

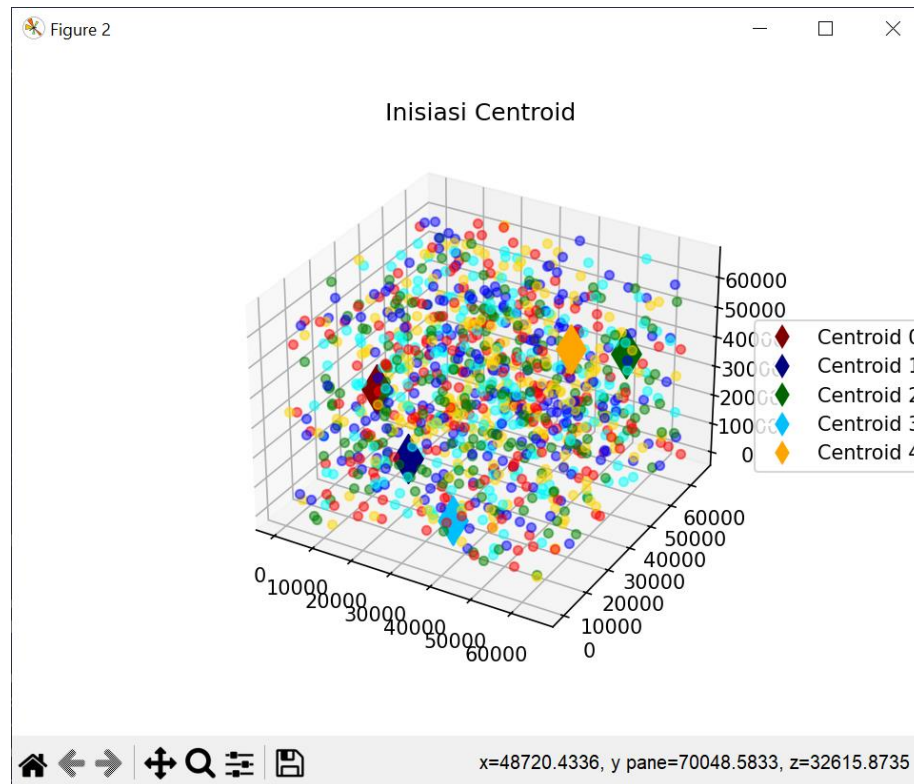
## DAFTAR PUSTAKA

- Abriyansyah, Muh (2022). Kendali Robot Humanoid Dengan Gelombang Otak Manusia [Unpublished undergraduate dissertation]. Universitas Hasanuddin.
- Akbar, Fahmi Malik Namus. Metode KNN (K-Nearest Neighbor) untuk Menentukan Kualitas Air. *Jurnal Tekno Kompak, Indonesia*, 18(1), 28-40.
- Argina, Andi Maulida (2020). Penerapan Metode Klasifikasi K-Nearest Neighbor pada Dataset Penderita Penyakit Diabetes. *Indonesian Journal of Data and Science, Indonesia*, 1(2), 29-33.
- Brilian, Ahmad Hayam (2015). Pengenalan Sandi Morse Dari Sinyal Electroencephalogram Yang Direkam Perangkat Neurosky Mindwave [Unpublished undergraduate dissertation]. Institut Teknologi Sepuluh Nopember.
- B. Prihambodo, A. W. F Y, E. Prayoga dan A. Jaffar (2023). Klasifikasi Kualitas Air Sungai Berbasis Teknik Data Mining Dengan Metode K-Nearest Neighbor (K-NN). *MITOR Jurnal Teknik Elektro, Indonesia*, 23, 31-36.
- C. Muhammad, R. Maulana dan M. H. H. Ichsan (2020). Purwarupa Perahu untuk Monitoring dan Klasifikasi Kualitas Air Bendungan dengan Metode K-Nearest Neighbor (KNN). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Indonesia*, 4, 651-659.
- Chrystinne Fernandes, C. L. (2017). Smart depth of anesthesia monitoring with EEG sensors and agent-based technology. *IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*. San Fransisco, USA.
- Dhingra, Hema (2023). *I.C.S.E. Robotics and Artificial Intelligence*. Goyal Brothers Prakashan, New Delhi.
- H. Said, N. H. Matondang, dan H. N. Irmanda (2022). Penerapan Algoritma K-Nearest Neighbor Untuk Memprediksi Kualitas Air Yang Dapat Dikonsumsi. *Techno.Com, Indonesia*, 21(2), 256–267.
- I. Handayani (2019). Application of K-Nearest Neighbor Algorithm on Classification of Disk Hernia and Spondylolisthesis in Vertebral Column. *J. Inf. Syst., Indonesia*, 2(1), 57–66.
- J. B. Chandra and D. Nasien (2023). Application Of Machine Learning K-Nearest Neighbour Algorithm To Predict Diabetes. *International Journal of Electrical, Energy and Power System Engineering*, 2, 134-139.

