akan memiliki nilai MAPE antara 10% dan 20%. (Shahid et al., 2021).

4. *R-Squared*

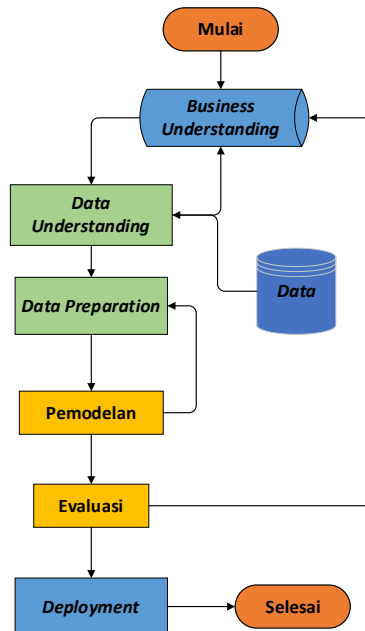
R-Squared adalah ukuran statistik yang menyatakan proporsi varian dari variabel dependen yang dinyatakan sebagai R^2 atau r^2 , diucapkan *r squared* adalah nilai yang menunjukkan bahwa semakin dekat nilai variabel independen R^2 adalah 0 atau 1 banding 1, semakin besar model menggunakan Nah, semakin mendekati 0, semakin buruk model yang digunakan (Rights & Sterba, 2020), rumusnya adalah sebagai berikut :

$$R^2 = 1 - \frac{\sum(Y_i - P_i)^2}{\sum(Y_i - A)^2} \quad (6)$$

Pada rumus (6) Untuk menghitung varians total, dapat mengurangi nilai aktual rata-rata dari setiap nilai aktual, Dari formula di atas Y' di sini adalah nilai sebenarnya. Kemudian P_i adalah nilai variabel dependen yang diprediksi. A merupakan Rata-rata data yang diprediksi atau diamati. menguadratkan Untuk menghitung R-kuadrat, caranya adalah dengan menghitung jumlah varian yang tidak dapat dijelaskan dari hasilnya, lalu menjumlahkannya dengan jumlah varian total. Kemudian, hasil tersebut dibagi dengan satu dan akan didapatkan nilai R-kuadrat (Rights et al. 2020). Untuk merancang tahapan penelitian menggunakan CRISP-DM, Adapun manfaat CRISP-DM sebagai berikut.

3.2.8 CRISP-DM

CRISP DM adalah standar industri yang digunakan secara luas untuk proses data mining yang dikembangkan di Eropa pada tahun 1996. Proses ini terdiri dari enam tahap, di mana tiga tahap awal bersifat eksklusif satu sama lain berdasarkan pengalaman penulis. Tahapan enam tahap dalam CRISP DM mencakup Pemahaman Bisnis, Pemahaman Data, Persiapan Data, Pemodelan, Pengujian, dan Penyebaran. (Huber et al., 2019).



Gambar 3.5 CRISP-DM

Pada gambar 2.5 berdasarkan (Huber et al., 2019) penjelasannya sebagai berikut :

1. Tahap pertama dalam CRISP DM adalah Pemahaman Bisnis (Business Understanding), yang bertujuan untuk memahami kebutuhan dan tujuan bisnis serta menerjemahkan pengetahuan bisnis menjadi definisi masalah dalam konteks data.
2. Tahap kedua dari CRISP DM adalah Pemahaman Data (Data Understanding), di mana dilakukan pengumpulan data, deskripsi data, dan evaluasi kualitas data.
3. Tahap ketiga dalam proses CRISP DM adalah Persiapan Data (Data Preparation), di mana dilakukan proses membangun dataset dari data

mentah melalui serangkaian langkah pembersihan, pemilihan, dan transformasi data.

4. Tahap Pemodelan (Modelling) melibatkan penerapan Machine learning untuk menentukan teknik data mining, alat bantu, dan algoritma data mining yang sesuai dengan tujuan analisis. Dalam penelitian ini, digunakan model Random Forest.
5. dalam proses CRISP-DM adalah evaluasi atau pengujian, yang melibatkan pemeriksaan performa pola data yang dihasilkan oleh algoritma menggunakan parameter seperti Confusion Matrix dengan nilai akurasi, presisi, dan recall.
6. Proses Penyebaran (Deployment) mencakup pembuatan laporan dan artikel jurnal yang memanfaatkan model yang telah dibuat.

Analisis dan perancangan sistem prediksi menggunakan UML Adapun penjelasan UML sebagai berikut.

3.2.9 UML (*Unified Modeling Language*)

UML (Unified Modeling Language) ialah sebuah bahasa standar yang digunakan untuk membantu visualisasi, definisi, pembangunan, dan dokumentasi dari sistem perangkat lunak berbasis OO (Object-Oriented). UML menggunakan gambar atau diagram sebagai alat untuk menampilkan informasi yang diperlukan. Selain itu, UML juga menyediakan standar penulisan blueprint sistem yang mencakup konsep proses bisnis, kode tertulis dalam bahasa pemrograman tertentu, skema database, dan komponen-komponen yang dibutuhkan untuk sistem perangkat lunak. Tujuan utama dari UML adalah untuk menyediakan cara yang standar dalam mendefinisikan, memvisualisasikan, mendokumentasikan, mengedit, dan membuat rencana sistem (perangkat lunak) (Suroso et al., n.d.).

UML (Unified Modeling Language) adalah sebuah bahasa model yang dapat digunakan untuk memodelkan sistem informasi.. Menurut UML terdiri dari beberapa macam diagram menurut (Suroso et al., n.d.) diantaranya:

- a. *Diagram Use Case* digunakan untuk memahami fungsi-fungsi dalam suatu sistem informasi dan siapa yang berhak menggunakannya.
- b. *Diagram Aktivitas* digunakan untuk menggambarkan alur kerja atau aktivitas dari suatu sistem atau proses bisnis.
- c. *Diagram Penyebaran* digunakan untuk menunjukkan konfigurasi komponen dalam proses eksekusi.

- d. *Component* Diagram digunakan untuk menggambarkan hubungan antara sekelompok komponen dalam sistem dan struktur organisasi dari ketergantungan di antara mereka.

Berikut adalah aplikasi dan perangkat yang digunakan dalam penyelesaian penelitian ini.

3.3 Alat dan aplikasi yang digunakan

Untuk menyelesaikan penelitian ini, beberapa alat dan aplikasi yang diperlukan menggunakan bahasa pemrograman Python, HTML, CSS, XAMPP, Django, Google Colab, dan Tableau. Berikut ini adalah alat yang digunakan dalam penelitian tersebut.

3.3.1 HTML

Software yang perlu digunakan adalah bahasa html untuk membangun sebuah aplikasi prediksi harga rumah.



Gambar 3.6 HTML

Hypertext Markup Language, atau yang lebih dikenal dengan sebutan HTML, adalah bahasa markup yang digunakan untuk membuat halaman web. Dengan menggunakan HTML, kita dapat menambahkan berbagai elemen seperti teks, gambar, video, dan tautan ke halaman web yang kita buat. HTML menggunakan tag untuk menandai elemen-elemen yang digunakan dalam halaman web, seperti judul, paragraf, dan tabel. HTML juga digunakan bersama dengan bahasa lain seperti CSS dan JavaScript untuk membuat halaman web yang lebih interaktif dan menarik.(Gur et al., 2022)

3.3.2 CSS



Gambar 3.7 CSS

Setelah mempelajari bahasa HTML, langkah selanjutnya adalah mempelajari bahasa yang dikenal sebagai Cascading Style Sheet (CSS), yang digunakan untuk merepresentasikan tampilan halaman web, termasuk warna, layout, dan font. CSS memungkinkan seorang pengembang web untuk membuat halaman web yang dapat menyesuaikan diri dengan ukuran layar yang berbeda. Biasanya, skrip CSS ditulis terpisah dari halaman HTML meskipun juga dapat dimasukkan ke dalam halaman HTML untuk mempermudah pengaturan halaman web dengan tampilan yang sama. CSS telah menjadi bahasa standar untuk merancang tampilan dan tata letak halaman web sejak diluncurkan pertama kali.(Gyani et al., 2022) bahasa telah dipersiapkan selanjutnya data dan host yang akan menjadi penopang data.

3.3.3 XAMPP



Gambar 3.8 XAMPP

XAMPP adalah perangkat lunak yang digunakan untuk menjalankan aplikasi web di komputer lokal. Ini adalah singkatan dari Apache, MariaDB, PHP, dan Perl. XAMPP menyediakan fasilitas yang diperlukan untuk menjalankan aplikasi web seperti Apache (web server), MariaDB (*database*), PHP (bahasa pemrograman), dan Perl (bahasa scripting) (Kotsifakos et al.,

2018). Untuk menggunakan XAMPP, Anda dapat mengikuti langkah-langkah berikut :

1. unduh XAMPP dari situs web resmi Apache Friends, ekstrak file yang diunduh ke folder yang diinginkan,
2. buka folder XAMPP yang diekstraksi dan jalankan file "xampp-control.exe",
3. di jendela yang muncul , klik tombol "Start" untuk menjalankan Apache dan MariaDB,
4. buka web browser dan masukkan " http://localhost " di bilah alamat, untuk mengakses folder htdocs,
5. Anda dapat mengetik " http://localhost/nama-folder -proyek" di bilah alamat, untuk menghentikan XAMPP, Anda dapat mengklik tombol "Stop" di Panel Kontrol XAMPP.

Harap diingat bahwa saat menjalankan XAMPP, diperlukan akses administrator, sehingga pastikan untuk menjalankannya dengan akun administrator atau sebagai administrator. Selain itu, perlu memastikan bahwa port yang digunakan oleh XAMPP tidak sedang digunakan oleh aplikasi lain seperti Skype atau program antivirus. Jika terjadi konflik, maka perlu mengubah port yang digunakan oleh aplikasi lain atau XAMPP (Sari et al., 2022). Bahasa pemrograman selanjutnya merupakan python yang digunakan dalam *google colab*.

3.3.4 Python

Software yang perlu digunakan adalah bahasa *python* untuk membangun sebuah aplikasi prediksi harga rumah.



Gambar 3.9 Python

Guido van Rossum developed Python programming language in 1989. Python is a highly popular programming language due to its easily comprehensible and user-friendly syntax, making it well-suited for both beginners and professionals (Burhan et al., 2020). Python digunakan dalam berbagai bidang seperti pengembangan aplikasi web, data science, machine learning, intelijen buatan, pengembangan desktop dan mobile, pembuatan

script, dll. Python adalah bahasa pemrograman yang populer digunakan dalam Machine Learning (ML) karena kemampuannya untuk menyediakan akses ke berbagai library dan framework yang dapat digunakan untuk melakukan proses pembelajaran mesin. Beberapa library populer (Raschka et al., 2020) yang digunakan dalam ML menggunakan Python adalah :

1. NumPy: Library ini berfungsi untuk melakukan perhitungan matematika pada array yang memiliki beberapa dimensi atau disebut juga dengan array multidimensional.
2. SciPy: Pustaka ini dimanfaatkan untuk melakukan operasi matematika yang lebih kompleks, termasuk optimasi, integrasi, dan interpolasi.
3. scikit-learn: Library ini digunakan untuk melakukan proses pembelajaran mesin, seperti klasifikasi, regresi, dan clustering.
4. TensorFlow: Library ini digunakan untuk melakukan proses pembelajaran mesin pada skala besar, seperti deep learning.
5. Keras: Library ini digunakan untuk membuat model deep learning dengan mudah.

Adapun beberapa aplikasi untuk mengimplementasi, fasilitas bahasa pemrograman *python* dari Google. Fasilitas ini memungkinkan para pengguna *Google Colab*.

3.3.5 Google Colab



Gambar 3.10 Google Colab

Google Colab adalah aplikasi yang dikembangkan oleh Google untuk menjalankan kode *Python* pada lingkungan yang disebut sebagai notebook. Ini sangat berguna untuk pembelajaran dan prototyping karena Anda dapat menjalankan kode *Python* secara interaktif dan melihat hasilnya secara *real-time*.

Untuk menggunakan *Google Colab*, Anda dapat mengikuti tata cara berikut:

1. Buka browser web dan masuk ke <https://colab.research.google.com>
2. Klik tombol "*New Notebook*" untuk membuat notebook baru.
3. Anda dapat memasukkan kode *Python* di sel yang tersedia dan menjalankannya dengan menekan tombol "*Shift + Enter*" atau mengklik tombol "*Run cells*" (panah hijau).
4. Anda juga dapat menambahkan sel baru dengan mengklik tombol "*Insert*" dan memilih "*Code*" atau "*Text*".
5. Sel-sel yang berisi kode *Python* akan dijalankan pada lingkungan yang disebut sebagai kernel, yang dapat diubah dengan mengklik tombol "*Runtime*" dan memilih "*Change runtime type*".
6. Anda juga dapat menyimpan notebook dengan mengklik tombol "*File*" dan memilih "*Save a copy in Drive*" atau "*Save a copy as Github Gist*".
7. Anda juga dapat memuat file dari lokal maupun dari cloud storage seperti google drive atau upload file dari komputer lokal dengan mengklik tombol "*Files*" dan memilih "*Upload*".

Perhatikan bahwa *Google Colab* menyediakan akses gratis ke lingkungan pemrosesan yang dikelola oleh Google, namun terdapat batasan-batasan dalam penggunaan. Jika Anda ingin menggunakan lingkungan yang lebih kuat atau memiliki kontrol yang lebih besar atas lingkungan, Anda dapat menggunakan solusi self-hosted seperti Jupyter (Vallejo et al., 2022).

3.3.6 VScode

Software selanjutnya yang digunakan untuk membuat webbase adalah Vscode.



Gambar 3.11 VScode

Visual Studio Code atau VSCode merupakan suatu software editor teks yang dikembangkan oleh Microsoft, digunakan untuk menulis kode pada berbagai bahasa pemrograman seperti Python, JavaScript, C++, dan

sebagainya. Software ini memiliki fitur-fitur yang mempermudah pengembangan aplikasi seperti IntelliSense yang memberikan saran kode serta dokumentasi, debugging, dan ekstensi yang dapat ditambahkan. Selain itu, VSCode juga terintegrasi dengan Git, fitur Live Share untuk kerja kolaboratif, dan banyak lagi. Penggunaan VSCode tidak terbatas pada pengembangan aplikasi pada satu platform saja, namun juga dapat digunakan untuk mengembangkan aplikasi web, mobile, desktop, dan IoT dengan teknologi seperti React, Node.js, Angular, JavaScript, dan lainnya. Dengan fitur-fitur yang tersedia, VSCode memudahkan pengembangan aplikasi dengan memberikan dukungan untuk saran kode, debugging, integrasi dengan Git, dan sebagainya.

3.3.7 Tableau



Gambar 3.12 Tableau

Tableau ialah suatu software yang difungsikan untuk membuat visualisasi data. Software tersebut memungkinkan pengguna untuk mengeksplorasi, menganalisis, dan mempresentasikan data dengan cara visual yang sederhana dipahami. *Tableau* dapat dimanfaatkan untuk menganalisis data dari bermacam sumber, seperti basis data, spreadsheet, atau file CSV..

Untuk menggunakan *Tableau*, Anda dapat mengikuti tata cara berikut:

1. Download dan instal *Tableau Desktop* dari situs resmi *Tableau*.
2. Buka *Tableau* dan klik "Connect to Data" untuk menghubungkan *Tableau* dengan data yang akan analisis. Anda dapat menghubungkan *Tableau* dengan berbagai jenis data, seperti *spreadsheet*, *database*, atau file CSV.
3. Setelah terhubung dengan data, Anda dapat mengeksplorasi data dengan mengklik dan menarik kolom yang diinginkan ke area "Drag field here" pada panel kiri. Anda juga dapat menambahkan filter dan ukuran untuk menganalisis data lebih detail.
4. Anda dapat menambahkan visualisasi data dengan mengklik tombol "Sheet" di panel kiri dan memilih tipe visualisasi yang diinginkan, seperti *bar chart*, *line chart*, atau *scatter plot*.

5. Anda dapat menyajikan data dengan mengklik tombol "*Story Points*" di panel kiri dan menambahkan halaman baru untuk menyajikan data dalam bentuk visual yang berbeda.
6. Anda dapat menyimpan proyek Anda dengan mengklik tombol "File" dan memilih "*Save As*".
7. Anda juga dapat membagikan proyek Anda dengan mengeksport dalam bentuk PDF, image, atau mengunggahnya ke *Tableau Server* atau *Tableau Public*.

Tableau menyediakan berbagai fitur yang dapat digunakan untuk menganalisis dan menyajikan data secara visual, sehingga dapat memudahkan dalam proses pengambilan keputusan dan pemahaman data. Namun, untuk menggunakan *Tableau* dengan baik dan benar, diperlukan pengetahuan tentang data dan visualisasi data (Coniglio et al., 2022).

3.3.8 Django



Gambar 3.13 Django

Django adalah framework web *Python* yang digunakan untuk membuat aplikasi web dengan efisien dan cepat. Django menyediakan struktur yang baik untuk mengelola aplikasi web besar dan kompleks dengan menyediakan fitur seperti ORM (Object-Relational Mapping), template engine, dan sistem routing (Sabita et al., 2022).

Untuk menggunakan Django, Anda dapat mengikuti tata cara berikut:

1. Instalasi Django: Anda harus menginstal Django terlebih dahulu dengan menjalankan perintah "pip install Django" di command prompt atau terminal.
2. Membuat proyek baru: setelah menginstal Django, Anda dapat membuat proyek baru dengan menjalankan perintah "django-admin startproject nama_proyek" di command prompt atau terminal. Ini akan membuat folder baru dengan nama proyek yang Anda masukkan, yang berisi file-file yang dibutuhkan untuk membuat aplikasi web.

