

Nama : Adella Septiana M.

NIM : 662017002

## TUGAS KOMPUTASI MATEMATIKA – PENGANTAR PROJECT 1

### A. REGRESI DENGAN MENORMALISASI DATA

#### 1. Simple Feature Scaling Normalization

a. Rumus Normalisasi

$$x_{new} = \frac{x_{old}}{x_{max}}$$

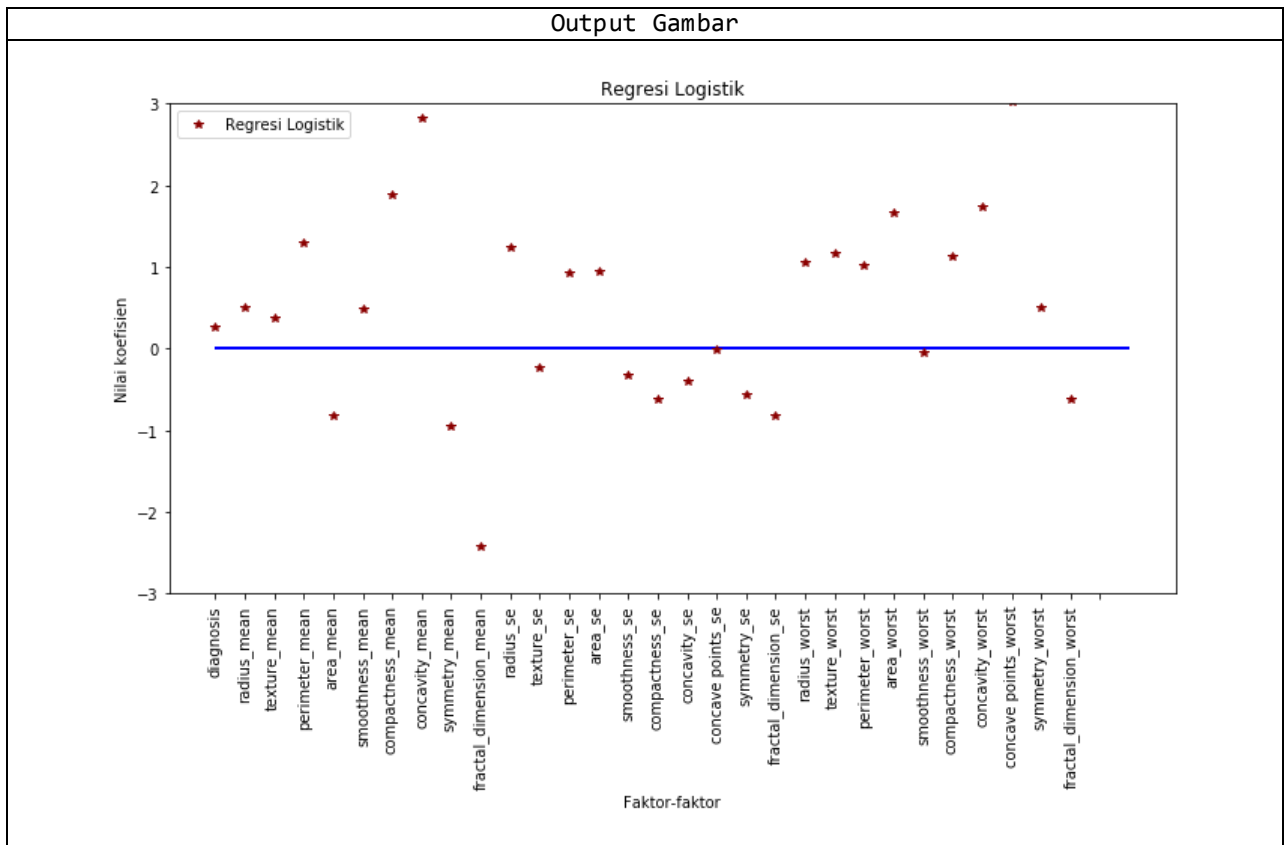
b. Hasil Pemrograman

Program	Output
<pre>import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns Bcancer = pd.read_csv('data.csv') Bcancer.drop(['id', 'Unnamed:32'],axis=1,inplace=True) Bcancer.columns</pre>	<pre>Index(['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se', 'radius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'symmetry_worst', 'fractal_dimension_worst'], dtype='object')</pre>
<pre>namafaktor = ['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se', 'radius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'symmetry_worst', 'fractal_dimension_worst'] X = Bcancer[namafaktor] y = Bcancer.diagnosis</pre>	
<pre>#Normalisasi Simple Feature Scaling X = X / (np.max(X)).values</pre>	
<pre># Untuk regresi bagi data yang sudah dinormalkan from sklearn.model_selection import train_test_split</pre>	

<pre>X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42)</pre>	
<pre>#perlu transpose X_train = X_train.T X_test = X_test.T y_train = y_train.T y_test = y_test.T</pre>	
<pre>#Logistik terhadap data yang dinormalisasi from sklearn import linear_model logreg=linear_model.LogisticRegression(random_state = 42,max_iter= 150) print("akurasi data uji: {} ".format(logreg.fit(X_train.T, y_train.T).score(X_test.T, y_test.T))) print("akurasi data latih: {} ".format(logreg.fit(X_train.T, y_train.T).score(X_train.T, y_train.T)))</pre>	<p>akurasi data uji: 0.9473684210526315 akurasi data latih: 0.9597989949748744</p>

### c. Hasil Visualisasi Gambar

Program
<pre>Bcancer_faktor = [x for i,x in enumerate(Bcancer.columns) if i!=8] plt.figure(figsize=(12,6)) #plt.plot(model.coef_.T, 'o', label="LogisticRegression()") #plt.plot(model_001.coef_.T, 'v', label="LogisticRegression(C=0.01)") plt.plot(logreg.coef_.T, '*', color='darkred',label="Regresi Logistik") plt.xticks(range(Bcancer.shape[1]), Bcancer_faktor, rotation=90) plt.hlines(0, 0, Bcancer.shape[1],color='blue', linewidth=2) plt.ylim(-3, 3) plt.title('Regresi Logistik') plt.xlabel("Faktor-faktor") plt.ylabel("Nilai koefisien") plt.legend() plt.show()</pre>



d. Analisa

Sebelum menghitung data menggunakan Regresi Logistik, perhitungan diatas menggunakan metode *Simple Feature Scaling Normalization* terlebih dahulu untuk menormalisasikan data. Setelah itu, barulah dalam perhitungan menggunakan regresi logistik, didapatkan hasil akurasi data uji (*data test*) dan akurasi data latih (*data train*) berturut-turut sebesar 0.9473684210526315 dan 0.9597989949748744. Dari data tersebut, bisa dilihat bahwa kedua nilai melebihi 0,9. Hal ini berarti semua faktor mulai dari rata-rata radius (*radius mean*) sampai dimensi fraktal terburuk (*fractal dimention worst*) saling berkorelasi dan sangat mempengaruhi diagnosa kanker payudara ini.

## 2. Z-Score Normalization

a. Rumus Normalisasi

$$x_{new} = \frac{x_{old} - \mu}{\sigma}$$

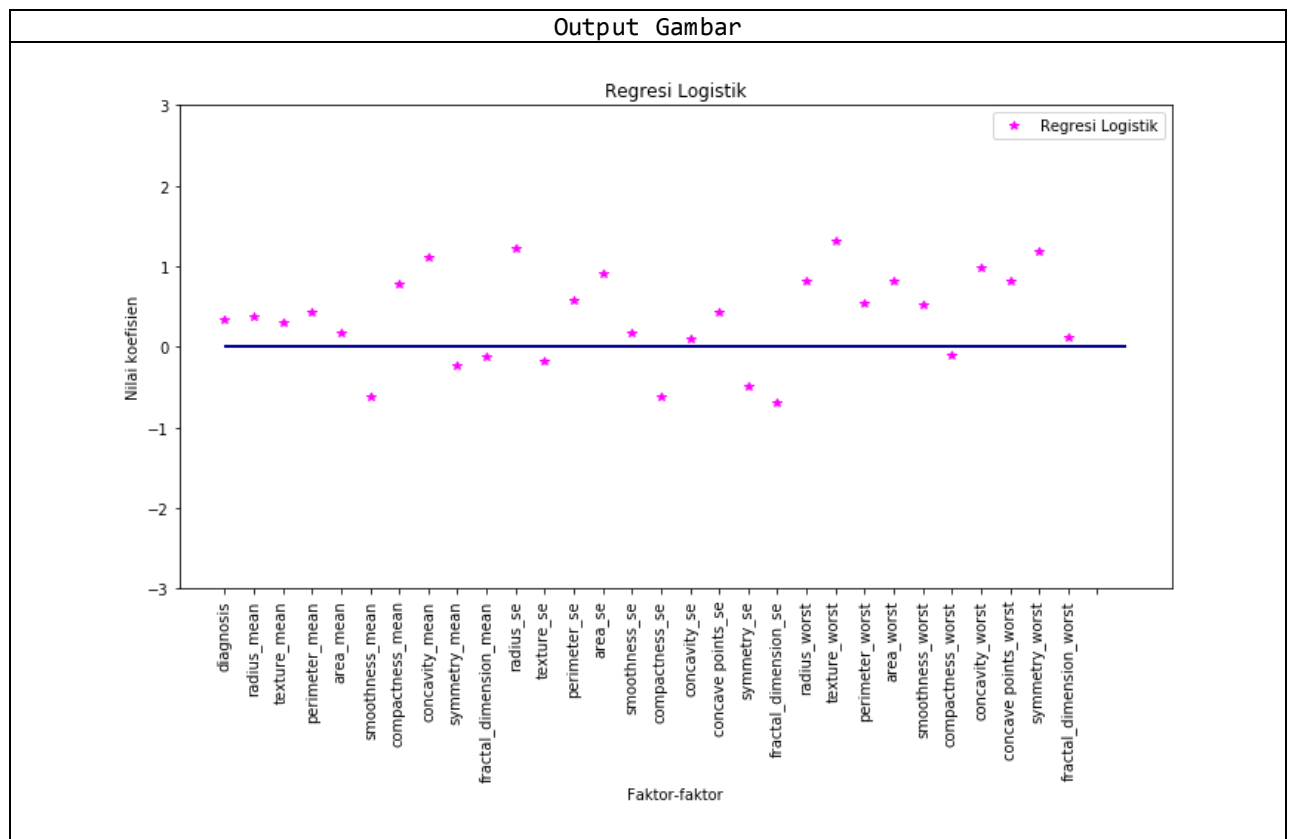
Dimana  $\mu$  adalah rerata atau Mean, dan  $\sigma$  adalah standar deviasi.

b. Hasil Pemrograman

Program	Output
<pre> namafaktor = ['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se', 'radius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'symmetry_worst', 'fractal_dimension_worst'] X = Bcancer[namafaktor] y = Bcancer.diagnosis </pre>	
<pre> #### normalisasi Z = (X - μ) / σ #### X = (X - np.mean(X))/np.std(X).values # Untuk regresi bagi data yang sudah dinormalkan from sklearn.model_selection import train_test_split X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42) </pre>	
<pre> #perlu transpose X_train = X_train.T X_test = X_test.T y_train = y_train.T y_test = y_test.T </pre>	
<pre> #Logistik terhadap data yang dinormalisasi from sklearn import linear_model logreg = linear_model.LogisticRegression(random_state = 42,max_iter= 150) print("akurasi data uji: {} ".format(logreg.fit(X_train.T, y_train.T).score(X_test.T, y_test.T))) print("akurasi data latih: {} ".format(logreg.fit(X_train.T, y_train.T).score(X_train.T, y_train.T))) </pre>	<pre> akurasi data uji: 0.9824561403508771 akurasi data latih: 0.9874371859296482 </pre>

c. Hasil Visualisasi Gambar

Program
<pre> Bcancer_faktor = [x for i,x in enumerate(Bcancer.columns) if i!=8] plt.figure(figsize=(12,6)) #plt.plot(model.coef_.T, 'o', label="LogisticRegression()") #plt.plot(model_001.coef_.T, 'v', label="LogisticRegression(C=0.01)") plt.plot(logreg.coef_.T, '*', color='magenta',label="Regresi Logistik") plt.xticks(range(Bcancer.shape[1]), Bcancer_faktor, rotation=90) plt.hlines(0, 0, Bcancer.shape[1],color='darkblue', linewidth=2) plt.ylim(-3, 3) plt.title('Regresi Logistik') plt.xlabel("Faktor-faktor") plt.ylabel("Nilai koefisien") plt.legend() plt.show() </pre>



d. Analisa

Sebelum menghitung data menggunakan metode *Z-Score Normalization* terlebih dahulu untuk menormalisasikan data. Setelah itu, barulah dalam perhitungan menggunakan regresi logistik, didapatkan hasil akurasi data uji (*data test*) dan akurasi data latih (*data train*) berturut-turut sebesar 0.9824561403508771 dan 0.9874371859296482. Dari data tersebut, bisa dilihat bahwa kedua nilai melebihi 0,9 bahkan mendekati 1. Hal ini berarti semua faktor mulai dari rata-rata radius (*radius mean*) sampai dimensi fraktal terburuk (*fractal dimention worst*) saling berkorelasi dan sangat mempengaruhi diagnosa kanker payudara ini.

### 3. Min - Max Normalization

a. Rumus Normalisasi

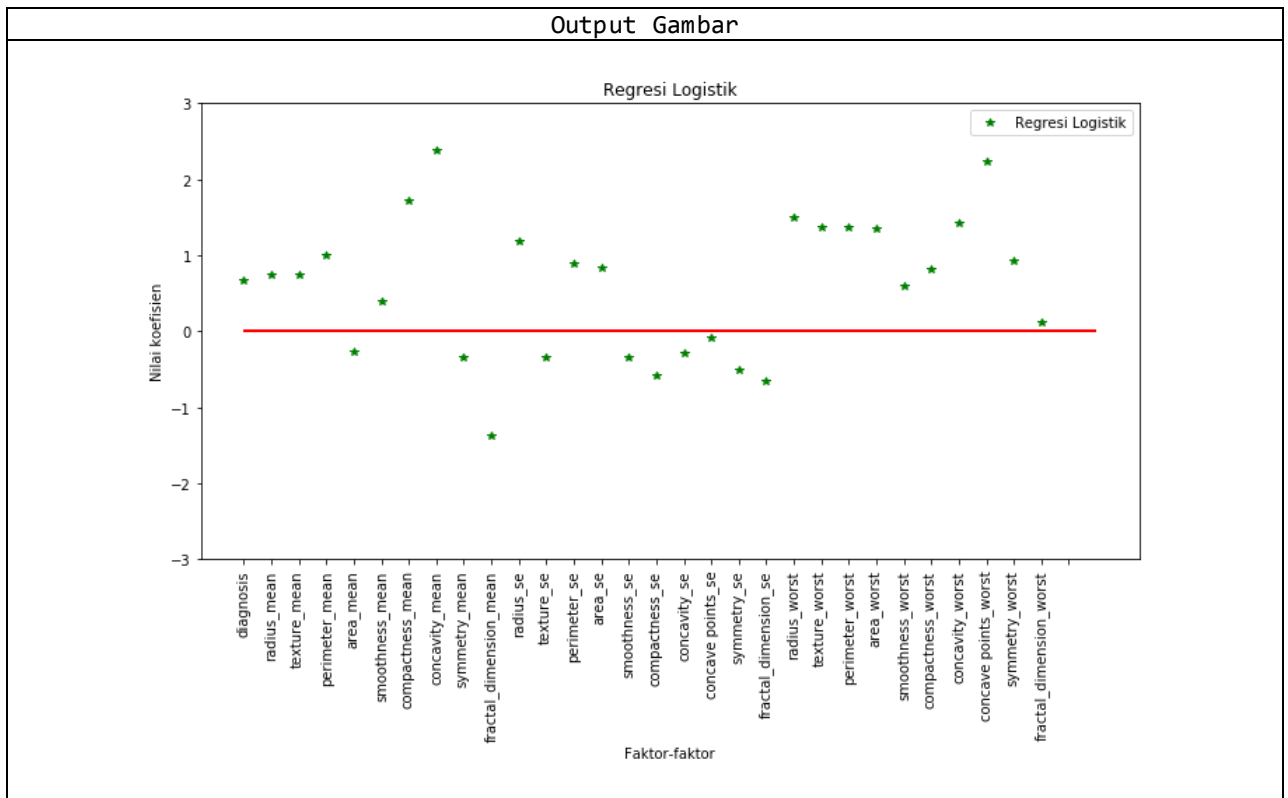
$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}}$$