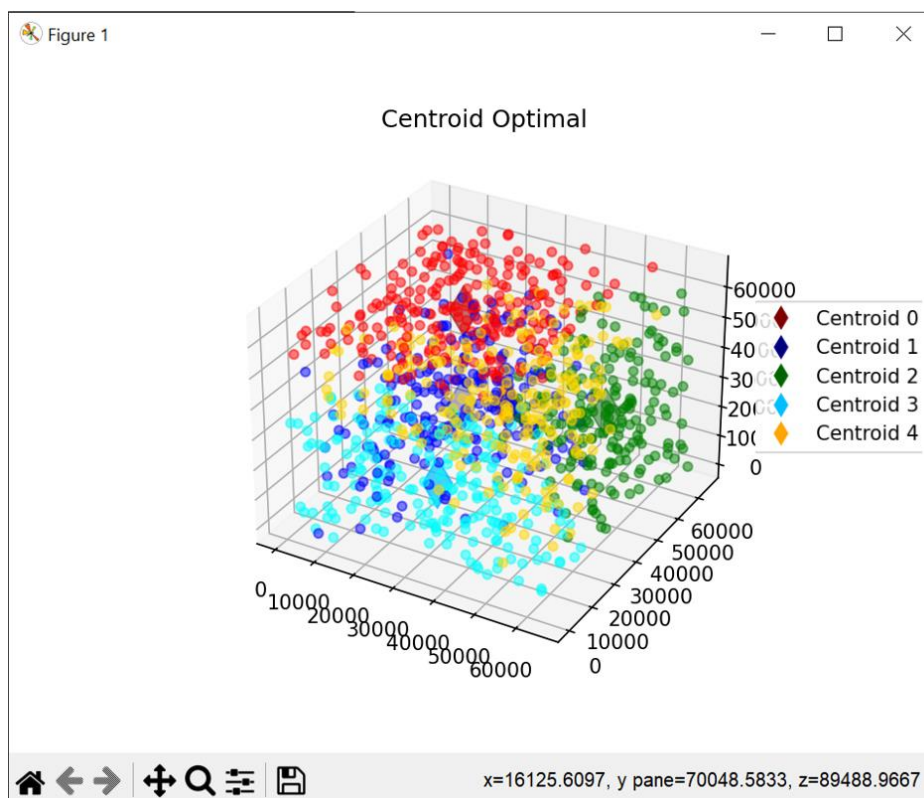
- K. Permana, S. K. (2019). Controlled wheelchair based on braincomputer interface using NeuroskyMindwave Mobile 2. AIP Conference Proceedings 2168.
- NeuroSky (2015). *MindWave Mobile : User Guide*. NeuroSky Inc, San Jose.
- Niku, Saeed B. (2011). *Introduction to Robotics : Analysis, Control, Applications 2nd Edition*. Wiley, New Jersey.
- P. Putra, A. M. H. Pardede dan S. Syahputra (2022). Analisis Metode K-Nearest Neighbour (Knn) Dalam Klasifikasi Data Iris Bunga. Jurnal Teknik Informatika Kaputama (JTIK), Indonesia, 6, 297-286.
- Primartha, Rifkie (2018). Belajar Machine Learning Teori dan Praktik. Informatika, Bandung.
- Purnama, Bedy (2019). Pengantar Machine Learning Konsep dan Praktikum dengan Contoh Latihan Berbasis R dan Python. Informatika, Bandung.
- Rohmansa, Reva Qintara (2024). Analisis Sentimen Ulasan Pengguna Aplikasi Discord Menggunakan Metode K-Nearest Neighbor. Jurnal Ilmiah Penelitian dan Pembelajaran Informatika, Indonesia, 9(1), 368-378.
- S. M. Prisca, B. Hidayat dan S. Darana (2018). Klasifikasi Untuk Deteksi Kualitas Keju Cheddar Menggunakan Pengolahan Citra Digital Dengan Metode Content Based Image Retrieval Dan K-Nearest Neighbor Berbasis Android. E-Proceeding of Engineering, Indonesia, 5, 2138-2146.
- Senai, Saeid dan Chambers, J. A. (2013). *EEG Signal Processing*. Wiley, New Jersey.