3. Membuat aplikasi baru: Setelah membuat proyek baru, Anda dapat membuat aplikasi baru dengan menjalankan perintah "python manage".

Django adalah salah satu kerangka kerja web *Python* paling populer. Kerangka kerja ini memungkinkan Anda membangun aplikasi web yang kompleks dengan kode *Python* minimal (Sabita et al., 2022). Django menyediakan struktur yang memungkinkan Anda membuat aplikasi web dengan cepat dan mudah. Untuk menggunakan Django, buat proyek Django baru. Proyek ini berisi semua file dan folder yang Anda perlukan untuk membangun situs web Anda. Setelah membuat proyek, Anda dapat mulai menambahkan kode *Python* untuk membuat situs web. Anda juga dapat menambahkan file HTML, CSS, dan JavaScript untuk menyesuaikan tampilan situs Anda. Setelah selesai, Anda dapat mengakses situs web Anda melalui server Django (Sabita et al., 2022).

BAB 4

HASIL DAN PEMBAHASAN

4.1 Analisis

Proses analisis sistem bertujuan untuk memecah sistem informasi menjadi bagian-bagian yang lebih kecil untuk mengevaluasi masalah atau kendala yang terjadi, agar dapat dilakukan perbaikan atau pengembangan sistem. Sementara itu, perancangan sistem melibatkan proses merancang sistem yang baik dengan memperhitungkan langkah-langkah operasi dan prosedur yang mendukung pengolahan data. Hal ini dilakukan untuk memenuhi kebutuhan pengguna dan memberikan panduan rinci bagi programmer dan ahli terkait dalam pengembangan sistem.

Pada tahap ini, akan dibahas mengenai prosedur analisis yang terdiri dari pembuatan flowmap BPMN, pengkodean, analisis sistem fungsional, serta analisis sistem non-fungsional termasuk perangkat keras dan lunak yang digunakan dan analisis pengguna yang terlibat dalam aplikasi. Tahap ini memegang peranan penting dalam mempersiapkan tahap selanjutnya, yaitu pemahaman bisnis. (*Business Understanding*).

4.2 Business Understanding

Dalam tahap ini, akan dilakukan perencanaan dan pemilihan metode untuk memprediksi harga rumah berdasarkan faktor-faktor spesifik yang ditentukan. Selain itu, juga akan dilakukan analisis terhadap alat dan bahan yang diperlukan untuk setiap fase kegiatan. Kegiatan ini bertujuan untuk memprediksi harga rumah dengan menggunakan data rumah di California yang diperoleh dari Kaggle, dan pendekatan yang digunakan adalah *Machine learning*. Setelah memprediksi harga rumah menggunakan *Machine learning*, model prediksi tersebut akan diterapkan dalam bentuk web base.

4.3 Data Understanding

Pemahaman data yang dimaksud mencakup pemahaman tentang dataset yang digunakan dalam penelitian ini. Data yang digunakan diambil dari salah satu platform, yaitu Kaggle, dengan judul "*Housing California.csv*". Tim peneliti mengumpulkan informasi tentang variabel dalam Sensus 1990 untuk semua

kelompok blok di California yang terdiri dari 1.425,5 orang yang tinggal di daerah yang padat secara geografis. Secara alami, daerah geografis yang dimasukkan dalam data akan menunjukkan kepadatan penduduk yang berlawanan.

Kami menghitung jarak antara centroid dari setiap kelompok blok yang diukur dalam lintang dan bujur. Kami mengecualikan semua grup blok yang melaporkan entri nol untuk variabel independen dan dependen. Data akhir berisi 20.640 pengamatan di 9 fitur, dataset dapat dilihat di kaggle yang sudah di cantumkan melalui qr code di bawah ini.

Tabel 4.1 Penjelasan Dataset

Judul Kolom	Deskripsi	Rentang	Jenis data
<i>longitude</i>	Ukuran seberapa jauh rumah berada di barat; nilai yang lebih negatif lebih jauh ke barat	Rentang nilai bujur dari -180 hingga +180 Set data min: -124,3 Set data maks: -114,3	float64
Latitude	Ukuran seberapa jauh rumah berada di bagian utara; nilai yang lebih tinggi lebih jauh ke utara	Rentang nilai lintang dari -90 hingga +90 Set data min: 32,5 Set data maks: 42,5	float64
Housing MedianAge	Usia rata-rata rumah dalam satu blok; angka yang lebih rendah adalah bangunan yang lebih baru	Set data min: 1,0 Set data maks: 52,0	float64
Total Rooms	Jumlah total ruang dalam satu blok	Set data min: 2,0 Set data maks: 37937,0	float64
Total Bedrooms	Jumlah total kamar tidur dalam satu blok	Set data min: 1,0	float64

		Set data maks: 6445,0	
Population	Jumlah total orang yang tinggal dalam satu blok	Set data min: 3,0 Kumpulan data maks: 35682,0	float64
Households	Jumlah total rumah tangga, yaitu sekelompok orang yang berada dalam satu unit rumah, untuk satu blok	Set data min: 1,0 Set data maks: 6082,0	float64
Median Income	Pendapatan rata-rata untuk anggota keluarga dalam satu blok rumah (diukur dalam puluhan ribu Dolar AS)	Set data min: 0,5 Set data maks: 15,0	float64
Median HouseValue	Nilai rumah median untuk anggota keluarga dalam satu blok (diukur dalam Dolar AS)	Set data min: 14999,0 Kumpulan data maks: 500.001,0	float64
Ocean Proximity	Kedekatan kota dengan laut	-	Object

4.4 Preprocessing Data

Berdasarkan data tersebut, berikut adalah pemaparan kode python pada *Google colab* dalam tahap Preprocessing Data berikut adalah pemaparan tentang preprocessing data.

4.4.1 Membaca Data (Pandas)

Pada Table di bawah ini adalah output codingan dari *pd.read_csv* dimana fitur atau fungsi ini bertujuan untuk membaca file data csv agar dapat di import kedalam *google colab*.

Tabel 4.2 Table Housing.head

no	Longitude	Latitude	Housing_ median_age	Total_rooms	Total_ bedrooms
0	-122.23	37.88	41.0	880.0	129.0

1	-122.22	37.86	21.0	7099.0	1106.0
2	-122.24	37.85	52.0	1467.0	190.0
3	-122.25	37.85	52.0	1274.0	235.0
4	-122.25	37.85	52.0	1627.0	280.0
no	population	household	Median_ income	Median_house_ value	Ocean_ proximity
0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	496.0	177.0	7.2574	352100.0	NEAR BAY
3	558.0	219.0	5.6431	341300.0	NEAR BAY
4	565.0	259.0	3.8462	342200.0	NEAR BAY

Pada Tabel 4.2 hasil dari membaca file CSV yang terletak di jalur file yang ditentukan dan membuat Pandas DataFrame dengan konten file CSV dan menugaskannya ke variabel "*housing*". Pandas adalah pustaka sumber terbuka populer untuk analisis dan manipulasi data dengan Python. Fungsi `read_csv` digunakan untuk membaca file CSV dan mengembalikan DataFrame. Dalam hal ini, file tersebut terletak di folder Google Drive "MyDrive" dan diberi nama "California.csv". selanjutnya dapat menampilkan data teratas (head), terbawah (tail), dan menampilkan typedata dari variabel "*housing*".

Tahapan selanjutnya setelah tahapan mendapatkan nilai outlier adalah memasukkan nilai pada kumpulan data Sebelum memasukkan nilai pada kumpulan data.

4.4.2 Input Nilai Pada Kumpulan Data (*Simple Imputer*)

Simple imputer mencari kolom mana yang terdapat *missing value* dengan menggunakan `.isnull()` dalam mencari data mana yang memiliki *missing value*.

Tabel 4.3 Table *Missing Value*

Longitude	0
Latitude	0
Housing_median_age	0
Total_rooms	0
Total_bedrooms	207
Population	0
Households	0
Median_income	0
Median_house_value	0
Ocean_proximity	0

Setelah dilakukan *missing value* terlihat pada tabel 4.3 bahwa *total_bedrooms* memiliki *missing value* yang berjumlah 207 caranya untuk menggantikan *missing value* yang terjadi pada kolom *total_bedrooms*.

Setelah dilakukan algoritma imputasi menggunakan nilai median data menjadi 0 pada tabel 4.4 untuk *missing valuenya* saat dilakukan pengecekan yang menandakan imputasi telah berhasil.

Tabel 4.4 Imputasi median

Total_bedrooms	Imputasi median
0	129.0
1	1106.0
2	190.0
3	235.0
4	280.0

Tabel 4.5 Setelah Imputasi

Longitude	0
Latitude	0
Housing_median_age	0
Total_rooms	0
Total_bedrooms	0
Population	0
Households	0
Median_income	0
Median_house_value	0
Ocean_proximity	0

Pada tabel 4.5 diatas kode memeriksa jumlah dari semua nilai yang hilang di DataFrame perumahan menggunakan metode `isnull().sum()` dan mencetaknya. Jika hasilnya 0 untuk semua kolom, berarti tidak ada nilai yang hilang di DataFrame.

4.4.3 Melakukan Encoding (Ocean Proximity)

Tahapan berikutnya adalah tahapan melakukann encoding dimana pada tahapan ini merubah kolom kategorikal menjadi nnumerik yang dimana pada dataset ini kolom *ocean_proximity* memiliki tipe data kategorikal maka perlu di lakukan encoding agar preproccesing data dapat berjalan dan input data pada *Machine learning* dapat diolah dengan baik sebagai contoh merah = “0”, biru = “1”.

ocean_proximity	
12655	INLAND
15502	NEAR OCEAN
2908	INLAND
14053	NEAR OCEAN
20496	<1H OCEAN
1481	NEAR BAY
18125	<1H OCEAN
5830	<1H OCEAN
17989	<1H OCEAN
4861	<1H OCEAN

Gambar 4.2 Sebelum
Encoding



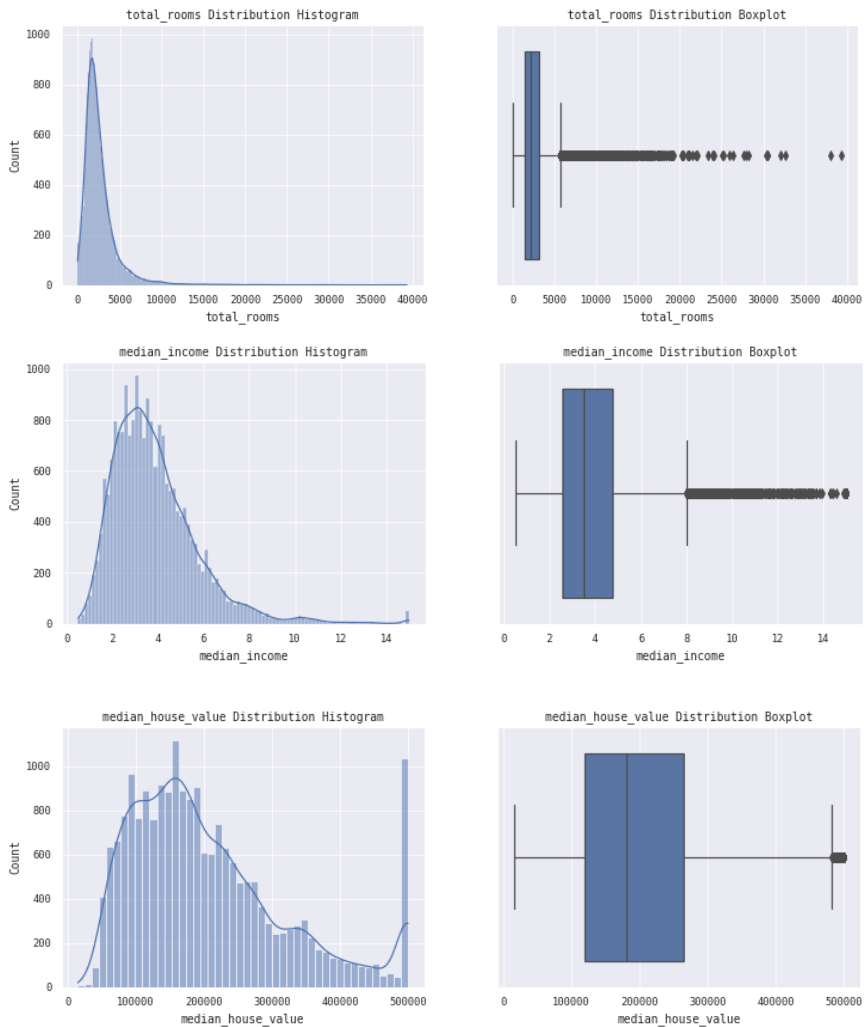
```
array([[1.],
       [4.],
       [1.],
       [4.],
       [0.],
       [3.],
       [0.],
       [0.],
       [0.],
       [0.]])
```

Gambar 4.1 Sesudah
Encoding

4.4.4 Mendapatkan Nilai *Outlier*

Berdasarkan visualisasi sebelumnya, dapat disimpulkan bahwa beberapa kolom dalam dataset, seperti *total_rooms*, *total_bedrooms*, *population*, *households*, *median_income*, *median_house_value*, dan *ocean_proximity*, memiliki nilai yang tergolong sebagai outlier. Untuk mengatasi hal tersebut, langkah-langkah yang dapat diambil adalah dengan menggunakan fungsi *num_dist* yang menerima argumen data dalam bentuk dataframe dan *var* dalam bentuk string yang merepresentasikan nama kolom dalam dataframe, membuat daftar variabel numerik yang akan digunakan nanti untuk melakukan plotting, dan membuat plot untuk menentukan kolom mana saja yang mengandung outlier. Selanjutnya, metode IQR dapat digunakan untuk menghapus data outlier yang ada dalam kolom tersebut..

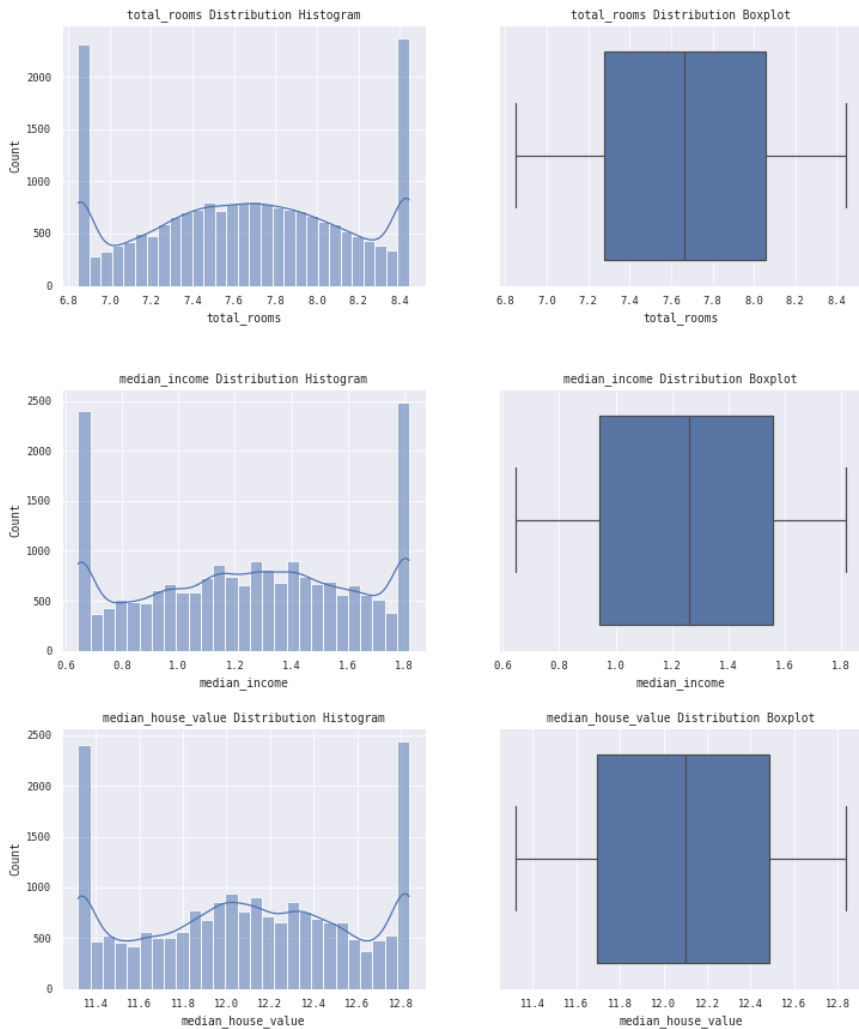
Berikut adalah contoh output dari codingan yang digunakan untuk mendapatkan dan menghapus outlier di kolom tersebut. Sebagai contoh dibawah ini pada gambar 4.3 merupakan 3 dari 6 gambar boxplot berisikan nilai outlier.



Gambar 4.3 Plot sebelum *cleaning data*

Selanjutnya cara mengatasinya dengan membuat variable `outlier_var` dengan didalamnya berisikan kolom yang berisikan data outlier didalamnya `['total_rooms', 'total_bedrooms', 'population', 'households', 'median_income', 'median_house_value']`.

Proses pembersihan data (data cleaning) untuk mengatasi outlier pada variabel yang ditentukan sebagai "outlier_var". Proses pembersihan data ini menggunakan lambda function yang digunakan untuk memotong (clip) nilai outlier pada data. Nilai threshold yang ditentukan sebagai 0.1 digunakan untuk memotong nilai outlier pada 10% data terbawah dan 10% data teratas. Setelah itu, variabel yang telah dibersihkan dari outlier dilakukan logaritmasi. Logaritmasi akan membantu memperbaiki distribusi data yang tidak normal dan membuat data lebih baik untuk dianalisis.



