INSCOM Data Mining Competition 2023

Implementasi Algoritma Decision Tree Sebagai Metode Penyeleksian Penerimaan Beasiswa pada Perguruan Tinggi ABC

18 Maret 2023

iSeng Team

Our Team

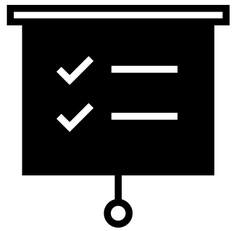
Sella Rikha Yasmin

Yuswar Aditya Permana

Isro' Rizky Wibowo

Muhammad Khusni Fikri

Overview



- Business Understanding
- Data Understanding
- Data Preparation
- Data Modeling
- Data Visualization
- Evaluation

Business Understanding

Seleksi Beasiswa

Perguruan Tinggi ABC melakukan seleksi penerima beasiswa dengan mengolah data yang ada. Banyak faktor yang mempengaruhi penerimaan beasiswa seperti, prestasi, ekonomi dan faktor lainya.

Masalah

Penyeleksian bersifat manual

Penyeleksian bersifat manual mengakibatkan pemilihan kandidat penerima yang tidak tepat sasaran.

Solusi

Machine Learning

Dengan Algoritma Decision Tree, diharapkan dapat membantu memilih kandidat dengan tepat sasaran.

Data Understanding

Read Data

```
data = pd.read_csv("Dataset.csv")
data
```

	NIM	Program Studi	Semester	IPK	Angkatan (Tahun Masuk)	Pekerjaan Ayah	Pekerjaan Ibu	Penghasilan Kotor Ayah/perbulan	Penghasilan Kotor Ibu/perbulan	Jumlah Anggota Keluarga Yang ditanggung	Label
0	12019130002	S1 Matematika	3	3.85	2019	Wiraswasta	Ibu Rumah Tangga	3000000	0	3	T
1	1520182015	D3 Kebidanan	5	3.59	2018	Wiraswasta	Ibu Rumah Tangga	1000000	0	2	T
2	2020181004	D3 Keperawatan	5	2.98	2018	Wiraswasta	Ibu Rumah Tangga	1000000	1000000	1	Y
3	52019050043	S1 Farmasi	3	3.10	2019	Wiraswasta	Ibu Rumah Tangga	3000000	0	1	Y
4	F420185015	S1 Farmasi	5	3.06	2018	Tenaga Honorer	Ibu Rumah Tangga	1500000	0	1	Y
...	...	...	...	...	...	...	...	...	...	...	...
772	212019010027	D3 Keperawatan	3	3.35	2019	Petani	Guru	100000	100000	3	T
773	12019120003	S1 Pgsd	3	3.55	2019	Buruh Harian Lepas	Wiraswasta	0	1000000	1	T
774	112019030093	S1 Keperawatan	3	2.60	2019	Guru	Karyawan Swasta	1000000	500000	5	T
775	1020183110	S1 Keperawatan	5	2.85	2018	Tidak Bekerja	PNS	0	3000000	3	T
776	212019010019	D3 Keperawatan	3	3.12	2019	Wiraswasta	Ibu Rumah Tangga	500000	500000	1	T

Data Explorations

```
[5]  #Data Informasi
     data.shape

     (777, 11)
```

```
[6]  #Data Informasi
     data.columns

     Index(['NIM', 'Program Studi', 'Semester', 'IPK', 'Angkatan (Tahun Masuk)',
            'Pekerjaan Ayah', 'Pekerjaan Ibu', 'Penghasilan Kotor Ayah/perbulan',
            'Penghasilan Kotor Ibu/perbulan',
            'Jumlah Anggota Keluarga Yang ditanggung', 'Label'],
           dtype='object')
```

Data Type

```
[4]  # Data Information
     data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 777 entries, 0 to 776
Data columns (total 11 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   NIM                                   777 non-null    object
 1   Program Studi                         777 non-null    object
 2   Semester                              777 non-null    int64
 3   IPK                                   777 non-null    float64
 4   Angkatan (Tahun Masuk)                777 non-null    int64
 5   Pekerjaan Ayah                        777 non-null    object
 6   Pekerjaan Ibu                         777 non-null    object
 7   Penghasilan Kotor Ayah/perbulan       777 non-null    int64
 8   Penghasilan Kotor Ibu/perbulan        777 non-null    int64
 9   Jumlah Anggota Keluarga Yang ditanggung  777 non-null  int64
 10  Label                                 777 non-null    object
dtypes: float64(1), int64(5), object(5)
memory usage: 66.9+ KB
```

