

**PENGEMBANGAN KLASIFIKASI ANGKA BAHASA ISYARAT DENGAN
MENGUNAKAN METODE K-NEAREST NEIGHBOR**

TUGAS AKHIR



AY FEE CONELIA

NIM: 311810006

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG
MALANG
2023**

**LEMBAR PENGESAHAN
TUGAS AKHIR**

**PENGEMBANGAN KLASIFIKASI ANGKA BAHASA ISYARAT DENGAN
MENGUNAKAN METODE K-NEAREST NEIGHBOR**

Oleh:

AY FEE CONELIA

NIM. 311810006

dari:

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG**

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan dan berhak mendapatkan gelar Sarjana Komputer

Dosen Pembimbing 1,

Dosen Pembimbing 2,

Dr. Eng. Romy Budhi Widodo
NIP. 20070035

Ir. Oesman Hendra K. M.Div., M.Cs.
NIP. 20110022

Dekan Fakultas Teknologi dan Desain,

Dr. Eng. Romy Budhi Widodo
NIP. 20070035

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “PENGEMBANGAN KLASIFIKASI ANGKA BAHASA ISYARAT DENGAN MENGGUNAKAN METODE K-NEAREST NEIGHBOR” adalah benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan – bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang, September 2023

Ay Fee Conelia
311810006

PENGEMBANGAN KLASIFIKASI ANGKA BAHASA ISYARAT DENGAN MENGGUNAKAN METODE K-NEAREST NEIGHBOR

Ay Fee Conelia, RomyBudhi Widodo, Oesman Hendra.K
Universitas Ma Chung

Abstrak

Komunikasi merupakan suatu aktivitas penting saling bertukar informasi pada dua orang atau lebih dan dilakukan setiap saat dengan berbagai cara, sehingga komunikasi menjadi kebutuhan bagi setiap masyarakat tidak luput bagi penyandang difabel khususnya tunarungu dan tunawicara dengan menggunakan bahasa yang tentunya dimengerti dan telah resmi yaitu bahasa isyarat.

Di Indonesia terdapat 2 jenis bahasa isyarat yaitu Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO) namun bahasa isyarat yang telah diresmikan oleh Presiden Republik Indonesia adalah SIBI sehingga bahasa isyarat SIBI lebih sering digunakan dan diajarkan di sekolah-sekolah SLB/B.

Pada penelitian ini penulis bertujuan untuk mengklasifikasikan angka berserta satuannya serta mengubah hasil klasifikasi menjadi suara atau bisa disebut *text to speech* dengan menggunakan *Flex Point Glove Kit* yang telah dilengkapi oleh sensor tekuk/sensor Bend yang nantinya akan diklasifikasikan dengan menggunakan metode KNN, total data yang akan digunakan adalah 120 data yang berasal dari 5 subjek, yang setiap subjeknya melakukan 24 gerakan, pembuatan model dari klasifikasi ini menggunakan bahasa pemrograman python. Pada penelitian ini tujuan yang akan dicapai telah berhasil dilakukan yaitu membuat model klasifikasi angka serta mengubah output hasil klasifikasi menjadi suara dalam Bahasa Indonesia, dengan tingkat akurasi model KNN sebesar 93%

Kata kunci: Bahasa Isyarat, SIBI, KNN, *Text to speech*, *Flex point glove kit*, sensor tekuk/sensor bend.

DEVELOPMENT OF SIGN LANGUAGE NUMBERS CLASSIFICATION USING K-NEAREST NEIGHBOR METHOD

Ay Fee Conelia, Romy Budhi Widodo, Oesman Hendra.K
Universitas Ma Chung

Abstract

Communication is an important activity of exchanging information with two or more people and is carried out at any time in various ways, so that communication becomes a necessity for every community, including persons with disabilities, especially the deaf and mute, by using a language that is understandable and official, namely sign language.

In Indonesia there are 2 types of sign language, namely the Indonesian Sign Language System (SIBI) and Indonesian Sign Language (BISINDO).

In this research the author aims to classify numbers along with their units and change the classification results to sound or can be called *text to speech* using the *Flex Point Glove Kit* which has been equipped with a bending sensor / Bend sensor which will later be classified using the KNN method, the total data will be 120 data were used from 5 subjects, each of subjects made 24 movements, modeling for this classification used the Python programming language. In this study, the objectives to be achieved have been successfully carried out, namely to create a number classification model and change the output of the classification results to sounds in Indonesian, with a KNN model accuracy rate of 93%.