Gambar 4.4 Plot setelah *cleaning data*

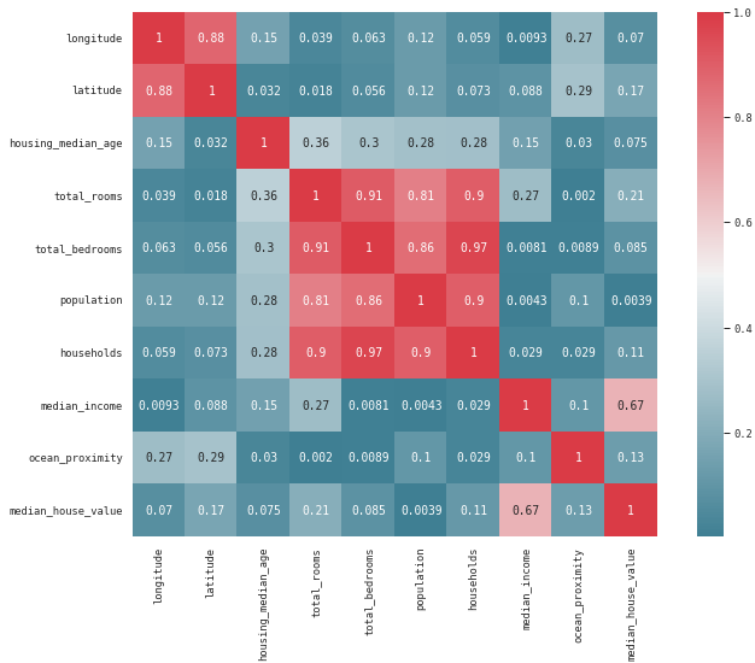
Pada Gambar 4.4 merupakan hasil dari proses yang akan menampilkan data yang telah dibersihkan dari outlier dan dilakukan logaritmasi. Merupakan hasil dari memplot data yang telah berhasil melalui tahap pembersihan.

4.4.5 Standarisasi Data

Standard Scaler adalah sebuah kelas dalam library sklearn yang berfungsi untuk melakukan normalisasi data, dengan tujuan agar data yang digunakan tidak mengalami perbedaan yang signifikan. Selain itu, Standard Scaler juga dapat membantu dalam mengurangi rata-rata fitur dan menskalakan fitur ke unit varians, sehingga proses normalisasi dapat dilakukan dengan lebih baik. Setelah proses normalisasi dilakukan, tahapan selanjutnya dapat dilihat pada gambar di bawah ini.

```
array([[ -1.32783522,  1.05254828,  0.98214266, ..., -1.6194315 ,
         1.51323811,  1.29108888 ],
       [ -1.32284391,  1.04318455, -0.60701891, ...,  1.57731823,
         1.51323811,  1.29108888 ],
       [ -1.33282653,  1.03850269,  1.85618152, ..., -1.6194315 ,
         1.51323811,  1.29108888 ],
       ...,
       [ -0.8237132 ,  1.77823747, -0.92485123, ...,  0.11615787,
        -1.5965484 , -0.11673923],
       [ -0.87362627,  1.77823747, -0.84539315, ..., -0.32121817,
        -1.5965484 , -0.11673923],
       [ -0.83369581,  1.75014627, -1.00430931, ...,  0.52610175,
        -0.99568451, -0.11673923]])
```

Gambar 4.5 Normalisasi Data



Gambar 4.6 Heatmap Normalisasi

Pada gambar diatas ini setelah diketahui oleh hitmap setelah normalisasi data. heatmap berguna untuk melihat korelasi antar kolom yang ada didalam dataset kolom yang memberikan angka 1.0 menggambarkan jika korelasi antar table itu baik namun sebaliknya jika korelasi di bawah 0.5 maka korelasi dianggap kurang baik. Jadi hal yang dilakukan dengan membuat DataFrame X_new baru dengan menghapus kolom 'median_house_value' dari df_scale DataFrame menggunakan metode drop() karena variabel tersebut sangat berpengaruh dan merupakan target yang ingin kita prediksi.

Selanjutnya, DataFrame baru y_new dibuat yang hanya berisi kolom 'median_house_value' dari DataFrame df_scale. Ini akan digunakan sebagai variabel target dalam model *machine learning* nantinya.

4.4.6 Train Test Split

Train-test split adalah teknik yang digunakan untuk membagi dataset menjadi dua bagian, yaitu data untuk pelatihan dan data untuk pengujian. Bagian data yang digunakan untuk pelatihan dikenal sebagai `train_data` dan digunakan untuk mengajarkan model, sedangkan bagian data test digunakan untuk menguji model yang telah dilatih. Untuk melakukan ini, dapat digunakan fungsi `"train_test_split"` dari pustaka `scikit-learn`, yang membagi dataset dalam variabel `"X_new"` dan `"y_new"` menjadi set pelatihan dan pengujian dengan rasio 80:20. Dalam kasus ini, 80% data digunakan untuk pelatihan dan 20% data digunakan untuk pengujian, dengan parameter `"test_size"` yang diatur ke 0,2. Bentuk dan ukuran dataset `X_train`, `X_test`, `y_train`, dan `y_test` menunjukkan bahwa data asli telah dibagi menjadi set pelatihan dan pengujian, seperti yang ditunjukkan pada tabel.

Tabel 4.6 parameter pelatihan dan pengujian

parameter	shape	size
X_train	(16512, 9)	148608
X_test	(4128, 9)	37152
y_train	(16512, 1)	16512
y_test	(4128, 1)	4128

Pada tabel 4.6 kumpulan data ini akan digunakan untuk melatih dan mengevaluasi model pembelajaran mesin. Dataset `X_train` dan `y_train` akan digunakan untuk melatih model, sedangkan dataset `X_test` dan `y_test` akan digunakan untuk mengevaluasi kinerjanya.

4.4.7 Pemodelan

Tahap modeling atau pemodelan adalah proses menentukan algoritma teknik pemodelan yang tepat untuk menyelesaikan masalah yang ada. Model pembelajaran mesin yang digunakan dalam contoh ini adalah algoritma *Random Forest Regressor* untuk memprediksi harga perumahan di California. Dalam proses ini, dataset dibagi menjadi dua bagian yaitu `train_labels` dan `train_prediction` untuk membantu proses prediksi. Setelah

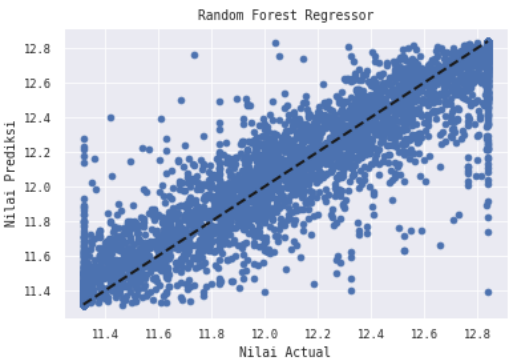
itu, algoritma *Random Forest Regressor* diimplementasikan dengan coding yang telah disediakan.

`n_estimator = 100` yang artinya peneliti membuat 100 decision tree angka ini bisa berubah sesuai kebutuhan, dan yang pasti akan ada perubahan dalam outputnya nanti, dari hasil prediksi yang ditampilkan adalah.

Tabel 4.7 Hasil harga aktual dan prediksi

Harga Aktual	Harga Prediksi
11.318126	11.318126
11.318126	11.379826
12.838939	12.809959
...	...
12.481049	12.300761
12.494255	12.385447
12.838939	12.825780

`Y_test` adalah variabel yang berisikan data aktual dari harga rumah california, sedangkan `rfReg_y_pred` adalah variabel yang berisikan data harga rumah setelah melalui tahap prediksi.



Gambar 4.7 Plot scatter nilai prediksi dan aktual

Pada plot gambar 4.7, dilakukan pembuatan plot scatter menggunakan metode `plt.scatter(y_test, y_pred)` dari perpustakaan "matplotlib". Sumbu x menunjukkan nilai sebenarnya (`y_test`), sedangkan sumbu y menunjukkan

nilai prediksi (y_{pred}). Selain itu, plot juga menampilkan garis terbaik yang didefinisikan sebagai garis yang menghubungkan nilai minimum dan maksimum aktual dengan nilai minimum dan maksimum prediksi yang sesuai.

4.5 Evaluasi Model

Setelah proses implementasi model Random Forest Regressor selesai dilakukan, langkah berikutnya adalah mengevaluasi performa model dengan menggunakan beberapa metrik evaluasi seperti R-Square, Root Mean Square Error (RMSE), dan Mean Absolute Percentage Error (MAPE). Gambar 4.8 menampilkan hasil dari evaluasi model yang telah dilakukan.

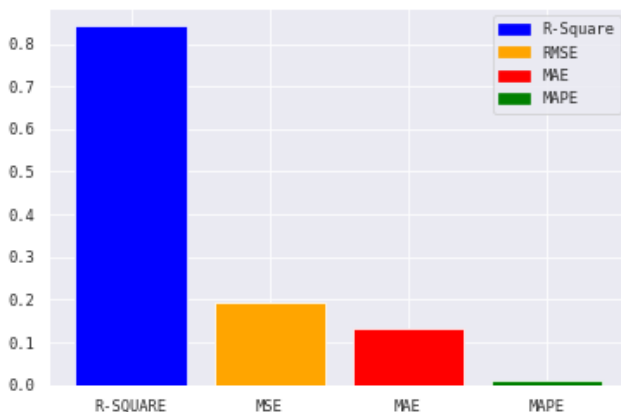
Dengan Nilai :

$R\text{-squared} = 0.8424999630150984$

$RMSE\ score = 0.1933382643380483$

$MAE = 0.13324184029644312$

$MAPE\ score = 0.011017254128622422$



Gambar 4.8 Visualisasi Model

4.6 Analisis Sistem

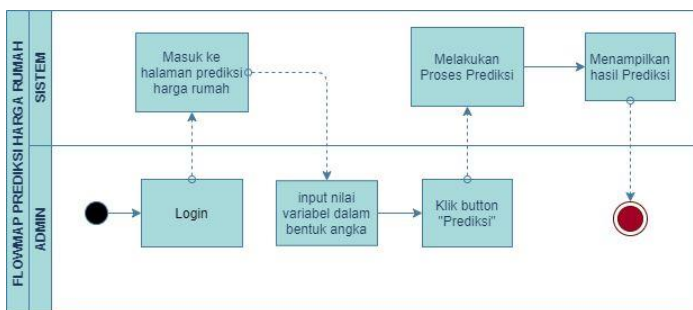
Perencanaan sistem melibatkan proses memecah sistem informasi menjadi bagian-bagian yang lebih kecil untuk mengevaluasi masalah atau kendala yang ada, serta melakukan perbaikan atau pengembangan pada sistem.

Sementara perancangan sistem adalah proses merancang dan mendesain sistem yang baik dengan mengikutsertakan langkah-langkah operasi dan prosedur yang digunakan. Fokus dari perancangan sistem adalah memenuhi kebutuhan pengguna sistem dan memberikan gambaran jelas dan rancangan lengkap kepada programmer dan ahli yang terlibat. Pada tahap ini, analisis prosedur diwujudkan dalam bentuk flowmap BPMN, pengkodean, analisis sistem fungsional dan non-fungsional, termasuk perangkat keras dan perangkat lunak yang digunakan, serta analisis pengguna yang terlibat dalam aplikasi. Tahap ini sangat penting untuk membantu memperjelas langkah selanjutnya dalam perancangan sistem..

4.6.1 Flowmap Prediksi Harga Rumah

Pada bagian ini dijelaskan tentang alur proses prediksi Harga Rumah yang dilakukan oleh admin, yang akan ditampilkan dalam bentuk *flowmap*. *Flowmap* ini menunjukkan alur proses yang akan diterapkan dalam aplikasi. Ada dua pelaku dalam kegiatan ini, yaitu admin sebagai pengguna sistem dan sistem sebagai komponen informasi yang digunakan.

Berikut adalah *flowmap* proses prediksi Harga Rumah yang akan diterapkan dalam pembuatan aplikasi.



Gambar 4.9 Flowmap Prediksi Harga Rumah

Keterangan :

1. Akses dapat dilakukan tanpa melakukan *login*.
2. Admin bisa langsung menginputkan data variabel berbentuk angka.
3. Lalu klik tombol prediksi pada bagian laman.
4. Maka sistem akan langsung melakukan prediksi.
5. Menampilkan hasil prediksi.

4.7 Perancangan Sistem (UML)

UML adalah sebuah bahasa pemodelan yang digunakan untuk menggambarkan sistem atau aplikasi yang didasarkan pada objek. UML dianggap sebagai standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. Selain itu, UML juga dikenal sebagai metodologi yang digunakan untuk mengembangkan sistem OOP (Object-Oriented Programming) dengan alat-alat yang membantu dalam pengembangan sistem tersebut. Oleh karena itu, UML dapat menjadi cara yang mudah untuk mengembangkan aplikasi secara berkelanjutan dan juga bermanfaat sebagai alat bantu untuk mentransfer pengetahuan tentang sistem yang akan dikembangkan dari satu pengembang ke pengembang lainnya.

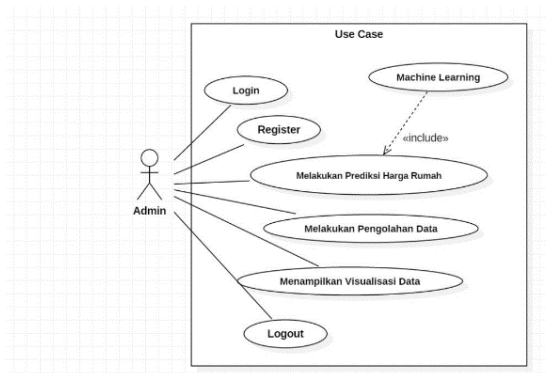
4.7.1 Use Case Diagram

Diagram use case merupakan salah satu jenis dari UML (Unified Modeling Language) yang digunakan untuk memvisualisasikan interaksi antara sistem dan aktor yang terlibat dalam sistem tersebut. Fokus dari diagram ini adalah pada fungsi-fungsi yang disediakan oleh sistem serta bagaimana aktor-aktor terlibat dalam interaksi dengan sistem tersebut.

Diagram *use case* terdiri dari beberapa elemen utama yaitu:

- 1) Aktor merupakan entitas yang terlibat dalam interaksi dengan sistem, yang dapat berupa manusia atau perangkat lain. Aktor dapat diwakili oleh berbagai hal seperti manusia, organisasi, atau perangkat yang menggunakan sistem.

- 2) Use case adalah tindakan atau kegiatan yang dapat dilakukan oleh sistem dengan bantuan aktor, dan digunakan untuk menggambarkan bagaimana sistem memenuhi kebutuhan aktor.
- 3) Association adalah garis yang menghubungkan aktor dengan use case, yang menggambarkan bagaimana aktor terlibat dalam use case.
- 4) Generalization adalah garis yang menghubungkan use case dengan use case lainnya, yang menggambarkan kesamaan atau kemiripan dalam tindakan antara use case yang terhubung..



Gambar 4.10 Use Case Diagram

1. Aktor

Pada bagian ini akan dijelaskan aktor-aktor yang terlibat dalam Sistem.

Tabel 4.8 Table Definisi Admin

No	Aktor	Deskripsi
1	Admin	<ul style="list-style-type: none"> Melakukan Prediksi Harga Rumah Melakukan Pengolahan Data

		<ul style="list-style-type: none">• Melakukan Visualisasi Data• <i>Logout</i>
--	--	--

2. Use case

Tabel 4.9 Table Definisi Use Case

No	Aktor	Deskripsi
1	<i>Machine learning</i>	Merupakan aktivitas interaksi antara manusia dengan mesin. Dalam hal ini, admin membuat model prediksi <i>Machine learning</i> .
2	<i>Login</i>	Merupakan aktivitas <i>login</i> yang dilakukan oleh admin. Sebelum admin masuk ke aplikasi ia harus melakukan <i>login</i> terlebih dahulu.
3	Melakukan Prediksi harga rumah	Merupakan aktivitas memprediksi harga rumah yang dilakukan oleh admin dengan parameter yang telah diperiksa oleh admin.
4	Melakukan Pengolahan Data	Merupakan aktivitas <i>insert, read, update</i> dan <i>delete</i> data rumah yang dilakukan oleh admin.
5	Menampilkan Visualisasi Data	Merupakan aktivitas visualisasi grafik data rumah yang dilakukan oleh admin.
6	<i>Logout</i>	Merupakan aktivitas <i>Logout</i> yang dilakukan oleh admin. Admin dapat keluar dari aplikasi jika telah selesai melakukan pekerjaan pada aplikasi.