## LAMPIRAN

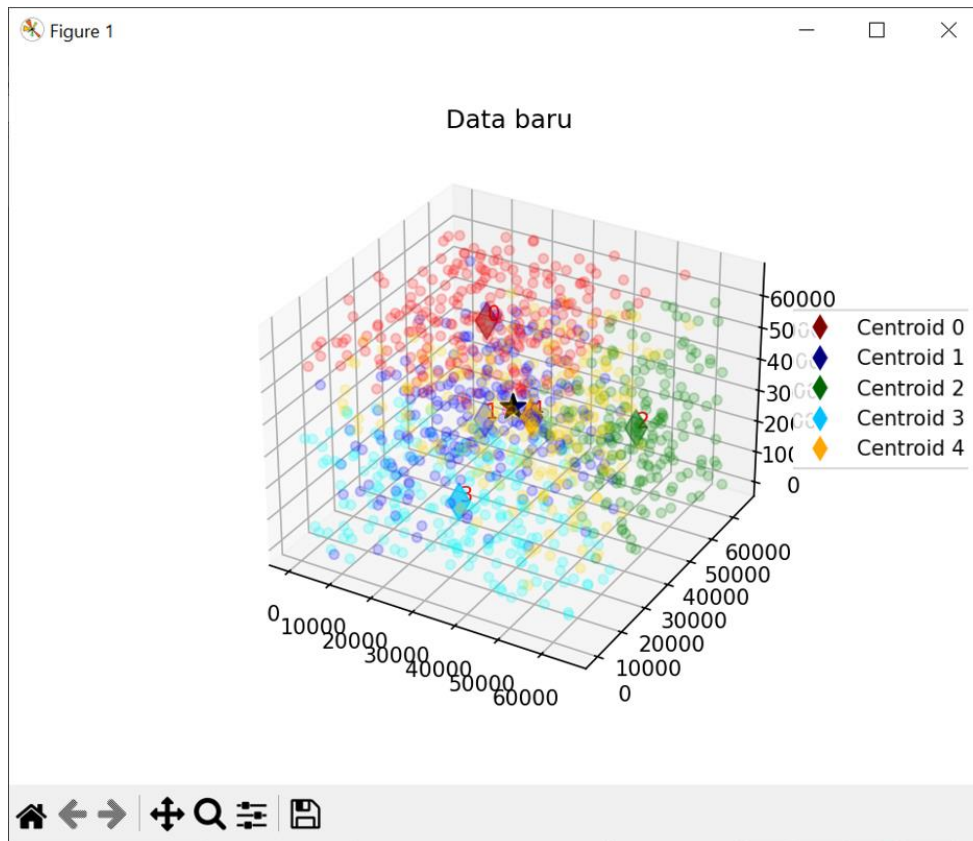
Lampiran 1 Grafik inisiasi *centroid* awal