b. Hasil Pemrograman

Program	Output
<pre> namafaktor = ['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se', 'radius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'symmetry_worst', 'fractal_dimension_worst'] X = Bcancer[namafaktor] y = Bcancer.diagnosis </pre>	
<pre> # KASUS 2 : data dinormalkan dulu X = (X - np.min(X))/(np.max(X)-np.min(X)).values # Untuk regresi bagi data yang sudah dinormalkan from sklearn.model_selection import train_test_split X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42) </pre>	
<pre> #perlu transpose X_train = X_train.T X_test = X_test.T y_train = y_train.T y_test = y_test.T </pre>	
<pre> #Logistik terhadap data yang dinormalisasi from sklearn import linear_model logreg = linear_model.LogisticRegression(random_state = 42,max_iter= 150) print("akurasi data uji: {} ".format(logreg.fit(X_train.T, y_train.T).score(X_test.T, y_test.T))) print("akurasi data latih: {} ".format(logreg.fit(X_train.T, y_train.T).score(X_train.T, y_train.T))) </pre>	<p>akurasi data uji: 0.9766081871345029 akurasi data latih: 0.9673366834170855</p>

c. Hasil Visualisasi Gambar

Program
<pre> Bcancer_faktor = [x for i,x in enumerate(Bcancer.columns) if i!=8] plt.figure(figsize=(12,6)) #plt.plot(model.coef_.T, 'o', label="LogisticRegression()") #plt.plot(model_001.coef_.T, 'v', label="LogisticRegression(C=0.01)") plt.plot(logreg.coef_.T, '*', color='green',label="Regresi Logistik") plt.xticks(range(Bcancer.shape[1]), Bcancer_faktor, rotation=90) plt.hlines(0, 0, Bcancer.shape[1],color='red', linewidth=2) plt.ylim(-3, 3) plt.title('Regresi Logistik') plt.xlabel("Faktor-faktor") plt.ylabel("Nilai koefisien") plt.legend() plt.show() </pre>



d. Analisa

Sebelum menghitung data menggunakan Regresi Logistik, perhitungan diatas menggunakan metode *Minimal-Maximal Normalization* terlebih dahulu untuk menormalisasikan data. Setelah itu, barulah dalam perhitungan menggunakan regresi logistik, didapatkan hasil akurasi data uji (*data test*) dan akurasi data latih (*data train*) berturut-turut sebesar 0.9766081871345029 dan 0.9673366834170855 . Dari data tersebut, bisa dilihat bahwa kedua nilai melebihi 0,9 bahkan mendekati 1. Hal ini berarti semua faktor mulai dari rata-rata radius (*radius mean*) sampai dimensi fraktal terburuk (*fractal dimention worst*) saling berkorelasi dan sangat mempengaruhi diagnosa kanker payudara ini.

## B. REGRESI DENGAN DATA TEST YANG BERBEDA

### 1. Data Test : 30 %

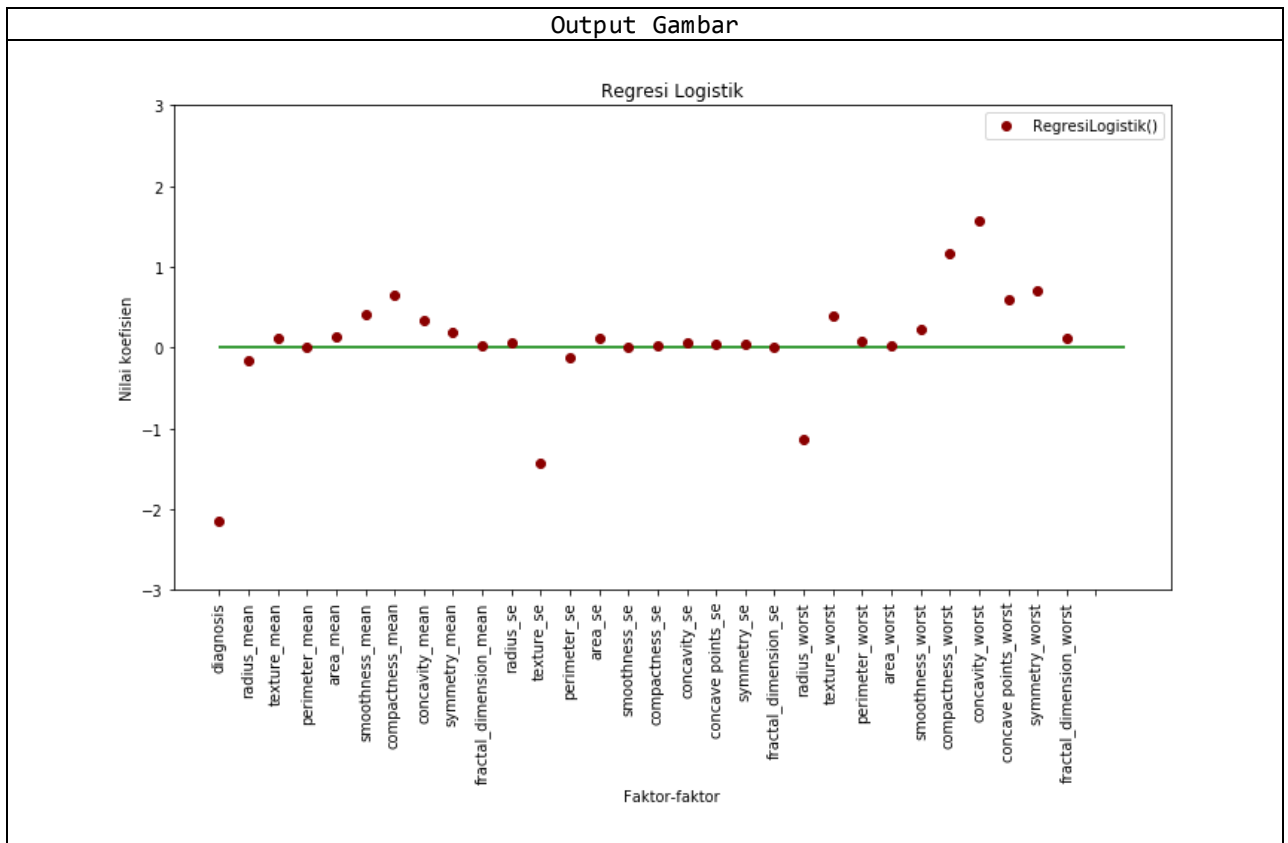
- Data Test yang Digunakan : 0.3  
Random State yang Digunakan : 42
- Hasil Pemrograman

