

Studi Algoritma Klasifikasi Untuk Mengenal Pola *Activity Daily Living* (ADL) Menggunakan K-Nearest Neighbor Pada Orang Dewasa Sehat

Manual Book



Disusun oleh :
Askawati (1301144215)

Program Studi Sarjana S1 Teknik Informatika
Fakultas Informatika
Universitas Telkom
Bandung
2018

DAFTAR ISI

Daftar Gambar	4
1. Data	5
1.1. Data Training	5
1.2. Data Pengujian	6
2. Program	7
2.1. Program Arduino	7
2.2. Program Pengolahan Data	10

Daftar Gambar

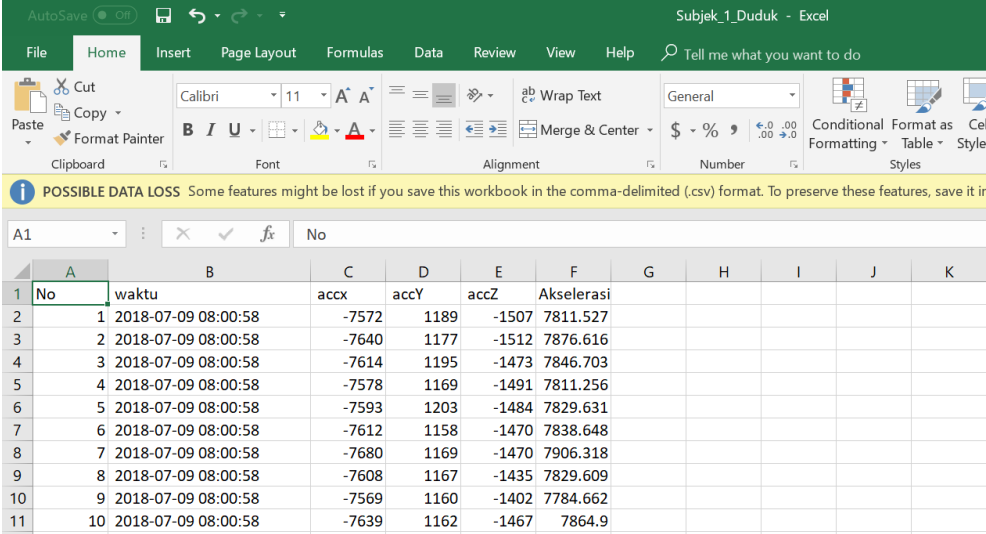
Gambar 1 Data sensor aktivitas duduk subjek 1	5
Gambar 2 Data hasil ekstraksi ciri	6
Gambar 3 Data pengujian aktivitas berjalan subjek 1	7
Gambar 4 Tampilan awal arduino	7
Gambar 5 Pemilihan board	8
Gambar 6 Pemilihan flash frequency dan upload speed	8
Gambar 7 Pemilihan port	9
Gambar 8 Konfigurasi server dan hotspot	9
Gambar 9 Serial monitor	10
Gambar 10 Tampilan awal pycharm edu 2018	10
Gambar 11 Menu settings	11
Gambar 12 Install packages	11
Gambar 13 Tampilan awal program file ekstraksi ciri	12
Gambar 14 Membuka file yang di ekstraksi	12
Gambar 15 Ubah label aktivitas pada ekstraksi ciri	13
Gambar 16 Simpan hasil ekstraksi	13
Gambar 17 Tampilan program file knn.py	13
Gambar 18 Membuka file dataset	14
Gambar 19 Pembagian data trainin dan testing	14
Gambar 20 pemilihan parameter k dan metrik	14
Gambar 21 Menjalankan program file knn.py	15
Gambar 22 Hasil menjalankan program file knn.py	15
Gambar 23 Tampilan awal program file Testing.py	16
Gambar 24 Membuka file prediksi	16
Gambar 25 Hasil menjalankan program file Testing.py	16

1. Data

Data diperoleh berdasarkan hasil percobaan menggunakan perangkat sensor MPU6050 dan mikrokontroler ESP32. Perangkat sensor diletakkan pada pinggang pengguna saat percobaan, percobaan menggunakan 30 subjek dengan aktivitas seperti :

- Aktivitas berjalan
- Aktivitas berdiri
- Aktivitas duduk
- Aktivitas turun tangga
- Aktivitas naik tangga

Data yang dihasilkan sensor berupa banyak data, waktu, akselerometer dengan sumbu x, sumbu y dan sumbu z. Data pada folder data sensor terdiri dari device/banyaknya data, waktu, accx, accy, accz, dan akselerasi. Berikut gambar data yang dihasilkan.



	A	B	C	D	E	F	G	H	I	J	K
	No	waktu	accx	accy	accz	Akselerasi					
1											
2	1	2018-07-09 08:00:58	-7572	1189	-1507	7811.527					
3	2	2018-07-09 08:00:58	-7640	1177	-1512	7876.616					
4	3	2018-07-09 08:00:58	-7614	1195	-1473	7846.703					
5	4	2018-07-09 08:00:58	-7578	1169	-1491	7811.256					
6	5	2018-07-09 08:00:58	-7593	1203	-1484	7829.631					
7	6	2018-07-09 08:00:58	-7612	1158	-1470	7838.648					
8	7	2018-07-09 08:00:58	-7680	1169	-1470	7906.318					
9	8	2018-07-09 08:00:58	-7608	1167	-1435	7829.609					
10	9	2018-07-09 08:00:58	-7569	1160	-1402	7784.662					
11	10	2018-07-09 08:00:58	-7639	1162	-1467	7864.9					

Gambar 1 Data sensor aktivitas duduk subjek 1

1.1. Data Training

Data *training* diperoleh berdasarkan hasil ekstraksi ciri menggunakan file Ekstraksi ciri.py pada setiap aktivitas. Data pada folder data *training* terdiri dari data *minimum*, *maksimum*, *mean*, standar deviasi, *median*, *variance*, dan label.