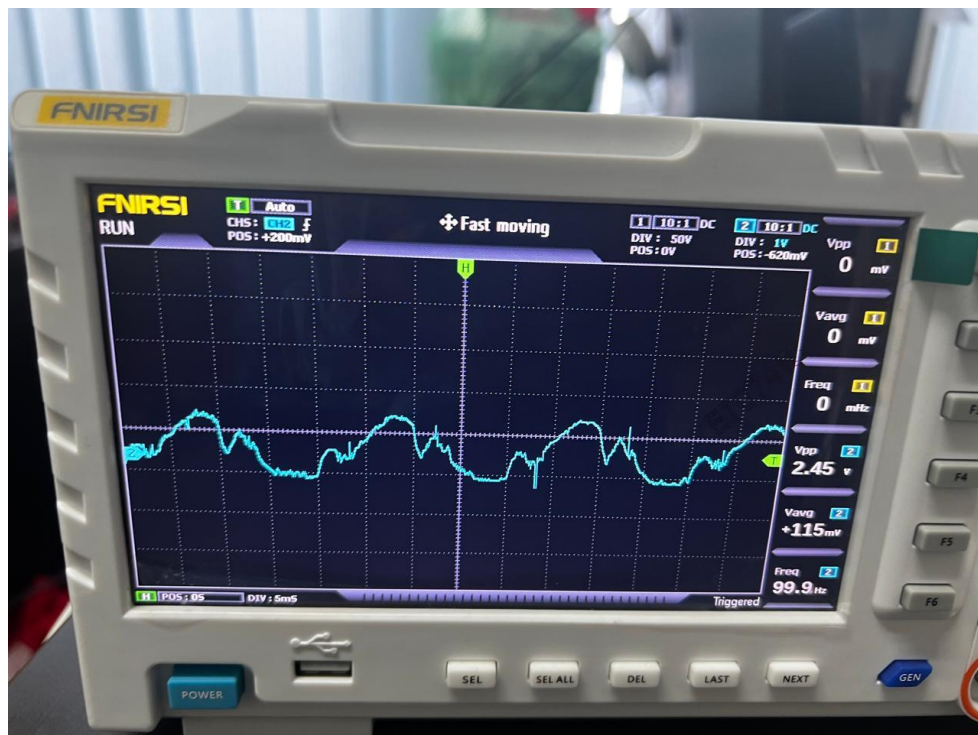
Lampiran 2 Grafik pengoptimalan *centroid*



Lampiran 3 Grafik ketika ada data yang masuk



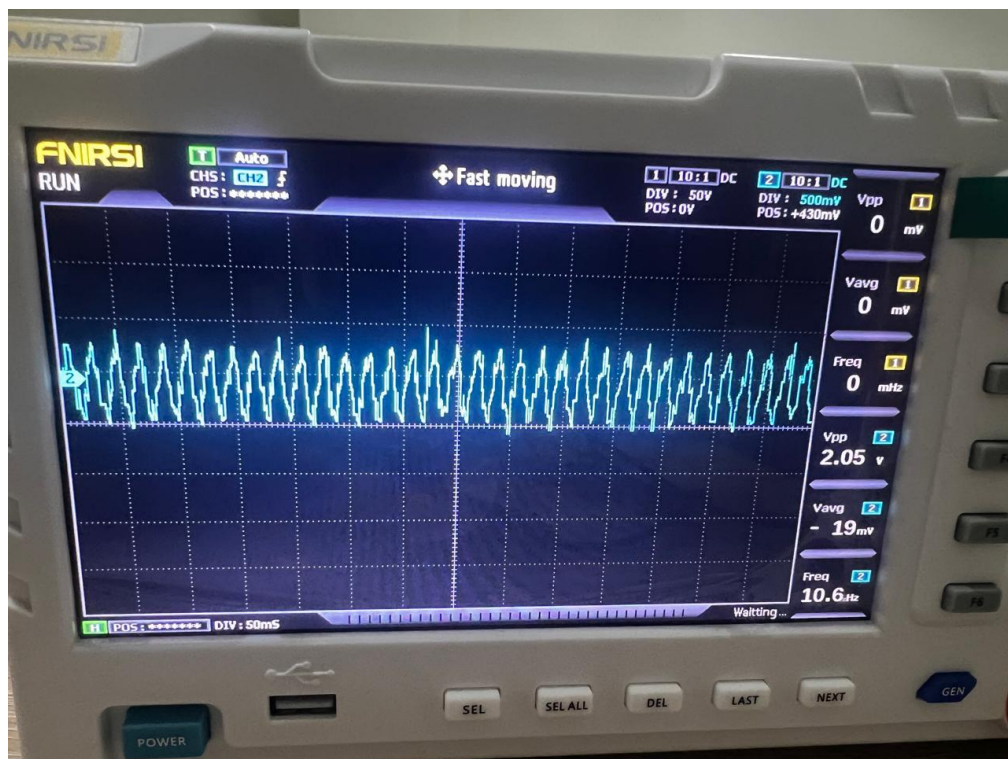
Lampiran 4 Pembacaan osiloskop ketika robot bergerak ke depan