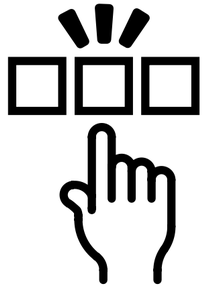
Data Describe

```
# Mendeskripsikan data yang terdapat pada dataset
data.describe()
```

	Semester	IPK	Angkatan (Tahun Masuk)	Penghasilan Kotor Ayah/perbulan	Penghasilan Kotor Ibu/perbulan	Jumlah Anggota Keluarga Yang ditanggung
count	777.000000	777.000000	777.000000	7.770000e+02	7.770000e+02	777.000000
mean	4.343629	3.169331	2018.333333	1.654835e+06	7.477323e+05	1.972973
std	1.463143	0.237392	0.732763	1.121924e+06	1.055638e+06	1.091457
min	3.000000	2.230000	2017.000000	0.000000e+00	0.000000e+00	0.000000
25%	3.000000	3.000000	2018.000000	1.000000e+06	0.000000e+00	1.000000
50%	5.000000	3.170000	2018.000000	1.500000e+06	4.000000e+05	2.000000
75%	5.000000	3.330000	2019.000000	2.000000e+06	1.000000e+06	2.000000
max	7.000000	3.850000	2020.000000	1.000000e+07	1.000000e+07	8.000000

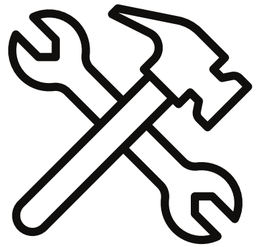
Data Preparation

Data Preparation

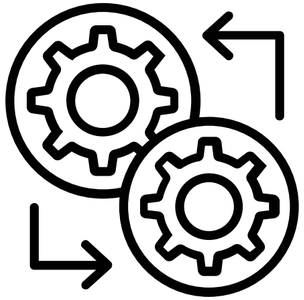


Select Data

Clean Data



Construct Data



Integrate Data

Select Data

```
data = pd.read_csv("Dataset.csv")
data
```

	NIM	Program Studi	Semester	IPK	Angkatan (Tahun Masuk)	Pekerjaan Ayah	Pekerjaan Ibu	Penghasilan Kotor Ayah/perbulan	Penghasilan Kotor Ibu/perbulan	Jumlah Anggota Keluarga Yang ditanggung	Label
0	12019130002	S1 Matematika	3	3.85	2019	Wiraswasta	Ibu Rumah Tangga	3000000	0	3	T
1	1520182015	D3 Kebidanan	5	3.59	2018	Wiraswasta	Ibu Rumah Tangga	1000000	0	2	T
2	2020181004	D3 Keperawatan	5	2.98	2018	Wiraswasta	Ibu Rumah Tangga	1000000	1000000	1	Y
3	52019050043	S1 Farmasi	3	3.10	2019	Wiraswasta	Ibu Rumah Tangga	3000000	0	1	Y
4	F420185015	S1 Farmasi	5	3.06	2018	Tenaga Honorer	Ibu Rumah Tangga	1500000	0	1	Y
...	...	...	...	...	...	...	...	...	...	...	...
772	212019010027	D3 Keperawatan	3	3.35	2019	Petani	Guru	100000	100000	3	T

Data Cleaning

```
# Menghitung banyaknya data yang null pada dataset
data.isna().sum()
```

```
NIM                                  0
Program Studi                        0
Semester                             0
IPK                                  0
Angkatan (Tahun Masuk)               0
Pekerjaan Ayah                       0
Pekerjaan Ibu                        0
Penghasilan Kotor Ayah/perbulan      0
Penghasilan Kotor Ibu/perbulan       0
Jumlah Anggota Keluarga Yang ditanggung    0
Label                                0
Total Penghasilan                    0
Alokasi Penghasilan per Anak         3
Kelayakan                            0
dtype: int64
```

```
data.dropna(how='any',inplace=True)
data
```

	NIM	Program Studi	Semester	IPK	Angkatan (Tahun Masuk)	Pekerjaan Ayah	Pekerjaan Ibu	Penghasilan Kotor Ayah/perbulan	Penghasilan Kotor Ibu/perbulan	Jumlah Anggota Keluarga Yang ditanggung	Label	Total Penghasilan
0	12019130002	S1 Matematika	3	3.85	2019	Wiraswasta	Ibu Rumah Tangga	3000000	0	3	T	3000000
1	1520182015	D3 Kebidanan	5	3.59	2018	Wiraswasta	Ibu Rumah Tangga	1000000	0	2	T	1000000
2	2020181004	D3 Keperawatan	5	2.98	2018	Wiraswasta	Ibu Rumah Tangga	1000000	1000000	1	Y	2000000
3	52019050043	S1 Farmasi	3	3.10	2019	Wiraswasta	Ibu Rumah Tangga	3000000	0	1	Y	3000000
4	F420185015	S1 Farmasi	5	3.06	2018	Tenaga Honorer	Ibu Rumah Tangga	1500000	0	1	Y	1500000
...	...	...	...	...	...	...	...	...	...	...	...	...

Drop Missing Value

```
[14] data["Jumlah Anggota Keluarga Yang ditanggung"].value_counts()[0]
```

```
23
```

```
[15] data.drop(data[data["Jumlah Anggota Keluarga Yang ditanggung"] == 0].index, inplace=True)
```

Construct Data

```
[8]  # Menambahkan kolom baru
     data['Total Penghasilan'] = data['Penghasilan Kotor Ayah/perbulan'] + data['Penghasilan Kotor Ibu/perbulan']
     data
```

	NIM	Program Studi	Semester	IPK	Angkatan (Tahun Masuk)	Pekerjaan Ayah	Pekerjaan Ibu	Penghasilan Kotor Ayah/perbulan	Penghasilan Kotor Ibu/perbulan	Jumlah Anggota Keluarga Yang ditanggung	Label	Total Penghasilan
0	12019130002	S1 Matematika	3	3.85	2019	Wiraswasta	Ibu Rumah Tangga	3000000	0	3	T	3000000
1	1520182015	D3 Kebidanan	5	3.59	2018	Wiraswasta	Ibu Rumah Tangga	1000000	0	2	T	1000000
2	2020181004	D3 Keperawatan	5	2.98	2018	Wiraswasta	Ibu Rumah Tangga	1000000	1000000	1	Y	2000000
3	52019050043	S1 Farmasi	3	3.10	2019	Wiraswasta	Ibu Rumah Tangga	3000000	0	1	Y	3000000
4	F420185015	S1 Farmasi	5	3.06	2018	Tenaga Honorer	Ibu Rumah Tangga	1500000	0	1	Y	1500000
...	...	...	...	...	...	...	...	...	...	...	...	...
772	212019010027	D3 Keperawatan	3	3.35	2019	Petani	Guru	100000	100000	3	T	200000
773	12019120003	S1 Pgsd	3	3.55	2019	Buruh Harian Lepas	Wiraswasta	0	1000000	1	T	1000000
774	112019030093	S1 Keperawatan	3	2.60	2019	Guru	Karyawan Swasta	1000000	500000	5	T	1500000

```
[9]  # Menambahkan kolom baru
     data['Alokasi Penghasilan per Anak'] = data['Total Penghasilan'] / data['Jumlah Anggota Keluarga Yang ditanggung']
     data['Alokasi Penghasilan per Anak']
```

```
0      1.000000e+06
1      5.000000e+05
2      2.000000e+06
3      3.000000e+06
4      1.500000e+06
          ...
772    6.666667e+04
773    1.000000e+06
774    3.000000e+05
775    1.000000e+06
776    1.000000e+06
Name: Alokasi Penghasilan per Anak, Length: 777, dtype: float64
```

Construct Data

```
data_selected.describe()
```

	Semester	IPK	Angkatan (Tahun Masuk)	Penghasilan Kotor Ayah/perbulan	Penghasilan Kotor Ibu/perbulan	Jumlah Anggota Keluarga Yang ditanggung	Total Penghasilan	Alokasi Penghasilan per Anak
count	751.000000	751.000000	751.000000	7.510000e+02	7.510000e+02	751.000000	7.510000e+02	7.510000e+02
mean	4.371505	3.165539	2018.319574	1.670449e+06	7.428602e+05	2.041278	2.413309e+06	1.410615e+06
std	1.467583	0.233650	0.735121	1.125155e+06	1.060901e+06	1.045448	1.602281e+06	1.145741e+06
min	3.000000	2.230000	2017.000000	0.000000e+00	0.000000e+00	1.000000	0.000000e+00	0.000000e+00
25%	3.000000	3.000000	2018.000000	1.000000e+06	0.000000e+00	1.000000	1.500000e+06	7.500000e+05
50%	5.000000	3.160000	2018.000000	1.500000e+06	4.000000e+05	2.000000	2.000000e+06	1.000000e+06
75%	5.000000	3.330000	2019.000000	2.000000e+06	1.000000e+06	3.000000	3.000000e+06	1.925000e+06

Construct Data

Klasifikasi Kelayakan Menggunakan Nilai Mean

```
kelayakan = []

for x in data["Alokasi Penghasilan per Anak"]:
  if x > 1410614: # Nilai Mean Alokasi Penghasilan per Anak
    kelayakan.append('0') #NON BEASISWA
  else:
    kelayakan.append('1') #BEASISWA (1)

data["Kelayakan"] = kelayakan
data["Kelayakan"]
```

```
0      1
1      1
2      0
3      0
4      0
```

Integrate Data

```
[ ]  selected = ["Semester","IPK", "Angkatan (Tahun Masuk)","Penghasilan Kotor Ayah/perbulan", "Penghasilan Kotor Ibu/perbulan"
                ,"Jumlah Anggota Keluarga Yang ditanggung","Total Penghasilan","Alokasi Penghasilan per Anak","Kelayakan"]
     data_selected = data[selected]

     X = data_selected.drop(columns="Kelayakan").values # atribut
     Y = data_selected["Kelayakan"].values #kelas/target
```

Data Modeling

Membagi Testing dan Membangun Model

```
[19] #Import library yang dibutuhkan
     from sklearn import model_selection

     #Mendefinisikan ukuran testing data dan seed untuk random state
     test_size = 0.30
     seed = 7
     X_train,X_test,y_train,y_test = model_selection.train_test_split(X,Y,test_size = test_size,random_state=seed)
```

Membangun Skenario Model

```
[21] #Melakukan import beberapa library berisi algoritma klasifikasi yang akan digunakan
     from sklearn.tree import DecisionTreeClassifier
     from sklearn.svm import SVC
     from sklearn.naive_bayes import GaussianNB
     from sklearn.tree import DecisionTreeClassifier
     from sklearn.neighbors import KNeighborsClassifier
     from sklearn.linear_model import LogisticRegression

     #Membuat array kosong untuk menampung beberapa algoritma klasifikasi
     models = []

     #Membuat array asosiatif berisi nama algoritma dan algoritma klasifikasinya
     models.append(('Decision Tree',DecisionTreeClassifier()))
     models.append(('SVM',SVC()))
     models.append(('Gaussian Naive Bayes',GaussianNB()))
     models.append(('KNN',KNeighborsClassifier(n_neighbors = 7, metric = 'euclidean')))
```

Membangun dan Menguji Model

Membangun dan Menguji Model

```
#Mendefinisikan ssed dan scoring yang digunakan untuk menguji model dengan validation data
seed = 7
scoring = 'accuracy'
results = []
names = []

#Membagi data menjadi traning dan validation set menggunakan k-fold cross validation, dengan k = 10
for name, model in models:
  kfold = model_selection.KFold(n_splits=7, random_state = seed, shuffle = True)

  #Menguji akurasi dari masing-masing model menggunakan validasi data
  cv_results = model_selection.cross_val_score(model,x_train,y_train.ravel(),cv=kfold,scoring = scoring)
  results.append(cv_results)
  msg = "%s : %f (%f)" % (name,cv_results.mean(),cv_results.std())
  print(msg)
```

```
Decision Tree : 1.000000 (0.000000)
SVM : 0.975238 (0.014998)
Gaussian Naive Bayes : 0.942857 (0.023328)
KNN : 0.923810 (0.034706)
```

```
#Menggunakan Decision Tree Classifier dengan kriteria splitting Gini impurity untuk membentuk model
clf_gini = DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=None)


# fit model
clf_gini.fit(X_train, y_train)
```

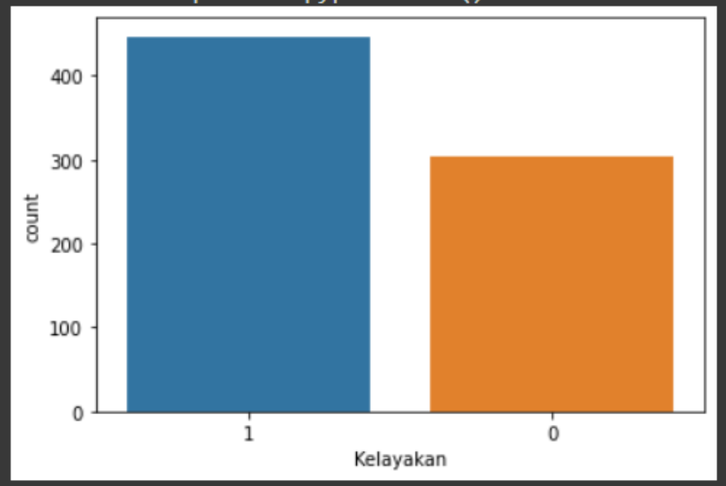
```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3)
```

Data Visualization

Data Visualization

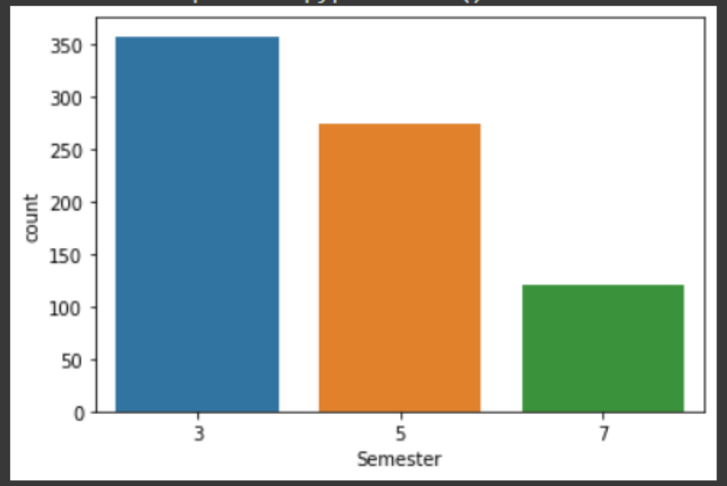
```
[30] sns.countplot(data['Kelayakan'])
     plt.ioff
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/_decorators.py:36:
  warnings.warn(
<function matplotlib.pyplot.ioff()>
```




```
[29] sns.countplot(data['Semester'])
     plt.ioff
```

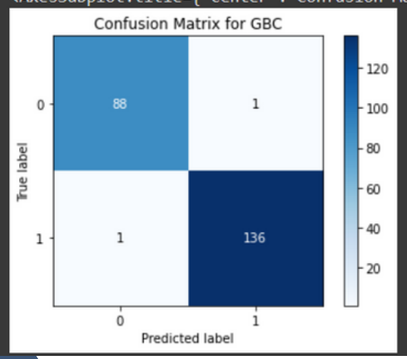
```
/usr/local/lib/python3.9/dist-packages/seaborn/_decorators.py:36:
  warnings.warn(
<function matplotlib.pyplot.ioff()>
```

