Algoritma Tree Decision

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

In [2]:
```
# read file
df = pd.read_csv('[Dataset]_Train_(Keryawan).csv')
df.head()
```

Out[2]:

	Employee_ID	Gender	Age	Education_Level	Relationship_Status	Hometown	Unit	Decisi
0	EID_23371	F	42.0	4	Married	Franklin	IT	
1	EID_18000	M	24.0	3	Single	Springfield	Logistics	
2	EID_3891	F	58.0	3	Married	Clinton	Quality	
3	EID_17492	F	26.0	3	Single	Lebanon	Human Resource Management	
4	EID_22534	F	31.0	1	Married	Springfield	Logistics	

5 rows × 24 columns

Preprocess Data

In [3]:
```
#data tidak lengkap
nan_data = df[df.isna().any(axis=1)]
nan_data
```

Out[3]:

	Employee_ID	Gender	Age	Education_Level	Relationship_Status	Hometown	Unit	De
3	EID_17492	F	26.0	3	Single	Lebanon	Human Resource Management	
7	EID_1235	F	NaN	3	Married	Springfield	Sales	
8	EID_10197	M	40.0	4	Single	Springfield	Production	
15	EID_20121	F	NaN	3	Married	Springfield	Logistics	
19	EID_12947	M	32.0	3	Single	Lebanon	IT	
...	...	...	...	...	...	...	...	
6969	EID_18566	F	NaN	1	Single	Washington	R&D	
6975	EID_2706	F	52.0	3	Married	Clinton	R&D	
6976	EID_15099	M	28.0	4	Married	Franklin	Operarions	

	Employee_ID	Gender	Age	Education_Level	Relationship_Status	Hometown	Unit	De
6981	EID_25181	M	NaN	3	Married	Springfield	Logistics	
6986	EID_17099	M	NaN	4	Single	Franklin	Human Resource Management	

1647 rows × 24 columns

In [4]:
```
# delete data tidak lengkap
df = df.dropna()
df
```

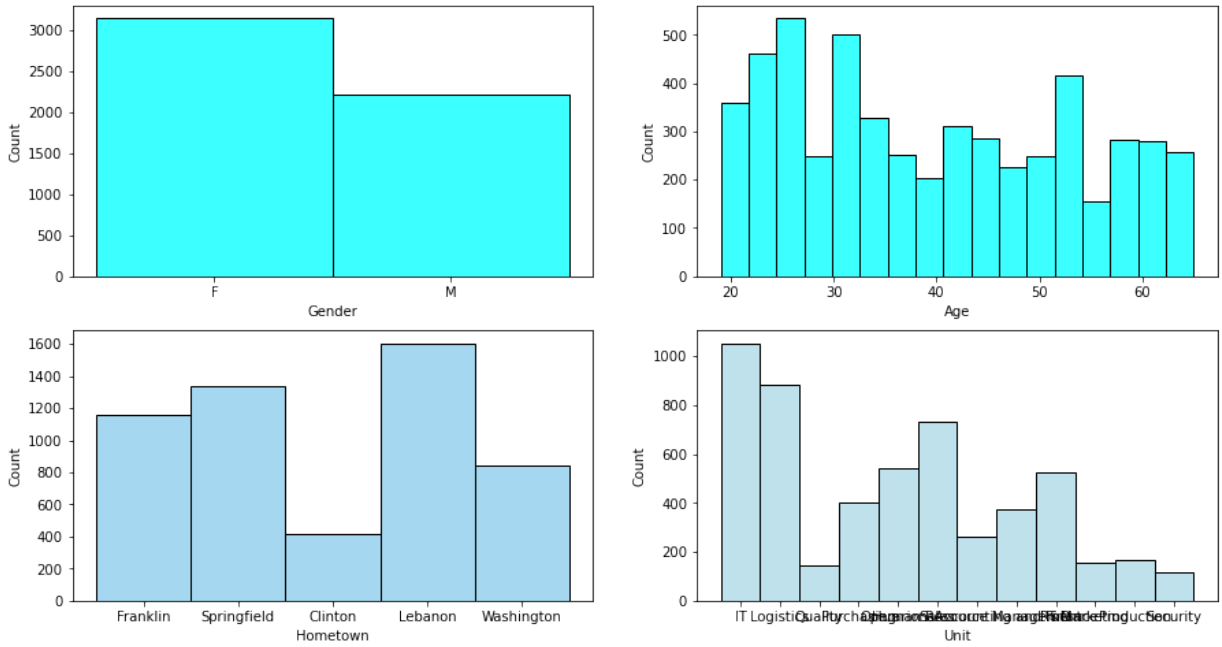
Out[4]:

	Employee_ID	Gender	Age	Education_Level	Relationship_Status	Hometown	Unit	Deci
0	EID_23371	F	42.0	4	Married	Franklin	IT	
1	EID_18000	M	24.0	3	Single	Springfield	Logistics	
2	EID_3891	F	58.0	3	Married	Clinton	Quality	
4	EID_22534	F	31.0	1	Married	Springfield	Logistics	
5	EID_2278	M	54.0	3	Married	Lebanon	Purchasing	
...	...	...	...	...	...	...	...	
6995	EID_16328	F	23.0	5	Married	Franklin	Operarions	
6996	EID_8387	F	44.0	1	Married	Lebanon	R&D	
6997	EID_8077	F	49.0	3	Single	Springfield	IT	
6998	EID_19597	F	47.0	3	Married	Washington	Sales	
6999	EID_1640	F	58.0	3	Married	Franklin	IT	

5353 rows × 24 columns

In [5]:
```
# Visualisasi data
fig,axes = plt.subplots(2,2,figsize=(15,8))

sns.histplot(data=df,x='Gender',ax=axes[0,0],color='aqua')
sns.histplot(data=df,x='Age',ax=axes[0,1],color='cyan')
sns.histplot(data=df,x='Hometown',ax=axes[1,0],color='skyblue')
sns.histplot(data=df,x='Unit',ax=axes[1,1],color='lightblue')
plt.show()
```



In [6]:
```
from sklearn.preprocessing import LabelEncoder
encode = LabelEncoder()
```

In [7]:
```
# Transformasi
df['Gender'] = encode.fit_transform(df['Gender'].values)
df['Relationship_Status'] = encode.fit_transform(df['Relationship_Status'].values)
df['Hometown'] = encode.fit_transform(df['Hometown'].values)
df['Unit'] = encode.fit_transform(df['Unit'].values)
df['Decision_skill_possess'] = encode.fit_transform(df['Decision_skill_possess'].val
df['Compensation_and_Benefits'] = encode.fit_transform(df['Compensation_and_Benefits
```

In [8]:
```
#hapus kolom tidak perlu
df = df.drop(['Employee_ID'],axis=1)
df = df.drop(['Attrition_rate'],axis=1)
```

Normalisasi

In [9]:
```
from sklearn.preprocessing import MinMaxScaler
kolom = [col for col in df.columns]
scaler = MinMaxScaler()
scaled = scaler.fit_transform(df[kolom])
df_scaled = pd.DataFrame(scaled,columns=kolom)
```

In [10]:
```
df_scaled
```

Out[10]:

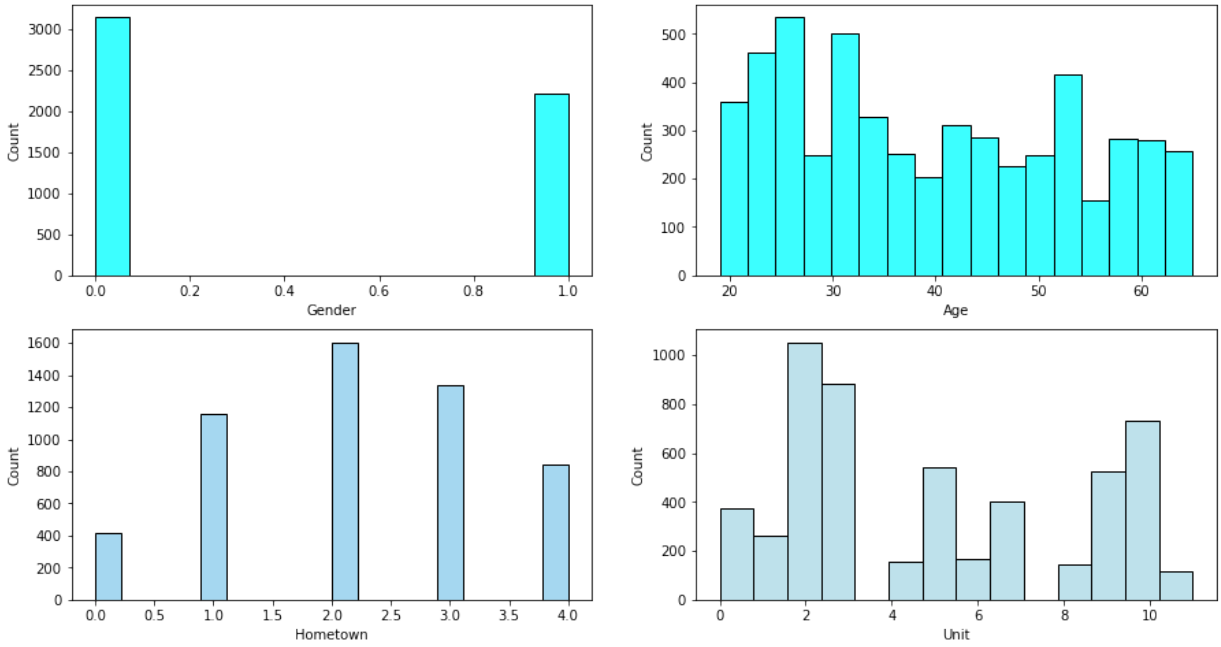
	Gender	Age	Education_Level	Relationship_Status	Hometown	Unit	Decision_skill_pos
0	0.0	0.500000	0.75	0.0	0.25	0.181818	0.66
1	1.0	0.108696	0.50	1.0	0.75	0.272727	0.00
2	0.0	0.847826	0.50	0.0	0.00	0.727273	0.66
3	0.0	0.260870	0.00	0.0	0.75	0.272727	0.66
4	1.0	0.760870	0.50	0.0	0.50	0.636364	0.66

	Gender	Age	Education_Level	Relationship_Status	Hometown	Unit	Decision_skill_pos
...	...	...	...	...	...	...	
5348	0.0	0.086957	1.00	0.0	0.25	0.454545	0.33
5349	0.0	0.543478	0.00	0.0	0.50	0.818182	0.00
5350	0.0	0.652174	0.50	1.0	0.75	0.181818	1.00
5351	0.0	0.608696	0.50	0.0	1.00	0.909091	0.33
5352	0.0	0.847826	0.50	0.0	0.25	0.181818	1.00

5353 rows × 22 columns

In [11]:
```
#Visualisasi data
fig,axes = plt.subplots(2,2,figsize=(15,8))

sns.histplot(data=df,x='Gender',ax=axes[0,0],color='aqua')
sns.histplot(data=df,x='Age',ax=axes[0,1],color='cyan')
sns.histplot(data=df,x='Hometown',ax=axes[1,0],color='skyblue')
sns.histplot(data=df,x='Unit',ax=axes[1,1],color='lightblue')
plt.show()
```



In [12]:
```
#input test data
test_data = pd.read_csv('[Dataset]_Test_(Karyawan).csv')
```

In [13]:
```
#pre processing
test_data = test_data.drop(['Employee_ID'],axis=1)
test_data = test_data.dropna()
#encode
test_data['Gender'] = encode.fit_transform(test_data['Gender'].values)
test_data['Relationship_Status'] = encode.fit_transform(test_data['Relationship_Stat
test_data['Hometown'] = encode.fit_transform(test_data['Hometown'].values)
test_data['Unit'] = encode.fit_transform(test_data['Unit'].values)
test_data['Decision_skill_possess'] = encode.fit_transform(test_data['Decision_skill_
test_data['Compensation_and_Benefits'] = encode.fit_transform(test_data['Compensatio
#normalisasi
```

```
test_data_kolom = [col for col in test_data.columns]
test_data_scaled = scaler.fit_transform(test_data[test_data_kolom])
test_data = pd.DataFrame(test_data_scaled,columns=test_data_kolom)
test_data
```