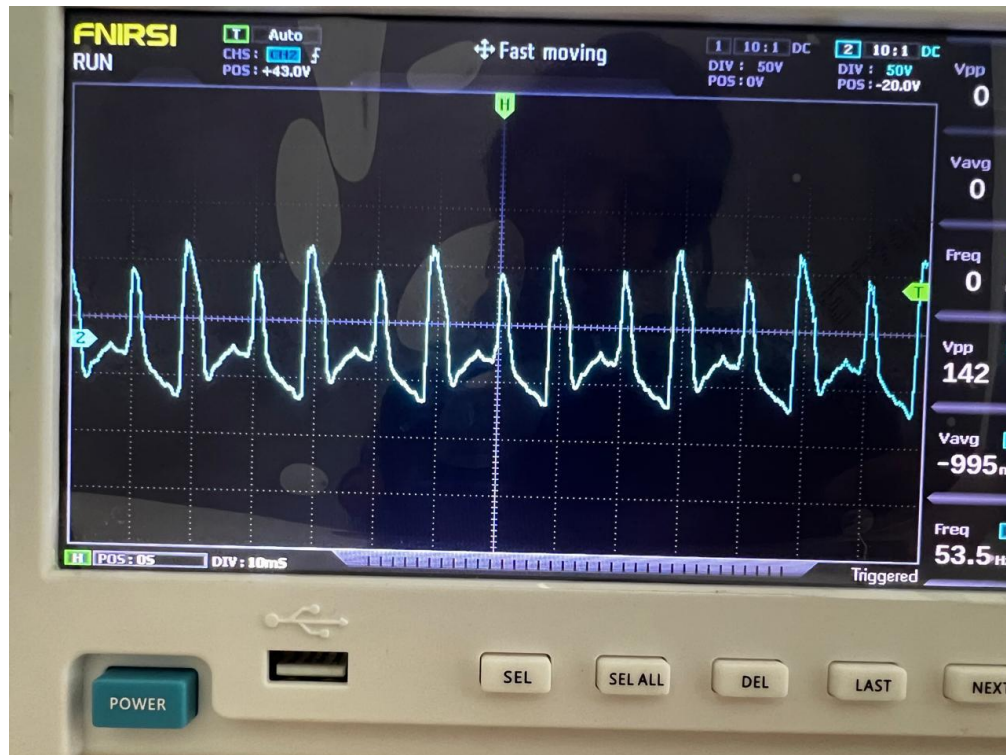
Lampiran 5 Pembacaan osiloskop ketika robot bergerak ke belakang