3. Alur *Use case*

Setelah fungsional *use case* dijalankan, diharapkan akan ada skenario *use case*. Selain itu, akan diberikan tanggapan dari sistem terkait aksi yang dilakukan oleh aktor. Setiap *use case* akan memiliki skenario yang menjelaskan secara rinci interaksi yang terjadi di dalamnya

Tabel 4.10 Skenario *Use Case* Pada *Machine learning*

Identifikasi	
Nomor	1
Nama	<i>Machine learning</i>
Tujuan	Pembuatan model prediksi
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Membuat model prediksi	Melakukan olah data rumah
Kondisi Akhir	Masuk pada aplikasi

Tabel 4.11 Skenario *Use Case* Melakukan Prediksi

Identifikasi	
Nomor	1
Tujuan	Melakukan Prediksi Harga Rumah
Deskripsi	
Aktor	Admin
Skenario Utama	

Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Menginput Data	Melakukan Prediksi Harga Rumah
Kondisi Akhir	Admin dapat memprediksi Harga Rumah

Tabel 4.12 Skenario *Use Case* Melakukan Visualisasi

Identifikasi	
Nomor	1
Nama	Melakukan Visualisasi Data
Tujuan	Melakukan visualisasi data Rumah
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Input data karyawan	Menampilkan grafik visualisasi data rumah
Kondisi Akhir	Admin dapat memvisualisasikan data

Tabel 4.13 Skenario *Use Case* Melakukan Pengolahan Data

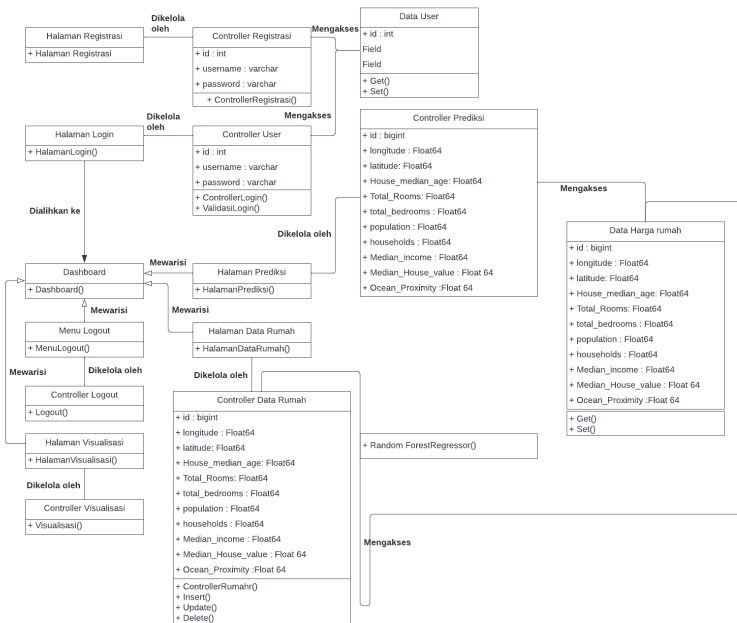
Identifikasi	
Nomor	1
Nama	Melakukan Pengolahan Data
Tujuan	Melakukan pengolahan data Rumah
Deskripsi	
Aktor	Admin
Skenario Utama	
Tahap Awal	
Aksi Aktor	Reaksi Sistem
<i>Insert, read, update, dan delete</i> data karyawan.	Mengubah data Rumah yang diubah oleh admin
Kondisi Akhir	Admin dapat mengelola data

Tabel 4.14 Skenario *Use Case* Melakukan *Logout*

Identifikasi	
Nomor	1
Nama	<i>Logout</i>
Tujuan	Keluar dari aplikasi
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Menekan tombol <i>Logout</i>	Melakukan proses keluar dari aplikasi
Kondisi Akhir	Keluar dari aplikasi

4.7.2 Class Diagram

Class diagram adalah salah satu jenis diagram dalam UML (Unified Modelling Language) yang digunakan untuk menggambarkan struktur dan hubungan antar class dalam suatu sistem. Diagram ini menunjukkan bagaimana kelas-kelas tersebut berinteraksi dan berhubungan satu sama lain dalam sistem yang dibuat. Diagram kelas terdiri dari beberapa elemen utama seperti kelas, atribut, operasi, asosiasi, dan generalisasi. Kelas mewakili objek dalam sistem dan menggambarkan atribut dan operasi yang dimilikinya. Atribut menggambarkan data yang disimpan oleh kelas, operasi menggambarkan tindakan atau aktivitas yang dapat dilakukan oleh kelas, asosiasi menggambarkan hubungan antar kelas, dan generalisasi menggambarkan bahwa kelas adalah subkelas dari superkelas yang terhubung. Diagram 4.11 menunjukkan class diagram dalam bentuk ilustrasi.



Gambar 4.11 Class Diagram

4.7.3 Activity Diagram

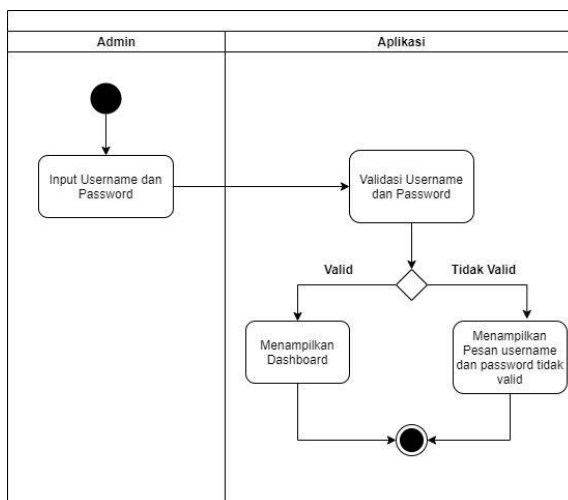
Activity Diagram merupakan sebuah diagram yang digunakan dalam UML (*Unified Modeling Language*) untuk menggambarkan aliran aktivitas dalam suatu sistem. Diagram ini menunjukkan langkah-langkah yang dijalankan dalam suatu proses atau kegiatan, serta kondisi yang dapat mempengaruhi aliran aktivitas.

a. Activity Diagram Login Admin

Berikut ini adalah *Activity diagram* yang menggambarkan proses *login* admin dalam suatu sistem. Diagram ini terdiri dari beberapa komponen seperti aktor admin dan aplikasi, satu initial node, empat activity yaitu:

1. Input *Password*,
2. validasi *username* dan *password*,
3. menampilkan *dashboard* dan menampilkan pesan *username* dan *password* tidak valid,
4. satu percabangan untuk valid dan tidak valid, dan satu *Activity Final Node*.

Dapat dilihat pada gambar 4.10 merupakan activity diagram *login* admin menggambarkan proses *login* admin.



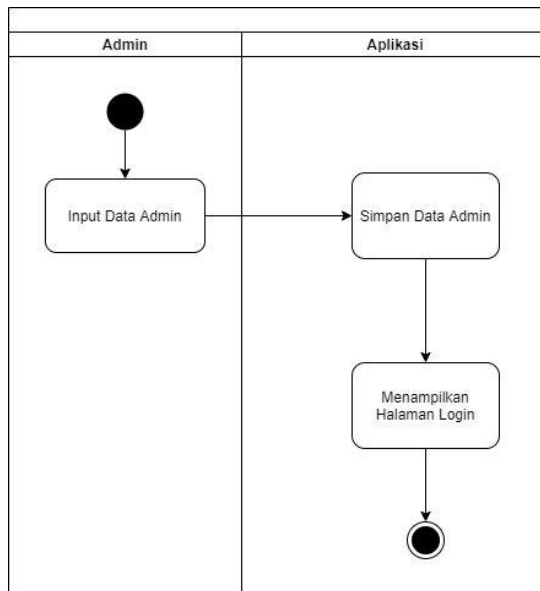
Gambar 4.12 Activity Login

Keterangan :

1. Admin menginput *username* dan *password*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid menampilkan pesan *username* dan *password* tidak valid.

b. Activity diagram Melakukan Registrasi

Diagram aktivitas registrasi admin yang menunjukkan berbagai aktivitas yang terjadi dalam suatu sistem, yang terdiri dari beberapa komponen.



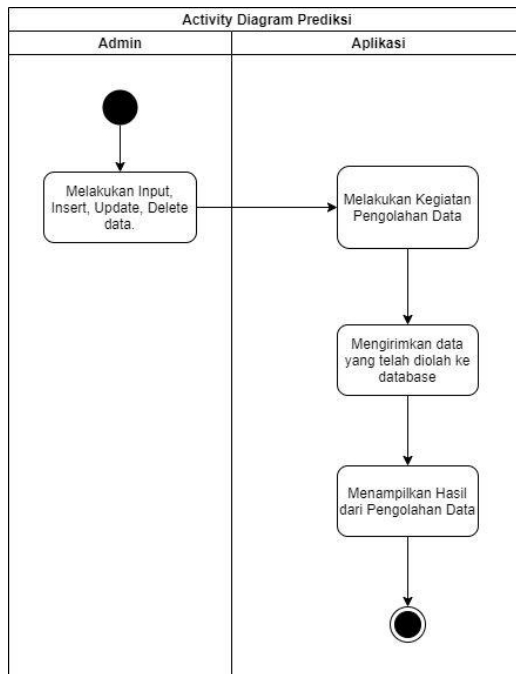
Gambar 4.13 Activity Registrasi

Keterangan :

1. Aktor : Admin dan Aplikasi
2. Satu initial node
3. *Activity* : Terdapat tiga activity yang dapat dipaparkan sebagai berikut.
 - a) Input data admin
 - b) Simpan data admin
 - c) Menampilkan halaman *login*

c. Activity Melakukan Pengolahan data

Diagram aktivitas pengolahan data yang menunjukkan berbagai tahap yang terjadi dalam suatu sistem, yang terdiri dari beberapa komponen:.



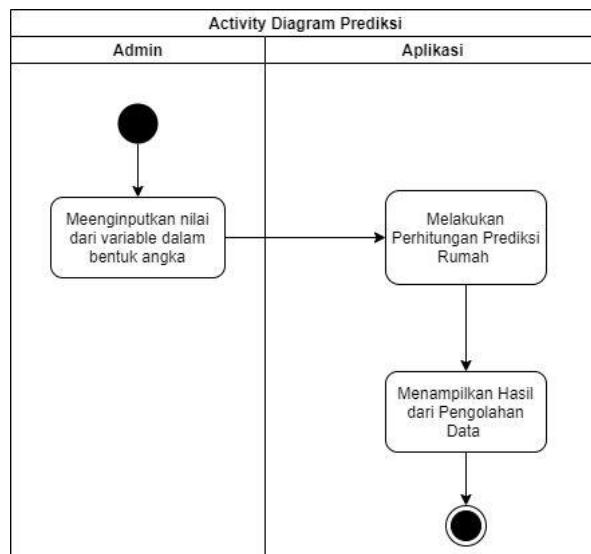
Gambar 4.14 Activity Pengolahan Data

Keterangan :

1. Admin melakukan olah data seperti insert, update, dan *delete* data.
2. Aplikasi melakukan proses olah data yang dilakukan.
3. Aplikasi mengirim data yang diolah ke *database* pegawai. Aplikasi menampilkan data hasil *create/update/delete*.

d. Activity Prediksi harga Rumah

Dibawah ini adalah tahapan dalam activity diagram untuk dilakuka prediksi oleh sistem:.



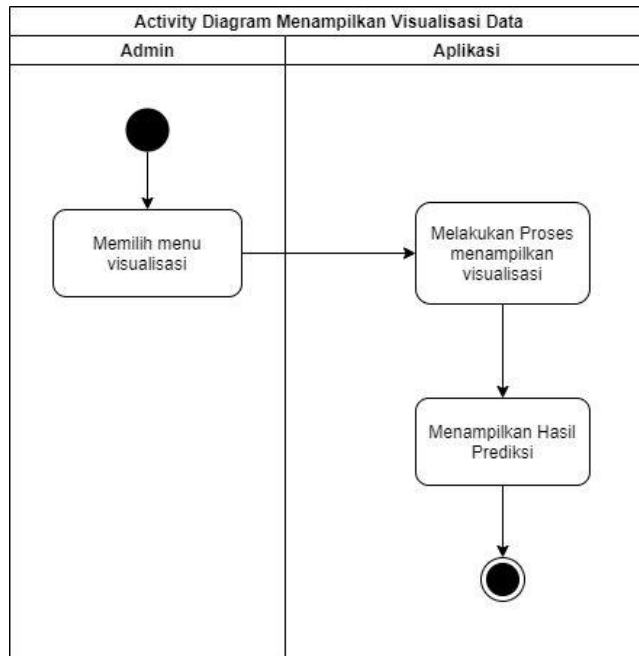
Gambar 4.15 Activity Prediksi Harga Rumah

Keterangan:

1. Admin menginputkan nilai dari variabel dalam bentuk angka pada form prediksi.
2. Aplikasi melakukan proses perhitungan prediksi rumah.
3. Aplikasi menampilkan hasil prediksi

e. **Activity Diagram Melakukan Visualisasi**

Berikut ini merupakan *Activity diagram* Visualisasi Data yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini.



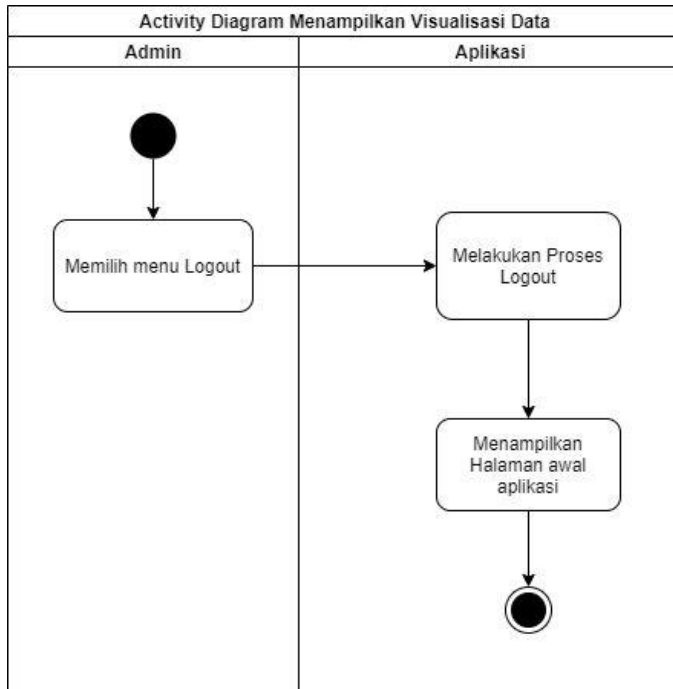
Gambar 4.16 Activity Diagram Visualisasi

Keterangan :

1. Admin memilih menu visualisasi.
2. Aplikasi melakukan proses menampilkan visualisasi.
3. Aplikasi menampilkan halamann visualisasi.

f. Activity Diagram Logout

Ini adalah diagram aktivitas *logout* admin yang menunjukkan proses yang terjadi dalam sistem.



Gambar 4.17 Activity Diagram Logout

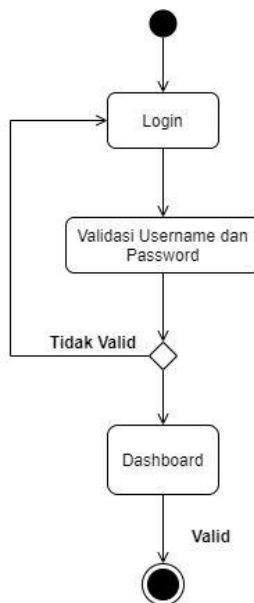
Keterangan :

1. Admin memilih menu *Logout*.
2. Aplikasi melakukan proses *Logout* aplikasi.
3. Aplikasi menampilkan halaman awal aplikasi.

4.7.4 Statechart Diagram

Diagram statechart dalam UML (Unified Modeling Language) digunakan untuk merepresentasikan transisi keadaan atau perubahan status objek dalam suatu sistem sebagai respons terhadap stimulus yang diterimanya. Diagram ini sering digunakan untuk menggambarkan perilaku dari suatu kelas tertentu dalam sistem dan satu kelas dapat memiliki lebih dari satu diagram statechart untuk memodelkan setiap aspek perilakunya.

a. *Statechart Diagram Login Admin*

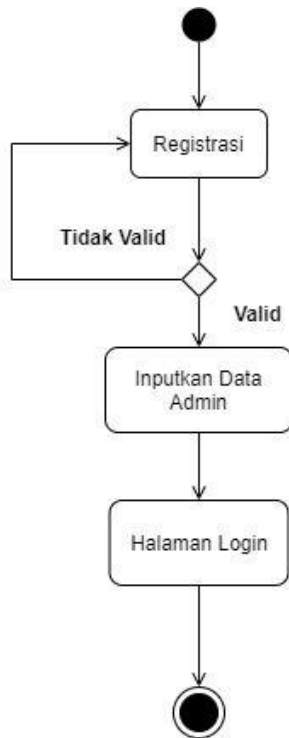


Gambar 4.18 *Statechart Diagram Login Admin*

Keterangan :

1. Admin melakukan login.
2. Aplikasi melakukan verifikasi username dan password yang dimasukkan.
3. Jika informasi tersebut valid, maka aplikasi akan menampilkan halaman dashboard.
4. Namun, jika informasi tersebut tidak valid, maka aplikasi akan mengarahkan kembali pengguna ke halaman login.

b. Statechart Diagram Register

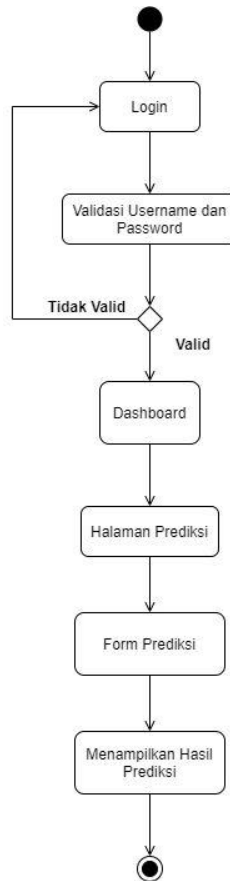


Gambar 4.19 Statechart Register

Keterangan :

1. Admin menginput data admin.
2. Jika valid maka data admin akan terinput ke *database*.
3. Jika tidak valid maka akan dikembalikan ke halaman registrasi admin.

c. **Statechart Diagram Melakukan Prediksi**

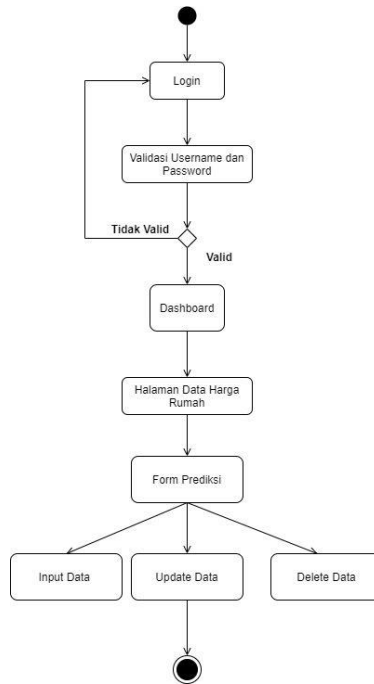


Gambar 4.20 Statechart Prediksi harga Rumah

Keterangan:

1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.
5. Aplikasi menampilkan form prediksi pada halaman prediksi
6. Aplikasi menampilkan hasil prediksi harga rumah.

d. *Statechart* Pengolahan Data

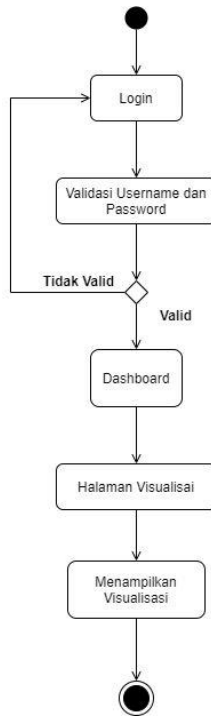


Gambar 4.21 *Statechart* pengolahan data

Keterangan :

1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.
5. Aplikasi akan menampilkan halaman data rumah
6. Pada halaman data karyawan, admin dapat memilih pengolahan antara input data rumah, *update* data rumah, ataupun *delete* data rumah.

e. **Statechart Diagram Visualisasi**



Gambar 4.22 Statechart Visualisasi Diagram

Keterangan :

1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.
5. Aplikasi menampilkan halaman visualisasi.

