PRAKTIKUM DATA WAREHOUSING DAN DATA MINING

MODUL 10

REGRESI LINIER SEDERHANA

Disusun oleh:

Adinda Aulia Hapsari

L200220037

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

TAHUN 2024

Setelah kegiatan selesai, lembar kerja ini dicetak (di-print) dan dikumpulkan ke asisten. NIM : L200220037 Nama : Adinda Aulia Hapsari Nama Asisten : Diva Halimah Tanggal Praktikum : 6 Desember 2024	(Diisi oleh Asisten) Nilai Praktek : Tanda Tangan :

KEGIATAN PRAKTIKUM

Contoh Kasus:

Perusahaan asuransi kesehatan hanya dapat menghasilkan uang jika pemasukan yang didapatkan lebih banyak dibanding dengan biaya yang dikeluarkan untuk pembiayaan kesehatan pasien penerima manfaat. Namun, memprediksi biaya kesehatan sangatlah sulit karena setiap individu memiliki karakteristik yang berbeda-beda. Tujuan dari eksperimen ini adalah untuk memprediksi biaya asuransi secara akurat berdasarkan data orang, termasuk usia, indeks masa tubuh, merokok atau tidak, dll. Selain itu, kita juga akan menentukan variabel terpenting yang mempengaruhi biaya asuransi. Prediksi ini akan bermanfaat bagi perusahaan asuransi untuk menentukan besaran premi yang harus dibayarkan per bulannya. Dari deskripsi di atas dapat kita simpulkan bahwa permasalahan kasus ini dapat diselesaikan dengan teknik regresi.

Dataset yang akan kita gunakan pada eksperimen ini tersedia di tautan yang tertera di atas dengan nama file insurance.csv. Dataset ini memiliki 1338 baris dan 7 kolom. Berikut deskripsi dari masing-masing kolom:

1. Usia: usia penerima manfaat asuransi
2. Jenis kelamin: jenis kelamin penerima manfaat asuransi
3. BMI : Indeks Massa Tubuh, memberikan pemahaman tentang berat badan yang relatif tinggi atau rendah relatif terhadap tinggi badan, indeks objektif berat badan (kg/m^2) menggunakan rasio tinggi terhadap berat, idealnya 18,5 hingga 24,9
4. Anak: jumlah anak yang ditanggung oleh asuransi Kesehatan
5. Perokok: merokok atau tidak
6. Wilayah: daerah perumahan penerima di Amerika Serikat yang meliputi timur laut, tenggara, barat daya, barat laut.
7. Biaya: biaya pengobatan individu yang ditagih oleh asuransi Kesehatan

Karena kita akan memprediksi biaya asuransi, maka yang akan menjadi target fitur (fitur yang akan diprediksi) adalah kolom biaya.

Memprediksi biaya asuransi berdasarkan fitur-fitur yang tersedia.

1. Pada Langkah pertama kita perlu memasukan beberapa library dan kelas yang akan kita gunakan pada eksperimen ini. Berikut beberapa library yang akan kita gunakan.

```
[1]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
```

2. Selanjutnya, kita akan membaca dataset yang sudah kita unduh dengan menggunakan library pandas. Untuk membaca dataset dan melihat 5 data teratas

```
[2]: df = pd.read_csv('insurance.csv')
     df.head()
```

[2]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

3. Selanjutnya kita juga bisa melihat gambaran dan ringkasan secara umum dari dataset yang kita gunakan.

```
[3]: df.describe()
```

[3]:

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

4. Sebelum kita bisa menggunakan dataset, kita perlu memastikan dulu bahwa tidak ada data yang null (kosong). Sehingga dapat dipastikan bahwa data konsisten dan tidak menyebabkan error.

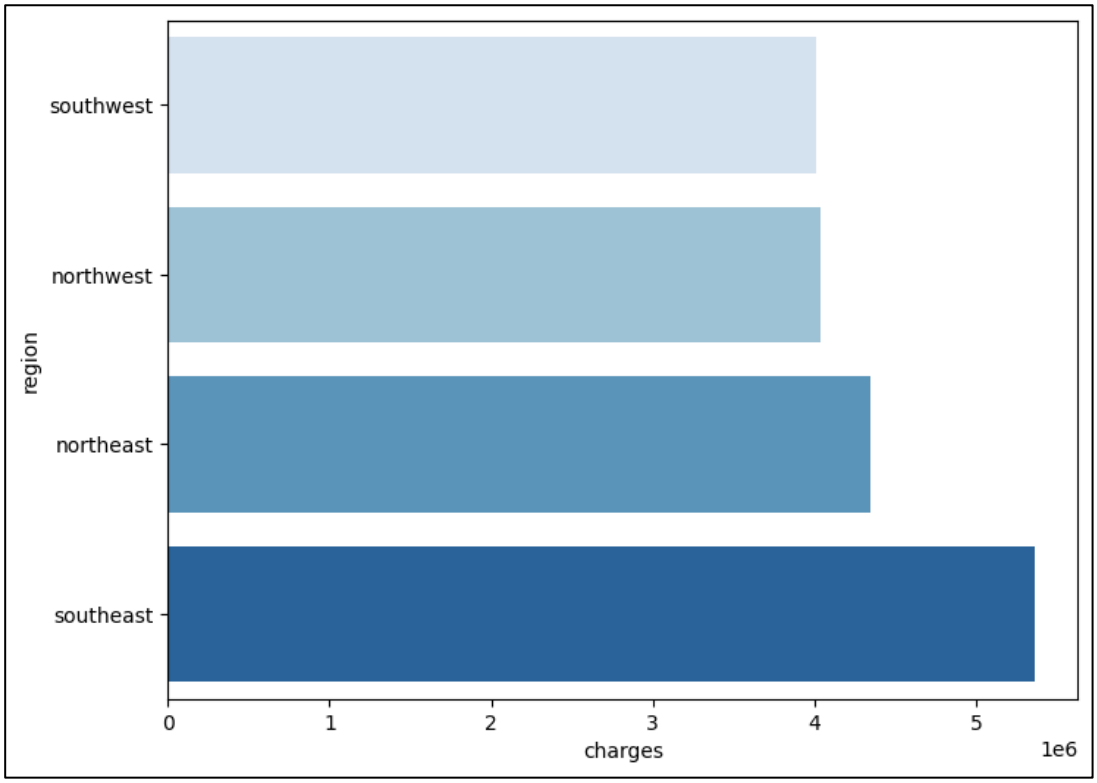
```
[4]: df.isnull().sum()

[4]: age         0
     sex         0
     bmi         0
     children    0
     smoker      0
     region      0
     charges     0
     dtype: int64
```

5. Selanjutnya kita bisa melakukan beberapa visualisasi terhadap data yang ada untuk meningkatkan pemahaman kita tentang dataset yang akan digunakan dalam eksperimen. Pertama kita bisa lihat distribusi biaya kesehatan untuk masing-masing daerah dengan kode dibawah ini.