Lampiran 6 Pembacaan osiloskop ketika robot bergerak ke kanan



### Lampiran 7 Pembacaan osiloskop ketika robot bergerak ke kiri



### Lampiran 8 Pseudocode Arduino

// Initialize constants and variables

initialize variables:

msright, msleft, msstand = 150, 150, 125

terkirim, tinyLEDState, pq, med = 0, HIGH, 0, 0

incomingData, meTime, isWait, motion, stateMind = "", 0, false, 0, 0

mindArray, mindCompare, summationmindCompare = arrays of specific sizes

various long int parameters = 0

initialize pins:

lednn = 12, tinyLEDpin = 13, focus = 8, mEDI = 9

right, front, back, left = 4, 5, 6, 7

setup() function:

```
end Serial, delay 100ms
call switcher with 31
begin Serial and Serial2 at 115200 baud
write 0xc2 to Serial2 for RF dongle pairing
set Serial for Dynamixel
begin Dynamixel at 1 Mbps, pin control 2
call dynamixelPing
delay 1500ms
call idling
```

```
set pin modes to OUTPUT
call sayHeadsetConnected
set initial states for digitalWrite
```

loop() function:

```
if no Serial data available:
  get current time
  get attention value from Mindwave, update aTT and mED
```

```
if aTT > 0:
  analogWrite mEDI and focus with calculated values
  if interval passed:
    update previousMillis
    switch stateMind:
      case 0:
        stateMind = 1
        call predictor
        read pdata from Serial
        if pdata is a movement command:
```

execute corresponding movement function, delay, call idling, delay  
reset stateMind to 0

if pdata is "stop":

call idling, delay, reset stateMind to 0

break

case 1:

stateMind = 2

break

default:

reset stateMind and terkirim to 0

call toggleTinyLed

predictor() function:

if terkirim == 0:

print EEG parameters and aTT, mED

set terkirim to 1

readOneByte() function:

wait for Serial2 data availability

return read byte from Serial2

wave(pin, frequency, duration) function:

calculate period from frequency

while duration not elapsed:

toggle pin HIGH and LOW with delayMicroseconds

switcher(gatePin) function:

set gatePin as OUTPUT, toggle LOW, delay, toggle HIGH



sayHeadsetConnected() function:

play sequence of tones

toggleTinyLed() function:

toggle tinyLEDState and set tinyLEDpin accordingly

idling() function:

move Dynamixel servos to idle positions with msstand speed

leftHand() function:

move Dynamixel servos for left hand motion

rightHand() function:

move Dynamixel servos for right hand motion

liftForward() function:

move Dynamixel servos for forward lift motion

liftBackward() function:

move Dynamixel servos for backward lift motion

dynamixelPing() function:

ping all Dynamixel servos

standBy() function:

delay, call preparing

preparing() function:

move Dynamixel servos to prepare positions

idlingTime() function:

move Dynamixel servos to idle time positions

getAttention() function:

initialize variables

read bytes from Serial2, validate checksum

analyze payload for attention, meditation, poor quality, and EEG parameters

return attention value

Lampiran 9 Pseudocode K Nearest Neighbor

IMPORT pustaka yang diperlukan (statistics, typing, pandas, numpy, matplotlib.pyplot, serial, datetime)

SETUP komunikasi serial pada 'COM4' dengan baud rate 115200

DEFINISIKAN label pergerakan sebagai dictionary

```
centroid_labels = {
    0: "Maju",
    1: "Mundur",
    2: "Kanan",
    3: "Kiri",
    4: "Berhenti"
}
```

LOAD data mentah dari file CSV

SET nama kolom untuk data mentah

KONVERSI data mentah menjadi list of lists

INISIALISASI sData sebagai tuple yang berisi 5 list kosong (untuk gerakan berbeda)

UNTUK setiap baris dalam list data mentah:

JIKA elemen terakhir adalah "Maju":

HAPUS elemen terakhir

TAMBAHKAN baris ke sData[0]

ELSE JIKA elemen terakhir adalah "Mundur":

HAPUS elemen terakhir

TAMBAHKAN baris ke sData[1]

ELSE JIKA elemen terakhir adalah "Kanan":

HAPUS elemen terakhir

TAMBAHKAN baris ke sData[2]

ELSE JIKA elemen terakhir adalah "Kiri":

HAPUS elemen terakhir

TAMBAHKAN baris ke sData[3]

ELSE JIKA elemen terakhir adalah "Berhenti":

HAPUS elemen terakhir

TAMBAHKAN baris ke sData[4]