4.7.5 ERD (Entity Relationship Diagram)

ERD (Entity Relationship Diagram) adalah suatu metode visualisasi yang digunakan untuk menggambarkan hubungan antara entitas data (Entity) dan relasinya (Relationship) dalam suatu model. Tujuan utama dari ERD adalah untuk memperlihatkan bagaimana entitas-entitas tersebut saling berhubungan.

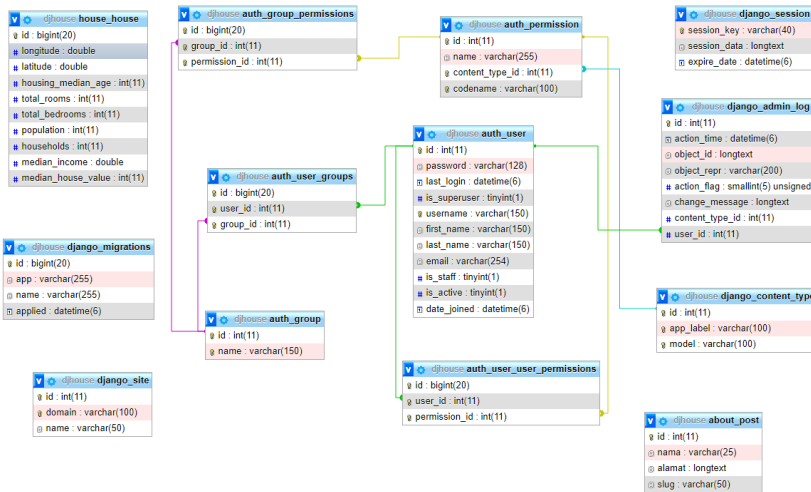
Berikut ini adalah simbol-simbol yang digunakan dalam ERD (Entity Relationship Diagram):

1. Dalam konteks ini, entitas merujuk pada objek konkret atau abstrak yang memiliki atribut yang dapat dicatat sebagai data.
2. Atribut adalah sifat yang menjelaskan karakteristik dari entitas tertentu, baik itu semua atau sebagian.
3. Relasi adalah hubungan alami antara satu atau lebih entitas dalam sistem.
4. Link adalah garis yang menghubungkan atribut dengan kelompok entitas dan kelompok entitas dengan relasi, membentuk diagram yang merepresentasikan struktur data dan hubungannya.

Pada ERD ini terdapat 13 (tiga belas) entitas (tabel), entitas tersebut dapat dijelaskan sebagai berikut.

1. Entitas berelasi
 - a) *Auth_user_groups*
 - b) *Auth_group*
 - c) *Auth_user*
 - d) *Auth_user_user_permissions*
 - e) *Auth_group_permissions*
 - f) *Django_admin_log*
 - g) *Django_content_type*
 - h) *Auth_permission*
 - i) *Django_site*
2. Entitas tidak berelasi
 - a) *Django_session*
 - b) *Django_migrations*
 - c) *House_house*
 - d) *About_post*

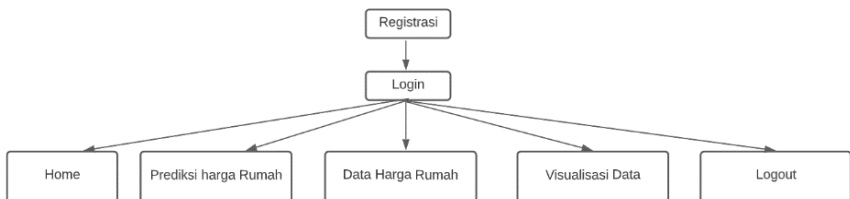
Berikut adalah Entity Relationship Diagram aplikasi prediksi harga rumah dalam bentuk gambar 4.23.



Gambar 4.23 Entity Relationship Diagram

4.7.6 Struktur Menu

Struktur menu merupakan aspek penting dalam desain aplikasi yang membantu pengguna untuk dengan mudah menjalankan aplikasi tanpa kesulitan dalam memilih menu yang diinginkan. Berikut adalah struktur menu dari aplikasi Prediksi Harga Rumah secara keseluruhan.



Gambar 4.24 Struktur Menu

Berdasarkan gambar 4.24, terdapat dijelaskan aplikasi memiliki halaman *dashboard* yang dapat mengakses menu lainnya yaitu sebagai berikut.

1. Halaman Registrasi merupakan halaman pendaftaran yang digunakan untuk memperoleh akses pada aplikasi.
2. Halaman Login digunakan untuk memvalidasi keabsahan pengguna.
3. Halaman Home merupakan halaman pertama yang ditampilkan setelah berhasil masuk ke aplikasi.
4. Halaman Prediksi Harga Rumah digunakan untuk melakukan prediksi harga rumah.
5. Halaman Data Rumah menampilkan data kepegawaian.
6. Halaman Visualisasi Data menampilkan visualisasi data.
7. Logout adalah menu untuk keluar dari aplikasi yang digunakan oleh pengguna.

BAB 5

IMPLEMENTASI APLIKASI

5.1 Implementasi Kode Aplikasi

Dalam bagian implementasi aplikasi, ditunjukkan bagaimana kode-kode beroperasi saat sistem berjalan. Juga ditunjukkan pola desain arsitektur sistem, termasuk Views, Template, dan URL.

5.1.1 Implementasi Package Yang Digunakan

```
from xml.etree.ElementInclude import include
from django.contrib import admin
from django.urls import path, include, re_path
from django.conf import settings
from django.shortcuts import render, HttpResponseRedirect,
redirect
from django.contrib.auth.models import User
from django.contrib.auth import authenticate, login,
logout
from django.contrib.auth.decorators import
login_required
from django.views import View
from django.http.response import JsonResponse
from django.utils.decorators import method_decorator
from django.views.decorators.csrf import csrf_exempt
from house import views
from django.conf.urls.static import static
```

Kode yang ditunjukkan di atas adalah bagian dari package yang digunakan dalam desain aplikasi. Package tersebut dipilih sesuai dengan kebutuhan dalam rancang bangun aplikasi.

5.1.2 Implementasi Proses *Login*

a) Views

```
def LoginPage(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        pass1 = request.POST.get('pass')
        User = authenticate(request, username=username, password=pass1)
        if User is not None:
            login(request, User)
            return redirect('home')
        else:
            return HttpResponse("Username or Password is incorrect!!!")

    return render(request, 'login.html')
```

b) Template

```
<body>
  <div class="background">
    <div class="shape"></div>
    <div class="shape"></div>
  </div>
  <form method="post">
    <h3>Login Here</h3>
    <!-- <center>
      <a href="{% provider_login_url 'google' %}?next=/" >Login With Google</a>
    </center> -->
    {% csrf_token %}
    <label for="username">Username</label>
    <input type="text" placeholder="Enter Username" id="username" name="username">

    <label for="password">Password</label>
    <input type="password" placeholder="Password" id="password" name="pass">

    <button type="submit">Log In</button>
    {% comment %} <button class="btn btn-outline-info">
      <a href="{% provider_login_url 'google' %}?next=/"
        <i class="fab fa-google"></i>
        Login With Google
      </a>
    </button> {% endcomment %}
    <center>
      <a href="{% url 'signup' %}" >Create a account</a>
    </center>
  </form>
</body>
```

c) URL

```
path('login/', views.LoginPage, name='login'),
```

5.1.3 Implementasi Proses Register

a) Views

```
def SignupPage(request):
    if request.method == 'POST':
        uname = request.POST.get('username')
        email = request.POST.get('email')
        pass1 = request.POST.get('password1')
        pass2 = request.POST.get('password2')

        if pass1 != pass2:
            return HttpResponse("You password and confirm password are not Same!! ")

        if User is not None:
            # return HttpResponse("Harap isi dengan benar !!")
            # else:
            my_user = User.objects.create_user(uname, email, pass1)
            my_user.save()
            return redirect('login')

    return render(request, 'signup.html')
```

b) Template

```
<form action="" method="post">
    {% csrf_token %}
    <h3>Signup Here</h3>

    <label for="username">Username</label>
    <input type="text" placeholder="Username" name="username" id="username" />

    <label for="email">Email</label>
    <input
        type="email"
        placeholder="Email or Phone"
        name="email"
        id="email"
    />

    <label for="password1">Password</label>
    <input
        type="password"
        placeholder="Password"
        id="password1"
        name="password1"
    />

    <label for="password2">Confirm Password</label>
    <input
        type="password"
        placeholder="Confirm Password"
        id="password2"
        name="password2"
    />

    <button type="submit">Signup</button>
    <center>
        <a href="{% url 'login' %}">i have already account ! </a>
    </center>
</form>
</body>
```

c) URL

```
path('', views.SignupPage, name='signup'),
```

5.1.4 Implementasi Proses Prediksi Harga Rumah

a) Views

```
def PredictPage(request):
    return render(request, 'predict.html', {'navbar': 'predict'})
```

b) Template

```
<head>
<title>House Price Prediction</title>

<link
  rel="stylesheet"
  href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
  integrity="sha384-Gn5384xqQ1ao/WXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGgFAH/dA1S6JXm"
  crossorigin="anonymous"
/>
<link rel="stylesheet" href="style.css" />
</head>
<body>
{% include "navbar.html" %}
<h1 class="text-dark">Input Nilai Prediksi</h1>
<br />
{% block content %}
<form method="POST" action="result">
  {% csrf_token %}
  <p>Masukkan Longitude dalam angka:</p>
  <input
    type="float"
    name="longitude"
    id="longitude"
    class="form-control"
    placeholder="example:-122.00"
    required
  />
  <br />
  <p>Masukkan latitude dalam angka:</p>
  <input
    type="float"
    name="latitude"
    id="latitude"
    class="form-control"
    placeholder="example:37.00"
    required
  />
  <br />
  <p>Masukkan housing median age dalam angka:</p>
  <input
    type="float"
    name="housing_median_age"
    id="housing_median_age"
    class="form-control"
    placeholder="example:53"
    required
  />
  <br />
  <p>Masukkan total rooms dalam angka:</p>
  <input
```

c) URL

```
path('predict/', views.PredictPage, name='predict'),
```

5.1.5 Implementasi Proses CRUD Data Rumah

5.1.5.a Halaman Tambah Data

a) Views

```
def rmh(request):
    if request.method == "POST":
        form = HouseForm(request.POST)
        if form.is_valid():
            try:
                form.save()
                return redirect('/view')
            except:
                pass
    else:
        form = HouseForm()
    return render(request, 'haltambah.html', {'form': form})
```

b) Template

```
<h3>Input Data Baru</h3>
</div>
<br></br>
</div>
<!-- <div class="form-group row">
    <label class="col-sm-2 col-form-label">House Id:</label>
    <div class="col-sm-4">{{ form.hId }}</div>
</div> -->
<div class="form-group row">
    <label class="col-sm-5 col-form-label">Garis Bujur :</label>
    <div class="col-sm-4">{{ form.longitude }}</div>
</div>
<div class="form-group row">
    <label class="col-sm-5 col-form-label">Garis mendatar :</label>
    <div class="col-sm-4">{{ form.latitude }}</div>
</div>
<div class="form-group row">
    <label class="col-sm-5 col-form-label">
        >Usia rata-rata perumahan :</label>
    <div class="col-sm-4">{{ form.housing_median_age }}</div>
</div>
<div class="form-group row">
    <label class="col-sm-5 col-form-label">Jumlah ruangan :</label>
    <div class="col-sm-4">{{ form.total_rooms }}</div>
</div>
<div class="form-group row">
    <label class="col-sm-5 col-form-label">Jumlah kamar :</label>
    <div class="col-sm-4">{{ form.total_bedrooms }}</div>
</div>
<div class="form-group row">
    <label class="col-sm-5 col-form-label">Populasi :</label>
```

c) URL

```
path('base/', views.base),
```


5.1.5.b Halaman Menampilkan Data

a) Views

```
def view(request):
    house = House.objects.all()
    return render(request, "view.html", {'house': house})
```

b) Template

```
<body>
{% include "navbar.html" %}
<h3 style="text-align: center; padding-top: 70px">
    Daftar Data Rumah Baru
</h3>
<div style="overflow-x: auto; margin-left: 50px; margin-right: 50px">
    <table class="table table-striped table-bordered table-sm">
        <thead class="thead-dark">
            <tr>
                <th>Garis Bujur</th>
                <th>Garis Mendatar</th>
                <th>Usia rata-rata perumahan</th>
                <th>Jumlah Ruangan</th>
                <th>Jumlah Kamar</th>
                <th>Populasi</th>
                <th>Rumah tangga</th>
                <th>Pendapatan Rata-rata</th>
                <th>Harga Rumah</th>
                <th>Action</th>
            </tr>
        </thead>
        {% for house in house %}
        <tr>
            <td>{{ house.longitude }}</td>
            <td>{{ house.latitude }}</td>
            <td>{{ house.housing_median_age }}</td>
            <td>{{ house.total_rooms }}</td>
            <td>{{ house.total_bedrooms }}</td>
            <td>{{ house.population }}</td>
            <td>{{ house.households }}</td>
            <td>{{ house.median_income }}</td>
            <td>{{ house.median_house_value }}</td>
            <td>
                <a href="/edit/{{ house.id }}"
                ><span class="btn btn-success">Edit</span></a>
                <a href="/delete/{{ house.id }}" class="btn btn-danger">Delete</a>
            </td>
        </tr>
        {% endfor %}
    </table>
</div>
</body>
```

c) URL

```
path('view/', views.view, name='view'),
```

5.1.5.c Halaman Edit Data

a) Views

```
def edit(request, id):  
    house = House.objects.get(id=id)  
    return render(request, 'edit.html', {'house': house})
```

b) Template

```
<body>  
{% include "navbar.html" %}  
<form method="POST" class="post-form" action="/update/{{ house.id }}">  
  <input type="hidden" name="id" id="id" required maxlength="20" value="{{ house.id }}">  
  {% csrf_token %}  
  <div class="container" style="padding-top: 70px; padding-left: 80px;">  
    <div class="form-group row">  
      <label class="col-sm-2 col-form-label"></label>  
      <div class="col-sm-8">  
        <h3>Edit Data Rumah</h3>  
      </div>  
    </div>  
    <div class="form-group row">  
      <label class="col-sm-5 col-form-label">Garis Bujur:</label>  
      <div class="col-sm-4"><input type="text" name="longitude" value="{{ house.longitude }}"></div>  
    </div>  
    <div class="form-group row">  
      <label class="col-sm-5 col-form-label">Garis mendatar:</label>  
      <div class="col-sm-4"><input type="text" name="latitude" value="{{ house.latitude }}"></div>  
    </div>  
    <div class="form-group row">  
      <label class="col-sm-5 col-form-label">Usia rata-rata perumahan:</label>  
      <div class="col-sm-4"><input type="text" name="housing_median_age" value="{{ house.housing_median_age }}"></div>  
    </div>  
    <div class="form-group row">  
      <label class="col-sm-5 col-form-label">Jumlah ruangan:</label>  
      <div class="col-sm-4"><input type="text" name="total_rooms" value="{{ house.total_rooms }}"></div>  
    </div>  
    <div class="form-group row">  
      <label class="col-sm-5 col-form-label">Jumlah kamar:</label>  
      <div class="col-sm-4"><input type="text" name="total_bedrooms" value="{{ house.total_bedrooms }}"></div>  
    </div>  
    <div class="form-group row">  
      <label class="col-sm-5 col-form-label">Populasi:</label>  
      <div class="col-sm-4"><input type="text" name="population" value="{{ house.population }}"></div>  
    </div>  
    <div class="form-group row">  
      <label class="col-sm-5 col-form-label">Rumah tangga:</label>  
      <div class="col-sm-4"><input type="text" name="households" value="{{ house.households }}"></div>  
    </div>  
    <div class="form-group row">  
      <label class="col-sm-5 col-form-label">Pendapatan Rata-rata:</label>  
      <div class="col-sm-4"><input type="text" name="median_income" value="{{ house.median_income }}"></div>  
    </div>  
</div>
```

c) URL

```
path('edit/<int:id>', views.edit),
```

5.1.5.d Halaman Menghapus Data

a) Views

```
def delete(request, id):
    house = House.objects.get(id=id)
    house.delete()
    return redirect("/view")
```