Program	Output
<pre>import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns Bcancer = pd.read_csv('data.csv') Bcancer.drop(['id', 'Unnamed:32'],axis=1, inplace=True) B.cancer.columns</pre>	<pre>Index(['diagnosis', 'radius_mean', 'tex ture_mean', 'perimeter_mean','area_mean ', 'smoothness_mean', 'compactness_mean ', 'concavity_mean', 'concave points_me an', 'symmetry_mean', 'fractal_dimensio n_mean', 'radius_se', 'texture_se', 'perimeter_s e', 'area_se', 'smoothness_se', 'compactness_se', 'concavity_se', 'conc ave points_se', 'symmetry_se', 'fractal_dimension_se', 'radius_worst', 'texture_worst','perimeter_worst', 'area_worst', 'smoothness_worst','compactness_worst', 'concavity_worst', 'concave</pre>

	<pre>points_worst','symmetry_worst', 'fractal_dimension_worst'],dtype='object')</pre>
<pre>namafaktor = ['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean','radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se','compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se','fractal_dimension_se', 'radius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst','compactness_worst', 'concavity_worst', 'concave points_worst','symmetry_worst', 'fractal_dimension_worst'] X = Bcancer[namafaktor] y = Bcancer.diagnosis</pre>	
<pre>#memisahkan data train dan test from sklearn.model_selection import train_test_split X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42) print(X_train.shape, y_train.shape) print(X_test.shape, y_test.shape)</pre>	<pre>(398, 30) (398,) (171, 30) (171,)</pre>
<pre>from sklearn.linear_model import LogisticRegression LogisticRegression().fit(X_train, y_train)  model = LogisticRegression().fit(X_train, y_train)  print(f"Regressi Logistik untuk data training dengan score: {format(model.score(X_train, y_train), '.4f')} ") print(f"Regressi Logistik untuk data test dengan score: {format(model.score(X_test, y_test), '.4f')} ")</pre>	<pre>Regressi Logistik untuk data training dengan score: 0.9523 Regressi Logistik untuk data test dengan score: 0.9649</pre>

### c. Hasil Visualisasi Gambar

Program
<pre>Bcancer_faktor = [x for i,x in enumerate(Bcancer.columns) if i!=8] plt.figure(figsize=(12,6)) #plt.plot(model.coef_.T, 'o', label="LogisticRegression()") #plt.plot(model_001.coef_.T, 'v', label="LogisticRegression(C=0.01)") plt.plot(model.coef_.T, 'o', color='darkred', label="RegresiLogistik()") plt.xticks(range(Bcancer.shape[1]), Bcancer_faktor, rotation=90) plt.hlines(0, 0, Bcancer.shape[1],color='green') plt.ylim(-3, 3) plt.title('Regresi Logistik') plt.xlabel("Faktor-faktor") plt.ylabel("Nilai koefisien") plt.legend() plt.show()</pre>



#### d. Analisa

Sebelum mengolah data menggunakan regresi logistik, terlebih dahulu kita memisahkan data antara set data yang harus dicapai (*data test*) dan set data yang bisa digunakan untuk mencapai tujuan (*data train*). *Data train* sendiri digunakan untuk membuat model *machine learning*, sedangkan *data test* akan digunakan untuk mengujikebenaran akan korelasi dalam model yang bersangkutan. Melalui perhitungan tersebut, didapatkan hasil bahwa ukuran dari data *X train* adalah 398 baris dan 30 kolom sedangkan *y train* sendiri berukuran 398 baris. Kemudian, untuk *X test* terdiri dari 171 baris dan 30 kolom, dan *y train* berukuran 171 baris. Pada perhitungan kali ini, juga digunakan *random state* sebesar 42 dan *test size* sebesar 30% atau 0,3. Nilai 30% sendiri menunjukkan persentase data yang harus disimpan dalam pengujian

Setelah memisah data, barulah dilakukan regresi logistik, didapatkan hasil sebesar 0.9523 untuk data training dan 0.9649 untuk data test. Karena keduanya bernilai lebih dari 0,9 bahkan mendekati 1, hal ini menandakan bahwa semua faktor mulai dari rata-rata radius (*radius mean*) sampai dimensi fraktal terburuk (*fractal dimention worst*) saling berkorelasi dan sangat mempengaruhi diagnosa kanker payudara ini.

## 2. Data Test : 20 %

- Data Test yang Digunakan : 0.2  
Random State yang Digunakan : 42
- Hasil Pemrograman

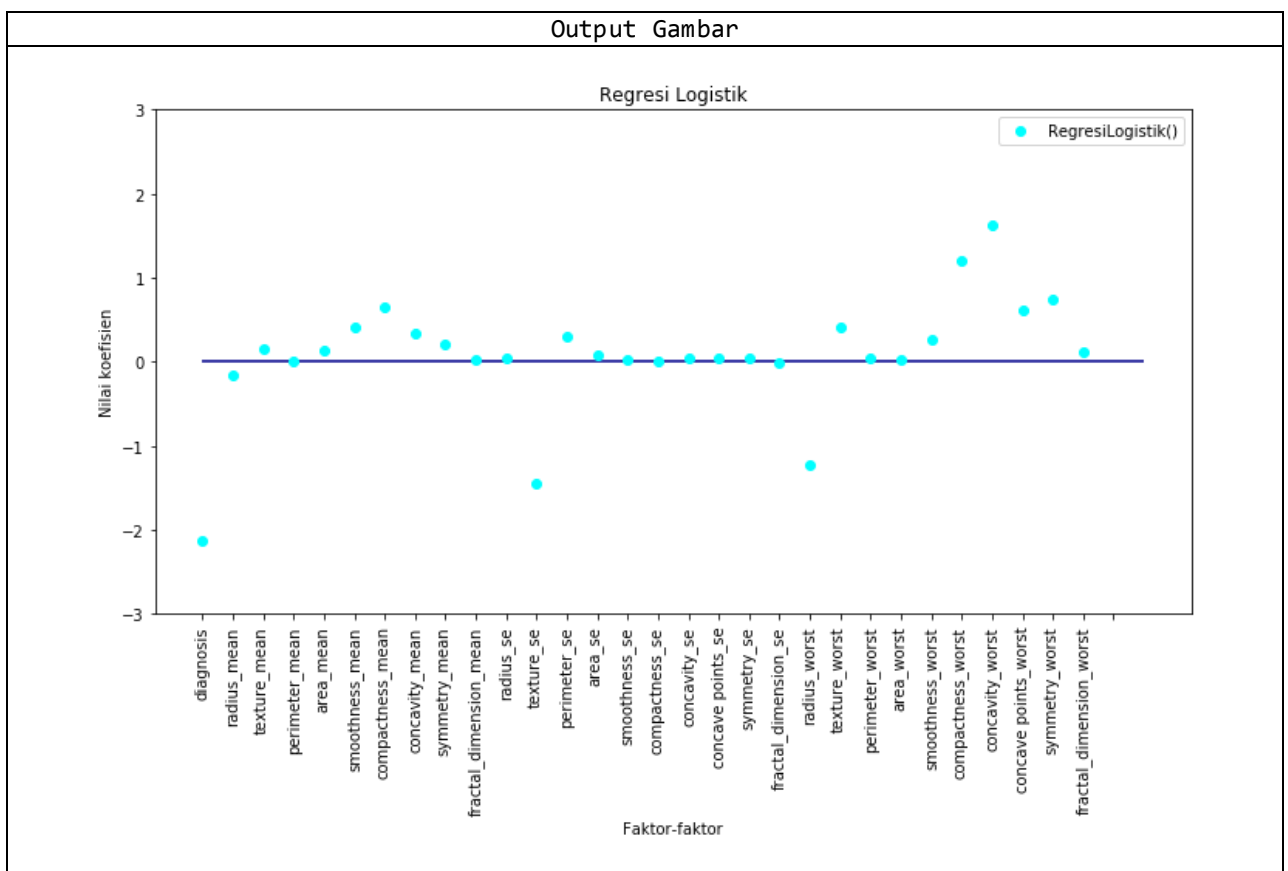
Program	Output
#memisahkan data train dan test from sklearn.model_selection import train_test_split X_train, X_test, y_train, y_test = train_test_split(X, y,	(455, 30) (455,) (114, 30) (114,)