#	min	max	mean	stdev	median	variance	label
1	125.7998	19903.19	5952.038	7188.597	4359.707	51675930	1
2	-44.517	8761.721	3101.966	3316.184	1139.93	10997077	1
3	-1829.65	3818.137	1278.592	2213.019	829.1261	4897455	1
4	-7889.56	-274.388	-2924.57	2737.808	-2607.81	7495595	1
5	-9453.96	387.2328	-3684.09	3688.292	-2022.94	13603495	1
6	-3368.83	265.5072	-1441.68	1497.752	-503.436	2243260	1
7	-1726.89	769.4407	-119.964	851.5978	185.3604	725218.8	1
8	13.23305	10925.39	3591.372	3943.929	2888.103	15554574	1
9	-106.266	968.3295	246.2684	383.0102	45.11055	146696.8	1
10	4673.03	4673.433	4673.433	0.000	4673.433	0.000	1

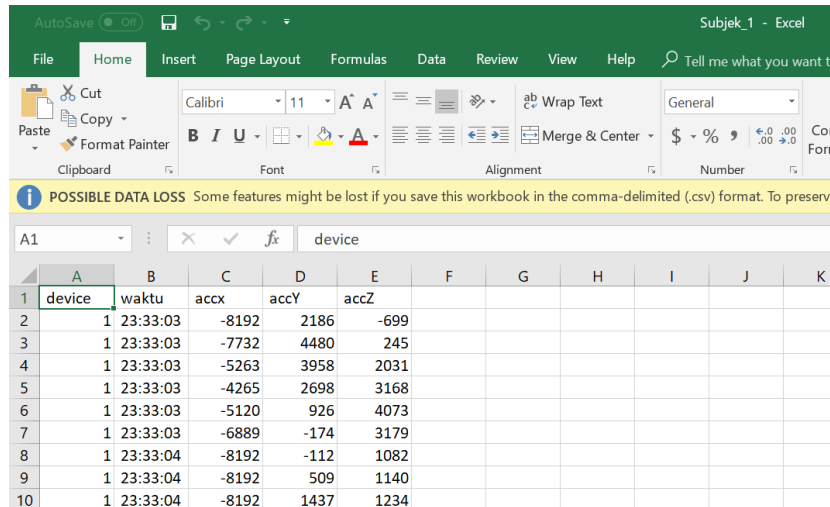
Gambar 2 Data hasil ekstraksi ciri

Berdasarkan gambar 2. Pada setiap aktivitas memiliki label yang berbeda-beda. Berikut penjelasan file pada folder data *training*.

- Data_Berjalan_1 (label 1)
- Data_Berdiri_2 (label 2)
- Data_NaikTangga_3 (label 3)
- Data_TurunTangga_4 (label 4)
- Data_Duduk_5 (label 5)
- Data_Allactivity, Kumpulan seluruh data aktivitas beserta label.

1.2. Data Pengujian

Data pengujian terdiri dari 10 subjek dengan 5 aktivitas yang sama. Data pada folder Data Pengujian terdiri dari device atau banyaknya data, waktu, accx, accy, dan accz. Berikut gambar data yang dihasilkan.



	A	B	C	D	E	F	G	H	I	J	K
1	device	waktu	accx	accY	accZ						
2	1	23:33:03	-8192	2186	-699						
3	1	23:33:03	-7732	4480	245						
4	1	23:33:03	-5263	3958	2031						
5	1	23:33:03	-4265	2698	3168						
6	1	23:33:03	-5120	926	4073						
7	1	23:33:03	-6889	-174	3179						
8	1	23:33:04	-8192	-112	1082						
9	1	23:33:04	-8192	509	1140						
10	1	23:33:04	-8192	1437	1234						

Gambar 3 Data pengujian aktivitas berjalan subjek 1

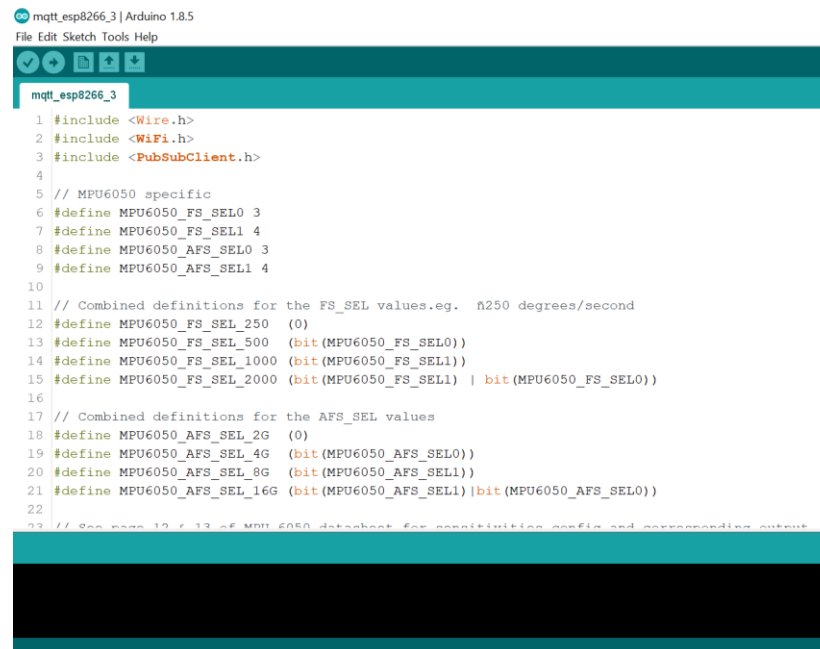
2. Program

Program pada penelitian ini terdiri dari dua program yaitu program Arduino untuk perangkat yang digunakan dan program pengolahan data menggunakan Bahasa pemrograman python.