b) URL

```
path('delete/<int:id>', views.delete),
```

5.1.6 Implementasi Proses Visualisasi

a) Views

```
def VisualizationPage(request):
    return render(request, 'visualization.html', {'navbar': 'visualization'})
```

b) Template

```
{% extends 'base.html' %}

<!-- Page content -->
{% block content %}
<center>
<br>
<br>
<h1 class="container text-info" id="visualization"></h1>

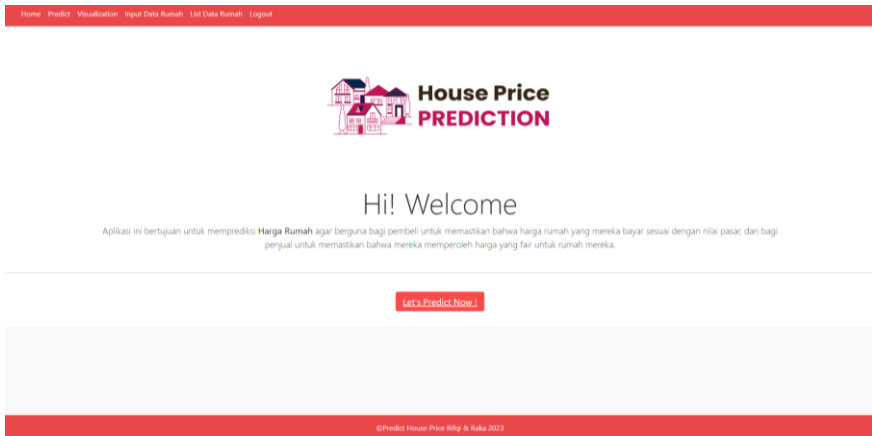
<div
class="tableauPlaceholder"
id="viz1675236215102"
style="position: relative"
>
<noscript>
<a href="#"
></a></noscript>
<object class="tableauViz" style="display: none">
<param name="host_url" value="https://public.tableau.com/" />
<param name="embed_code_version" value="3" />
<param name="site_root" value="" />
<param name="name" value="proyek3/Dashboard1" />
<param name="tabs" value="no" />
<param name="toolbar" value="yes" />
<param
name="static_image"
value="https://public.tableau.com/static/images/pr/projek3/Dashboard1&1.png"
/>
<param name="animate_transition" value="yes" />
<param name="display_static_image" value="yes" />
<param name="display_spinner" value="yes" />
<param name="display_overlay" value="yes" />
<param name="display_count" value="yes" />
<param name="language" value="en-US" />
<param name="filter" value="publish=yes" />
</object>
</div>
<script type="text/javascript">
var divElement = document.getElementById("viz1675236215102");
var vizElement = divElement.getElementsByTagName("object")[0];
if (divElement.offsetWidth > 800) {
vizElement.style.minWidth = "420px";
vizElement.style.maxWidth = "650px";
vizElement.style.width = "100%";
vizElement.style.minHeight = "587px";
```

c) URL

```
path('visualization/', views.VisualizationPage, name='visualization'),
```

5.2 Implementasi Halaman Antar Muka

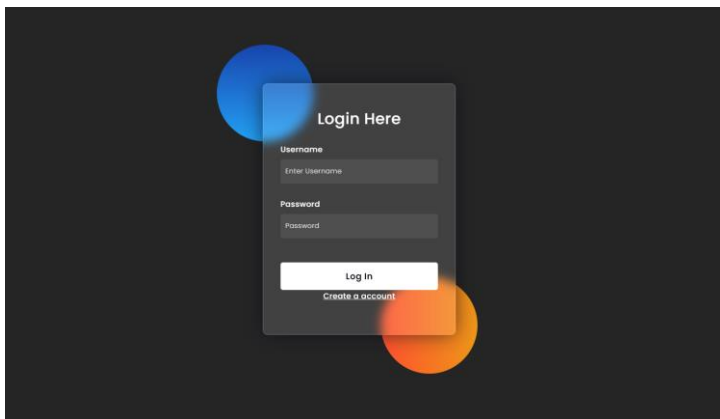
5.2.1 Implementasi Halaman Depan



Gambar 5.1 Tampilan UI Halaman depan

Gambar diatas merupakan implementasi dari halaman depan sistem. Pada halaman ini, *user* dapat menekan button “Login” pada navbar untuk masuk ke dalam sistem.

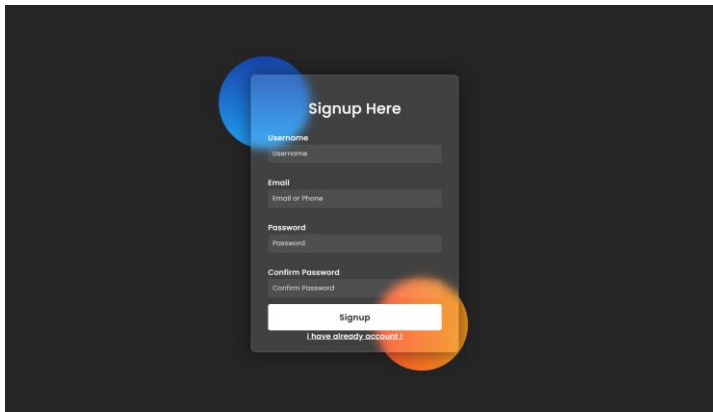
5.2.2 Implementasi Halaman Login



Gambar 5.2 Tampilan UI Depan

Implementasi dari halaman login terlihat pada gambar di atas. Sebelum dapat mengakses halaman dashboard, pengguna harus memasukkan username dan password yang tepat. Jika data yang dimasukkan sesuai, pengguna akan diarahkan ke halaman dashboard. Jika tidak, pengguna akan diminta untuk memasukkan username dan password yang sesuai kembali.

5.2.3 Implementasi Halaman Register



Gambar 5.3 Tampilan UI Register

Pada gambar di atas terlihat implementasi dari halaman pendaftaran akun. Jika pengguna belum memiliki akun untuk masuk ke sistem, maka dapat mendaftar dengan cara memasukkan username dan password, kemudian menekan tombol register.

5.2.4 Implementasi Halaman Prediksi Rumah

Home Predict Visualization Input Data Rumah List Data Rumah Logout

Input Nilai Prediksi

Masukkan Longitude dalam angka:
min: -124.3 set data max: -114.3

Masukkan latitude dalam angka:
min: 32.5 set data max: 42.5

Masukkan housing median age dalam angka:
min: 1.0 Set data max: 52.0

Masukkan total rooms dalam angka:
min: 2.0 Set data max: 37937.0

Masukkan total bedrooms dalam angka:
min: 1.0 Set data max: 6445.0

Masukkan population dalam angka:
min: 3.0 Set data max: 35682.0

Masukkan households dalam angka:
min: 1.0 Set data max: 6082.0

Masukkan Median income dalam angka:
min: 0.5 Set data max: 15.0

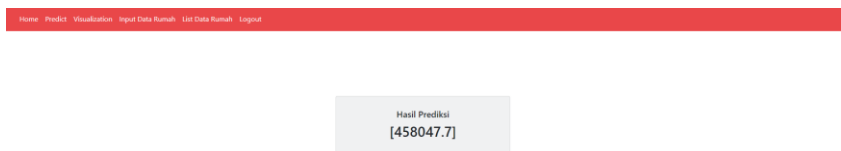
Gambar 5.4 Halaman UI Input nilai prediksi

Gambar diatas merupakan implementasi dari halaman prediksi dari sistem yang dirancang. Pada halaman ini, terdapat form untuk melakukan prediksi rumah. *User* dapat menginputkan angka ke dalam form untuk mengetahui hasil prediksi rumah dan diterapkan *text placeholder* sebagai acuan pada *field*. *Field* tersebut diantaranya adalah sebagai berikut :

- 1) *longitude*
- 2) *latitude*
- 3) *housing_median_age*
- 4) *total_rooms*
- 5) *total_bedrooms*
- 6) *population*
- 7) *households*
- 8) *median_income*
- 9) *ocean_proximity*

Kemudian *user* dapat menekan button “Lakukan Prediksi” untuk melihat hasil prediksi berdasarkan data yang diinputkan.

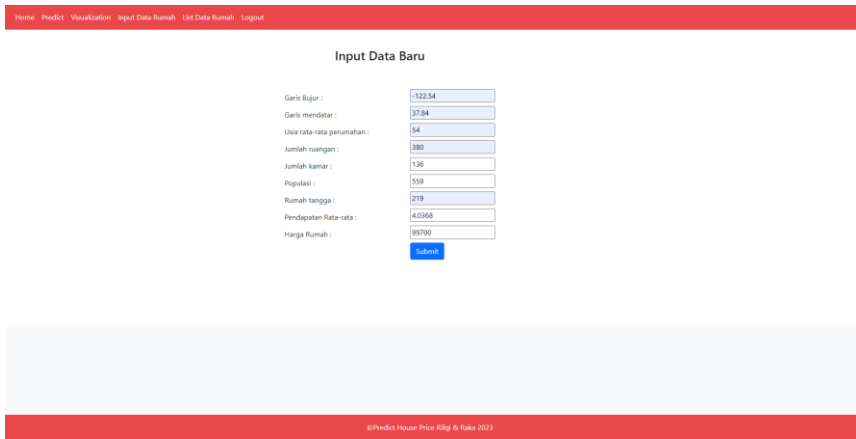
5.2.5 Implementasi Halaman Hasil Prediksi Rumah



Gambar 5.5 Halaman Hasil Prediksi Rumah

Gambar diatas merupakan implementasi dari halaman hasil prediksi dari aplikasi yang dirancang. Pada halaman ini, akan ditampilkan hasil prediksi rumah dari yang sebelumnya telah menginputkan nilai pada form prediksi.

5.2.6 Implementasi Halaman Menambahkan Data Rumah Baru



Gambar 5.6 Halaman Tambah data rumah baru

Gambar diatas merupakan implementasi dari halaman tambah data rumah dari aplikasi yang dirancang. Pada halaman ini, akan ditampilkan form tambah data untuk menambah data rumah. *User* dapat menginputkan data-data yang

ada pada atribut data rumah pada form. Lalu *user* dapat menekan button Simpan untuk menyimpan data yang diinputkan.

5.2.7 Implementasi Halaman Daftar data rumah baru

Home | Profile | Visualization | Input Data Rumah | **Uat Data Rumah** | Logout

Daftar Data Rumah Baru

Garis Bujur	Garis Mendatar	Usia rata-rata perumahan	Jumlah Ruangan	Jumlah Kamar	Populasi	Rumah tangga	Pendapatan Rata-rata	Harga Rumah	Action
-122.54	37.84	54	380	800	213	188	8.33	355200	<button>Edit</button> <button>Delete</button>
-122.0	37.86	54	1120	230	1100	2000	7.25	213100	<button>Edit</button> <button>Delete</button>
-122.25	37.85	52	1274	235	558	219	5.6431	143200	<button>Edit</button> <button>Delete</button>
-122.48	37.84	53	320	560	589	220	6.34	255200	<button>Edit</button> <button>Delete</button>
-122.54	37.84	54	380	136	559	219	4.0368	99700	<button>Edit</button> <button>Delete</button>

Gambar 5.7 Halaman Daftar data rumah baru

Gambar diatas merupakan implementasi dari halaman data rumah baru dari aplikasi yang dirancang. Pada halaman ini, akan ditampilkan data dalam bentuk tabel yang terdiri dari atribut yang di inputkan, Adapun tombol edit untuk mengubah inputan pada data rumah dan *delete* untuk menghapus data rumah baru.

5.2.8 Implementasi Halaman Mengedit Data Rumah Baru

Home | Profile | Visualization | Input Data Rumah | **Uat Data Rumah** | Logout

Uat Data Rumah

Garis Bujur:

-122.54

Garis mendatar:

37.84

Usia rata-rata perumahan:

54

Jumlah ruangan:

380

Jumlah kamar:

136

Populasi:

569

Rumah tangga:

217

Pendapatan Rata-rata:

4.238

Harga Rumah:

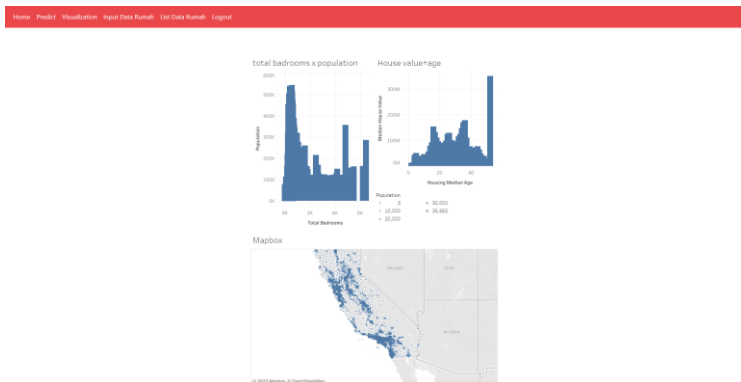
96500

Submit

Gambar 5.8 Halaman Edit Data rumah baru

Gambar diatas merupakan implementasi dari halaman edit data rumah dari aplikasi yang dirancang. Pada halaman ini, akan ditampilkan form edit data untuk mengubah data rumah. *User* dapat mengubah nilai dari atribut yang dipilih pada data form. Lalu *user* dapat menekan button Simpan untuk menyimpan data yang telah diubah.

5.2.9 Implementasi Halaman Visualisasi



Gambar 5.9 Halaman Visualisasi dari Tableau

Gambar diatas merupakan implementasi dari halaman visualisasi dari aplikasi yang dirancang. Pada halaman ini, akan ditampilkan bentuk visualisasi diagram batang populasi, harga rumah, map box daerah pada dataset.

BAB 6

RANGKUMAN MATERI

6.1 RANGKUMAN MATERI

Berdasarkan studi kasus dan pembahasan yang telah dilakukan, dapat diambil beberapa kesimpulan materi pada buku ini sebagai berikut :

1. Variabel Median *House Value* memiliki koefisien determinasi sebesar 84% dan merupakan variabel yang optimal dalam merekomendasikan harga rumah.
2. Model *Random Forest Regressor* memiliki akurasi sebesar 84% , nilai RMSE sebesar 0.193 yang dapat dikatakan sangat baik, nilai MAE sebesar 0.133 dapat dikatakan sangat baik, dan nilai MAPE sebesar 0.011 dapat dikatakan sangat baik karena mendekati 0 untuk performa model dalam memprediksi nilai aktual.
3. Hasil pemodelan dapat diintegrasikan ke dalam aplikasi berbasis web menggunakan framework Django.

6.2 Saran

Adapun Saran yang dapat diberikan kepada pembaca yang akan mengembangkan studi kasus ini adalah :

1. Nilai akurasi model yang perlu dikaji lebih lanjut untuk peningkatan hasil.
2. Sumber data yang digunakan dapat ditingkatkan dengan menggunakan data real dari perumahan yang ada, bukan hanya data dari kaggle.

BAB 7

LATIHAN

7.1 LATIHAN

Berikut adalah beberapa soal latihan untuk memperdalam pemahaman siswa dan menguji kemampuan siswa dalam memahami materi yang disampaikan dalam buku tentang Analisis Data Perumahan Untuk Memprediksi Harga Rumah Menggunakan Machine Learning :

- 1) Apa yang dimaksud dengan machine learning dalam analisis data perumahan?
- 2) Bagaimana cara membuat pemodelan dalam memprediksi harga rumah menggunakan machine learning?
- 3) Apa yang dimaksud dengan Random Forest Regressor dan bagaimana cara menggunakannya?
- 4) Bagaimana menganalisis variabel-variabel yang memiliki korelasi signifikan untuk memprediksi harga rumah?
- 5) Apa saja indeks yang digunakan untuk mengevaluasi model prediksi harga rumah?
- 6) Bagaimana cara menentukan model prediksi harga rumah yang baik dan efektif?
- 7) Apa yang dimaksud dengan overfitting dan bagaimana cara mencegahnya dalam memprediksi harga rumah?
- 8) Bagaimana memperbaiki kinerja model prediksi harga rumah yang kurang baik?

Dengan menjawab soal-soal ini, diharapkan siswa dapat memperdalam pemahaman dan mempraktekan pengetahuan mereka mengenai analisis data perumahan untuk memprediksi harga rumah menggunakan machine learning.

DAFTAR PUSTAKA

- Adetunji, A. B., Akande, O. N., Ajala, F. A., Oyewo, O., Akande, Y. F., & Oluwadara, G. (2021). House Price Prediction using Random Forest Machine Learning Technique. *Procedia Computer Science*, 199, 806–813. <https://doi.org/10.1016/j.procs.2022.01.100>
- Ahrens, A., Hansen, C. B., & Schaffer, M. E. (2020). lassopack: Model selection and prediction with regularized regression in Stata. *Stata Journal*, 20(1), 176–235. <https://doi.org/10.1177/1536867X20909697>
- Alwated, B., & El-Amin, M. F. (2021). Enhanced oil recovery by nanoparticles flooding: From numerical modeling improvement to machine learning prediction. *Advances in Geo-Energy Research*, 5(3). <https://doi.org/10.46690/ager.2021.03.06>
- Alzubi, J., Nayyar, A., & Kumar, A. (2018). Machine Learning from Theory to Algorithms: An Overview. *Journal of Physics: Conference Series*, 1142(1). <https://doi.org/10.1088/1742-6596/1142/1/012012>
- Bhamani, S., Makhdoom, A. Z., Bharuchi, V., Ali, N., Kaleem, S., & Ahmed, D. (2020). Home Learning in Times of COVID: Experiences of Parents. *Journal of Education and Educational Development*, 7(1), 9. <https://doi.org/10.22555/joeeed.v7i1.3260>
- Botchkarev, A. (2018). *Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology*.
- Burhan, H., Haq, U., Ur, H., Kayani, R., Khalil Toor, S., Zafar, S., & Khalid, I. (2020). The Popular Tools Of Data Sciences: Benefits, Challenges and Applications. In *IJCSNS International Journal of Computer Science and Network Security* (Vol. 20, Issue 5). <https://www.researchgate.net/publication/349624351>
- Cabras, I., Sohns, F., Canduela, J., & Toms, S. (2021). Public houses and house prices in Great Britain: a panel analysis. *European Planning Studies*, 29(1), 163–180. <https://doi.org/10.1080/09654313.2020.1726294>
- Chazar, C., & Widhiaputra, B. E. (2020). *Machine Learning Diagnosis Kanker Payudara Menggunakan Algoritma Support Vector Machine*.