GABUNGKAN semua sData menjadi satu list allData

KONVERSI allData ke array NumPy

KONVERSI allData ke DataFrame pandas

FUNGSI initialize\_centroids(k, data):

DAPATKAN jumlah dimensi dari data

DAPATKAN nilai minimum dan maksimum untuk setiap dimensi dari data

INISIALISASI list kosong untuk centroids

UNTUK k kali:

GENERATE centroid acak dalam rentang min-max

TAMBAHKAN centroid ke list centroids

KONVERSI list centroids ke DataFrame dengan kolom yang sama seperti data

RETURN DataFrame centroids

FUNGSI `calculate_error(a, b)`:

HITUNG error kuadrat antara a dan b

RETURN error

FUNGSI `assign_centroid(data, centroids)`:

INISIALISASI list kosong untuk assign centroid dan error

UNTUK setiap observasi dalam data:

INISIALISASI array kosong untuk error

UNTUK setiap centroid:

HITUNG error antara centroid dan observasi (10 kolom pertama)

TAMBAHKAN error ke array errors

TEMUKAN indeks error minimum dalam array errors

TAMBAHKAN indeks ke list assign centroid

TAMBAHKAN error minimum ke list centroid errors

RETURN list assign centroid dan list centroid errors

FUNGSI `kmeans(data, k)`:

INISIALISASI centroids dengan fungsi `initialize_centroids`

INISIALISASI list kosong untuk error

SET `compr` ke True

SET counter iterasi `i` ke 0

WHILE `compr` is True:

ASSIGN centroid ke data menggunakan fungsi assign\_centroid

APPEND jumlah iterasi errors ke list error

UPDATE centroid dengan menghitung rata-rata dari data yang diassign

IF list error memiliki kurang dari 2 elemen:

    SET compr ke True

ELSE:

    IF error saat ini dan sebelumnya tidak sama (dibulatkan ke 3 desimal):

        SET compr ke True

    ELSE:

        SET compr ke False

INCREMENT counter iterasi i

ASSIGN centroid akhir dan hitung error

UPDATE centroid dengan menghitung rata-rata dari data yang diassign

RETURN list assign centroid, iterasi errors, dan centroids yang diupdate

FUNGSI online\_kmeans\_update(data\_point, allData\_pd, centroids):

HITUNG errors antara data\_point dan setiap centroid

TEMUKAN indeks error minimum

TAMBAHKAN indeks centroid terdekat ke data\_point

BUAT DataFrame baru dengan data\_point\_with\_centroid

APPEND DataFrame baru ke allData\_pd

DAPATKAN poin yang diassign untuk centroid terdekat

HITUNG nilai centroid baru sebagai rata-rata dari poin yang diassign

UPDATE centroid dengan nilai baru

```
    RETURN allData_pd dan centroids yang diupdate
JALANKAN kmeans pada allData_pd dengan 5 centroids
PRINT assign centroid awal dan centroids
INISIALISASI prediction index ke 0
INISIALISASI move forward count ke 0
INISIALISASI init_time ke None
SET gerakan target ke "Kiri"

TRY:
    SET init_time ke datetime saat ini

WHILE True:
    IF ada data yang tersedia dari port serial:
        BACA baris dari port serial
        KONVERSI baris ke list of floats

        HITUNG errors antara data_point dan setiap centroid
        TEMUKAN indeks error minimum
        DAPATKAN timestamp saat ini

        DAPATKAN label prediksi dari centroid_labels

        IF label prediksi sesuai dengan gerakan target:
            INCREMENT move forward count

            IF move forward count sama dengan 5:
                HITUNG durasi sejak init_time
                PRINT waktu yang dibutuhkan untuk 5 kemunculan
                RESET move forward count
```

UPDATE init\_time ke timestamp saat ini

PRINT timestamp saat ini, prediction index, dan label prediksi

KIRIM perintah yang sesuai ke port serial berdasarkan label prediksi

UPDATE allData\_pd dan centroids dengan data point baru menggunakan fungsi online\_kmeans\_update

INCREMENT prediction index

EXCEPT KeyboardInterrupt:

CLOSE koneksi serial