<pre>test_size = 0.2, random_state = 42) print(X_train.shape, y_train.shape) print(X_test.shape, y_test.shape)</pre>	
<pre>from sklearn.linear_model import LogisticRegression LogisticRegression().fit(X_train, y_train)  model = LogisticRegression().fit(X_train, y_train)  print(f"Regressi Logistik untuk data training dengan score: {format(model.score(X_train, y_train), '.4f')} ") print(f"Regressi Logistik untuk data test dengan score: {format(model.score(X_test, y_test), '.4f')} ")</pre>	Regressi Logistik untuk data training dengan score: 0.9582 Regressi Logistik untuk data test dengan score: 0.9561

### c. Hasil Visualisasi Gambar

Program
<pre>Bcancer_faktor = [x for i,x in enumerate(Bcancer.columns) if i!=8] plt.figure(figsize=(12,6)) #plt.plot(model.coef_.T, 'o', label="LogisticRegression()") #plt.plot(model_001.coef_.T, 'v', label="LogisticRegression(C=0.01)") plt.plot(model.coef_.T, 'o', color='cyan', label="RegresiLogistik()") plt.xticks(range(Bcancer.shape[1]), Bcancer_faktor, rotation=90) plt.hlines(0, 0, Bcancer.shape[1],color='darkblue') plt.ylim(-3, 3) plt.title('Regresi Logistik') plt.xlabel("Faktor-faktor") plt.ylabel("Nilai koefisien") plt.legend() plt.show()</pre>



### d. Analisa

Sebelum mengolah data menggunakan regresi logistik, terlebih dahulu kita memisahkan data antara set data yang harus dicapai (*data test*) dan set data yang bisa digunakan untuk mencapai tujuan (*data train*). Melalui perhitungan tersebut, didapatkan hasil bahwa ukuran dari data *X*

*train* adalah 455 baris dan 30 kolom sedangkan *y train* sendiri berukuran 455 baris. Kemudian, untuk *X test* terdiri dari 144 baris dan 30 kolom, dan *y train* berukuran 144 baris. Pada perhitungan kali ini, juga digunakan *random state* sebesar 42 dan *test size* sebesar 20% atau 0,2. Nilai 20% sendiri menunjukkan persentase data yang harus disimpan dalam pengujian

Setelah memisah data, barulah dilakukan regresi logistik, didapatkan hasil sebesar 0.9561 untuk *data training* dan 0.9582 untuk *data test*. Karena keduanya bernilai lebih dari 0,9 bahkan mendekati 1, hal ini menandakan bahwa semua faktor mulai dari rata-rata radius (*radius mean*) sampai dimensi fraktal terburuk (*fractal dimention worst*) saling berkorelasi dan sangat mempengaruhi diagnosa kanker payudara ini.

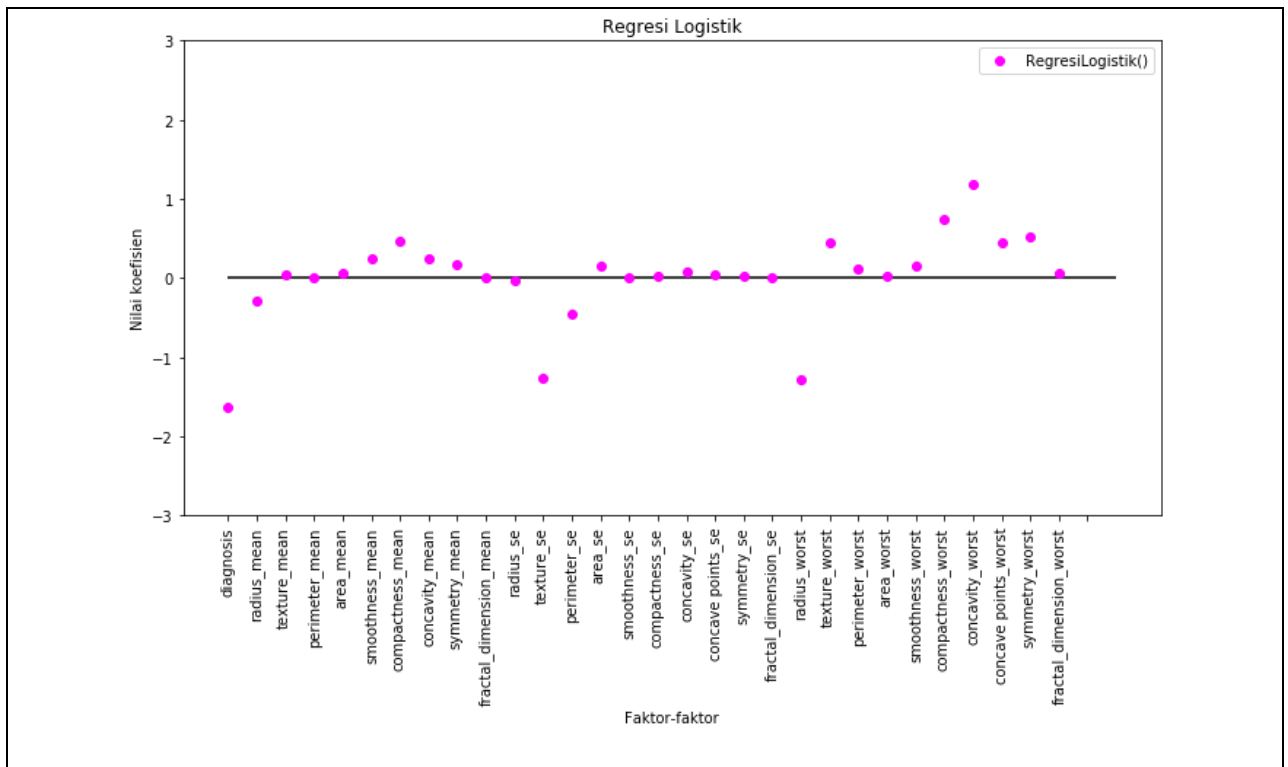
### 3. Data Test : 50 %

- a. Data Test yang Digunakan : 0.5  
Random State yang Digunakan : 42
- b. Hasil Pemrograman

Program	Output
<pre>#memisahkan data train dan test from sklearn.model_selection import train_test_split X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42) print(X_train.shape, y_train.shape) print(X_test.shape, y_test.shape)</pre>	<pre>(284, 30) (284,) (285, 30) (285,)</pre>
<pre>from sklearn.linear_model import LogisticRegression LogisticRegression().fit(X_train, y_train)  model = LogisticRegression().fit(X_train, y_train)  print(f"Regresi Logistik untuk data training dengan score: {format(model.score(X_train, y_train), '.4f')} ") print(f"Regresi Logistik untuk data test dengan score: {format(model.score(X_test, y_test), '.4f')} ")</pre>	<pre>Regresi Logistik untuk data training dengan score: 0.9542 Regresi Logistik untuk data test dengan score: 0.9509</pre>