- Chen, W., Li, Y., Xue, W., Shahabi, H., Li, S., Hong, H., Wang, X., Bian, H., Zhang, S., Pradhan, B., & Ahmad, B. bin. (2020). Modeling flood susceptibility using data-driven approaches of naïve Bayes tree, alternating decision tree, and random forest methods. *Science of the Total Environment*, 701. <https://doi.org/10.1016/j.scitotenv.2019.134979>
- Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7, 1–24. <https://doi.org/10.7717/PEERJ-CS.623>
- Coniglio, M. E., del Cerro, L. F., & Peron, N. M. (2022). *Tableau systems for some Iqvlev-like (quantified) modal logics*. <http://arxiv.org/abs/2202.02136>
- Diadjeng S, W. (2021). Team Convergence in Prevention and Reduction of Stunting Rate in Malang District, East Java, Indonesia. *Bioscience Biotechnology Research Communications*, 14(5), 133–140. <https://doi.org/10.21786/bbrc/14.5/26>
- Dimitriadis, S. I., Liparas, D., & Tsolaki, M. N. (2018). Random forest feature selection, fusion and ensemble strategy. *Journal of Neuroscience Methods*, 302, 14–23. <https://doi.org/10.1016/j.jneumeth.2017.12.010>
- Duca, J. v., Muellbauer, J., & Murphy, A. (2021). What drives house price cycles? International experience and policy issues. *Journal of Economic Literature*, 59(3), 773–864. <https://doi.org/10.1257/JEL.20201325>
- Fachri, S., & Ramdan, P. J. (2022). Pemodelan Machine Learning : Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM Menggunakan Data Twitter. In *Jurnal Sistem Cerdas*. <https://t.co/IEnuGFFuuj>
- Grech, V. (2018). WASP (Write a Scientific Paper) using Excel –5: Quartiles and standard deviation. *Early Human Development*, 118, 56–60. <https://doi.org/10.1016/j.earlhumdev.2018.01.012>
- Gur, I., Nachum, O., Miao, Y., Safdari, M., Huang, A., Chowdhery, A., Narang, S., Fiedel, N., & Faust, A. (2022). *Understanding HTML with Large Language Models*. <http://arxiv.org/abs/2210.03945>
- Gyani, J., Ahmed, A., & Haq, M. A. (2022). MCDM and Various Prioritization Methods in AHP for CSS: A Comprehensive Review. In *IEEE Access* (Vol. 10,

pp. 33492–33511). Institute of Electrical and Electronics Engineers Inc.
<https://doi.org/10.1109/ACCESS.2022.3161742>

- Heidari, M., Zad, S., & Rafatirad, S. (2021, April 21). Ensemble of supervised and unsupervised learning models to predict a profitable business decision. *2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021* - *Proceedings*.
<https://doi.org/10.1109/IEMTRONICS52119.2021.9422649>
- Hong, J., Choi, H., & Kim, W. S. (2020). A house price valuation based on the random forest approach. *International Journal of Strategic Property Management*, 24(3), 140–152. <https://doi.org/10.3846/ijspm.2020.11544>
- Huber, S., Wiemer, H., Schneider, D., & Ihlenfeldt, S. (2019). DMME: Data mining methodology for engineering applications - A holistic extension to the CRISP-DM model. *Procedia CIRP*, 79, 403–408.
<https://doi.org/10.1016/j.procir.2019.02.106>
- Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688.
<https://doi.org/10.1016/j.ijforecast.2006.03.001>
- Kafil, M. (2019). Penerapan Metode K-Nearest neighbors untuk Prediksi penjualan berbasis web pada Boutiq Dealove Bondowoso. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 3, Issue 2).
- Kang, Y., Zhang, F., Peng, W., Gao, S., Rao, J., Duarte, F., & Ratti, C. (2021). Understanding house price appreciation using multi-source big geo-data and machine learning. *Land Use Policy*, 111.
<https://doi.org/10.1016/j.landusepol.2020.104919>
- Kotsifakos, D., Magetos, D., & Douligeris, C. (2018). Utilization of Web-Based Services and Applications for Educational Purposes in Vocational Education and Training (VET). *European Journal of Engineering Research and Science, CIE*, 38. <https://doi.org/10.24018/ejers.2018.0.cie.642>
- Lu, S., Li, Z., Qin, Z., Yang, X., & Goh, R. S. M. (2018). A hybrid regression technique for house prices prediction. *IEEE International Conference on Industrial Engineering and Engineering Management, 2017-December*, 319–323. <https://doi.org/10.1109/IEEM.2017.8289904>

- Maxence Valentin. (2022). *Essays on the Deductibility of Housing Expenses*.
- Meng, T., Jing, X., Yan, Z., & Pedrycz, W. (2020). A survey on machine learning for data fusion. *Information Fusion*, 57, 115–129. <https://doi.org/10.1016/j.inffus.2019.12.001>
- Mohammady, M., Pourghasemi, H. R., & Amiri, M. (2019). Land subsidence susceptibility assessment using random forest machine learning algorithm. *Environmental Earth Sciences*, 78(16). <https://doi.org/10.1007/s12665-019-8518-3>
- Moons, K. G. M., Wolff, R. F., Riley, R. D., Whiting, P. F., Westwood, M., Collins, G. S., Reitsma, J. B., Kleijnen, J., & Mallett, S. (2019). PROBAST: A tool to assess risk of bias and applicability of prediction model studies: Explanation and elaboration. In *Annals of Internal Medicine* (Vol. 170, Issue 1, pp. W1–W33). American College of Physicians. <https://doi.org/10.7326/M18-1377>
- Muhammad, L. J., Algehyne, E. A., Usman, S. S., Ahmad, A., Chakraborty, C., & Mohammed, I. A. (2021). Supervised Machine Learning Models for Prediction of COVID-19 Infection using Epidemiology Dataset. *SN Computer Science*, 2(1). <https://doi.org/10.1007/s42979-020-00394-7>
- Paschen, J., Kietzmann, J., & Kietzmann, T. C. (2019). Artificial intelligence (AI) and its implications for market knowledge in B2B marketing. *Journal of Business and Industrial Marketing*, 34(7), 1410–1419. <https://doi.org/10.1108/JBIM-10-2018-0295>
- Prasetyo, B., & Trisyanti, D. (2018). *Strategi Pembangunan Nasional Menghadapi Revolusi Industri 4.0*. <http://dx.doi.org/10.12962/j23546026.y2018i5.4417>
- R. Saravanan, & Pothula, S. (2018). *A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification*.
- Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. In *Information (Switzerland)* (Vol. 11, Issue 4). MDPI AG. <https://doi.org/10.3390/info11040193>

- Rashid, A., & Chaturvedi, A. (2019). Cloud Computing Characteristics and Services A Brief Review. *International Journal of Computer Sciences and Engineering*, 7(2), 421–426. <https://doi.org/10.26438/ijcse/v7i2.421426>
- Retnoningsih, E., & Pramudita, R. (2020). Mengenal Machine Learning Dengan Teknik Supervised dan Unsupervised Learning Menggunakan Python. *BINA INSANI ICT JOURNAL*, 7(2), 156–165. <https://www.python.org/>
- Rights, J. D., & Sterba, S. K. (2020). New Recommendations on the Use of R-Squared Differences in Multilevel Model Comparisons. *Multivariate Behavioral Research*, 55(4), 568–599. <https://doi.org/10.1080/00273171.2019.1660605>
- Sabita, H., Herwanto, R., Syafitri, Y., Dwi Prasetyo, B., & Komputer, F. I. (2022). PENGEMBANGAN APLIKASI AKREDITASI PROGRAM STUDI BERBASIS FRAMEWORK DJANGO. In *Jurnal Informatika* (Vol. 22, Issue 01). Bulan Juni.
- Sari, I. P., Jannah, A., Meuraxa, A. M., Syahfitri, A., & Omar, R. (2022). Perancangan Sistem Informasi Penginputan Database Mahasiswa Berbasis Web. *Hello World Jurnal Ilmu Komputer*, 1(2), 106–110. <https://doi.org/10.56211/helloworld.v1i2.57>
- Sen, P. C., Hajra, M., & Ghosh, M. (2020). Supervised Classification Algorithms in Machine Learning: A Survey and Review. *Advances in Intelligent Systems and Computing*, 937, 99–111. https://doi.org/10.1007/978-981-13-7403-6_11
- Shahid, N., Shah, M. A., Khan, A., Maple, C., & Jeon, G. (2021). Towards greener smart cities and road traffic forecasting using air pollution data. *Sustainable Cities and Society*, 72. <https://doi.org/10.1016/j.scs.2021.103062>
- Siarni-Namini, S., Tavakoli, N., & Siarni Namin, A. (2019). A Comparison of ARIMA and LSTM in Forecasting Time Series. *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, 1394–1401. <https://doi.org/10.1109/ICMLA.2018.00227>
- Suroso, F., Rahmah, G. M., & Ap, D. R. (n.d.). *PEMODELAN SISTEM PERAMALAN KEBUTUHAN SPARE PART MENGGUNAKAN UNIFIED MODELING LANGUAGE*.

- Tien Bui, D., Nhu, V. H., & Hoang, N. D. (2018). Prediction of soil compression coefficient for urban housing project using novel integration machine learning approach of swarm intelligence and Multi-layer Perceptron Neural Network. *Advanced Engineering Informatics*, 38, 593–604. <https://doi.org/10.1016/j.aei.2018.09.005>
- Vallejo, W., Díaz-Urbe, C., & Fajardo, C. (2022). Google Colab and Virtual Simulations: Practical e-Learning Tools to Support the Teaching of Thermodynamics and to Introduce Coding to Students. *ACS Omega*, 7(8), 7421–7429. <https://doi.org/10.1021/acsomega.2c00362>
- Varma, A., Sarma, A., Doshi, S., & Nair, R. (2018). House Price Prediction Using Machine Learning and Neural Networks. *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, 1936–1939. <https://doi.org/10.1109/ICICCT.2018.8473231>
- Zhang, W., Wu, C., Li, Y., Wang, L., & Samui, P. (2021). Assessment of pile drivability using random forest regression and multivariate adaptive regression splines. *Georisk*, 15(1), 27–40. <https://doi.org/10.1080/17499518.2019.1674340>
- Zolfaghari, M., & Golabi, M. R. (2021). Modeling and predicting the electricity production in hydropower using conjunction of wavelet transform, random forest models. *Renewable Energy*, 170, 1367–1381. <https://doi.org/10.1016/j.renene.2021.02.017>

GLOSARIUM

A

Artificial Intellegence : Cabang dari ilmu komputer yang berfokus pada pembuatan mesin dan perangkat lunak yang dapat melakukan tugas yang biasanya dilakukan oleh manusia, seperti memahami bahasa, mengambil keputusan, dan melakukan tugas-tugas yang melibatkan pembelajaran dan pemahaman. AI mencoba untuk menirukan kapasitas intelektual manusia, seperti pemecahan masalah, pemahaman, dan pembelajaran, dengan menggunakan algoritma dan teknik matematis. Ada beberapa jenis AI, seperti machine learning, deep learning, computer vision, dan natural language processing.

Activity : Tindakan atau serangkaian tindakan yang dilakukan untuk mencapai tujuan tertentu. Dalam hal teknologi informasi, istilah activity sering digunakan untuk menggambarkan serangkaian tindakan yang dilakukan oleh pengguna atau sistem untuk mencapai tujuan tertentu.

B

Business Understanding : Tahap awal dalam proses Data Mining, dimana tujuan dan masalah bisnis yang akan diselesaikan dengan menggunakan data didefinisikan dan diklarifikasi. Dalam tahap ini, analis data dan tim bisnis bekerja sama untuk memahami dan menentukan apa yang harus dicapai dari proyek data mining. Tahap ini sangat penting karena memastikan bahwa proyek data mining berfokus pada masalah bisnis yang benar dan memastikan bahwa hasil akhir proyek berguna bagi bisnis.

C

Classification : Salah satu tugas pembelajaran mesin yang membutuhkan sistem AI untuk memasukkan item baru ke dalam salah satu kelas yang telah ditentukan sebelumnya. Tujuannya adalah untuk memprediksi kelas item baru berdasarkan fitur atau atribut yang diketahui sebelumnya.

CRISP-DM : standar lintas industri untuk proses data mining yang dikembangkan di Eropa sejak tahun 1996. Proses ini terdiri dari 6 tahap, di mana 3 tahap awal saling eksklusif menurut pengalaman penulis. Enam tahap CRISP DM tersebut adalah Pemahaman bisnis, data, persiapan data, pemodelan, pengujian dan penyebaran

CSS (Cascading Style Sheet) : bahasa yang digunakan untuk merepresentasikan halaman web. Representasi yang dilakukan meliputi warna, layout, dan font. Dengan menggunakan CSS, seorang web developer dapat membuat halaman web yang dapat beradaptasi dengan berbagai ukuran layar. Penulisan skrip CSS biasanya dilakukan secara terpisah dari halaman HTML, meskipun skrip CSS juga dapat ditambahkan ke dalam halaman HTML

D

Data : Informasi atau fakta yang dikumpulkan, diolah, dan disimpan untuk tujuan analisis atau pembuatan keputusan. Data dapat berupa angka, teks, gambar, suara, atau tipe informasi lainnya. Dalam dunia teknologi informasi, data adalah aset penting karena memungkinkan untuk memahami dan membuat keputusan berdasarkan analisis data yang tepat. Data dapat dikumpulkan melalui berbagai sumber seperti formulir, survey, transaksi bisnis, atau pemantauan sistem. Data dapat dianalisis dengan menggunakan teknik statistik, data mining, atau analisis big data untuk memperoleh informasi dan wawasan yang berguna.

Diagram : Sebuah representasi visual dari informasi atau konsep. Digram membantu dalam memahami dan menyampaikan informasi dengan lebih efektif dan mudah daripada menggunakan teks saja. Ada banyak jenis diagram, seperti diagram alir, diagram organisasi, diagram flowchart, diagram Gantt, dan lainnya.

Django : framework web *Python* yang digunakan untuk membuat aplikasi web dengan efisien dan cepat. Django menyediakan struktur yang baik untuk mengelola aplikasi web besar dan kompleks dengan menyediakan fitur seperti ORM (Object-Relational Mapping), template engine, dan sistem routing (Sabita et al., 2022).

Data Understanding : tahap dalam proses Data Science dimana pemahaman yang mendalam tentang data yang akan dianalisis dilakukan. Tujuan dari tahap

ini adalah untuk menentukan apa yang akan dianalisis dan bagaimana data tersebut akan digunakan untuk memecahkan masalah atau menjawab pertanyaan yang diajukan.

Data Preparation : tahap dalam proses Data Science yang bertujuan untuk mempersiapkan data agar siap digunakan dalam analisis. Tahap ini meliputi tugas-tugas seperti pembersihan data, imputasi data yang hilang, merubah tipe data yang salah, dan menangani data yang tidak sesuai. Tahap ini juga melibatkan normalisasi data, memilih variabel yang sesuai, dan membuat transformasi yang diperlukan pada data.

Deployment : tahap dalam proses Data Science yang bertujuan untuk mengeluarkan hasil analisis data dan membuatnya tersedia bagi pengguna akhir. Tahap ini melibatkan tugas seperti pembuatan model atau algoritma yang akan digunakan untuk membuat ramalan atau mengambil keputusan, memvalidasi model dan memastikan bahwa model tersebut bekerja dengan baik, dan melakukan implementasi model ke dalam sistem produksi.

E

Evaluasi data : Proses penilaian dan pengukuran kualitas dan keakuratan data. Tujuannya adalah untuk memastikan bahwa data yang digunakan dalam suatu sistem atau proses memenuhi standar dan spesifikasi yang ditentukan. Evaluasi data dapat melibatkan berbagai metode, seperti statistik deskriptif, analisis kualitatif, dan pengujian hipotesis. Evaluasi data juga melibatkan pengujian dan validasi data untuk memastikan bahwa data yang digunakan benar dan dapat dipercaya. Proses ini penting karena data yang buruk atau tidak akurat dapat mengarah pada kesalahan dalam pengambilan keputusan dan solusi yang tidak efektif.

ERD (Entity Relationship Diagram) : Sebuah diagram yang digunakan untuk menggambarkan hubungan antar entitas dalam suatu sistem. ERD menggambarkan entitas, atribut-atribut entitas, dan hubungan antar entitas. ERD juga menggambarkan jenis hubungan antar entitas, seperti satu-ke-satu, satu-ke-banyak, atau banyak-ke-banyak. Ini berguna dalam proses perancangan basis data untuk memahami dan menentukan hubungan antar entitas dalam suatu sistem.

F

Forecast: Proyeksi pendapatan, beban, serta pemerolehan dan penyusutan/pembuangan aset perusahaan di masa depan.

Flowmap : sebuah diagram alir (flowchart) yang menggambarkan alur atau tahapan proses yang ada dalam suatu sistem atau aktivitas. Flowmap dapat membantu untuk memahami bagaimana suatu proses berjalan dan mengidentifikasi potensi masalah atau kesalahan dalam proses.

G

Google Colab : aplikasi yang dikembangkan oleh Google untuk menjalankan kode *Python* pada lingkungan yang disebut sebagai notebook. Ini sangat berguna untuk pembelajaran dan prototyping karena Anda dapat menjalankan kode *Python* secara interaktif dan melihat hasilnya secara *real-time*

H

HTML : *Hypertext Markup Language* merupakan bahasa markup yang digunakan untuk membuat halaman web. Dengan menggunakan HTML, kita dapat menambahkan berbagai elemen seperti teks, gambar, video, dan tautan ke halaman web yang kita buat. HTML menggunakan tag untuk menandai elemen-elemen yang digunakan dalam halaman web, seperti judul, paragraf, dan tabel.

I

IQR (*Interquartile Range*) : sebuah ukuran dispersi atau variasi data yang digunakan dalam statistik. IQR mengukur jarak antara nilai paling bawah (25th percentile) dan nilai paling tinggi (75th percentile) dalam suatu distribusi data. IQR memberikan informasi tentang bagaimana data tersebar dan menunjukkan seberapa jauh setengah dari data terletak dari median. IQR digunakan untuk mengidentifikasi outlier (nilai yang sangat berbeda dari data lain) dan membantu dalam interpretasi data.

J

K

L

Login : proses otentikasi pengguna di dalam sebuah sistem atau aplikasi dengan memasukkan informasi identifikasi pribadi seperti username dan password, atau metode otentikasi lainnya. Tujuan dari login adalah memberikan akses yang terbatas hanya kepada pengguna yang telah terautentikasi dan memiliki otoritas untuk mengakses data atau sumber daya yang dilindungi.

Logout : proses keluar dari suatu sistem atau aplikasi dengan cara menghentikan akses ke sumber daya yang dilindungi. Ketika pengguna logout, sistem akan menghapus informasi otentikasi pengguna dan tidak lagi memberikan akses ke sumber daya tersebut kecuali pengguna melakukan login kembali. Logout merupakan tindakan yang penting untuk menjaga keamanan informasi pribadi dan data pada saat tidak digunakan dan mencegah akses yang tidak sah pada akun pengguna.

M

Machine Learning : Salah satu aplikasi dari Artificial Intelligent (AI) yang fokus pada pengembangan sebuah sistem yang dapat belajar sendiri tanpa harus diprogram berulang kali. Dengan menggunakan *Machine learning*, komputer dapat meningkatkan performa dalam menyelesaikan tugas tertentu dengan cara menganalisis dan menyesuaikan diri terhadap input data yang diberikan.

MSE (Mean Squared Error) : Metrik yang digunakan untuk mengukur kualitas hasil dari sebuah model prediksi. MSE mengukur rata-rata kuadrat dari selisih antara nilai aktual dan nilai prediksi. Ini berarti bahwa MSE mengukur seberapa besar kesalahan yang terjadi pada setiap prediksi dan mengevaluasi secara keseluruhan performa dari model prediksi. Semakin kecil nilai MSE, maka semakin baik performa dari model tersebut dalam memprediksi nilai aktual.

MAE (*Mean Absolute Error*) : metrik yang digunakan untuk mengukur kualitas hasil dari sebuah model prediksi. MAE mengukur rata-rata dari selisih absolut antara nilai aktual dan nilai prediksi. Ini berarti bahwa MAE mengukur seberapa besar kesalahan yang terjadi pada setiap prediksi dan mengevaluasi secara keseluruhan performa dari model prediksi. Semakin kecil nilai MAE, maka semakin baik performa dari model tersebut dalam memprediksi nilai aktual.

MAPE (*Mean Absolute Percentage Error*) : metrik yang digunakan untuk mengukur kualitas hasil dari sebuah model prediksi. MAPE mengukur rata-rata dari selisih absolut dalam bentuk persentase antara nilai aktual dan nilai prediksi. Ini berarti bahwa MAPE mengukur seberapa besar kesalahan yang terjadi pada setiap prediksi dalam bentuk persentase dan mengevaluasi secara keseluruhan performa dari model prediksi.

N

Normalisasi : proses memperbaiki atau memodifikasi data dengan memastikan bahwa data tersebut memenuhi beberapa aturan tertentu. Normalisasi data digunakan untuk mengatasi masalah seperti data yang tidak teratur, data dengan skala yang berbeda, dan data dengan distribusi yang tidak seimbang

Numerik : istilah yang digunakan untuk menjelaskan sesuatu yang berhubungan dengan angka atau nomor. Dalam dunia komputer dan teknologi informasi, numerik sering digunakan untuk menjelaskan tipe data numerik seperti integer, float, dan double. Tipe data numerik digunakan untuk menyimpan dan memproses angka-angka yang dapat digunakan dalam berbagai aplikasi seperti perhitungan matematika, statistika, dan analisis data.

O

Outlier : memiliki nilai ekstrim dibandingkan dengan data rata-rata, dimana penyimpangan dapat muncul berupa angka yang terlalu tinggi atau terlalu rendah. Berikut adalah jenis-jenis *outlier*: *Global Outlier* adalah tipe data *outlier* global yang nilainya jauh melampaui keseluruhan tempat data berada *Outlier* kontekstual yang menyimpang dari data lain Penyimpangan kolektif dalam kumpulan data tertentu, beberapa poin signifikan dari kumpulan data lainnya.

P

Prediksi : proses membuat ramalan tentang suatu peristiwa atau kondisi di masa depan berdasarkan data yang ada. Dalam Data Science, prediksi dapat dilakukan dengan menggunakan berbagai metode dan teknik, seperti Regresi, Klasifikasi, dan Analisis Kluster. Proses prediksi melibatkan tugas seperti pemahaman tentang masalah yang akan diprediksi, pemilihan metode dan teknik yang sesuai, pembuatan model dan validasi model, dan implementasi model ke dalam sistem produksi.

Price : harga dari suatu produk atau jasa. Harga adalah jumlah uang yang harus dibayar oleh pembeli untuk memperoleh produk atau jasa tersebut.

Pemodelan : proses membuat suatu model matematis atau statistik untuk menjelaskan hubungan antara variabel-variabel dan memprediksi hasil. Dalam Data Science, pemodelan merupakan tahap penting dalam proses analisis data, yang memungkinkan data scientist untuk memahami dan membuat ramalan tentang data. Proses pemodelan melibatkan tugas seperti pemilihan metode dan teknik pemodelan yang sesuai, pembuatan model dan validasi model, dan interpretasi hasil model.

Python : bahasa pemrograman yang dikembangkan oleh Guido van Rossum pada tahun 1989. Python adalah bahasa pemrograman yang sangat populer karena sintaksnya yang mudah dipahami dan mudah digunakan, sangat cocok untuk pemula dan profesional (Burhan et al., 2020). Python digunakan dalam berbagai bidang seperti pengembangan aplikasi web, data science, machine learning, intelijen buatan, pengembangan desktop dan mobile, pembuatan script, dll.

Q

R

Random Forest : algoritma *Machine learning supervised* yang dikeluarkan oleh Breiman pada tahun 2001. Ia dapat digunakan untuk klasifikasi dan regresi. Metode ini membangun beberapa pohon keputusan dari dataset pelatihan yang

terdiri dari pasangan input nilai dan output, kemudian menggabungkan hasil dari masing-masing pohon keputusan untuk menghasilkan prediksi yang lebih akurat. (Mohammady et al., 2019)

Regression : metode analisis statistik yang digunakan untuk memahami hubungan antara variabel dependen (target variable) dan variabel independen (predictor variable) dalam suatu data. Tujuan dari regresi adalah untuk membuat model matematis yang menerangkan hubungan antara variabel dependen dan independen dan memprediksi nilai dari variabel dependen berdasarkan nilai dari variabel independen.

R-Squared : ukuran statistik yang menyatakan proporsi varian dari variabel dependen yang dinyatakan sebagai R^2 atau r^2 , diucapkan r squared adalah nilai yang menunjukkan bahwa semakin dekat nilai variabel independen R^2 adalah 0 atau 1 banding 1, semakin besar model menggunakan Nah, semakin mendekati 0, semakin buruk model yang digunakan (Rights & Sterba, 2020),

Register/Signup : proses untuk membuat akun baru di dalam sebuah sistem atau aplikasi. Dalam proses sign-up, pengguna diharuskan untuk memasukkan informasi pribadi seperti nama, alamat email, nomor telepon, dan memilih username dan password yang akan digunakan untuk masuk ke sistem atau aplikasi tersebut. Setelah proses sign-up selesai, pengguna akan mendapatkan akses ke sistem atau aplikasi dan dapat menggunakannya sesuai dengan fungsinya.

S

Standarisasi : proses mengubah skala suatu data menjadi skala yang standar. Dalam statistik, standarisasi sering digunakan untuk membandingkan data yang memiliki skala yang berbeda-beda dan membuat interpretasi lebih mudah.

Statechart : metode model-driven untuk menentukan perilaku suatu sistem atau aplikasi yang menggambarkan bagaimana suatu objek bereaksi terhadap suatu input dan berpindah dari satu keadaan ke keadaan lain. Statechart digunakan untuk menentukan perilaku suatu sistem yang dinamis, di mana perilaku dapat berubah tergantung pada keadaan dan input yang diterima.

Spearman : sebuah ukuran korelasi antar dua variabel dalam statistik. Spearman's rho digunakan untuk mengukur hubungan antara dua variabel yang

tidak harus linear dan dapat berupa hubungan curvilinear. Spearman's rho mengukur korelasi berdasarkan ranking relatif dari data dan bukan nilai absolut dari data. Nilai korelasi yang dapat berkisar antara -1 dan 1, dengan nilai 0 menunjukkan bahwa tidak ada korelasi antar variabel, nilai positif menunjukkan korelasi positif antar variabel, dan nilai negatif menunjukkan korelasi negatif antar variabel.

Studi Literatur : Kumpulan penelitian terdahulu (state of the art) yang membahas prediksi menggunakan Machine learning. Adapun proses mempelajari dan meninjau informasi yang tersedia dalam bentuk tulisan atau publikasi ilmiah yang berhubungan dengan topik atau masalah tertentu. Studi literatur melibatkan pencarian, seleksi, dan analisis sumber-sumber informasi seperti jurnal ilmiah, buku, laporan, dan dokumen lain yang terkait dengan topik yang diteliti.

T

Tableau : perangkat lunak yang digunakan untuk visualisasi data. *Tableau* memungkinkan pengguna untuk mengeksplorasi, analisis, dan menyajikan data dalam bentuk visual yang mudah dipahami. *Tableau* dapat digunakan untuk menganalisis data dari berbagai sumber, seperti *database*, *spreadsheet*, atau file CSV.

Train data : sekumpulan data yang digunakan untuk membuat model machine learning. Data training merupakan bagian dari data yang digunakan untuk melatih algoritma machine learning untuk memahami pola dan hubungan antar fitur dalam data. Model machine learning dibuat berdasarkan data training dan digunakan untuk membuat prediksi pada data baru yang belum dilihat sebelumnya (data testing).

Test data : sekumpulan data yang digunakan untuk menguji kemampuan model machine learning. Data testing berbeda dari data training, yaitu data yang tidak digunakan dalam proses pelatihan model dan baru dilihat oleh model setelah pelatihan selesai. Data testing digunakan untuk memvalidasi kemampuan model dalam membuat prediksi yang akurat pada data baru yang belum dilihat sebelumnya.

U

UML (*Unified Modeling Language*) : bahasa yang menyediakan standar untuk memvisualisasikan, mendefinisikan, membangun, dan mendokumentasikan sistem perangkat lunak berbasis OO (Object-Oriented).

Use case : representasi dari interaksi antara sistem atau aplikasi dan pengguna atau lingkungan yang berisi spesifikasi fungsionalitas yang diinginkan. Use case digunakan untuk menggambarkan bagaimana sistem akan digunakan untuk memenuhi kebutuhan pengguna atau untuk memecahkan masalah tertentu.

V

VScode (*Visual Studio Code*) : software editor teks yang dikembangkan oleh Microsoft, digunakan untuk menulis kode dalam berbagai bahasa pemrograman seperti JavaScript, Python, C++, dan lain-lain. Software ini dilengkapi dengan fitur-fitur yang memudahkan pengembangan aplikasi. Secara umum, VSCode digunakan untuk memudahkan pengembangan aplikasi dengan menyediakan fitur-fitur yang dibutuhkan oleh developer, seperti penyuguhan saran kode, debugging, integrasi dengan Git dll.

W

X

XAMPP : singkatan dari Apache, MariaDB, PHP, dan Perl. XAMPP menyediakan fasilitas yang diperlukan untuk menjalankan aplikasi web seperti Apache (web server), MariaDB (*database*), PHP (bahasa pemrograman), dan Perl (bahasa scripting)

Y

Z

KREDIT GAMBAR

- Gambar 3.1 : https://miro.medium.com/max/1400/0*unFpv2rld3EJNWt9.png
- Gambar 3.2 : https://miro.medium.com/v2/resize:fit:720/format:webp/1*ASYPFfDh7XnreU-ygqXonw.png
- Gambar 3.3 : https://miro.medium.com/v2/resize:fit:720/format:webp/1*UrRNiThLDzdpYqnI0RA1w.png
- Gambar 3.4 : <https://i0.wp.com/geekycodes.in/wp-content/uploads/2022/03/main-qimg-104b107465543d694b0822ef5843c65d.png?w=520&ssl=1>
- Gambar 3.6 : https://www.allangray.co.za/globalassets/information-technology/w3_html5-card.png
- Gambar 3.7 : https://miro.medium.com/v2/resize:fit:720/format:webp/1*f2i47G95nVCx71KzM1iXlg.png
- Gambar 3.8 : https://upload.wikimedia.org/wikipedia/commons/thumb/0/03/Xampp_logo.svg/2560px-Xampp_logo.svg.png
- Gambar 3.9 : https://www.python.org/static/community_logos/python-logo-master-v3-TM-flattened.png
- Gambar 3.10 : <https://algotrading101.com/learn/wp-content/uploads/2021/05/Google-Colab-Guide-1024x683.jpg>
- Gambar 3.11 : <https://seeklogo.com/images/V/visual-studio-code-logo-284BC24C39-seeklogo.com.png>
- Gambar 3.12 : <https://dev3lop.com/wp-content/uploads/2017/04/tableau-logo-tableau-software.jpg>
- Gambar 3.13 : <https://static.djangoproject.com/img/logos/django-logo-negative.png>

TENTANG PENULIS



Syafrial Fachri Pane, S.T.,M.T.I.,EBDP. Lahir di Medan 16 April 1988. Lulus D3 di Program Studi Teknik Informatika Politeknik Pos Indonesia tahun 2009, lulus S1 di Program Studi Teknik Informatika Universitas Pasundan tahun 2013, dan lulus S2 di Teknik Informatika Universitas Bina Nusantara. Saat ini adalah dosen tetap Program Studi Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Mengampu matakuliah Database I dan Database II. Aktif menulis di berbagai jurnal ilmiah. Bidang yang ditekuni adalah ilmu data. Saat ini sedang melanjutkan Pendidikan S3 di Telkom University.



Rakasona. Lahir di Kuningan, 21 Oktober 2002. Saat ini adalah mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Kini sedang menekuni bidang ilmu data.



Rifqi Fathurrohman. Lahir di Jakarta, 16 Maret 2001. Saat ini adalah mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Kini sedang menekuni bidang ilmu data.

ANALISIS DATA PERUMAHAN UNTUK MEMPREDIKSI HARGA RUMAH MENGUNAKAN MACHINE LEARNING

Sinopsi buku "Analisis Data Perumahan untuk Memprediksi Harga Rumah Menggunakan Machine Learning" menjelaskan tentang bagaimana teknologi machine learning dapat digunakan untuk memprediksi harga rumah. Buku ini membahas bagaimana data perumahan dapat dianalisis dan diterjemahkan menjadi informasi yang berguna bagi para pemain industri properti dalam membuat keputusan bisnis.

Buku ini juga membahas bagaimana machine learning dapat mengatasi masalah-masalah yang sering dijumpai dalam analisis data perumahan, seperti data yang tidak lengkap dan tidak akurat. Dengan memahami bagaimana teknologi machine learning dapat mengatasi masalah tersebut, para pembaca dapat membuat keputusan yang lebih baik dalam bidang properti.

Di dalam buku ini juga dibahas tentang bagaimana membangun model machine learning untuk memprediksi harga rumah dan bagaimana model tersebut dapat diuji dan divalidasi. Selain itu, buku ini juga membahas tentang bagaimana model machine learning dapat digunakan secara efektif dalam dunia nyata.

Secara keseluruhan, sinopsi buku ini menjelaskan bagaimana teknologi machine learning dapat digunakan untuk memprediksi harga rumah dan membantu para pemain industri properti dalam membuat keputusan yang lebih baik dengan faktor penentu dan model evaluasi yang ada.



Syafrial Fachri Pane, S.T., M.T.I., EBDP. Lahir di Medan 16 April 1988. Lulus D3 di Program Studi Teknik Informatika Politeknik Pos Indonesia tahun 2009, lulus S1 di Program Studi Teknik Informatika Universitas Pasundan tahun 2013, dan lulus S2 di Teknik Informatika Universitas Bina Nusantara. Saat ini adalah dosen tetap Program Studi Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Mengampu matakuliah Database I dan Database II. Aktif menulis di berbagai jurnal ilmiah. Bidang yang diteliti adalah ilmu data. Saat ini sedang melanjutkan Pendidikan S3 di Telkom University.



Rakasana. Lahir di Kuningan, 21 Oktober 2002. Saat ini adalah mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Kini sedang menekuni bidang ilmu data.



Rifqi Fathurrohman. Lahir di Jakarta, 16 Maret 2001. Saat ini adalah mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Kini sedang menekuni bidang ilmu data.