

## LAMPIRAN

### Lampiran 1. Syntax Pemodelan *Random Forest* Menggunakan *Python*

```
In [4]: import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle

from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report, f1_score, accuracy_score, recall_score, precision_score
```

```
In [5]: dataframe = pd.read_excel('Data Awal.xlsx')
dataframe.head()
```

Out[5]:

	Nama	Email	Alamat Domisili	Provinsi Domisili	Kota Domisili	Kecamatan Domisili	Kelurahan Domisili	Alamat Ktp	Provinsi Ktp	Kota Ktp	...	Kemampuan 1 (Komunikasi)	Kemampuan 2 (Negotiasi)
0	Indriyani	yaniindri8298@gmail.com	Jl. Duri B VII No. 7 RT/003 RW/004	DKI Jakarta	Kota Jakarta Pusat	Gambir	Duri Pulo	Jl. Duri B VII No. 7 RT/003 RW/004	DKI Jakarta	Kota Jakarta Pusat	...	Iya	Iya
1	Indriyani	indriyani.indri294@gmail.com	Serua Residence Blok A1 E1 Jalan Cinarangka Boj...	Jawa Barat	Kota Depok	Bojongsari	Pondok Petir	Cidodol RT 013 / RW 006 Grogol Selatan Kebayor...	DKI Jakarta	Kota Jakarta Selatan	...	Iya	Iya
2	Rima Rahmawati	rimarahma@gmail.com	Jl. Bendungan Hilir No. 13, Ket. Bendungan HI...	Banten	Kota Tangerang Selatan	Ciputat	Ciputat	Jl. Kusuma Bhakti No. 49, Kel. Kubu Galai Banc...	Sumatera Barat	Kota Bukittinggi	...	Tidak	Tidak
3	Siti nuraisyah	ainursiti14@gmail.com	Kp bambu baru, RT013/003	Banten	Kab. Tangerang	Pakuheji	Rawa Bori	Kp bambu baru, RT013/003	Banten	Kab. Tangerang	...	Iya	Iya
4	Fitriah	Fitriahh14@gmail.com	Kp.keranggan	Banten	Kota Tangerang Selatan	Selu	Keranggan	Kp.keranggan	Banten	Kota Tangerang Selatan	...	Iya	Iya

5 rows x 35 columns

```
In [6]: dataframe.shape
```

Out[6]: (1155, 35)

```
In [7]: dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1155 entries, 0 to 1154
Data columns (total 35 columns):
#   Column              Non-Null Count  Dtype
---  ---
0   Nama                 1155 non-null   object
1   Email                1155 non-null   object
2   Alamat Domisili      1121 non-null   object
3   Provinsi Domisili    1094 non-null   object
4   Kota Domisili        1094 non-null   object
5   Kecamatan Domisili  553 non-null    object
6   Kelurahan Domisili  551 non-null    object
7   Alamat Ktp           1187 non-null   object
8   Provinsi Ktp         1061 non-null   object
9   Kota Ktp             1061 non-null   object
10  Kecamatan Ktp        535 non-null    object
11  Kelurahan Ktp        534 non-null    object
12  Tempat Lahir         1124 non-null   object
13  Tanggal Lahir        1124 non-null   object
14  Usia                 1121 non-null   float64
15  Telp 1                1154 non-null   object
16  Telp 2                454 non-null    object
17  Tingkat Pendidikan  1143 non-null   object
18  Institut Pendidikan  1143 non-null   object
19  Bidang Pendidikan    1143 non-null   object
20  Tahun Lulus          1143 non-null   float64
21  Nilai Akhir          1143 non-null   float64
22  Posisi Yang Dilamar  1155 non-null   object
23  Level                 1155 non-null   object
24  Pengalaman Pekerjaan 1155 non-null   object
25  Kemampuan 1 (Komunikasi) 1155 non-null   object
26  Kemampuan 2 (Negosiasi) 1155 non-null   object
27  Kemampuan 3 (IT Umum) 1155 non-null   object
28  Tanggal Daftar       1155 non-null   object
29  Bulan Daftar         1155 non-null   object
30  Jenis Kelamin        1155 non-null   object
31  Gaji Yang Diharapkan 644 non-null    object
32  Status Kelengkapan  1155 non-null   object
33  Agama                1125 non-null   object
34  Tahapan              1155 non-null   object
dtypes: float64(3), object(32)
memory usage: 315.9+ KB
```

```
In [8]: dataframe.isnull().sum()
```

```
Out[8]: Nama 0
Email 0
Alamat Domisili 34
Provinsi Domisili 61
Kota Domisili 61
Kecamatan Domisili 602
Kelurahan Domisili 604
Alamat Ktp 48
Provinsi Ktp 94
Kota Ktp 94
Kecamatan Ktp 620
Kelurahan Ktp 621
Tempat Lahir 31
Tanggal Lahir 31
Usia 34
Telp 1 1
Telp 2 701
Tingkat Pendidikan 12
Institut Pendidikan 12
Bidang Pendidikan 12
Tahun Lulus 12
Nilai Akhir 12
Posisi Yang Dilamar 0
Level 0
Pengalaman Pekerjaan 0
Kemampuan 1 (Komunikasi) 0
Kemampuan 2 (Negosiasi) 0
Kemampuan 3 (IT Umum) 0
Tanggal Daftar 0
Bulan Daftar 0
Jenis Kelamin 0
Gaji Yang Diharapkan 511
Status Kelengkapan 0
Agama 30
Tahapan 0
dtype: int64
```

```
In [9]: df = dataframe.drop(['Nama', 'Email', 'Alamat Domisili', 'Provinsi Domisili', 'Kecamatan Domisili', 'Kelurahan Domisili', 'Alamat Domisili']).head()
```

```
Out[9]:
```