- c. Hasil Visualisasi Gambar

Program
<pre>Bcancer_faktor = [x for i,x in enumerate(Bcancer.columns) if i!=8] plt.figure(figsize=(12,6)) #plt.plot(model.coef_.T, 'o', label="LogisticRegression()") #plt.plot(model_001.coef_.T, 'v', label="LogisticRegression(C=0.01)") plt.plot(model.coef_.T, 'o', color = 'magenta', label="RegresiLogistik()") plt.xticks(range(Bcancer.shape[1]), Bcancer_faktor, rotation=90) plt.hlines(0, 0, Bcancer.shape[1]) plt.ylim(-3, 3) plt.title('Regresi Logistik') plt.xlabel("Faktor-faktor") plt.ylabel("Nilai koefisien") plt.legend() plt.show()</pre>
Output Gambar



#### d. Analisa

Sebelum mengolah data menggunakan regresi logistik, terlebih dahulu kita memisahkan data antara set data yang harus dicapai (*data test*) dan set data yang bisa digunakan untuk mencapai tujuan (*data train*). Melalui perhitungan tersebut, didapatkan hasil bahwa ukuran dari data *X train* adalah 284 baris dan 30 kolom sedangkan *y train* sendiri berukuran 284 baris. Kemudian, untuk *X test* terdiri dari 285 baris dan 30 kolom, dan *y train* berukuran 285 baris. Pada perhitungan kali ini, juga digunakan *random state* sebesar 42 dan *test size* sebesar 50% atau 0,5. Nilai 50% sendiri menunjukkan persentase data yang harus disimpan dalam pengujian

Setelah memisah data, barulah dilakukan regresi logistik, didapatkan hasil sebesar 0.9542 untuk *data training* dan 0.9509 untuk *data test*. Karena keduanya bernilai lebih dari 0,9 bahkan mendekati 1, hal ini menandakan bahwa semua faktor mulai dari rata-rata radius (*radius mean*) sampai dimensi fraktal terburuk (*fractal dimention worst*) saling berkorelasi dan sangat mempengaruhi diagnosa kanker payudara ini.

### C. REGRESI DENGAN RANDOM STATE YANG BERBEDA

#### 1. Random State : 10

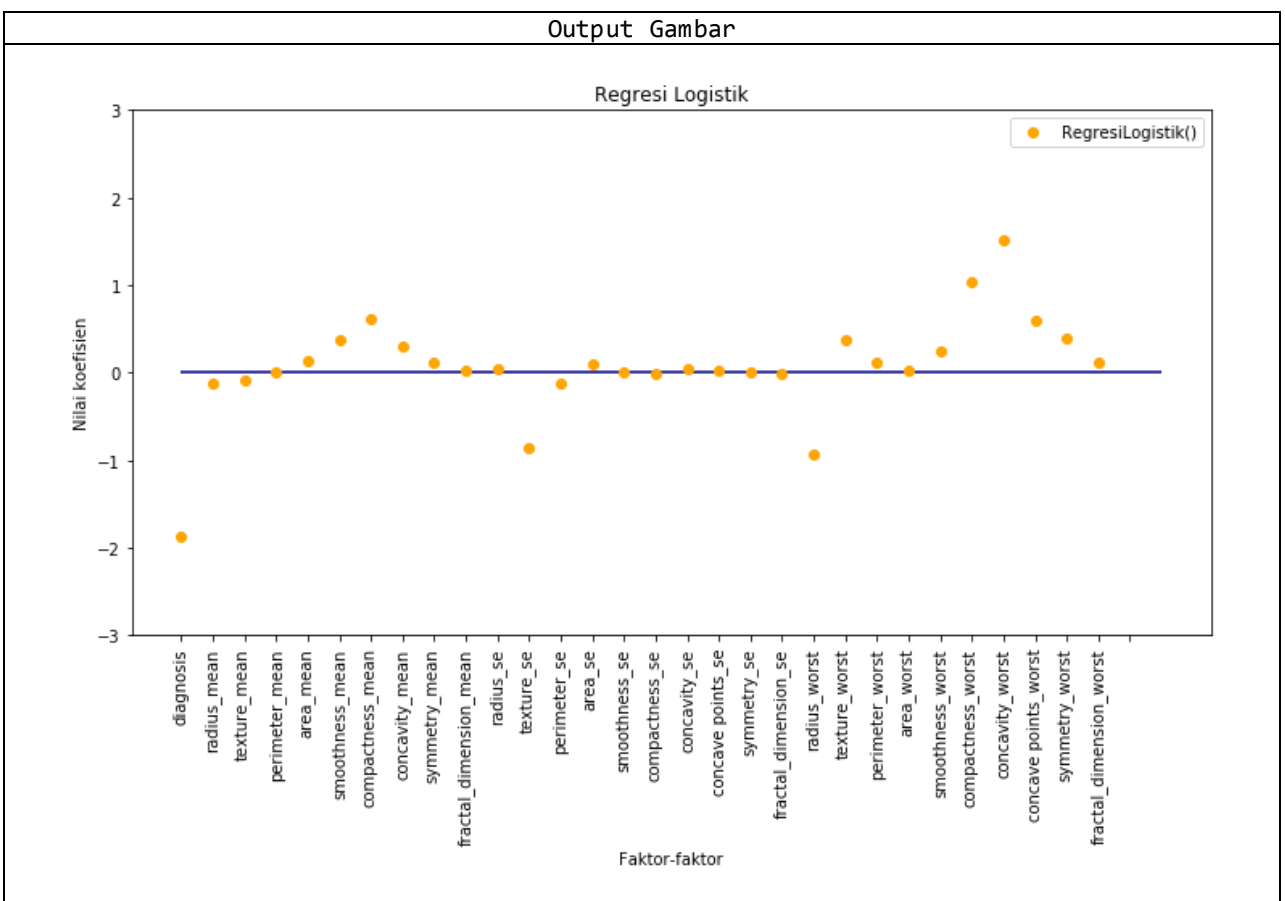
- Data Test yang Digunakan : 0.2  
Random State yang Digunakan : 10
- Hasil Pemrograman

Program	Output
<pre>#memisahkan data train dan test from sklearn.model_selection import train_test_split X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 10) print(X_train.shape, y_train.shape) print(X_test.shape, y_test.shape)</pre>	<pre>(455, 30) (455,) (114, 30) (114,)</pre>

<pre> from sklearn.linear_model import LogisticRegression LogisticRegression().fit(X_train, y_train)  model = LogisticRegression().fit(X_train, y_train)  print(f"Regressi Logistik untuk data training dengan score: {format(model.score(X_train, y_train), '.4f')} ") print(f"Regressi Logistik untuk data test dengan score: {format(model.score(X_test, y_test), '.4f')} ") </pre>	Regressi Logistik untuk data training dengan score: 0.9626 Regressi Logistik untuk data test dengan score: 0.9386
--	--

### c. Hasil Visualisasi Gambar

Program
<pre> Bcancer_faktor = [x for i,x in enumerate(Bcancer.columns) if i!=8] plt.figure(figsize=(12,6)) #plt.plot(model.coef_.T, 'o', label="LogisticRegression()") #plt.plot(model_001.coef_.T, 'v', label="LogisticRegression(C=0.01)") plt.plot(model.coef_.T, 'o', color='orange', label="RegresiLogistik()") plt.xticks(range(Bcancer.shape[1]), Bcancer_faktor, rotation=90) plt.hlines(0, 0, Bcancer.shape[1],color='darkblue') plt.ylim(-3, 3) plt.title('Regresi Logistik') plt.xlabel("Faktor-faktor") plt.ylabel("Nilai koefisien") plt.legend() plt.show() </pre>



### d. Analisa

Sebelum mengolah data menggunakan regresi logistik, terlebih dahulu kita memisahkan data antara set data yang harus dicapai (*data test*) dan set data yang bisa digunakan untuk mencapai tujuan (*data train*). Melalui perhitungan tersebut, didapatkan hasil bahwa ukuran dari data *X train* adalah 455 baris dan 30 kolom sedangkan *y train* sendiri berukuran 455 baris. Kemudian,

untuk *X test* terdiri dari 144 baris dan 30 kolom, dan *y train* berukuran 144 baris. Pada perhitungan kali ini, juga digunakan *test size* sebesar 20% atau 0,2 dan *random state* sebesar 10. Nilai *random state* yaitu 10 sendiri menunjukan sumber ke generator acak atau *random generator*, yang akan membuat pemisahan data atau *split train-test data* akan selalu bersifat deterministik atau tidak akan berbeda setiap kali dihitung.

Setelah memisah data, barulah dilakukan regresi logistik, didapatkan hasil akurasi sebesar 0.9626 untuk *data training* dan 0.9386 untuk *data test*. Karena keduanya bernilai lebih dari 0,9, hal ini menandakan bahwa semua faktor mulai dari rata-rata radius (*radius mean*) sampai dimensi fraktal terburuk (*fractal dimention worst*) saling berkorelasi dan sangat mempengaruhi diagnosa kanker payudara ini.

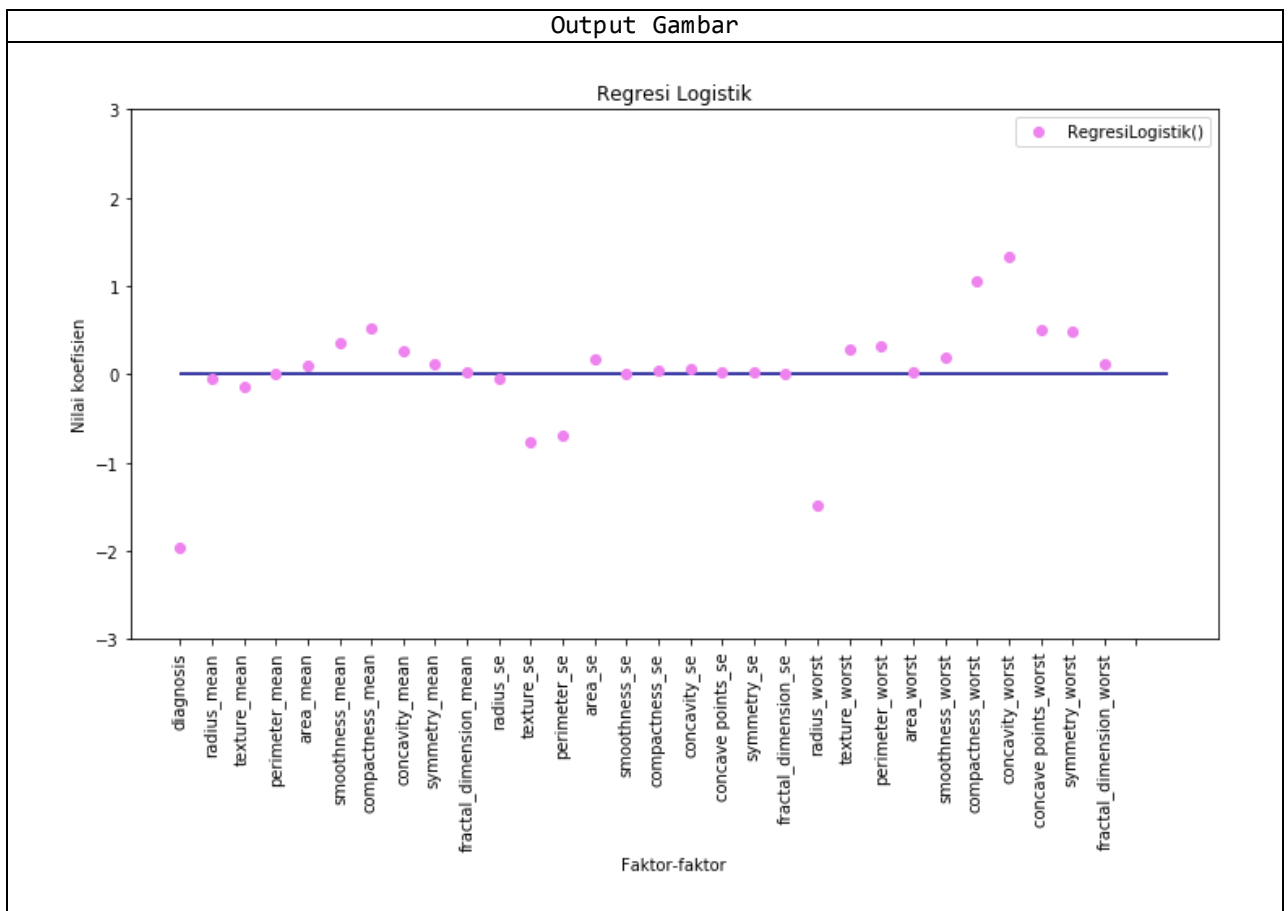
## 2. Random State : 30

- a. Data Test yang Digunakan : 0.2  
Random State yang Digunakan : 30
- b. Hasil Pemrograman

Program	Output
<pre>#memisahkan data train dan test from sklearn.model_selection import train_test_split X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 30) print(X_train.shape, y_train.shape) print(X_test.shape, y_test.shape)</pre>	<pre>(455, 30) (455,) (114, 30) (114,)</pre>
<pre>from sklearn.linear_model import LogisticRegression LogisticRegression().fit(X_train, y_train)  model = LogisticRegression().fit(X_train, y_train)  print(f"Regressi Logistik untuk data training dengan score: {format(model.score(X_train, y_train), '.4f')} ") print(f"Regressi Logistik untuk data test dengan score: {format(model.score(X_test, y_test), '.4f')} ")</pre>	<pre>Regressi Logistik untuk data training dengan score: 0.9670 Regressi Logistik untuk data test dengan score: 0.9035</pre>

- c. Hasil Visualisasi Gambar

Program
<pre>Bcancer_faktor = [x for i,x in enumerate(Bcancer.columns) if i!=8] plt.figure(figsize=(12,6)) #plt.plot(model.coef_.T, 'o', label="LogisticRegression()") #plt.plot(model_001.coef_.T, 'v', label="LogisticRegression(C=0.01)") plt.plot(model.coef_.T, 'o', color='violet', label="RegresiLogistik()") plt.xticks(range(Bcancer.shape[1]), Bcancer_faktor, rotation=90) plt.hlines(0, 0, Bcancer.shape[1],color='darkblue') plt.ylim(-3, 3) plt.title('Regresi Logistik') plt.xlabel("Faktor-faktor") plt.ylabel("Nilai koefisien") plt.legend() plt.show()</pre>



#### d. Analisa

Sebelum mengolah data menggunakan regresi logistik, terlebih dahulu kita memisahkan data antara set data yang harus dicapai (*data test*) dan set data yang bisa digunakan untuk mencapai tujuan (*data train*). Melalui perhitungan tersebut, didapatkan hasil bahwa ukuran dari data yang masih sama seperti perhitungan sebelumnya yaitu  $X_{train}$  adalah 455 baris dan 30 kolom sedangkan  $y_{train}$  sendiri berukuran 455 baris. Kemudian, untuk  $X_{test}$  terdiri dari 144 baris dan 30 kolom, dan  $y_{train}$  berukuran 144 baris. Pada perhitungan kali ini, juga digunakan *test size* sebesar 20% atau 0,2 dan *random state* sebesar 20. Nilai *random state* yaitu 20 sendiri menunjukan sumber ke generator acak atau *random generator*, yang akan membuat pemisahan data atau *split train-test data* akan selalu bersifat deterministik atau tidak akan berbeda setiap kali dihitung.