2.1. Program Arduino

Berikut langkah-langkah menjalankan program arduino.

1. Buka arduino dan buka file mqtt_esp8266_3



```

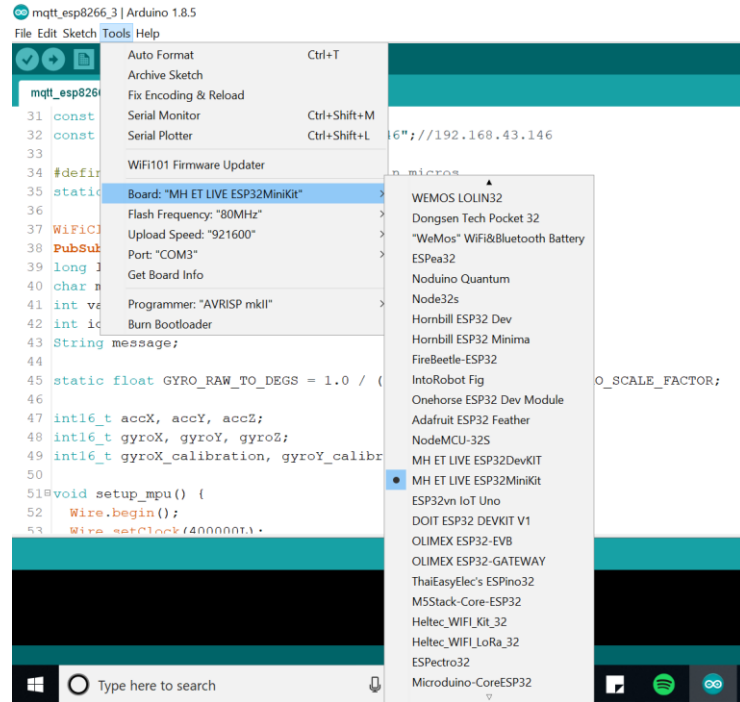
mqtt_esp8266_3 | Arduino 1.8.5
File Edit Sketch Tools Help

mqtt_esp8266_3
1 #include <Wire.h>
2 #include <WiFi.h>
3 #include <PubSubClient.h>
4
5 // MPU6050 specific
6 #define MPU6050_FS_SEL0 3
7 #define MPU6050_FS_SEL1 4
8 #define MPU6050_AFS_SEL0 3
9 #define MPU6050_AFS_SEL1 4
10
11 // Combined definitions for the FS_SEL values.eg. 1250 degrees/second
12 #define MPU6050_FS_SEL_250 (0)
13 #define MPU6050_FS_SEL_500 (bit(MPU6050_FS_SEL0))
14 #define MPU6050_FS_SEL_1000 (bit(MPU6050_FS_SEL1))
15 #define MPU6050_FS_SEL_2000 (bit(MPU6050_FS_SEL1) | bit(MPU6050_FS_SEL0))
16
17 // Combined definitions for the AFS_SEL values
18 #define MPU6050_AFS_SEL_2G (0)
19 #define MPU6050_AFS_SEL_4G (bit(MPU6050_AFS_SEL0))
20 #define MPU6050_AFS_SEL_8G (bit(MPU6050_AFS_SEL1))
21 #define MPU6050_AFS_SEL_16G (bit(MPU6050_AFS_SEL1) | bit(MPU6050_AFS_SEL0))
22
23 // See page 12 & 13 of MPU 6050 datasheet for sensitivities, offsets and corresponding output