	Kota Domisili	Usia	Tingkat Pendidikan	Posisi Yang Dilamar	Pengalaman Pekerjaan	Kemampuan 1 (Komunikasi)	Kemampuan 2 (Negosiasi)	Kemampuan 3 (IT Umum)	Jenis Kelamin	Status Kelengkapan	Tahapan
0	Kota Jakarta Pusat	24.0	S1	Telemarketing	Ada	Iya	Iya	Tidak	Perempuan	Lengkap	Ditolak
1	Kota Depok	22.0	S1	Telemarketing	Ada	Iya	Tidak	Tidak	Perempuan	Tidak Lengkap	Ditolak
2	Kota Tangerang Selatan	24.0	S1	Telemarketing	Ada	Tidak	Tidak	Tidak	Perempuan	Lengkap	Ditolak
3	Kab. Tangerang	25.0	SMA/SMK	Telesales	Ada	Iya	Iya	Iya	Perempuan	Lengkap	Diajukan
4	Kota Tangerang Selatan	24.0	SMA/SMK	Sales Marketing	Ada	Iya	Iya	Iya	Perempuan	Lengkap	Diajukan

```
In [10]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1155 entries, 0 to 1154
Data columns (total 11 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Kota Domisili       1094 non-null   object
1   Usia                1121 non-null   float64
2   Tingkat Pendidikan 1143 non-null   object
3   Posisi Yang Dilamar 1155 non-null   object
4   Pengalaman Pekerjaan 1155 non-null   object
5   Kemampuan 1 (Komunikasi) 1155 non-null  object
6   Kemampuan 2 (Negosiasi) 1155 non-null  object
7   Kemampuan 3 (IT Umum) 1155 non-null  object
8   Jenis Kelamin       1155 non-null  object
9   Status Kelengkapan 1155 non-null  object
10  Tahapan             1155 non-null  object
dtypes: float64(1), object(10)
memory usage: 99.4+ KB
```

```
In [11]: df.isnull().sum().sort_values(ascending=False)
```

```
Out[11]: Kota Domisili 61
Usia 34
Tingkat Pendidikan 12
Posisi Yang Dilamar 0
Pengalaman Pekerjaan 0
Kemampuan 1 (Komunikasi) 0
Kemampuan 2 (Negosiasi) 0
Kemampuan 3 (IT Umum) 0
Jenis Kelamin 0
Status Kelengkapan 0
Tahapan 0
dtype: int64
```

```
In [12]: present = df['Kota Domisili'].isna().mean()*100
print("Presentase data yang hilang: {:.2f}%".format(present))
```

```
Presentase data yang hilang: 5.28%
```

```
In [13]: df['Kota Domisili'].fillna(df['Kota Domisili'].mode()[0],inplace=True)
df['Usia'].fillna(df['Usia'].mean(),inplace=True)
df['Tingkat Pendidikan'].fillna(df['Tingkat Pendidikan'].mode()[0],inplace=True)
df.isnull().sum().sort_values(ascending=False)
```

```
Out[13]: Kota Domisili      0
Usia                      0
Tingkat Pendidikan       0
Posisi Yang Dilamar      0
Pengalaman Pekerjaan     0
Kemampuan 1 (Komunikasi) 0
Kemampuan 2 (Negosiasi)  0
Kemampuan 3 (IT Umum)    0
Jenis Kelamin            0
Status Kelengkapan      0
Tahapan                  0
dtype: int64
```