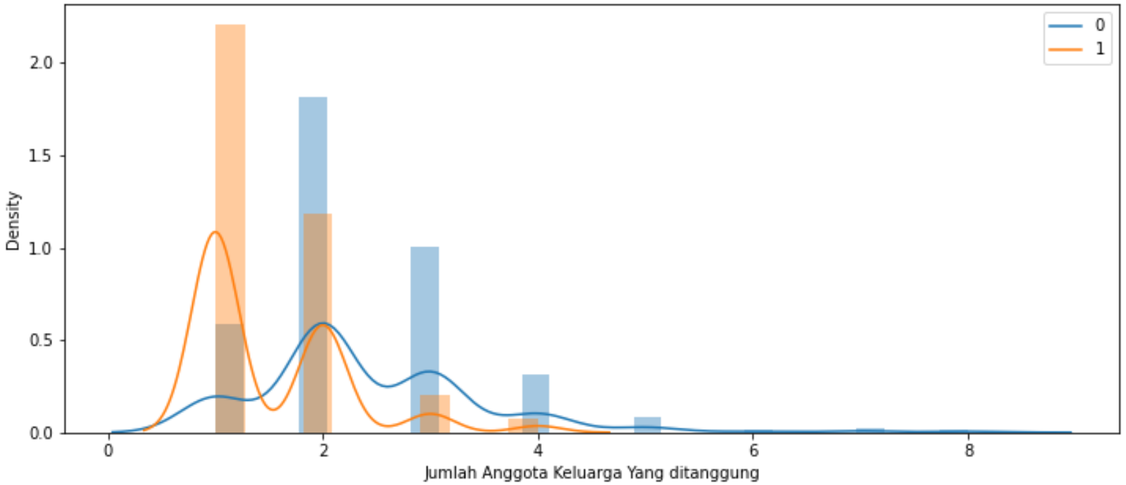


Data Visualization

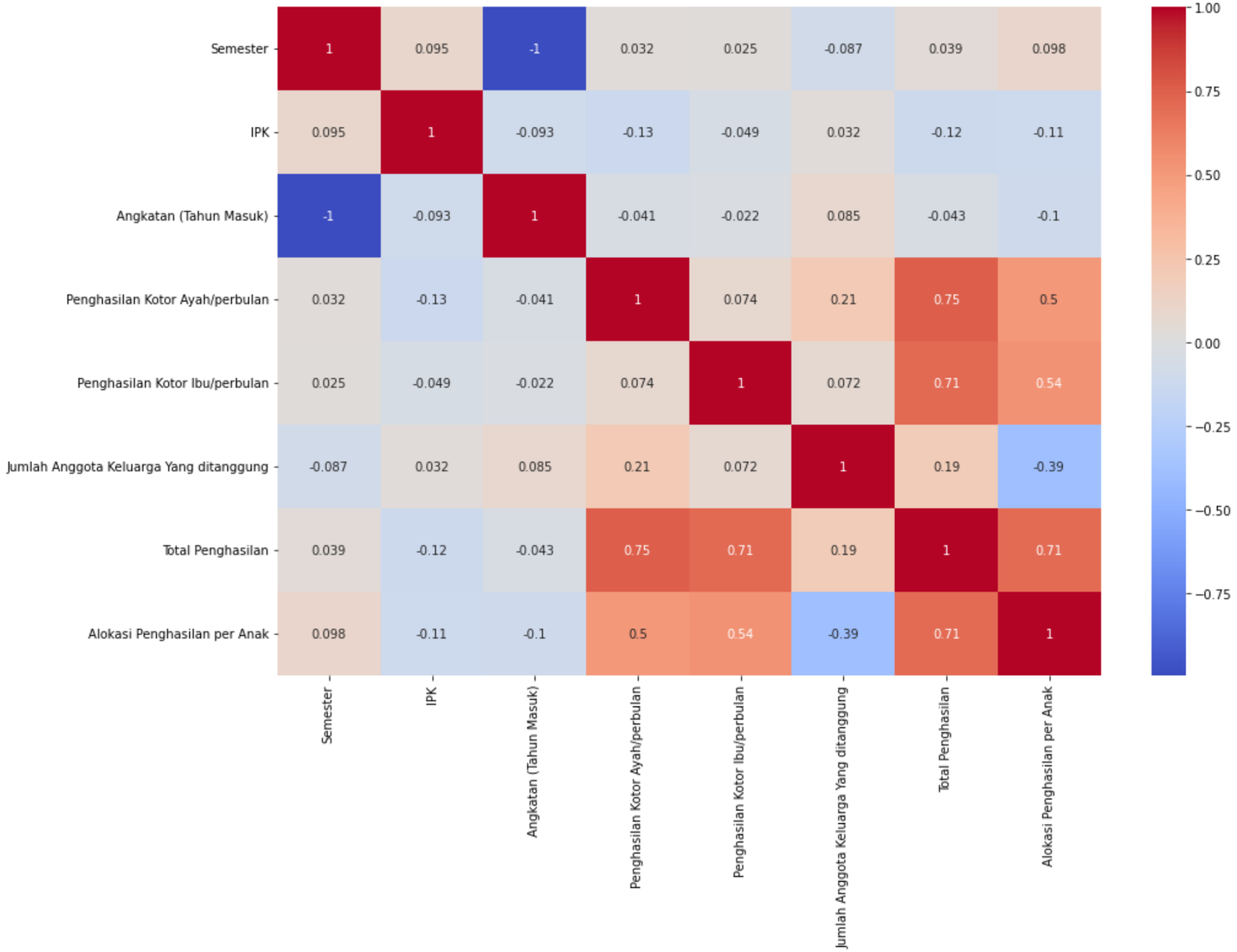
```
[26] from sklearn.metrics import confusion_matrix
     import scikitplot as skplt
     cm = confusion_matrix(y_test,y_pred)
     print(cm)
     skplt.metrics.plot_confusion_matrix(y_test, y_pred, normalize=False, title = 'Confusion Matrix for GBC')
```

```
[[ 88   1]
 [  1 136]]
<AxesSubplot:title={'center':'Confusion Matrix for GBC'}, xlabel='Predicted label', ylabel='True label'>
```