```

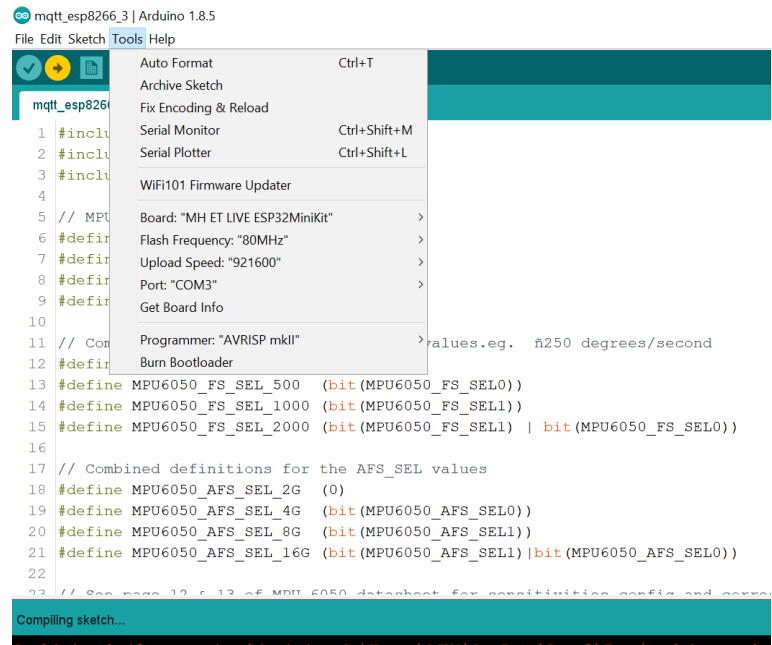
Gambar 4 Tampilan awal arduino

2. Pilih board yang digunakan. Pilih tools dan pilih board MH ET LIBE ESP32MiniKit.



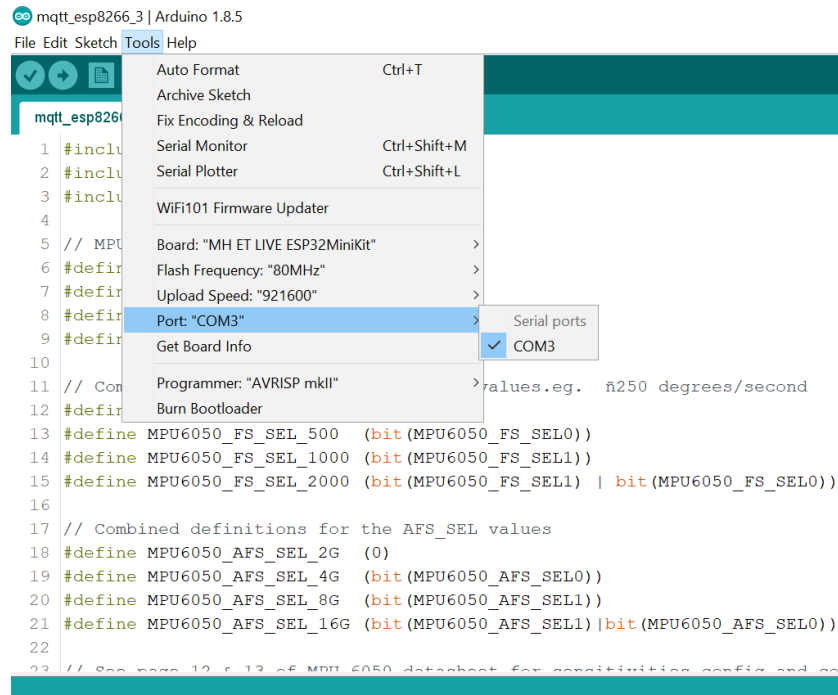
Gambar 5 Pemilihan board

3. Pada menu tools, pilih flash frequency 80MHz dan upload speed 921600.



Gambar 6 Pemilihan flash frequency dan upload speed

4. Setelah itu hubungkan perangkat dan pilih port yang digunakan. Setiap perangkat memiliki port yang berbeda-beda.



Gambar 7 Pemilihan port

5. Ubah koneksi yang digunakan sehingga data dapat melakukan upload.

```

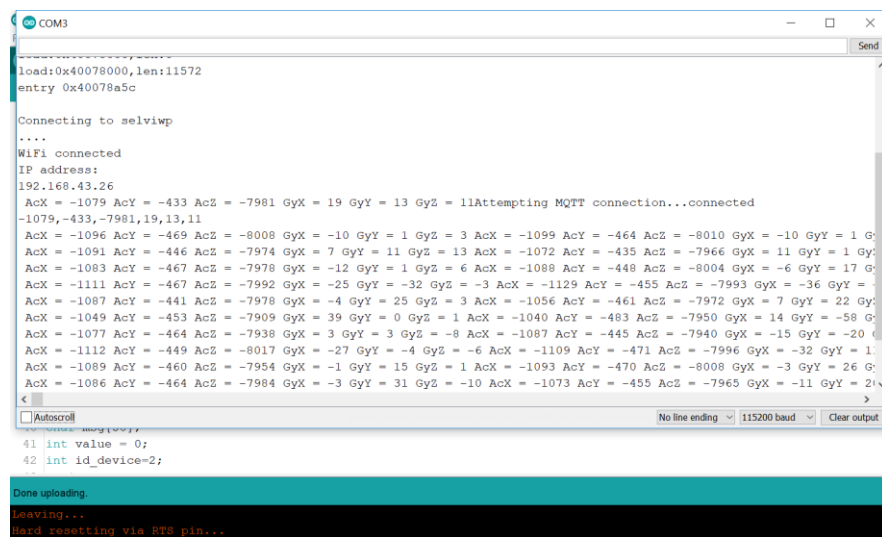
29 |
30 | const char* ssid = "AndroidAPJ";
31 | const char* password = "tugasakhir";
32 | const char* mqtt_server = "192.168.43.146";//192.168.43.146
33 |

```

Gambar 8 Konfigurasi server dan hotspot

Berdasarkan gambar 8. Ssid merupakan nama hotspot yang digunakan, password merupakan password dari hotspot yang digunakan dan mqtt_server merupakan alamat IP dari komputer yang akan menjadi server.

6. Jika sudah di konfigurasi selanjutnya upload file tersebut dan buka serial monitor.

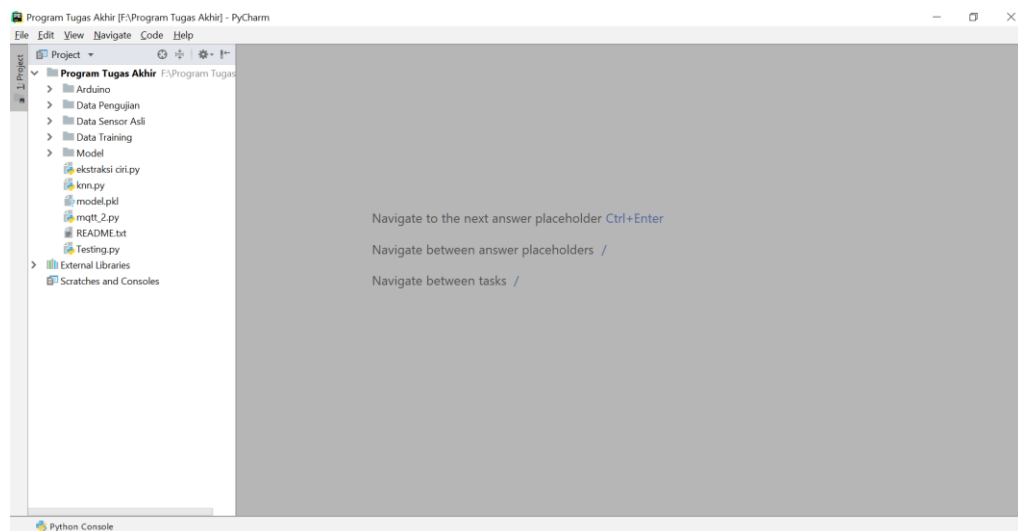


Gambar 9 Serial monitor

2.2. Program Pengolahan Data

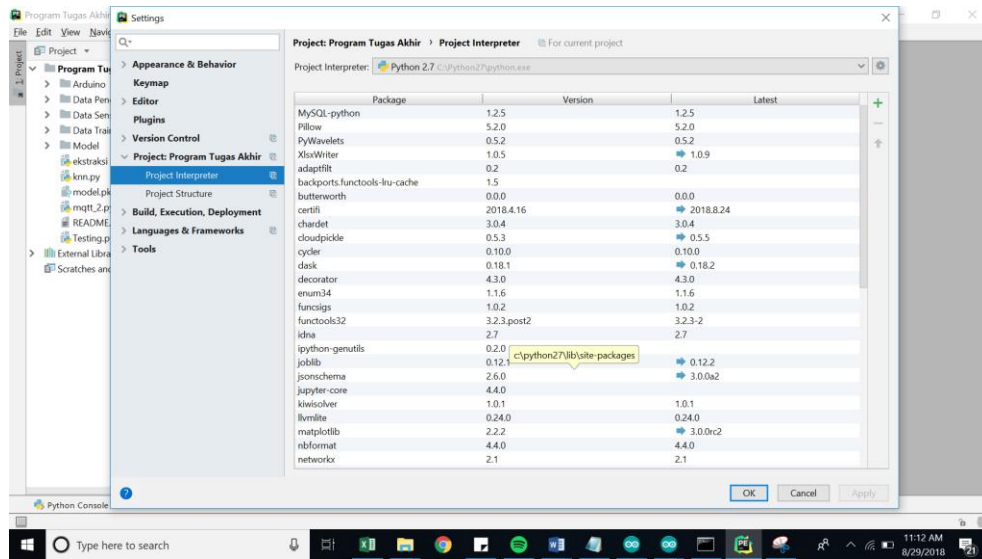
Program ini menggunakan python 2.7 dengan software pycharm edu 2108. Berikut langkah-langkah menjalankan program menggunakan pycharm edu 2018 :

1. Buka pycharm edu 2018, dan buka folder program tugas akhir.



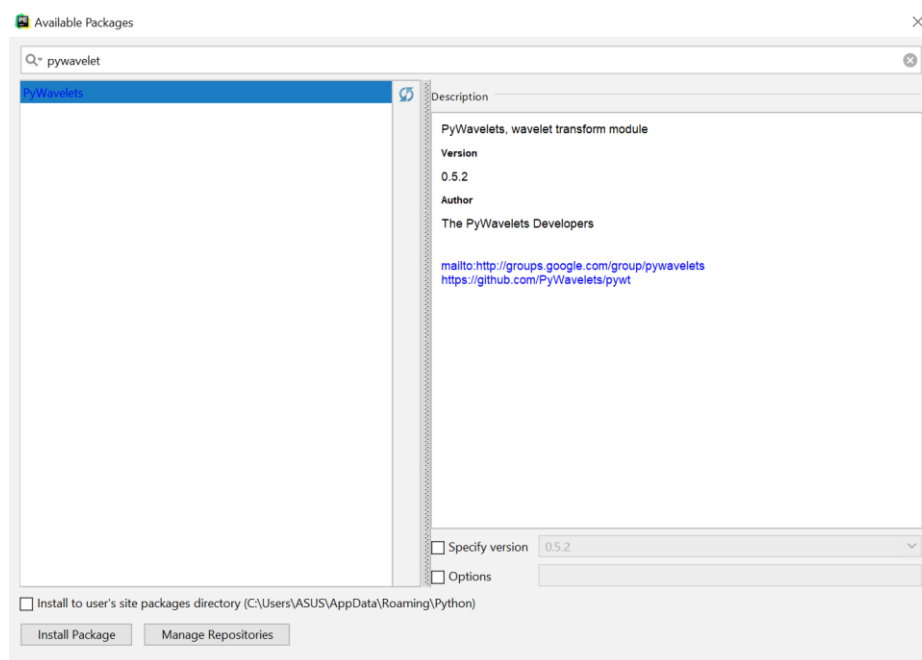
Gambar 10 Tampilan awal pycharm edu 2018

2. Selanjutnya sebelum membuka dan menjalankan program terdapat konfigurasi pada menu file pilih settings.



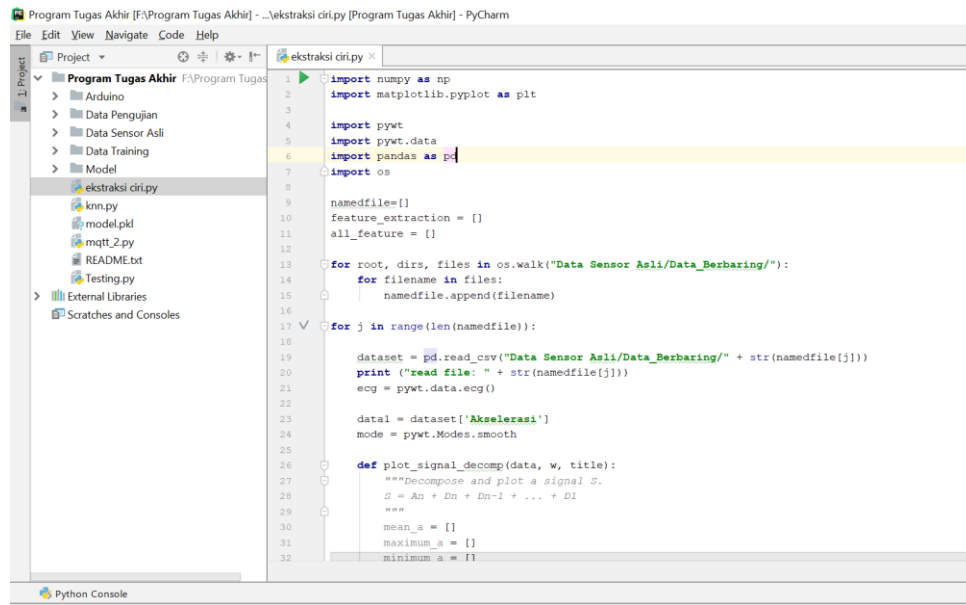
Gambar 11 Menu settings

Setelah ada tampilan seperti pada gambar 10. Pilih menu Project:Program Tugas Akhir. Lalu pilih Project Interpreter. Ubah project interpreter menjadi python 2.7 atau versi pyton lain yang dapat digunakan. Dalam tugas akhir ini digunakan python 2.7. Selanjutnya, install package-package yang diperlukan. Package yang digunakan pada program ini yaitu PyWavelet, joblib, matplotlib, seaborn, sklearn, pandas, numpy.



Gambar 12 Install packages

3. Setelah semua package telah terinstall. Buka program file Ekstraksi ciri.py.



Gambar 13 Tampilan awal program file ekstraksi ciri

4. Ubah file yang akan di ekstraksi.

```
for root, dirs, files in os.walk("Data Sensor/Data_Berbaring/"):
    for filename in files:
        namedfile.append(filename)

for j in range(len(namedfile)):

    dataset = pd.read_csv("Data Sensor/Data_Berbaring/" + str(namedfile[j]))
    print("read file: " + str(namedfile[j]))
    ecg = pywt.data.ecg()

    data1 = dataset['Akselerasi']
    mode = pywt.Modes.smooth
```

Gambar 14 Membuka file yang di ekstraksi

File yang digunakan untuk melakukan ekstraksi fitur yaitu pada folder data sensor. Masukkan semua aktivitas yang ada sehingga dapat diperoleh ciri dari setiap aktivitas yang digunakan. Program akan membaca data pada kolom “Akselerasi” dan menghasilkan ciri data dari aktivitas tersebut seperti data *minimum*, *maksimum*, *mean*, standar deviasi, *median*, *variance*.

5. Ubah label pada setiap aktivitas.

```
feature_extraction = [min_coefD[4], max_coefD[4], mean_coefD[4], stdev_coefD[4], median_coefD[4], var_coefD[4], 6]  
all_feature.append(feature_extraction)  
print("completed")
```

Gambar 15 Ubah label aktivitas pada ekstraksi ciri

Berikut adalah daftar label yang digunakan pada setiap aktivitas :

- Data_Berjalan = label 1
 - Data_Berdiri = label 2
 - Data_NaikTangga = label 3
 - Data_TurunTangga = label 4
 - Data_Duduk = label 5
6. Jika telah mengubah label dari aktivitas yang akan di ekstrak. Selanjutnya simpan data hasil ekstraksi ciri dengan format .csv seperti pada gambar 16.

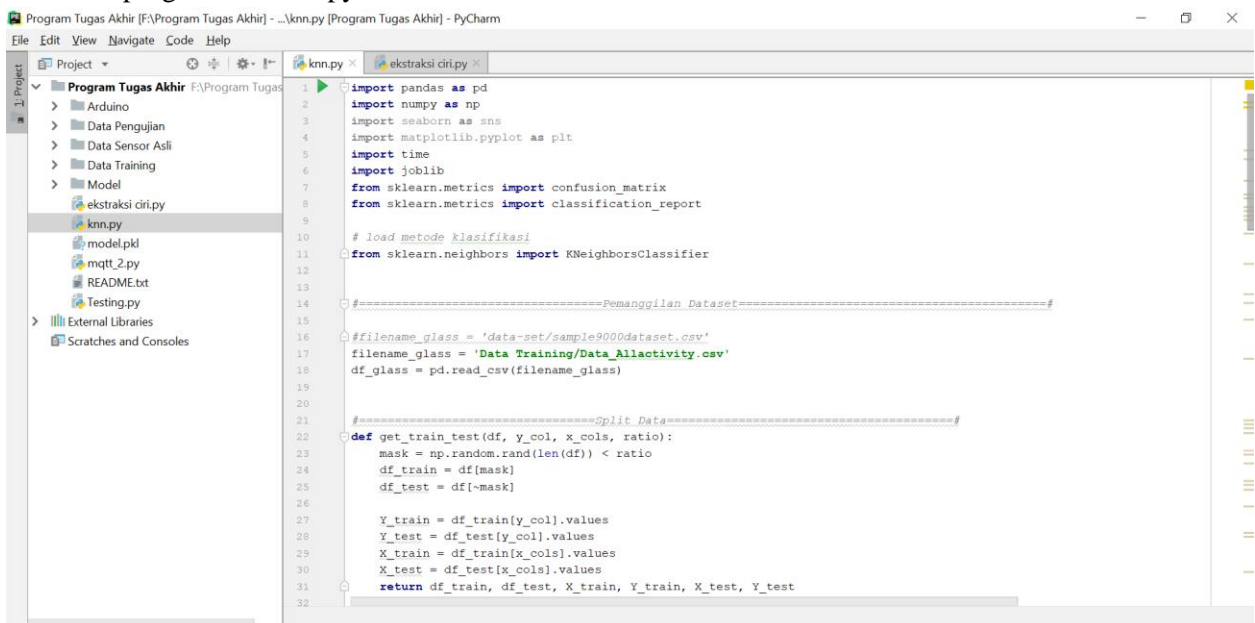
```
print(all_feature)  
np.savetxt("Data Training/Data_Berjalan_1.csv", all_feature, header="min,max,mean,stdev,median,variance,label", delimiter=",")
```

Gambar 16 Simpan hasil ekstraksi

Catatan :

Run semua file pada folder data sensor sehingga mendapatkan ciri dan label data setiap aktiitas.

7. Buka program file knn.py.



Gambar 17 Tampilan program file knn.py

8. Buka file yang akan digunakan untuk mencari akurasi, sensitifitas, dan spesifisitas seperti pada gambar 18.

```
}#=====Pemanggilan Dataset=====
|
|#filename_glass = 'data-set/sample9000dataset.csv'
|filename_glass = 'Data Training/Data_Allactivity.csv'
|df_glass = pd.read_csv(filename_glass)
```

Gambar 18 Membuka file dataset

Catatan :

File Data_Allactivity.csv merupakan data hasil pengabungan dari aktivitas-aktivitas yang digunakan beserta label pada setiap aktivitas.

9. Jika diinginkan ubah train_test_ratio untuk pembagian data *training* dan *testing* pada. Pada tugas akhir ini menggunakan pembagian data 0.6, 0.7 dan 0.8. Pada gambar 19. Train_test_ratio yang digunakan yaitu 0.8 yang artinya data training 0.8 dan sisanya adalah data *testing*.

```
train_test_ratio = 0.8
df_train, df_test, X_train, Y_train, X_test, Y_
#=====
```

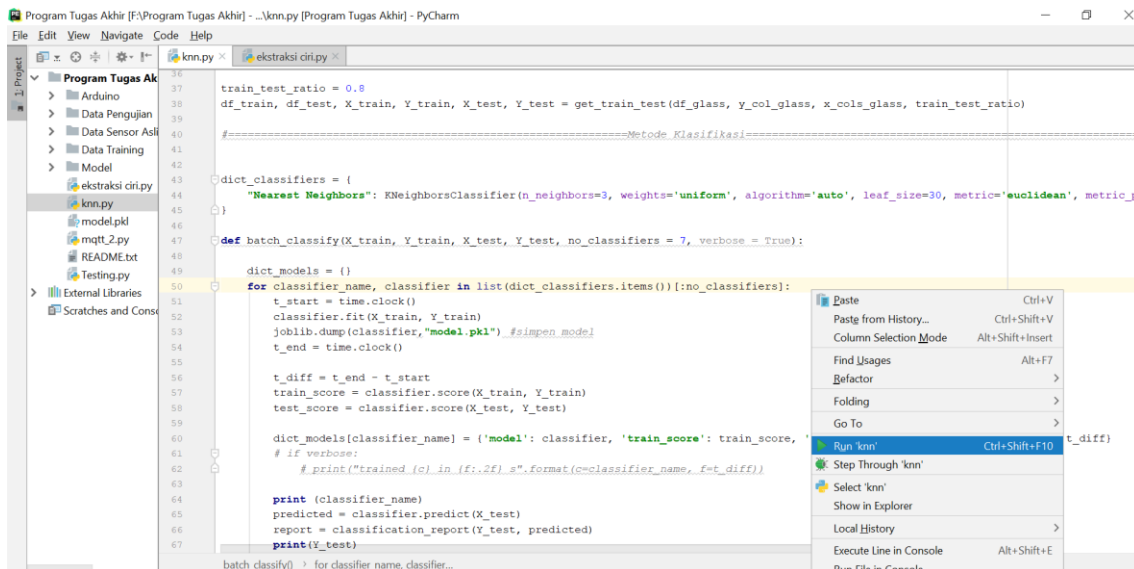
Gambar 19 Pembagian data trainin dan testing

10. pada dict_classifier, ubah n_neighbors yang merupakan parameter k . Contoh n_neighbors=3. Ubah metric atau metode pengukuran jarak yang digunakan. Contoh: (Metric='euclidean') .

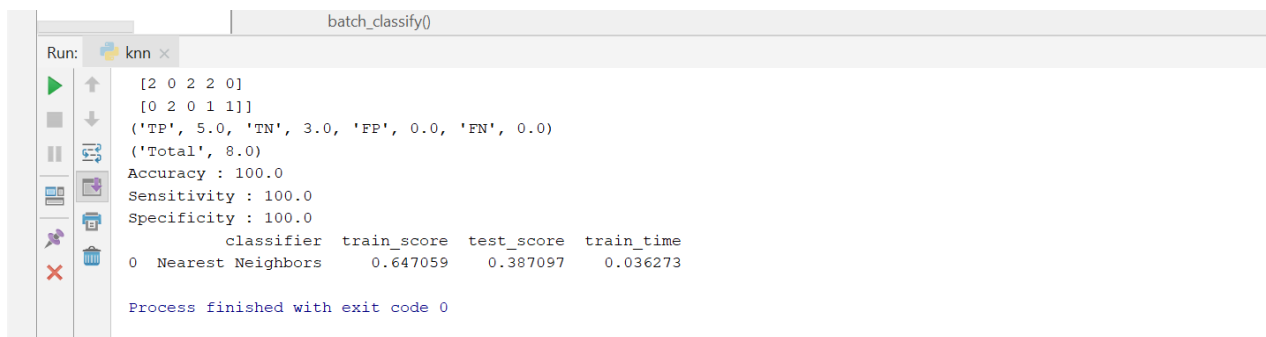
```
dict_classifiers = {
    "Nearest Neighbors": KNeighborsClassifier(n_neighbors=3, weights='uniform', algorithm='auto', leaf_size=30, metric='euclidean',
}
```