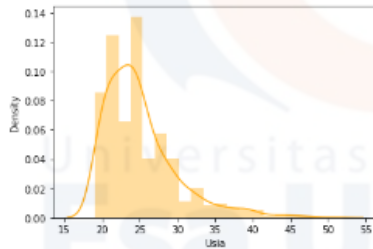
```
In [14]: for col in df:
print(df[col].value_counts())
```

```
Name: Pengalaman Pekerjaan, dtype: int64
Iya      859
Tidak   296
Name: Kemampuan 1 (Komunikasi), dtype: int64
Iya      806
Tidak   349
Name: Kemampuan 2 (Negosiasi), dtype: int64
Iya      605
Tidak   550
Name: Kemampuan 3 (IT Umum), dtype: int64
Perempuan 697
Laki-Laki 458
Name: Jenis Kelamin, dtype: int64
Lengkap  1015
Tidak Lengkap 140
Name: Status Kelengkapan, dtype: int64
Ditolak  653
Diajukan 502
Name: Tahapan, dtype: int64
```

```
In [15]: np.sort(df['Usia'].unique())
```

```
Out[15]: array([[19. , 20. , 21. , 22. , 23. ,
24. , 25. , 25.05530776, 26. , 27. ,
28. , 29. , 30. , 31. , 32. ,
33. , 34. , 35. , 36. , 37. ,
38. , 39. , 40. , 41. , 42. ,
43. , 44. , 45. , 46. , 47. ,
48. , 51. ]])
```

```
In [16]: sn.distplot(df['Usia'],bins=20,color='orange')
plt.show()
```