Correlation Map



Evaluation

Evaluation Result

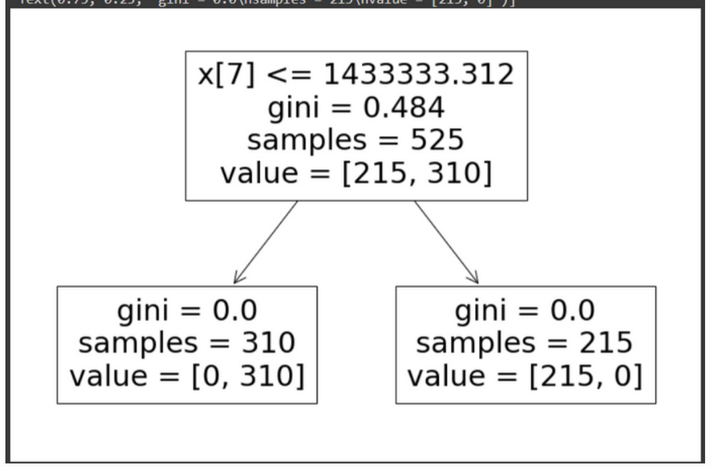
Mengevaluasi Hasil Permodelan

```
[35] matrix = classification_report(y_test,y_pred,labels=[1,0])
     print('<--------------- Classification report --------------->\n', matrix)
```

```
     <--------------- Classification report --------------->
                   precision    recall  f1-score   support

                1       0.99      0.99      0.99       137
                0       0.99      0.99      0.99        89

        micro avg       0.99      0.99      0.99       226
        macro avg       0.99      0.99      0.99       226
     weighted avg       0.99      0.99      0.99       226
```

```
#Plot the tree
plt.figure(figsize=(12,8))

tree.plot_tree(clf_gini.fit(X_train, y_train))
```

```
[Text(0.5, 0.75, 'x[7] <= 1433333.312\ngini = 0.484\nsamples = 525\nvalue = [215, 310]'),
 Text(0.25, 0.25, 'gini = 0.0\nsamples = 310\nvalue = [0, 310]'),
 Text(0.75, 0.25, 'gini = 0.0\nsamples = 215\nvalue = [215, 0]')]
```




```
[36] # informasi yang didapatkan dari klasifikasi
     from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score, confusion_matrix, roc_curve, roc_auc_score
     hasil = classifier.predict(x_test)
     print('accuracy score :',accuracy_score(hasil,y_test))
     print('Precision Score :',precision_score(y_test, hasil, pos_label='1'))
     print('Recall score :',recall_score(y_test, hasil,pos_label='1'))
     print('F1 score :',f1_score(y_test, hasil,pos_label='1'))
```

```
accuracy score : 0.9911504424778761
Precision Score : 0.9927007299270073
Recall score : 0.9927007299270073
F1 score : 0.9927007299270073
```

Kesimpulan

Berdasarkan hasil penelitian yang telah dipaparkan pada bab sebelumnya, maka dapat ditarik kesimpulan bahwa model Decision Tree yang telah dibuat dan diterapkan pada dataset mendapatkan hasil,

- Accuracy = 0.991,
- Recall = 0.992,
- Precision = 0.992,
- dan F1 Score = 0.992.

Berdasarkan hasil tersebut, membuktikan bahwa dengan menggunakan algoritma Decision Tree mendapatkan nilai evaluasi performa yang tinggi sehingga dapat digunakan untuk melakukan penyeleksian terhadap mahasiswa yang berhak mendapatkan beasiswa.