Gambar 20 pemilihan parameter k dan metrik

11. Jika semua telah diubah sesuai yang diinginkan. Jalankan program seperti pada gambar 21.



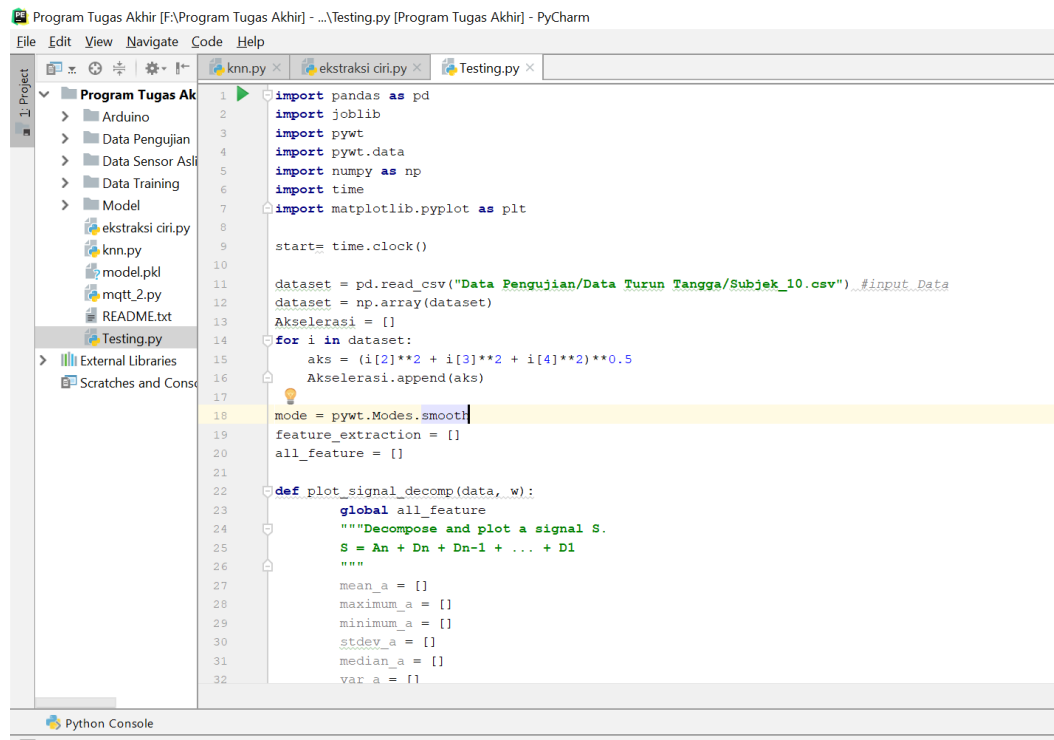
Gambar 21 Menjalankan program file knn.py



Gambar 22 Hasil menjalankan program file knn.py

Gambar 21. Akan dilakukan sebanyak 10 kali untuk mendapatkan hasil rata-rata dari setiap aktivitas berdasarkan parameter k dan metode yang digunakan. Hasil rata-rata terbaik yang dihasilkan akan menjadi inputan dalam melakukan prediksi yang akan dilakukan pada program file `testing.py`.

12. Buka program file `Testing.py`



Gambar 23 Tampilan awal program file Testing.py

13. Pilih data untuk melakukan prediksi aktivitas pada folder data testing seperti pada gambar 24.

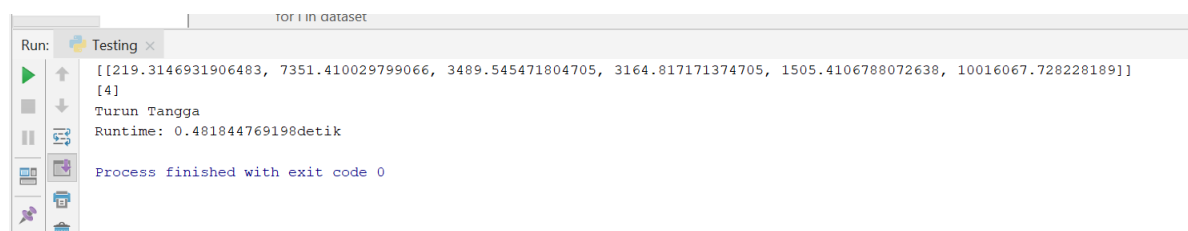
```

dataset = pd.read_csv("Data Pengujian/Data Turun Tangga/Subjek_10.csv") #input Data
dataset = np.array(dataset)
Akselerasi = []
for i in dataset:

```

Gambar 24 Membuka file prediksi

14. Jalankan program file Testing.py untuk melihat hasil prediksi dari data testing yang diperoleh seperti pada gambar 25.



Gambar 25 Hasil menjalankan program file Testing.py