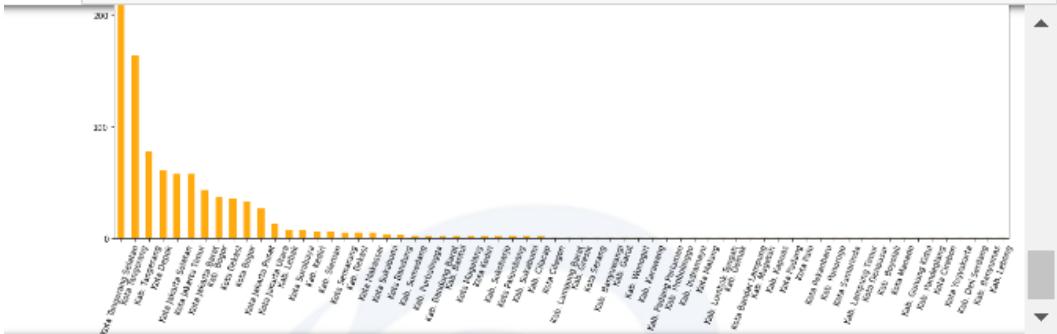
```
In [17]: df['Usia'] = df['Usia'].replace((19.0, 20.0, 21.0, 22.0, 23.0, 24.0, 25.0, 25.055307760927743, 26.0, 27.0, 28.0, 29.0, 30.0), '1')
df['Usia'] = df['Usia'].replace((31.0, 32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 39.0, 40.0, 41.0, 42.0, 43.0, 44.0, 45.0, 46.0), '2')
```

```
Out[17]: 0      <30
1      <30
2      <30
3      <30
4      <30
...
1150   <30
1151   <30
1152   <30
1153   <30
1154   <30
Name: Usia, Length: 1155, dtype: object
```

```
In [18]: np.sort(df['Kota Domisili'].unique())
```

```
Out[18]: array(['Kab. Bandung Barat', 'Kab. Bantul', 'Kab. Banyumas',  
'Kab. Banyuwangi', 'Kab. Bekasi', 'Kab. Bogor', 'Kab. Boyolali',  
'Kab. Cilacap', 'Kab. Deli Serdang', 'Kab. Demak', 'Kab. Garut',  
'Kab. Gresik', 'Kab. Gunung Kidul', 'Kab. Indramayu',  
'Kab. Kapuas', 'Kab. Karawang', 'Kab. Kediri',  
'Kab. Lampung Barat', 'Kab. Lampung Timur', 'Kab. Lebak',  
'Kab. Lebong', 'Kab. Lombok Tengah', 'Kab. Magetan',  
'Kab. Padang Pariaman', 'Kab. Pandeglang', 'Kab. Ponorogo',  
'Kab. Probolinggo', 'Kab. Purbalingga', 'Kab. Sleman',  
'Kab. Sukabumi', 'Kab. Sukoharjo', 'Kab. Sumedang',  
'Kab. Tangerang', 'Kab. Wonogiri', 'Kota Bandar Lampung',  
'Kota Bandung', 'Kota Bekasi', 'Kota Bogor', 'Kota Cilegon',  
'Kota Cirebon', 'Kota Denpasar', 'Kota Depok',  
'Kota Jakarta Barat', 'Kota Jakarta Pusat', 'Kota Jakarta Selatan',  
'Kota Jakarta Timur', 'Kota Jakarta Utara', 'Kota Kediri',  
'Kota Magelang', 'Kota Makassar', 'Kota Malang', 'Kota Manado',  
'Kota Padang', 'Kota Palembang', 'Kota Palu', 'Kota Pekanbaru',  
'Kota Samarinda', 'Kota Semarang', 'Kota Serang', 'Kota Sukabumi',  
'Kota Surabaya', 'Kota Tangerang', 'Kota Tangerang Selatan',  
'Kota Yogyakarta'], dtype=object)
```

```
In [19]: df['Kota Domisili'].value_counts().plot.bar(figsize=(20,12), color='orange')  
plt.xticks(rotation=70)
```



```
In [20]: df['Kota Domisili'] = df['Kota Domisili'].replace(('Kota Tangerang Selatan', 'Kota Tangerang', 'Kab. Tangerang'), 'Wilayah Tange  
df['Kota Domisili'] = df['Kota Domisili'].replace(('Kota Depok', 'Kota Jakarta Selatan', 'Kota Jakarta Timur',  
'Kota Jakarta Barat', 'Kota Bekasi', 'Kab. Bogor', 'Kota Bogor',  
'Kota Jakarta Pusat', 'Kota Jakarta Utara', 'Kab. Lebak',  
'Kota Surabaya', 'Kab. Kediri', 'Kab. Sleman', 'Kota Semarang',  
'Kab. Bekasi', 'Kota Makassar', 'Kota Sukabumi', 'Kab. Sumedang',  
'Kab. Purbalingga', 'Kab. Bandung Barat', 'Kab. Bantul',  
'Kota Magelang', 'Kota Kediri', 'Kab. Sukoharjo', 'Kota Bandung',  
'Kota Palembang', 'Kab. Sukabumi', 'Kab. Cilacap', 'Kota Cilegon',  
'Kab. Lampung Barat', 'Kab. Gresik', 'Kota Serang', 'Kab. Banyuwangi',  
'Kab. Garut', 'Kab. Wonogiri', 'Kab. Karawang', 'Kab. Padang Pariaman',  
'Kab. Probolinggo', 'Kab. Indramayu', 'Kota Malang',  
'Kab. Lombok Tengah', 'Kab. Demak', 'Kota Bandar Lampung',  
'Kab. Magetan', 'Kab. Kapuas', 'Kota Padang', 'Kota Palu',  
'Kota Pekanbaru', 'Kab. Ponorogo', 'Kota Samarinda',  
'Kab. Lampung Timur', 'Kota Denpasar', 'Kab. Boyolali', 'Kota Manado',  
'Kab. Gunung Kidul', 'Kab. Pandeglang', 'Kota Cirebon',  
'Kota Yogyakarta', 'Kab. Deli Serdang', 'Kab. Banyumas', 'Kab. Lebong'), 'Luar Wilayah Tangerang')
```

```
df['Kota Domisili']
```

```
Out[20]: 0    Luar Wilayah Tangerang  
1    Luar Wilayah Tangerang  
2    Wilayah Tangerang  
3    Wilayah Tangerang  
4    Wilayah Tangerang  
...  
1150   Wilayah Tangerang  
1151   Wilayah Tangerang  
1152   Luar Wilayah Tangerang  
1153   Luar Wilayah Tangerang  
1154   Luar Wilayah Tangerang  
Name: Kota Domisili, Length: 1155, dtype: object
```

In [21]: df.head()

Out[21]:

	Kota Domisili	Usia	Tingkat Pendidikan	Posisi Yang Dilamar	Pengalaman Pekerjaan	Kemampuan 1 (Komunikasi)	Kemampuan 2 (Negosiasi)	Kemampuan 3 (IT Umum)	Jenis Kelamin	Status Kelengkapan	Tahapan
0	Luar Wilayah Tangerang	<30	S1	Telemarketing	Ada	Iya	Iya	Tidak	Perempuan	Lengkap	Ditolak
1	Luar Wilayah Tangerang	<30	S1	Telemarketing	Ada	Iya	Tidak	Tidak	Perempuan	Tidak Lengkap	Ditolak
2	Wilayah Tangerang	<30	S1	Telemarketing	Ada	Tidak	Tidak	Tidak	Perempuan	Lengkap	Ditolak
3	Wilayah Tangerang	<30	SMA/SMK	Telesales	Ada	Iya	Iya	Iya	Perempuan	Lengkap	Diajukan
4	Wilayah Tangerang	<30	SMA/SMK	Sales Marketing	Ada	Iya	Iya	Iya	Perempuan	Lengkap	Diajukan

In [22]: for col in df:

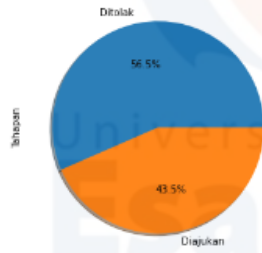
```
print(df[col].value_counts())
```

```
Wilayah Tangerang      689
Luar Wilayah Tangerang  466
Name: Kota Domisili, dtype: int64
<30      1018
>30      137
Name: Usia, dtype: int64
SMA/SMK    815
S1         269
D3         71
Name: Tingkat Pendidikan, dtype: int64
Telesales   1021
Telemarketing  110
Sales Marketing  24
Name: Posisi Yang Dilamar, dtype: int64
Ada         976
Tidak ada   179
Name: Pengalaman Pekerjaan, dtype: int64
Iya         859
Tidak      296
Name: Kemampuan 1 (Komunikasi), dtype: int64
Iya         806
Tidak      349
Name: Kemampuan 2 (Negosiasi), dtype: int64
Iya         605
Tidak      550
Name: Kemampuan 3 (IT Umum), dtype: int64
Perempuan   697
Laki-Laki   458
Name: Jenis Kelamin, dtype: int64
Lengkap    1015
Tidak Lengkap  140
Name: Status Kelengkapan, dtype: int64
Ditolak    653
Diajukan   502
Name: Tahapan, dtype: int64
```

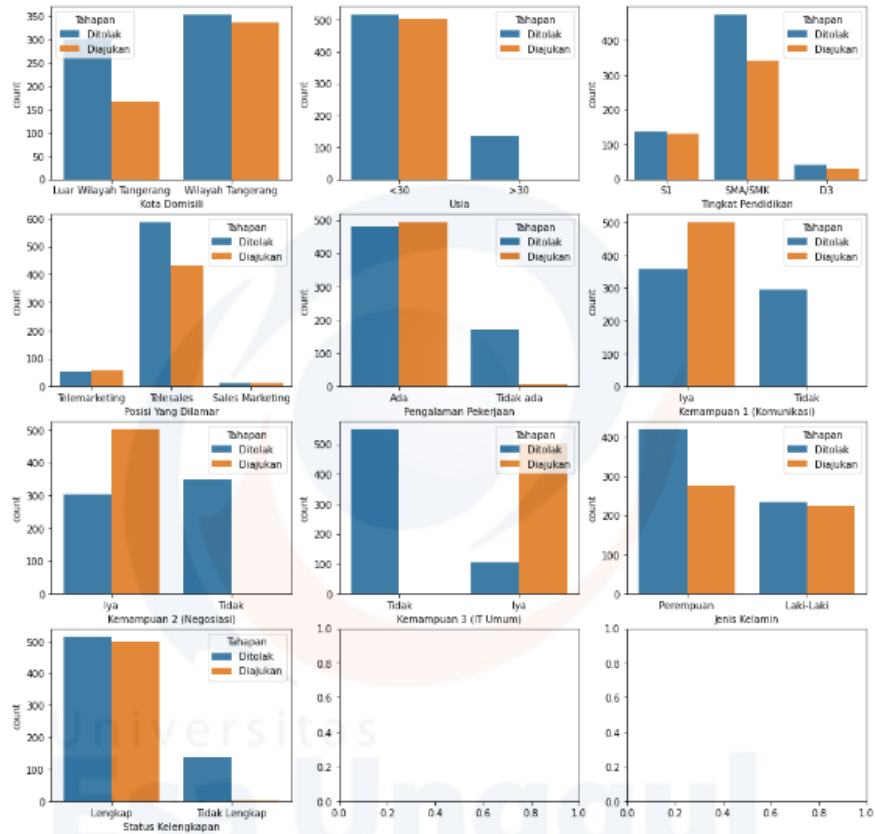
In [23]:

```
fig = plt.subplots(figsize=(15,5))
df['Tahapan'].value_counts().plot.pie(autopct='%1.1f%%', shadow=True)
```

Out[23]: <AxesSubplot:ylabel='Tahapan'>



```
In [24]: cat_var = ['Kota Domisili', 'Usia', 'Tingkat Pendidikan', 'Posisi Yang Dilamar', 'Pengalaman Pekerjaan', 'Kemampuan 1 (Komunikas  
fig, axes = plt.subplots(4, 3, figsize=(15, 15))  
for cat, ax in zip(cat_var, axes.flatten()):  
    sn.countplot(cat, data=df, hue="Tahapan", ax=ax)
```



```
In [25]: obj_kd = {'Luar Wilayah Tangerang':1, 'Wilayah Tangerang':2}
df['Kota Domisili'] = df['Kota Domisili'].replace(obj_kd)

# encode column usia
obj_usia = {'>30':1, '<30':2}
df['Usia'] = df['Usia'].replace(obj_usia)

# encode column pengalaman pekerjaan
obj_pp = {'Tidak ada':1, 'Ada':2}
df['Pengalaman Pekerjaan'] = df['Pengalaman Pekerjaan'].replace(obj_pp)

# encode column jenis kelamin
obj_jk = {'Perempuan':1, 'Laki-Laki':2}
df['Jenis Kelamin'] = df['Jenis Kelamin'].replace(obj_jk)

# encode column status kelengkapan
obj_sk = {'Tidak Lengkap':1, 'Lengkap':2}
df['Status Kelengkapan'] = df['Status Kelengkapan'].replace(obj_sk)

# encode column tahapan
obj_tahapan = {'Ditolak':1, 'Diajukan':2, }
df['Tahapan'] = df['Tahapan'].replace(obj_tahapan)

# encode column tahapan
obj_skill1 = {'Tidak':1, 'Iya':2}
df['Kemampuan 1 (Komunikasi)'] = df['Kemampuan 1 (Komunikasi)'].replace(obj_skill1)

# encode column tahapan
obj_skill2 = {'Tidak':1, 'Iya':2}
df['Kemampuan 2 (Negosiasi)'] = df['Kemampuan 2 (Negosiasi)'].replace(obj_skill2)

# encode column tahapan
obj_skill3 = {'Tidak':1, 'Iya':2}
df['Kemampuan 3 (IT Umum)'] = df['Kemampuan 3 (IT Umum)'].replace(obj_skill3)

obj_tkpd = {'SMA/SMK':1, 'D3':2, 'S1':3}
df['Tingkat Pendidikan'] = df['Tingkat Pendidikan'].replace(obj_tkpd)

obj_posisi = {'Telesales':1, 'Telemarketing':2, 'Sales Marketing':3}
df['Posisi Yang Dilamar'] = df['Posisi Yang Dilamar'].replace(obj_posisi)
```

```
In [26]: df.head()
```

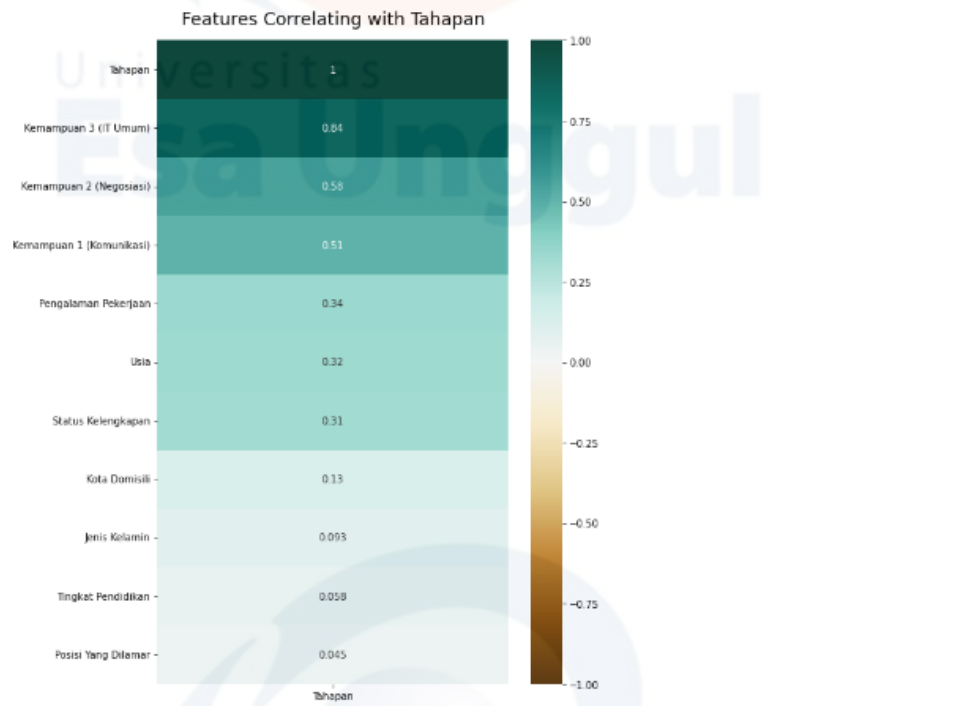
Out[26]:

	Kota Domisili	Usia	Tingkat Pendidikan	Posisi Yang Dilamar	Pengalaman Pekerjaan	Kemampuan 1 (Komunikasi)	Kemampuan 2 (Negosiasi)	Kemampuan 3 (IT Umum)	Jenis Kelamin	Status Kelengkapan	Tahapan
0	1	2	3	2	2	2	2	1	1	2	1
1	1	2	3	2	2	2	1	1	1	1	1
2	2	2	3	2	2	1	1	1	1	2	1
3	2	2	1	1	2	2	2	2	1	2	2
4	2	2	1	3	2	2	2	2	1	2	2

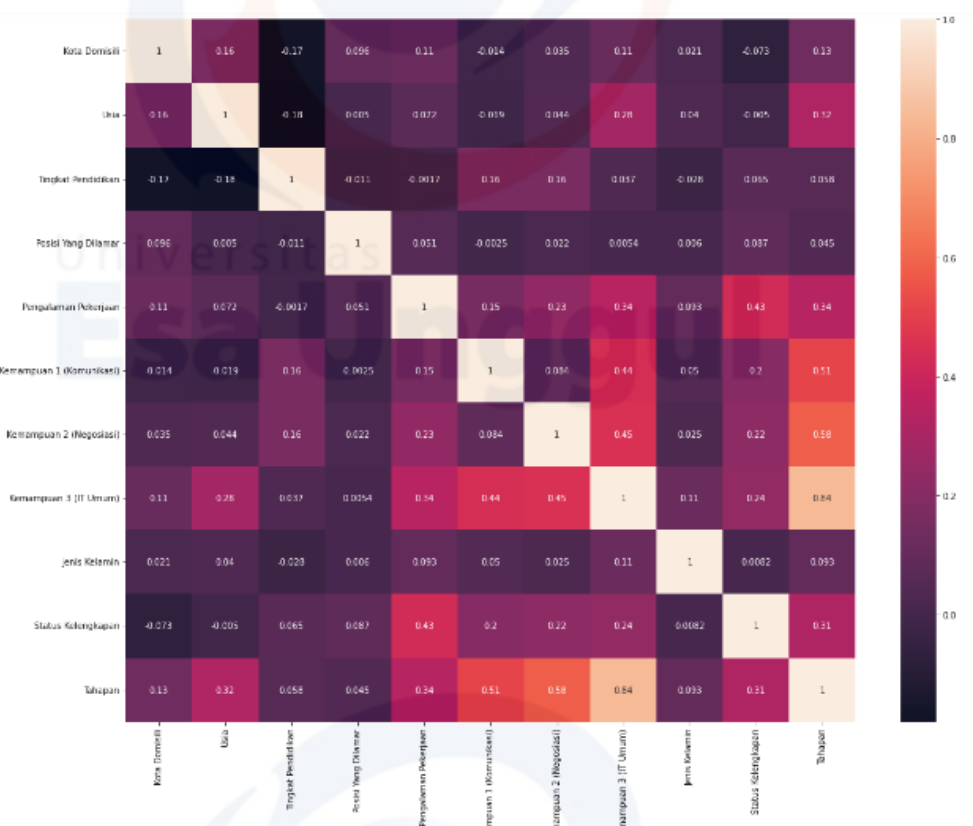
```
In [27]: for col in df:
print(df[col].value_counts())
```

```
2    689
1    466
Name: Kota Domisili, dtype: int64
2    1018
1    137
Name: Usia, dtype: int64
1    815
3    269
2    71
Name: Tingkat Pendidikan, dtype: int64
1    1021
2    110
3    24
Name: Posisi Yang Dilamar, dtype: int64
2    976
1    179
Name: Pengalaman Pekerjaan, dtype: int64
2    859
1    296
Name: Kemampuan 1 (Komunikasi), dtype: int64
2    886
1    349
Name: Kemampuan 2 (Negosiasi), dtype: int64
2    685
1    550
Name: Kemampuan 3 (IT Umum), dtype: int64
1    697
2    458
Name: Jenis Kelamin, dtype: int64
2    1015
1    148
Name: Status Kelengkapan, dtype: int64
1    653
2    502
Name: Tahapan, dtype: int64
```

```
In [28]: plt.figure(figsize=(8, 12))
heatmap = sn.heatmap(df.corr()[['Tahapan']].sort_values(by='Tahapan', ascending=False), vmin=-1, vmax=1, annot=True, cmap='BrBG')
heatmap.set_title('Features Correlating with Tahapan', fontdict={'fontsize':18}, pad=16);
```



```
In [29]: plt.figure(figsize=(19, 15))
corrMatrix = df.corr()
sn.heatmap(corrMatrix, annot=True)
plt.show()
```







In [48]: CV\_rfc.best\_score\_

Out[48]: 0.9950617283950617

```
In [38]: clf = RandomForestClassifier(random_state=42, max_features='auto', n_estimators= 50, max_depth=4, criterion='entropy')
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
print("Accuracy Score:", accuracy_score(y_test,y_pred))
print("Recall Score:", recall_score(y_test, y_pred))
print("Precision Score:", precision_score(y_test, y_pred))
```

```
[[204  1]
 [ 0 142]]
```

	precision	recall	f1-score	support	
1	1.00	1.00	1.00	205	
2	0.99	1.00	1.00	142	
accuracy				1.00	347
macro avg				1.00	347
weighted avg				1.00	347

Accuracy Score: 0.9971181556195965  
Recall Score: 0.9951219512195122  
Precision Score: 1.0

```
In [34]: cm_matrix = pd.DataFrame(data=confusion_matrix(y_test, y_pred),
                                columns=['Predict Positive:1',
                                         'Predict Negative:0'],
                                index=['Actual Positive:1',
                                       'Actual Negative:0'])
cm_matrix
```

Out[34]:

	Predict Positive:1	Predict Negative:0
Actual Positive:1	204	1
Actual Negative:0	0	142

In [35]: sn.heatmap(cm\_matrix,annot=True,fmt='d')

Out[35]: <AxesSubplot:>





## Deploy Flask Script (app.py)

```
1 import pandas as pd      You, 3 hours ago • first commit .
2 from jcopml.utils import load_model
3 from flask import Flask, render_template, request
4
5 app = Flask(__name__)
6 model = load_model("model/model_datapelamar.pkl")
7
8 @app.route("/", methods=["GET", "POST"])
9 def index():
10     if request.method == "GET":
11         return render_template("upload.html")
12     elif request.method == "POST":
13         excel_file = request.files.get("file")
14         X_test = pd.read_excel(excel_file)
15         X_test["Tahapan"] = model.predict(X_test)
16         return X_test.to_html()
17
18 if __name__ == "__main__":
19     app.run(debug=True)
```

## Deploy Aplikasi Menggunakan Heroku

```
-----> Building on the Heroku-20 stack
-----> Using buildpack: heroku/python
-----> Python app detected
-----> No Python version was specified. Using the same version as the last build: python-3.10.6
    To use a different version, see: https://devcenter.heroku.com/articles/python-runtimes
-----> No change in requirements detected, installing from cache
-----> Using cached install of python-3.10.6
-----> Installing pip 22.1.2, setuptools 60.10.0 and wheel 0.37.1
-----> Installing SQLite3
-----> Installing requirements with pip
-----> Discovering process types
    Procfile declares types -> web
-----> Compressing...
    Done: 256.4M
-----> Launching...
    Released v11
    https://ta-datapelamar.herokuapp.com/ deployed to Heroku
This app is using the Heroku-20 stack, however a newer stack is available.
To upgrade to Heroku-22, see:
https://devcenter.heroku.com/articles/upgrading-to-the-latest-stack
Build finished
```