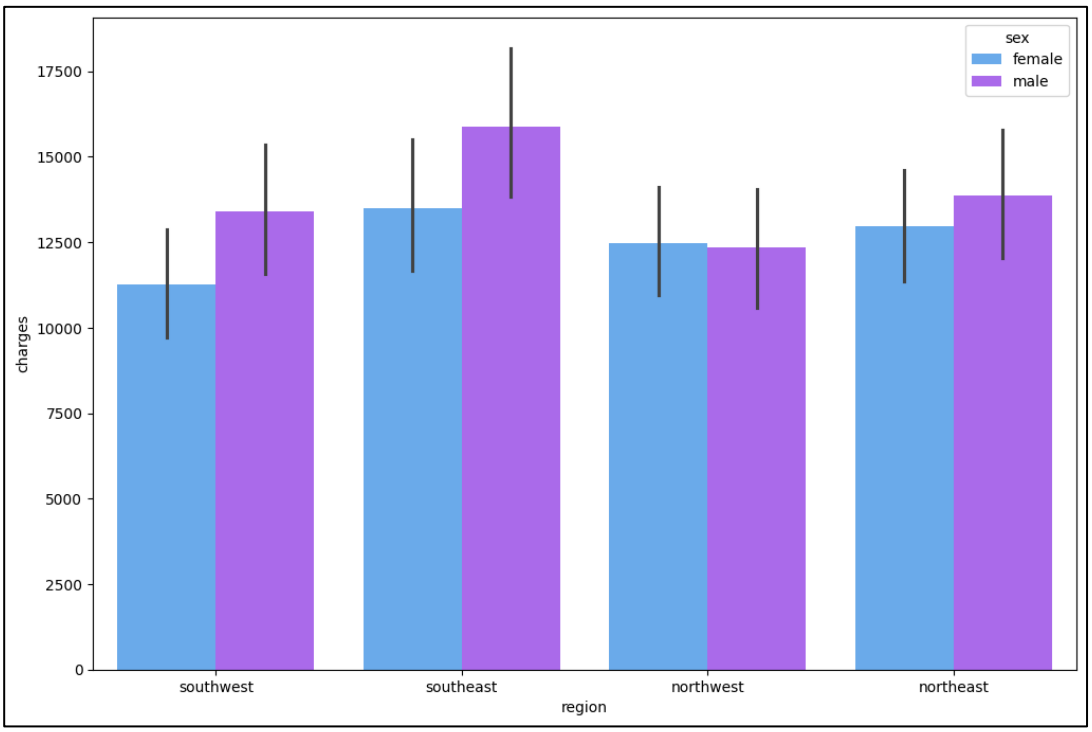
```
[9]: charges = df['charges'].groupby(df.region).sum().sort_values(ascending = True)
     f, ax = plt.subplots(1, 1, figsize=(8, 6))
     ax = sns.barplot(x = charges.head(), y = charges.head().index, palette='Blues')
```

Jadi secara keseluruhan biaya pengobatan tertinggi ada di wilayah southeast dan terendah di southwest.



6. Selanjutnya kita juga bisa melakukan analisa terhadap data-data yang lain meliputi jenis kelamin, merokok atau tidak dan jumlah anak (khusus data yang dalam format kategorikal). Dalam hal ini kita akan menganalisa data tersebut berdasarkan wilayah yang berbeda. Sekarang kita mulai dengan jenis kelamin terlebih dahulu.

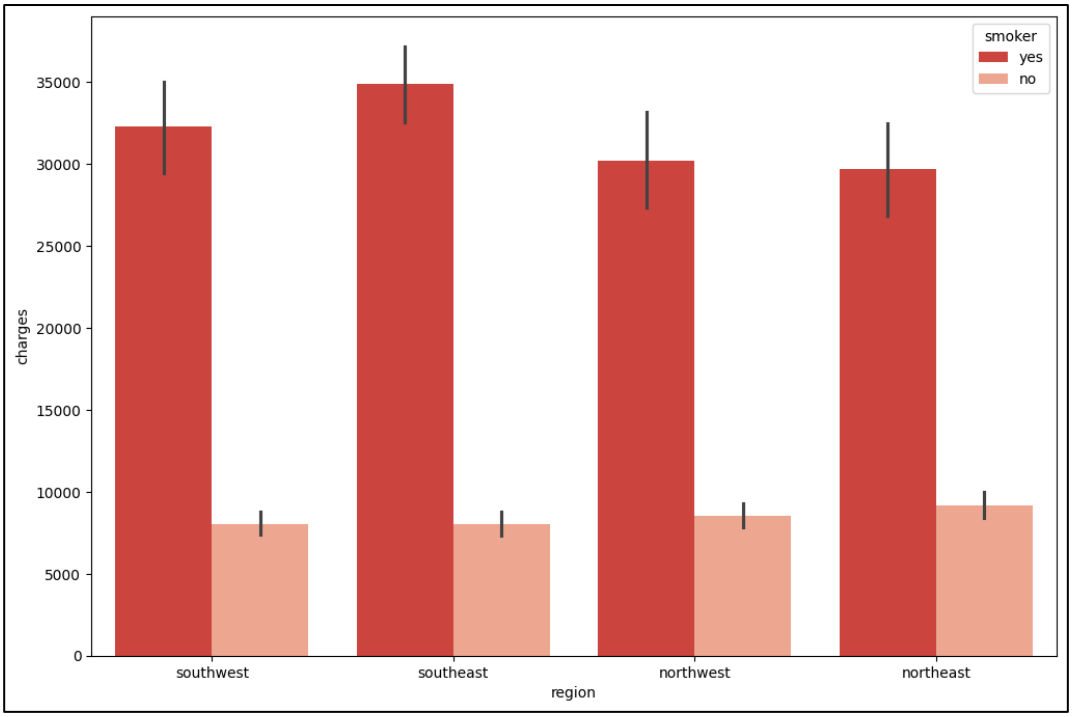
```
[10]: f, ax = plt.subplots(1, 1, figsize=(12, 8))
      ax = sns.barplot(x='region', y='charges', hue='sex', data=df, palette='cool')
```



Dari gambar di atas dapat dilihat bahwa pasien laki-laki memiliki biaya lebih banyak di 3 dari 4 wilayah.