Out[13]:

	Gender	Age	Education_Level	Relationship_Status	Hometown	Unit	Decision_skill_pos
0	0.0	0.282609	1.00	1.0	0.75	0.818182	0.66
1	1.0	1.000000	0.25	1.0	0.50	0.181818	1.00
2	1.0	0.717391	0.50	0.0	0.75	0.909091	1.00
3	1.0	0.673913	1.00	1.0	1.00	0.363636	0.00
4	0.0	0.543478	0.50	0.0	0.25	0.818182	0.66
...	...	...	...	...	...	...	
2316	1.0	0.608696	0.50	0.0	0.25	0.454545	0.33
2317	0.0	0.347826	0.75	1.0	1.00	0.363636	0.66
2318	0.0	0.282609	0.50	1.0	0.25	0.909091	1.00
2319	0.0	0.695652	0.00	0.0	0.75	0.181818	0.33
2320	0.0	0.565217	0.00	1.0	0.25	0.909091	1.00

2321 rows × 22 columns

In [14]:

```
# Splitting Data
X_train, X_test, y_train, y_test = train_test_split(df_scaled, df_scaled['Relationsh
```

In [15]:

```
# Build Model
model = DecisionTreeClassifier(random_state=42)
```

In [16]:

```
# Train Model
model.fit(X_train, y_train)
```

Out[16]:

```
DecisionTreeClassifier(random_state=42)
```

Evaluasi kinerja model dengan metrik

In [17]:

```
# Evaluate Model
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
conf_matrix = confusion_matrix(y_test, y_pred)
print(f'Confusion Matrix:\n{conf_matrix}')
class_report = classification_report(y_test, y_pred)
print(f'Classification Report:\n{class_report}')
```

```
Accuracy: 1.0
Confusion Matrix:
[[677   0]
 [  0 394]]
Classification Report:
```

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	677
1.0	1.00	1.00	1.00	394
accuracy			1.00	1071
macro avg	1.00	1.00	1.00	1071
weighted avg	1.00	1.00	1.00	1071

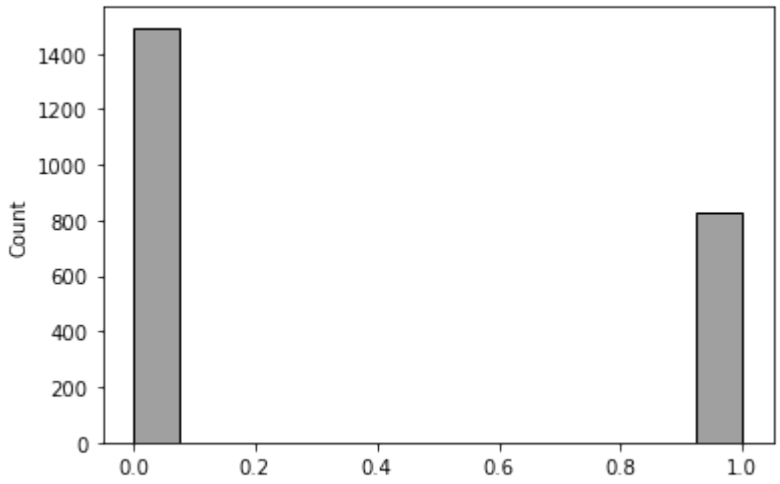
Uji Coba Model

In [18]:
```
# Make predictions
new_prediction = model.predict(test_data)
print(f'New Prediction: {new_prediction}')
```

```
New Prediction: [1. 1. 0. ... 1. 0. 1.]
```

In [22]:
```
sns.histplot(data=new_prediction,color='gray')
```

Out[22]:
```
<AxesSubplot:ylabel='Count'>
```



In [ ]: