

DAFTAR PUSTAKA

- Ambarwari, A., Jafar Adrian, Q., Herdiyeni, Y., 2020. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi) 4, 117–122.
- Azeroual, O., 2020. Data 5, 1–9.
- BPS, 2019. Berita Resmi Statistik Keadaan Ketenagakerjaan Indonesia.pdf.
- Chawla, N. v., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Journal of Artificial Intelligence Research 16, 321–357.
- Cybertech Ltd, 2020. Navicat Premium [WWW Document]. PremiumSoft Cybertech Ltd.
- Digmi, I., 2018. Google Colab Gratis Untuk Belajar Deep Learning [WWW Document]. imam.digmi.id. URL <https://imam.digmi.id/post/google-colab-gratis-untuk-belajar-deep-learning/>
- Direktorat Kelembagaan Kemdikbud, 2020. Panduan API Laporan Kerjasama PT [WWW Document]. URL <https://documenter.getpostman.com/view/2615716/TzsYPVQp>
- Direktorat Pembelajaran dan Kemahasiswaan, 2020. tracer study [WWW Document]. Kemdikbudristek. URL <http://tracerstudy.kemdikbud.go.id/>
- Dirjen Dikti Kemdikbud, 2018. Pedoman Teknis Penyelenggaraan Program Kerja Sama Perguruan Tinggi Indonesia Dengan Perguruan Tinggi Atau Lembaga Lain Di Dalam Dan Luar Negeri, Angewandte Chemie International Edition, 6(11), 951–952. Indonesia.
- Dirjen Dikti Kemdikbud, 2020. Buku Panduan Merdeka Belajar - Kampus Merdeka. Indonesia.

- Faqih, F., 2018. Belajar Python Dasar : Memahami Jupyter Notebook dan cara menggunakannya [WWW Document]. Ngodingdata.
- Feldman, D., Gross, S., 2004. SSRN Electronic Journal.
- Gök, E.C., Olgun, M.O., 2021. Neural Computing and Applications 33, 15693–15707.
- Goldman, Ian. and Pabari, M., 2021. 6.
- Goller, C., Löning, J., Will, T., Wolff, W., 2000. Internationalen Symposiums für Informationswissenschaft 8, 145–162.
- Hayaty, M., Muthmainah, S., Ghufran, S.M., 2021. International Journal of Artificial Intelligence Research 4, 86.
- Indonesia, P.R., 2012. Undang-undang Republik Indonesia Nomor 12. Indonesia.
- Katore, L.S., Ratnaparkhi, B.S., Umale, J.S., 2015. Global Conference on Communication Technologies, GCCT 2015 503–506.
- Kementerian Pendidikan, R. dan T., 2016. Peraturan Menteri Ristekdikti nomor 61 tahun 2016 tentang Pangkalan Data Pendidikan Tinggi. Indonesia.
- Kusrini, Taufiq Luthfi, E., 2019. Algoritma Data Mining. Andi Offset.
- Muhson, A., Wahyuni, D., Mulyani, E., 2012. Jurnal Economia 8, 42–52.
- Mukherjee, M., Khushi, M., 2021. Applied System Innovation 4, 1–15.
- Nugroho, A.S., Witarto, A.B., Handoko, D., 2003. Support Vector Machine - Teori dan Aplikasinya dalam Bioinformatika, in: Ilmu Komputer. pp. 842–847.

- Patil, M.M., Hiremath, B.N., 2018. International Journal of Information Technology and Computer Science 10, 32–39.
- Rangnekar, R.H., Suratwala, K.P., Krishna, S., Dhage, S., 2018a. Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018 1–6.
- Rangnekar, R.H., Suratwala, K.P., Krishna, S., Dhage, S., 2018b. Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018 1–6.
- Santosa, B., 2007. Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis. Garah Ilmu, Yogyakarta.
- Saputra Ginting, V., Taufiq Luthfi, E., 2020. Jurnal Teknologi Informasi 4.
- Sinaga, A.R., 2012. Jurnal Pelita Informatika Budi Darma 4.
- Tair, M.M.A., El-halees, A.M., 2012. International Journal of Information and Communication technology Research 2, 140–146.
- Tembusai, Z.R., Mawengkang, H., Zarlis, M., 2021. International Journal of Advances in Data and Information Systems 2.
- Untoro, M.C., Buliali, J.L., 2018. Register: Jurnal Ilmiah Teknologi Sistem Informasi 4, 23–29.
- Virmani, D., Taneja, S., Malhotra, G., n.d. Normalization based K means Clustering Algorithm.
- Wibowo, P., Fatichah, C., 2021. Register: Jurnal Ilmiah Teknologi Sistem Informasi 7, 63–71.
- Wijaya, J., Soleh, A.M., Rizki, A., 2018. Xplore: Journal of Statistics 2, 32–40.

Wijaya, M.B., 2019. Mengenal Anaconda, distribusi bahasa pemrograman Python dan R [WWW Document]. M.B - Wijaya Blogpage.

Xinjuan, Z., Jing, Z., 2013a. INTERNATIONAL JOURNAL ON Advances in Information Sciences and Service Sciences 5, 96–102.

Xinjuan, Z., Jing, Z., 2013b. INTERNATIONAL JOURNAL ON Advances in Information Sciences and Service Sciences 5, 96–102.

LAMPIRAN

Code Aplikasi Prediksi menggunakan Django

```
from django.shortcuts import render, redirect
from django.core.files.storage import FileSystemStorage
from django.http import HttpResponseRedirect

import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import accuracy_score, classification_report

import pydotplus
from IPython.display import Image

from tesis_ali import datasets

# Create your views here.

def upload(request):
    if request.method == "POST":
        uploaded_file = request.FILES['document']
        fs = FileSystemStorage()
        ds = fs.exists('datasets.csv')
        if ds:
```

```

        fs.delete('datasets.csv')
        fs.save('datasets.csv', uploaded_file)

        return HttpResponseRedirect('/home/split')

    return render(request, 'home/upload_datasets.html')

def split(request):
    if request.method == 'POST':
        test = request.POST.get('test')

        fs = FileSystemStorage()
        dataSets = fs.exists('datasets.csv')
        if dataSets:
            df = pd.read_csv('media/datasets.csv')
            df.dropna(inplace=True)

            df['notasi_masa_tunggu'] =
pd.factorize(df['notasi_masa_tunggu'])[0]

            df_numpy = df.to_numpy()

            datasets.cols_label.clear()
            col_len = len(df.columns.values)
            for i in range(col_len - 3) :
                datasets.cols_label.append(df.columns.values[i])

            x = df_numpy[:, 0:18]

```

```
    yH = df_numpy[:, 18]
    yV = df_numpy[:, 19]
    y = df_numpy[:, 20]

    datasets.x_h_Train, datasets.x_h_Test, datasets.y_h_Train,
datasets.y_h_Test = train_test_split(x, yH, test_size=float(int(test)/100))
    datasets.x_v_Train, datasets.x_v_Test, datasets.y_v_Train,
datasets.y_v_Test = train_test_split(x, yV, test_size=float(int(test)/100))
    datasets.x_Train, datasets.x_Test, datasets.y_Train,
datasets.y_Test = train_test_split(x, y, test_size=float(int(test)/100))

    return HttpResponseRedirect('/home/set')

    return HttpResponseRedirect('/home/upload')

df = pd.read_csv('media/datasets.csv')
df.dropna(inplace=True)

context = {
    'datasets': df,
}
return render(request, 'home/split_datasets.html', context)

def set(request):
    return render(request, 'home/parameter.html')

def proses(request):
    if request.method == "POST":
```



```

# c45=====

criterion = request.POST.get('criterion')
splitter = request.POST.get('splitter')
mss = request.POST['min_samples_split']
msl = request.POST['min_samples_leaf']
mmaxDepth = request.POST['max_depth']

datasets.x_h_Train = MinMaxScaler().fit_transform(datasets.x_h_Train)
datasets.x_v_Train = MinMaxScaler().fit_transform(datasets.x_v_Train)
datasets.x_Train = MinMaxScaler().fit_transform(datasets.x_Train)

datasets.x_h_Test = MinMaxScaler().fit_transform(datasets.x_h_Test)
datasets.x_v_Test = MinMaxScaler().fit_transform(datasets.x_h_Test)
datasets.x_Test = MinMaxScaler().fit_transform(datasets.x_Test)

decisionTree = DecisionTreeClassifier(criterion=criterion,
splitter=splitter,
max_depth=int(mmaxDepth),
min_samples_split=int(mss), min_samples_leaf=int(msl))

model_h = decisionTree.fit(datasets.x_h_Train, datasets.y_h_Train)
model_v = decisionTree.fit(datasets.x_v_Train, datasets.y_v_Train)
model_nmt = decisionTree.fit(datasets.x_Train, datasets.y_Train)

print(datasets.cols_label)

```

```
dot_data_h = tree.export_graphviz(model_h, out_file=None,
feature_names=datasets.cols_label, class_names=['Sangat Erat', 'Erat',
'Cukup Erat', 'Kurang Erat', 'Tidak Sama Sekali', ])

description_h = tree.export_text(decisionTree,
feature_names=datasets.cols_label)

dot_data_v = tree.export_graphviz(model_v, out_file=None,
feature_names=datasets.cols_label, class_names=['Sangat Erat', 'Erat',
'Cukup Erat', 'Kurang Erat', 'Tidak Sama Sekali', ])

description_v = tree.export_text(decisionTree,
feature_names=datasets.cols_label)

dot_data_nmt = tree.export_graphviz(model_nmt, out_file=None,
feature_names=datasets.cols_label, class_names=['<=6 Bulan', '<=1
tahun', '<=2 tahun', ])

description_nmt = tree.export_text(decisionTree,
feature_names=datasets.cols_label)

graph_h = pydotplus.graph_from_dot_data(dot_data_h)
graph_v = pydotplus.graph_from_dot_data(dot_data_v)
graph_nmt = pydotplus.graph_from_dot_data(dot_data_nmt)

Image(graph_h.create_png())
Image(graph_v.create_png())
Image(graph_nmt.create_png())

filename_h = 'media/graph_h.png'
filename_v = 'media/graph_v.png'
filename_nmt = 'media/graph_nmt.png'
graph_h.write_png(filename_h)
graph_v.write_png(filename_v)
graph_nmt.write_png(filename_nmt)
```

```

predicC45_h = model_h.predict(datasets.x_h_Test)
predicC45_v = model_v.predict(datasets.x_v_Test)
predicC45_nmt = model_nmt.predict(datasets.x_Test)

akurasiC45_h = classification_report(datasets.y_h_Test,
predicC45_h)
akurasiC45_v = classification_report(datasets.y_v_Test,
predicC45_v)
akurasiC45_nmt = classification_report(datasets.y_Test,
predicC45_nmt)

scorec45_h = accuracy_score(datasets.y_h_Test, predicC45_h)
scorec45_v = accuracy_score(datasets.y_v_Test, predicC45_v)
scorec45_nmt = accuracy_score(datasets.y_Test, predicC45_nmt)

#end c45=====

# ann=====

hls = request.POST['hls']
lri = request.POST['lri']
iterasi = request.POST['iterasi']

mlp = MLPClassifier(hidden_layer_sizes=int(hls),
learning_rate_init=float(int(lri)/10), max_iter=int(iterasi))
clf_h = mlp.fit(datasets.x_h_Train, datasets.y_h_Train)
clf_v = mlp.fit(datasets.x_v_Train, datasets.y_v_Train)
clf_nmt = mlp.fit(datasets.x_Train, datasets.y_Train)

predicAnn_h = clf_h.predict(datasets.x_h_Test)
predicAnn_v = clf_v.predict(datasets.x_v_Test)

```

```
predicAnn_nmt = clf_nmt.predict(datasets.x_Test)

    akurasiAnn_h      =      classification_report(datasets.y_h_Test,
predicAnn_h)
    akurasiAnn_v      =      classification_report(datasets.y_v_Test,
predicAnn_v)
    akurasiAnn_nmt    =      classification_report(datasets.y_Test,
predicAnn_nmt)

scoreAnn_h = accuracy_score(datasets.y_h_Test, predicAnn_h)
scoreAnn_v = accuracy_score(datasets.y_v_Test, predicAnn_v)
scoreAnn_nmt = accuracy_score(datasets.y_Test, predicAnn_nmt)

#end ann=====

context = {
    'status': True,
    'img_h': filename_h,
    'img_v': filename_v,
    'img_nmt': filename_nmt,
    'des_h': description_h,
    'des_v': description_v,
    'des_nmt': description_nmt,
    'akurasiC45_h': akurasiC45_h,
    'akurasiC45_v': akurasiC45_v,
    'akurasiC45_nmt': akurasiC45_nmt,
    'scoreC45_h': round((scorec45_h*100), 2),
    'scoreC45_v': round((scorec45_v*100), 2),
    'scoreC45_nmt': round((scorec45_nmt*100), 2),
    'akurasiAnn_h': akurasiAnn_h,
```

```
'akurasiAnn_v': akurasiAnn_v,  
'akurasiAnn_nmt': akurasiAnn_nmt,  
'scoreAnn_h': round((scoreAnn_h*100), 2),  
'scoreAnn_v': round((scoreAnn_v*100), 2),  
'scoreAnn_nmt': round((scoreAnn_nmt*100), 2),  
}  
return render(request, 'home/hasil.html', context)  
  
return redirect('/home/set')
```