7. Selanjutnya kita akan melihat pengaruh variabel merokok atau tidak terhadap biaya kesehatan. Berikut potongan kode yang digunakan untuk memvisualisasikan pengaruh variabel merokok.

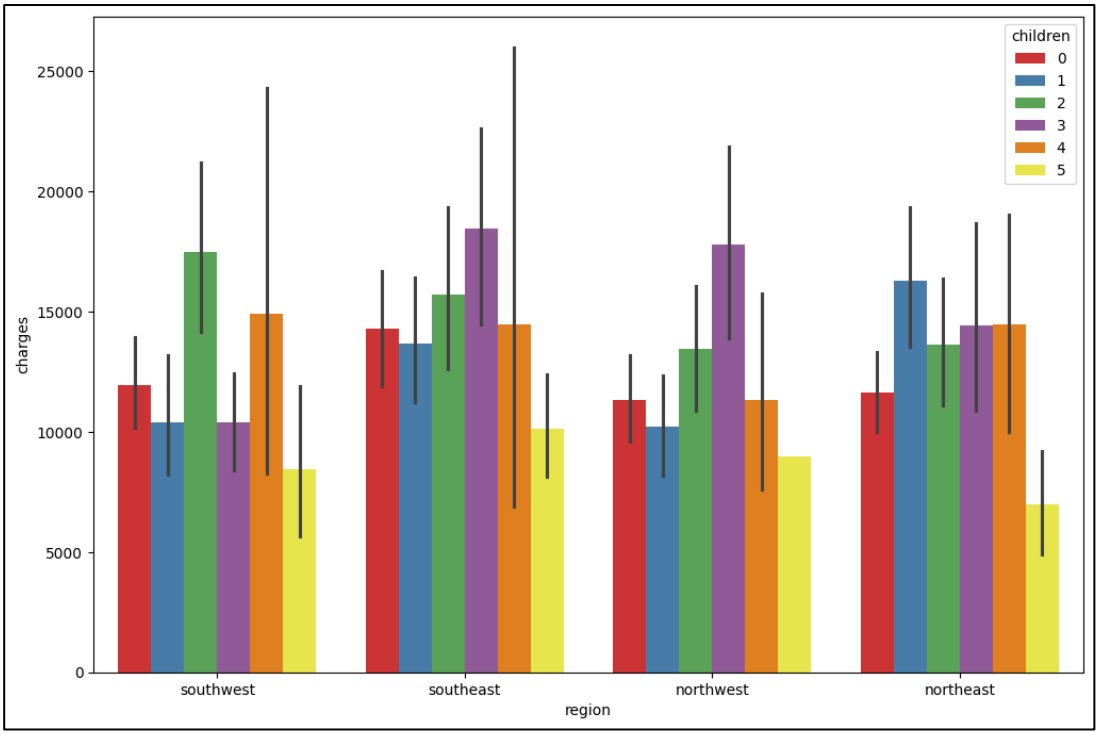
```
[12]: f, ax = plt.subplots(1,1, figsize=(12,8))
      ax = sns.barplot(x = 'region', y = 'charges',
                       hue = 'smoker', data=df, palette='Reds_r')
```



Dari gambar di atas dapat dilihat bahwa pasien merokok jauh menggunakan biaya kesehatan yang lebih tinggi dibandingkan yang tidak merokok di semua wilayah.

8. Terakhir kita akan melihat pengaruh jumlah anak yang dimiliki oleh pasien terhadap biaya kesehatan yang diklaim. Untuk melihat analisa tersebut bisa dengan menggunakan kode di bawah ini.

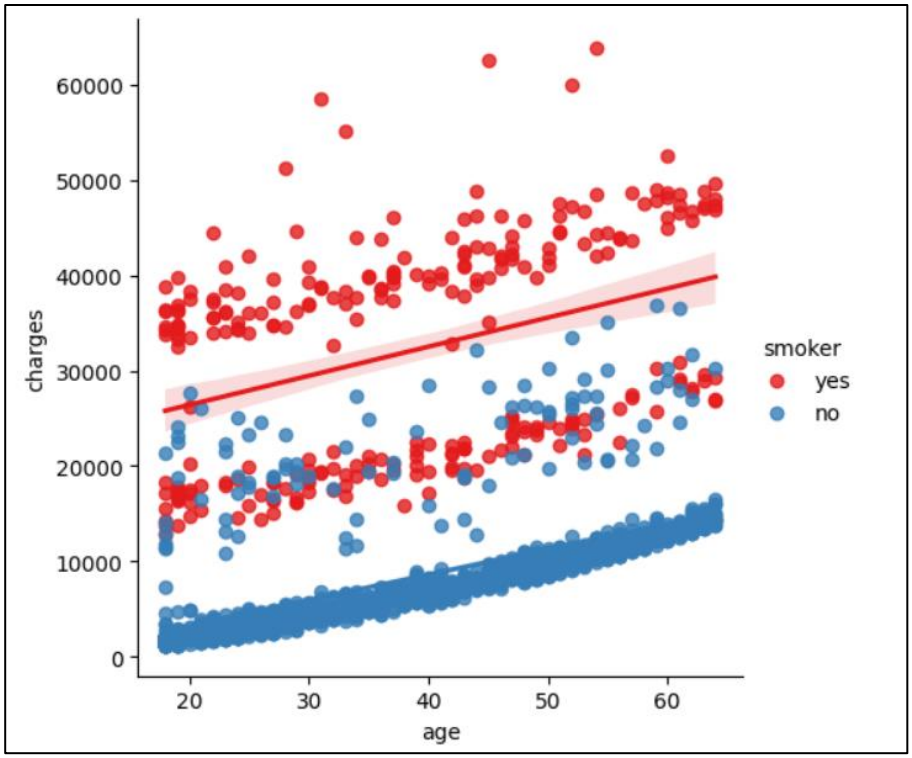
```
[14]: f, ax = plt.subplots(1, 1, figsize=(12, 8))
      ax = sns.barplot(x='region', y='charges',
                       hue='children', data=df,
                       palette='Set1')
```

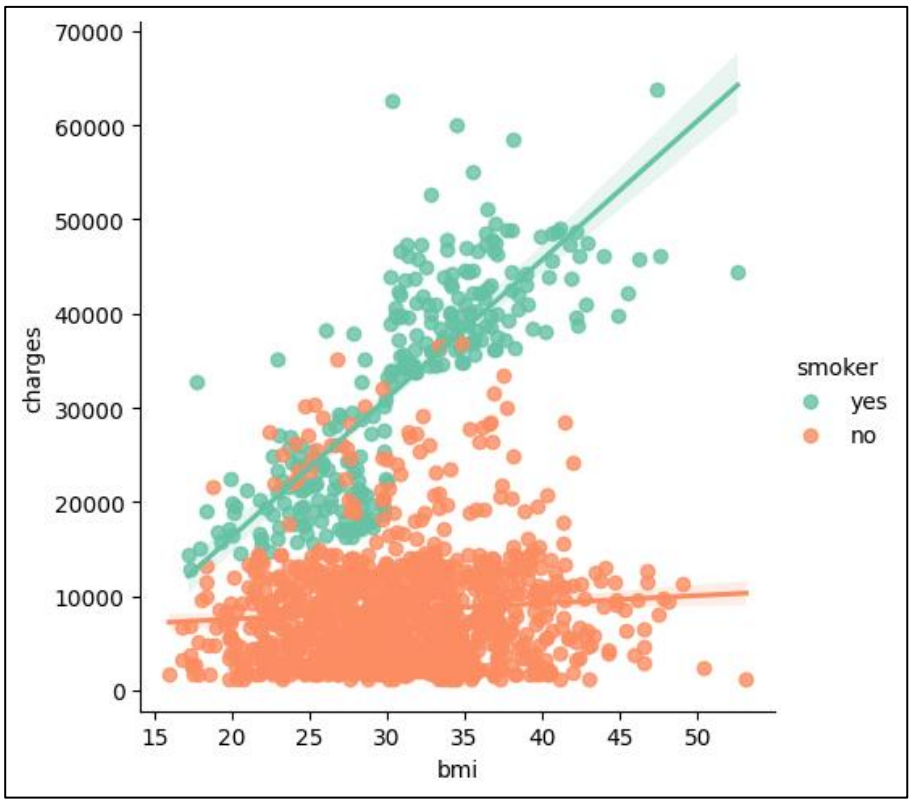


Dari gambar di atas dapat disimpulkan juga bahwa orang dengan anak-anak cenderung memiliki biaya medis yang lebih tinggi secara keseluruhan.

9. Dari sini kita sudah paham bahwa faktor merokok atau tidak merupakan variabel yang sangat penting dan berpengaruh terhadap biaya medis pasien. Untuk menganalisa dua variabel selanjutnya yaitu umur dan BMI, kita akan sekaligus menyertakan variabel merokok atau tidak. Untuk melakukan itu, silahkan gunakan kode di bawah ini.

```
[16]: ax = sns.lmplot(x='age', y='charges', data=df,
                     hue='smoker', palette='Set1')
      ax = sns.lmplot(x='bmi', y='charges', data=df,
                     hue='smoker', palette='Set2')
```





Dari kedua gambar di atas kita bisa simpulkan bahwa merokok memiliki dampak tertinggi pada biaya medis, meskipun biayanya meningkat seiring bertambahnya usia dan BMI.

10. Setelah kita selesai untuk menganalisa data dengan melakukan visualisasi dan interaksi antar data, selanjutnya kita akan melakukan prediksi biaya medis suatu pasien dengan menggunakan algoritma regresi linear. Langkah pertama yang harus kita lakukan adalah

mengubah data yang yang sifatnya label atau teks menjadi data kategorikal. Dari data di atas yang memiliki data label ada 3 yaitu jenis kelamin, merokok atau tidak, dan wilayah. Untuk mengubah dari teks menjadi kategorikal dapat dilakukan dengan kode di bawah ini.

```
[18]: df[['sex', 'smoker', 'region']] = df[['sex', 'smoker',
                                            'region']].astype('category')
```

Kode di atas akan mengubah data jenis kelamin, merokok atau tidak, dan wilayah menjadi data kategorikal.

11. Selanjutnya kita perlu mengubah data yang sudah diubah menjadi kategorikal pada langkah sebelumnya ke data numerikal. Untuk melakukan hal tersebut, kita bisa gunakan library LabelEncoder seperti potongan kode di bawah ini.

```
[21]: from sklearn.preprocessing import LabelEncoder
      label = LabelEncoder()
      label.fit(df.sex.drop_duplicates())
      df.sex = label.transform(df.sex)
      label.fit(df.smoker.drop_duplicates())
      df.smoker = label.transform(df.smoker)
      label.fit(df.region.drop_duplicates())
      df.region = label.transform(df.region)
```

12. Untuk data yang lain tidak perlu dilakukan perubahan karena sudah dalam format numerikal. Setelah semua data telah siap digunakan, kita dapat menggunakannya untuk eksperimen prediksi biaya kesehatan. Seperti yang sudah disampaikan sebelumnya, bahwa kita akan menggunakan algoritma linear regression. Pertama kita perlu melakukan pemanggilan terhadap library yang akan digunakan untuk mengeksekusi algoritma linear regression dengan kode seperti berikut.

```
[22]: from sklearn.model_selection import train_test_split as holdout
      from sklearn.linear_model import LinearRegression
      from sklearn import metrics
```

Baris pertama kode di atas digunakan untuk membagi data yang kita miliki menjadi dua porsi yaitu data training dan data testing. Baris kedua digunakan untuk memanggil algoritma linear regression. Baris ketiga digunakan untuk mengevaluasi performa dari algoritma linear regression dalam melakukan prediksi.

13. Langkah selanjutnya, kita perlu mendefinisikan data training dan data testing yang akan kita gunakan dalam eksperimen. Untuk melakukan itu, gunakan kode berikut ini.

```
[24]: x = df.drop(['charges'], axis = 1)
      y = df['charges']
      x_train, x_test, y_train, y_test = holdout(x, y, test_size=0.2,
                                                 random_state=0)
```

Variabel ‘x’ digunakan untuk menampung atribut yang digunakan dalam eksperimen, di mana semua data digunakan kecuali data ‘charges’. Kemudian variabel ‘y’ digunakan untuk menampung atribut target, yaitu atribut yang akan diprediksi, di mana yang digunakan sebagai atribut target adalah data ‘charges’. Kode baris ketiga digunakan untuk membagi data menjadi data training dan data testing, dimana 80% data akan digunakan sebagai data training dan 20% data digunakan sebagai data testing.

14. Selanjutnya kita bisa lakukan proses training (fitting) model linear regression kita dengan menggunakan data training yaitu x_train dan y_train. Untuk melakukan training bisa digunakan kode di bawah ini.

```
[25]: linear_reg = LinearRegression()
      linear_reg.fit(x_train, y_train)

[25]: LinearRegression
      LinearRegression()
```

Baris kode pertama digunakan untuk menginisialisasi model linear regression. Selanjutnya baris kedua digunakan untuk melakukan training dengan menggunakan data training, dimana x_train (berisi f itur yang akan dipelajari berisi data-data karakteristik pasien) akan di fitting dengan y_train (berisi target fitur berupa biaya kesehatan).

15. Setelah proses training, kita bisa melakukan testing dengan menggunakan data testing yang telah kita persiapkan. Pada proses testing ini, kita akan memprediksi data testing dengan menggunakan model linear regression yang telah kita train. Untuk melakukan testing dapat digunakan kode di bawah ini.

```
[26]: y_pred = linear_reg.predict(x_test)
```

Hasil prediksi akan ditampung di variabel ‘y_pred’.

16. Selanjutnya sudah tentu kita ingin melakukan evaluasi terhadap performa prediksi model linear regression yang telah kita kembangkan. Untuk melakukan evaluasi regresi, metrik yang paling umum digunakan adalah R2 dan RMSE (root mean squared error). Gunakan kode berikut untuk mengevaluasi hasil prediksi.

```
[27]: R2 = metrics.r2_score(y_test, y_pred)
      rmse = (np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
      print('R2 : {0:0.3f}'.format(R2))
      print('RMSE : {0:0.3f}'.format(rmse))

      R2 : 0.800
      RMSE : 5643.220
```

17. Selanjutnya kita juga masih bisa mencoba meningkatkan performa model linear regression dalam melakukan prediksi dengan mengeliminasi fitur yang tidak terlalu berpengaruh. Sebelum mengeliminasi fitur yang kurang penting, kita perlu mengetahui peranan masing-

masing fitur dalam eksperimen ini. Untuk melihat peranan masing-masing fitur, gunakan kode di bawah ini.

```
[31]: importance = linear_reg.coef_
      variables = ['age', 'sex', 'bmi', 'children',
                   'smoker', 'region']
      for i,v in zip(variables,importance) :
          print('Future: %s, Score: %.5f' % (i,v))

      Future: age, Score: 253.99185
      Future: sex, Score: -24.32455
      Future: bmi, Score: 328.40262
      Future: children, Score: 443.72930
      Future: smoker, Score: 23568.87948
      Future: region, Score: -288.50857
```

18. Dari hasil analisa peranan masing-masing fitur dapat disimpulkan bahwa jenis kelamin dan wilayah memiliki peranan yang tidak signifikan dalam menentukan biaya medis pasien asuransi. Selanjutnya kita juga bisa menyimpulkan bahwa faktor merokok atau tidaknya pasien merupakan fitur paling penting untuk menentukan biaya medis pasien. Penemuan ini sesuai dengan analisa yang telah kita lakukan pada bagian awal eksperimen ini.

TUGAS

Dikerjakan saat ini, jika tidak selesai bisa dilanjutkan di rumah.

Kasus:

Pada tugas ini, anda akan diberikan sebuah dataset tentang karakteristik beberapa rumah beserta dengan harganya. Tugas anda adalah memprediksi harga rumah dengan menggunakan algoritma regresi linear. Dataset dapat diunduh pada tautan yang tercantum bagian alat dan bahan dengan nama file Real estate.csv. Pada dataset ini terdapat beberapa kolom diantaranya:

1. transaction date
2. house age
3. distance to the nearest MRT station
4. number of convenience stores
5. latitude
6. longitude
7. house price of unit area

Dari data di atas data nomor 7 yang akan menjadi kelas target. Silahkan ikuti langkah-langkah praktikum di atas untuk mengerjakan tugas ini. Beberapa hal yang perlu kalian perhatikan adalah:

1. Silahkan kerjakan tugas ini dengan mengikuti semua langkah langkah di atas. Untuk visualisasi dan analisa data (langkah 5 sampai langkah 9) bersifat opsional. Namun yang mengerjakan langkah 5 sampai 9 akan mendapatkan nilai yang maksimal. 188 Data Warehousing dan Data Mining (Modul Praktikum)

```
[46]: #tugas
df = pd.read_csv('Real estate.csv')
df.head(10)
```

[46]:

	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
0	1	2012.917	32.0	84.87882	10	24.98298	121.54024	37.9
1	2	2012.917	19.5	306.59470	9	24.98034	121.53951	42.2
2	3	2013.583	13.3	561.98450	5	24.98746	121.54391	47.3
3	4	2013.500	13.3	561.98450	5	24.98746	121.54391	54.8
4	5	2012.833	5.0	390.56840	5	24.97937	121.54245	43.1
5	6	2012.667	7.1	2175.03000	3	24.96305	121.51254	32.1
6	7	2012.667	34.5	623.47310	7	24.97933	121.53642	40.3
7	8	2013.417	20.3	287.60250	6	24.98042	121.54228	46.7
8	9	2013.500	31.7	5512.03800	1	24.95095	121.48458	18.8
9	10	2013.417	17.9	1783.18000	3	24.96731	121.51486	22.1

```
[47]: df.describe()
```

[47]:

	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
count	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000
mean	207.500000	2013.148971	17.712560	1083.885689	4.094203	24.969030	121.533361	37.980193
std	119.655756	0.281967	11.392485	1262.109595	2.945562	0.012410	0.015347	13.606488
min	1.000000	2012.667000	0.000000	23.382840	0.000000	24.932070	121.473530	7.600000
25%	104.250000	2012.917000	9.025000	289.324800	1.000000	24.963000	121.528085	27.700000
50%	207.500000	2013.167000	16.100000	492.231300	4.000000	24.971100	121.538630	38.450000
75%	310.750000	2013.417000	28.150000	1454.279000	6.000000	24.977455	121.543305	46.600000
max	414.000000	2013.583000	43.800000	6488.021000	10.000000	25.014590	121.566270	117.500000

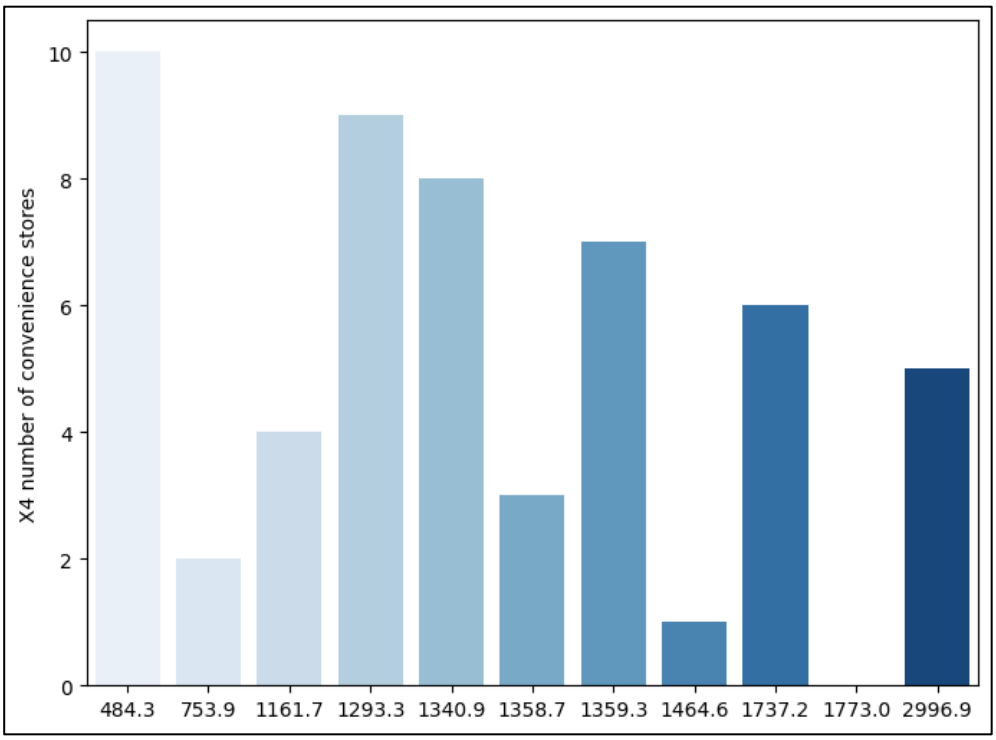
[48]:
```
df.isnull().sum()
```

[48]:
```
No                                      0
X1 transaction date                     0
X2 house age                            0
X3 distance to the nearest MRT station  0
X4 number of convenience stores         0
X5 latitude                             0
X6 longitude                            0
Y house price of unit area              0
dtype: int64
```

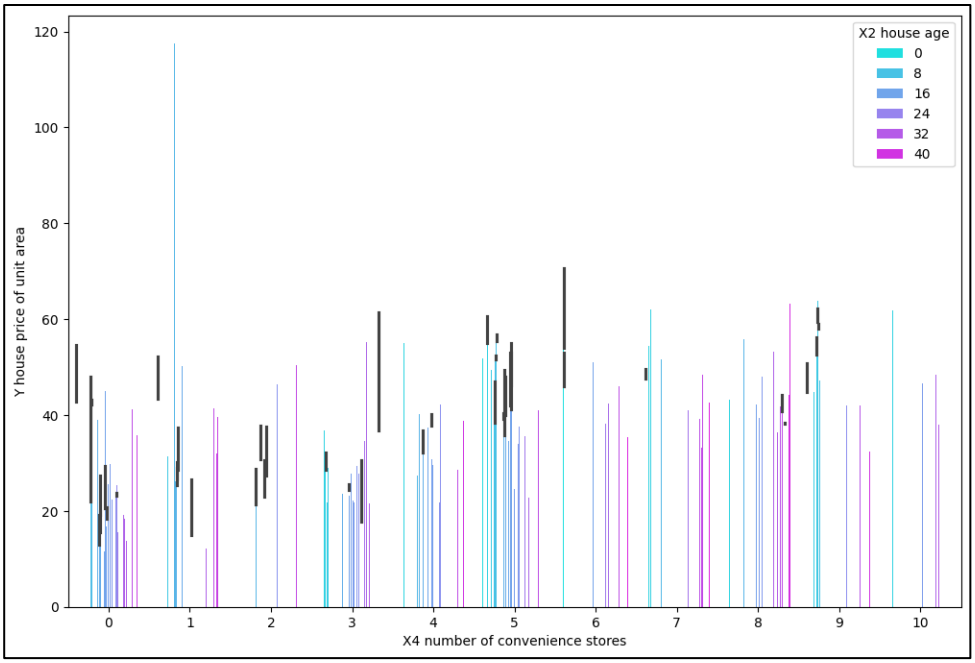
[49]:
```
# Menghitung total harga rumah per jumlah toko terdekat, lalu mengurutkan
charges = df['Y house price of unit area'].groupby(df['X4 number of convenience stores']).sum().sort_values(ascending=True)

# Membuat visualisasi barplot horizontal
f, ax = plt.subplots(1, 1, figsize=(8, 6))
sns.barplot(x=charges.values, y=charges.index, palette='Blues', ax=ax)

# Menampilkan plot
plt.show()
```

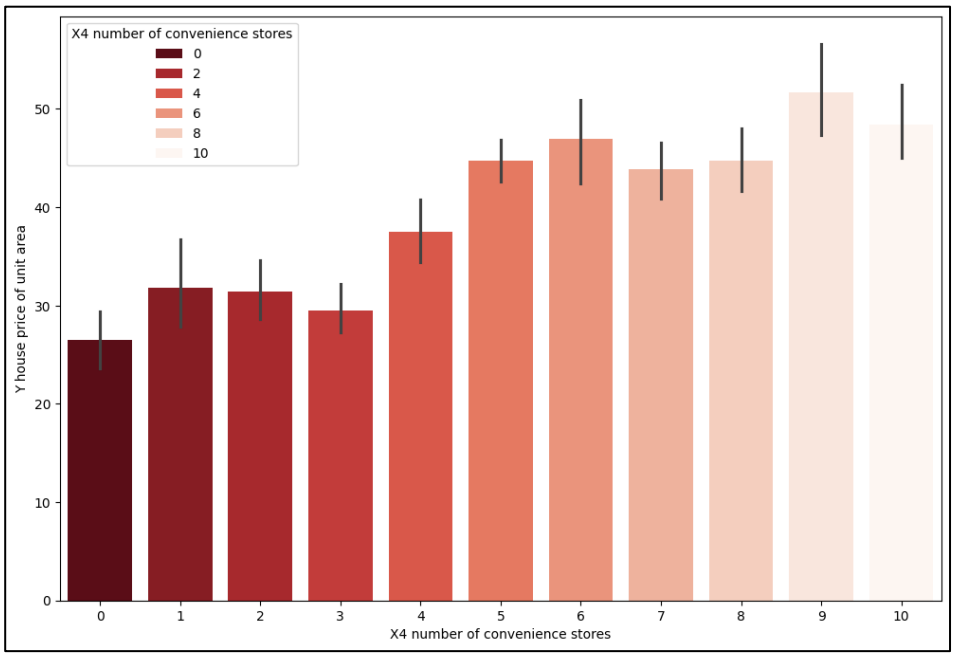


[52]:
```
f, ax = plt.subplots(1, 1, figsize=(12, 8))
ax = sns.barplot(x='X4 number of convenience stores',
                 y='Y house price of unit area', hue='X2 house age',
                 data=df, palette='cool')
```



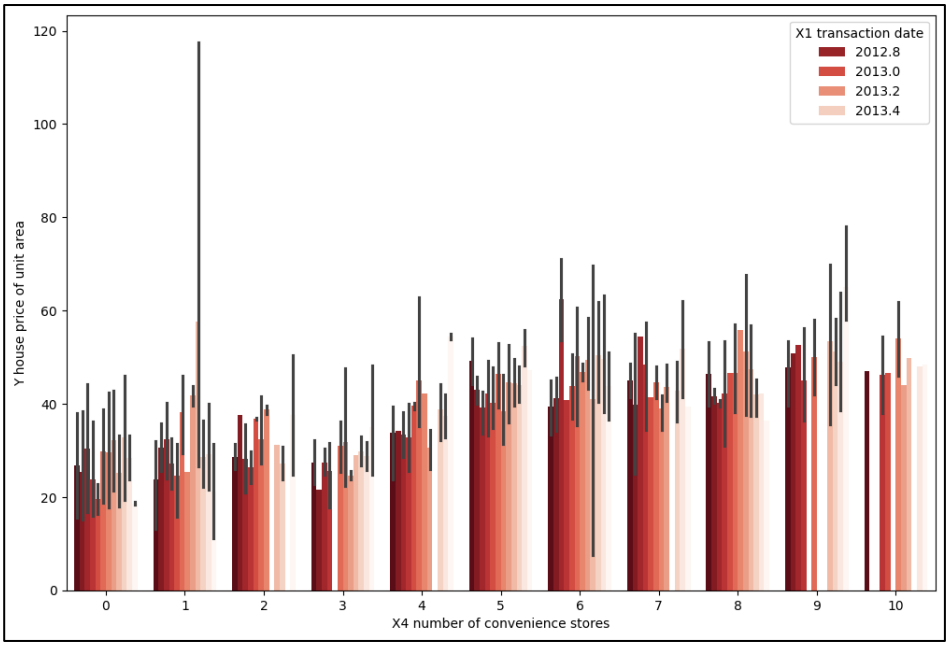

```
[53]: f, ax = plt.subplots(1, 1, figsize=(12, 8))
      sns.barplot(x='X4 number of convenience stores',
                  y='Y house price of unit area',
                  hue='X4 number of convenience stores',
                  data=df, palette='Reds_r')
```

```
[53]: <Axes: xlabel='X4 number of convenience stores', ylabel='Y house price of unit area'>
```

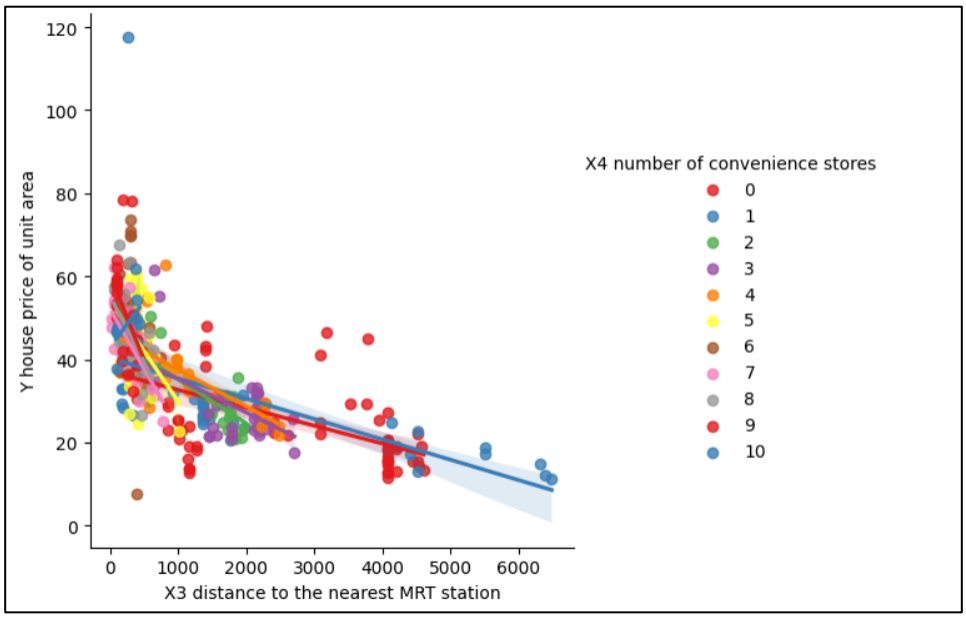


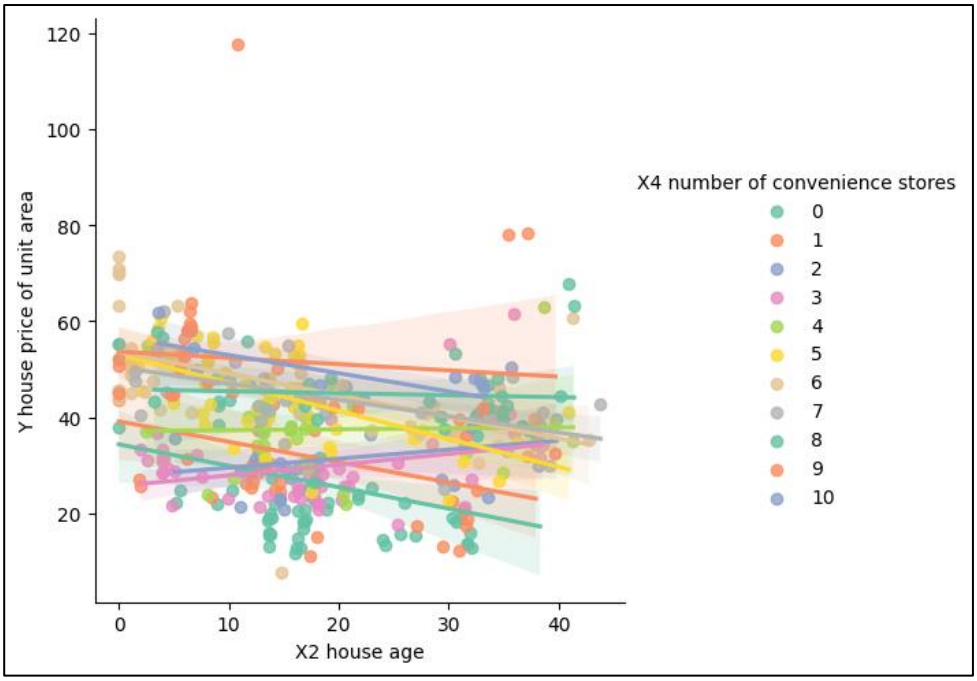

```
[54]: f, ax = plt.subplots(1, 1, figsize=(12, 8))
      sns.barplot(x='X4 number of convenience stores',
                  y='Y house price of unit area',
                  hue='X1 transaction date',
                  data=df, palette='Reds_r')
```

```
[54]: <Axes: xlabel='X4 number of convenience stores', ylabel='Y house price of unit area'>
```




```
[55]: ax = sns.lmplot(x='X3 distance to the nearest MRT station',
                      y='Y house price of unit area', data=df,
                      hue='X4 number of convenience stores', palette='Set1')
      ax = sns.lmplot(x='X2 house age', y='Y house price of unit area',
                      data=df, hue='X4 number of convenience stores', palette='Set2')
```






```
[56]: df['X4 number of convenience stores'] = df['X4 number of convenience stores'].astype('category')

[57]: from sklearn.preprocessing import LabelEncoder

      label = LabelEncoder()
      df['X4 number of convenience stores'] = label.fit_transform(df['X4 number of convenience stores'])

[58]: from sklearn.model_selection import train_test_split as holdout
      from sklearn.linear_model import LinearRegression
      from sklearn import metrics

[59]: x = df.drop(['Y house price of unit area'], axis=1)
      y = df['Y house price of unit area']
      x_train, x_test, y_train, y_test = holdout(x, y, test_size=0.2, random_state=0)

[60]: linear_reg = LinearRegression()
      linear_reg.fit(x_train, y_train)

[60]: LinearRegression
      LinearRegression()

[61]: y_pred = linear_reg.predict(x_test)
```

```
[62]: R2 = metrics.r2_score(y_test, y_pred)
      rmse = (np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
      print('R2    : {0:.3f}'.format(R2))
      print('RMSE  : {0:.3f}'.format(rmse))

      R2    : 0.656
      RMSE  : 7.733

[63]: importance = linear_reg.coef_
      variables = ['X1 transaction date', 'X2 house age', 'X3 distance to the nearest MRT station',
                   'X4 number of convenience stores', 'X5 latitude', 'X6 longitude', 'Y house price of unit area']
      for i, v in zip(variables, importance):
          print('Feature: %s, Score: %.5f' % (i, v))

      Feature: X1 transaction date, Score: -0.00446
      Feature: X2 house age, Score: 4.87352
      Feature: X3 distance to the nearest MRT station, Score: -0.26278
      Feature: X4 number of convenience stores, Score: -0.00452
      Feature: X5 latitude, Score: 1.08106
      Feature: X6 longitude, Score: 226.07761
      Feature: Y house price of unit area, Score: -9.84541
```

2. Jika menurut anda ada fitur atau data yang tidak berpengaruh terhadap harga rumah, anda bisa mengeliminasi fitur tersebut. Kemudian bandingkan hasilnya dengan ketika menggunakan semua fitur.

[67]:
```
#2
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
```

[68]:
```
# Dataset dengan semua fitur
X_all = df[['X1 transaction date', 'X2 house age', 'X3 distance to the nearest MRT station',
            'X4 number of convenience stores', 'X5 latitude', 'X6 longitude']]
y = df['Y house price of unit area']

# Dataset tanpa fitur dengan pengaruh rendah
X_reduced = df[['X2 house age', 'X5 latitude', 'X6 longitude']]  # Fitur dengan pengaruh signifikan
```

[69]:
```
# Membagi dataset menjadi training dan testing
X_train_all, X_test_all, y_train, y_test = train_test_split(X_all, y, test_size=0.2, random_state=42)
X_train_reduced, X_test_reduced, _, _ = train_test_split(X_reduced, y, test_size=0.2, random_state=42)

# Model dengan semua fitur
model_all = LinearRegression()
model_all.fit(X_train_all, y_train)
y_pred_all = model_all.predict(X_test_all)

# Model tanpa fitur dengan pengaruh rendah
model_reduced = LinearRegression()
model_reduced.fit(X_train_reduced, y_train)
y_pred_reduced = model_reduced.predict(X_test_reduced)

# Evaluasi performa
mse_all = mean_squared_error(y_test, y_pred_all)
mse_reduced = mean_squared_error(y_test, y_pred_reduced)

print(f"MSE dengan semua fitur: {mse_all:.5f}")
print(f"MSE tanpa fitur dengan pengaruh rendah: {mse_reduced:.5f}")
```

```
MSE dengan semua fitur: 53.50562
MSE tanpa fitur dengan pengaruh rendah: 72.01936
```