Setelah memisah data, barulah dilakukan regresi logistik, didapatkan hasil akurasi sebesar 0.9670 untuk *data training* dan 0.9035 untuk *data test*. Karena keduanya bernilai lebih dari 0,9, hal ini menandakan bahwa semua faktor mulai dari rata-rata radius (*radius mean*) sampai dimensi fraktal terburuk (*fractal dimension worst*) saling berkorelasi dan sangat mempengaruhi diagnosa kanker payudara ini.

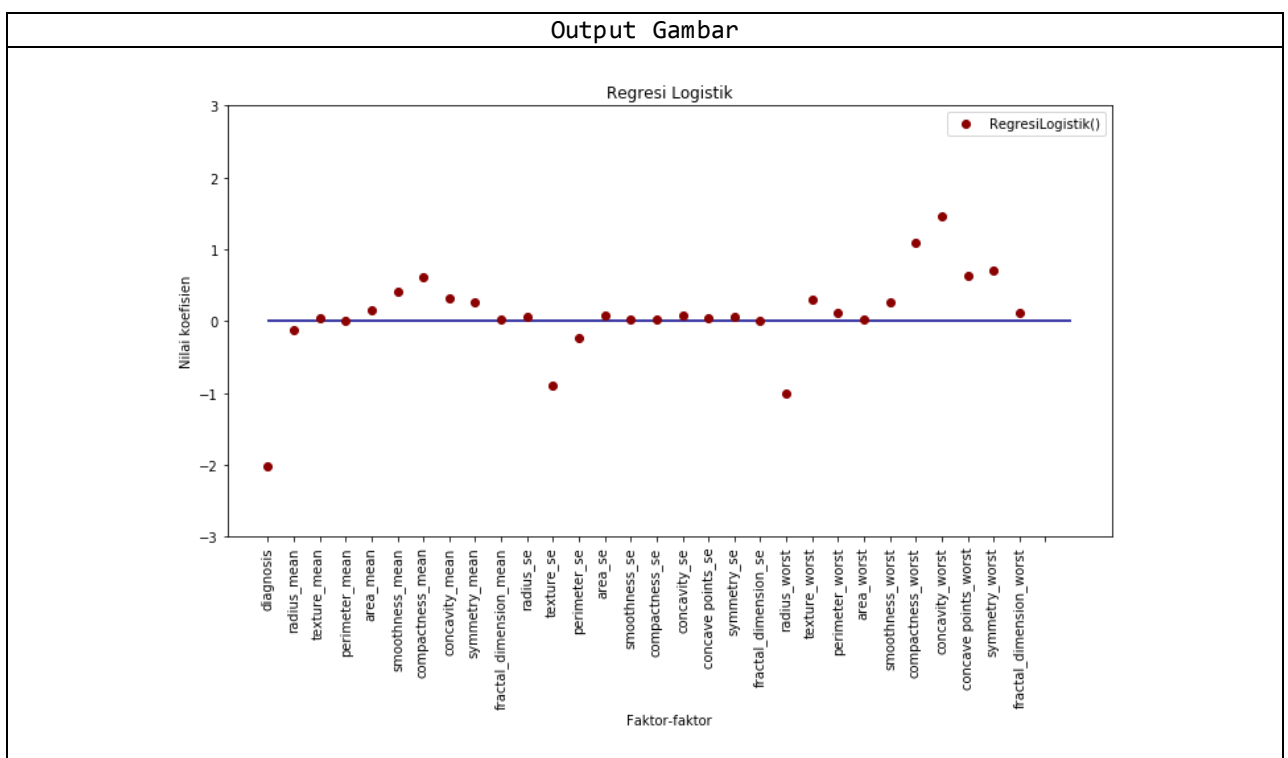
### 3. Random State : 50

- Data Test yang Digunakan : 0.2  
Random State yang Digunakan : 50
- Hasil Pemrograman

Program	Output
<pre>#memisahkan data train dan test from sklearn.model_selection import train_test_split X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 50) print(X_train.shape, y_train.shape) print(X_test.shape, y_test.shape)</pre>	<pre>(455, 30) (455,) (114, 30) (114,)</pre>
<pre>from sklearn.linear_model import LogisticRegression LogisticRegression().fit(X_train, y_train)  model = LogisticRegression().fit(X_train, y_train)  print(f"Regressi Logistik untuk data training dengan score: {format(model.score(X_train, y_train), '.4f')} ") print(f"Regressi Logistik untuk data test dengan score: {format(model.score(X_test, y_test), '.4f')} ")</pre>	<pre>Regressi Logistik untuk data training dengan score: 0.9538 Regressi Logistik untuk data test dengan score: 0.9649</pre>

- Hasil Visualisasi Gambar

Program
<pre>Bcancer_faktor = [x for i,x in enumerate(Bcancer.columns) if i!=8] plt.figure(figsize=(12,6)) #plt.plot(model.coef_.T, 'o', label="LogisticRegression()") #plt.plot(model_001.coef_.T, 'v', label="LogisticRegression(C=0.01)") plt.plot(model.coef_.T, 'o', color = 'darkred', label="RegresiLogistik()") plt.xticks(range(Bcancer.shape[1]), Bcancer_faktor, rotation=90) plt.hlines(0, 0, Bcancer.shape[1],color='darkblue') plt.ylim(-3, 3) plt.title('Regresi Logistik') plt.xlabel("Faktor-faktor") plt.ylabel("Nilai koefisien") plt.legend() plt.show()</pre>



d. Analisa

Sebelum mengolah data menggunakan regresi logistik, terlebih dahulu kita memisahkan data antara set data yang harus dicapai (*data test*) dan set data yang bisa digunakan untuk mencapai tujuan (*data train*). Melalui perhitungan tersebut, didapatkan hasil bahwa ukuran dari data yang masih sama seperti perhitungan sebelumnya yaitu *X train* adalah 455 baris dan 30 kolom sedangkan *y train* sendiri berukuran 455 baris. Kemudian, untuk *X test* terdiri dari 144 baris dan 30 kolom, dan *y train* berukuran 144 baris. Pada perhitungan kali ini, juga digunakan *test size* sebesar 20% atau 0,2 dan *random state* sebesar 50. Nilai *random state* yaitu 50 sendiri menunjukan sumber ke generator acak atau *random generator*, yang akan membuat pemisahan data atau *split train-test data* akan selalu bersifat deterministik atau tidak akan berbeda setiap kali dihitung.

Setelah memisah data, barulah dilakukan regresi logistik, didapatkan hasil akurasi sebesar 0.9538 untuk *data training* dan 0.9649 untuk *data test*. Karena keduanya bernilai lebih dari 0,9, hal ini menandakan bahwa semua faktor mulai dari rata-rata radius (*radius mean*) sampai dimensi fraktal terburuk (*fractal dimention worst*) saling berkorelasi dan sangat mempengaruhi diagnosa kanker payudara ini.