Keywords: *Sign language, SIBI, KNN, Text to speech, Flex point glove kit, sensor tekuk/sensor bend.*

KATA PENGANTAR

Puji Syukur kehadiran Tuhan Yang Maha Esa karena atas Rahmat dan hidayah-Nya sehingga penelitian ini dapat berjalan dengan baik, Laporan ini berjudul pengembangan klasifikasi Bahasa isyarat angka dengan menggunakan metode K- Nearest Neighbor, laporan ini disusun untuk memenuhi salah satu persyaratan dalam menyelesaikan TA (Tugas Akhir) bagi Mahasiswa dari Prodi Teknik Informatika Universitas Ma Chung Malang.

Atas dukungan moral dan materi yang telah diberikan dalam penyusunan laporan ini, penulis menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu terutama kepada :

1. Orang tua yang telah mendukung dalam pelaksanaan praktik kerja lapangan ini.
2. Bapak Dr. Eng. Romy Budhi Widodo, ST, MT. selaku dosen prodi Teknik Informatika dan pembimbing I Tugas Akhir.
3. Bapak Ir. Oesman Hendra.K selaku dosen prodi Teknik Informatika dan pembimbing II Tugas Akhir.
4. Bapak Mochamad Subianto selaku Dosen Prodi Teknik Informatika dan penguji Tugas Akhir.
5. Pusat Studi HMI Universitas Ma Chung Malang yang telah memberikan kesempatan dan fasilitas untuk melakukan Tugas Akhir.
6. Universitas Ma Chung Malang yang telah memberikan kesempatan untuk melakukan Tugas Akhir.
7. Saudara dan teman-teman yang telah memberikan dukungan moral.

Akhir kata, penulis berharap kepada Tuhan Yang Maha Esa membalas segala kebaikan semua pihak yang telah membantu. Semoga laporan Tugas Akhir ini dapat bermanfaat dan menambah ilmu pembaca.

Malang, September 2023

Ay Fee Conelia
311810006

Daftar Isi

LEMBAR PENGESAHAN	ii
PERNYATAAN KEASLIAN TUGAS AKHIR	iii
Abstrak	iv
Abstract.....	v
KATA PENGANTAR.....	vi
Daftar Gambar	ix
DAFTAR TABEL	xi
Bab I	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah	2
1.4 Rumusan Masalah	2
1.5 Tujuan Penelitian.....	2
1.6 Luaran.....	2
1.7 Manfaat.....	3
1.8 Sistematika Penulisan	3
Bab II	5
2.1 Flexpoint USB Glove kit	5
2.2 Sensor Bend.....	6
2.3 Sensor Magnetometer	7
2.4 Sensor Akselerometer.....	7
2.5 Bahasa Isyarat.....	9
2.6 Artificial Intelligence.....	10
2.7 Machine Learning	11
2.8 K-Nearest Neighbors (KNN).....	11
2.9 Python.....	13
2.10 Scikit Learn	13
2.11 Numerical Python (NumPy)	14
2.12 Pandas.....	15
2.13 Pyttsx3	16
2.14 Confusion Matrix.....	16

2.15	Google Colaboratory	19
2.16	gTTS (Google Text-to-Speech)	19
2.17	Anaconda	20
2.18	Jupyter Notebook.....	21
Bab III.....		24
3.1	Alur penelitian	24
3.2	Desain Sistem	25
3.3	Penelitian Terdahulu.....	26
3.4	Analisis Kebutuhan	27
3.5	Studi Pustaka	27
3.6	Gerakan Gestur Tangan	27
3.7	Pembacaan Data oleh Sensor.....	28
3.8	USB Glove Kit	29
3.9	Pengumpulan Data.....	29
3.10	Pembelajaran Data Menggunakan K-Nearest Neighbor	30
3.11	Implementasi Sistem	31
3.12	Rancangan Pengujian	31
3.13	Text to Speech	34
Bab IV		35
4.1	Pengumpulan Data.....	35
4.2	Proses Pengolahan Data	37
4.3	Proses Pembuatan Model KNN.....	39
4.4	Hasil Pengujian KNN	42
4.5	Pengujian Konversi	43
4.6	Pengujian Menggabungkan 2 data.....	44
4.7	Pengujian Menggabungkan 2 data lebih.....	45
4.8	Pengujian Text to Speech	48
Bab V		50
5.1	Kesimpulan.....	50
5.2	Saran	50
Daftar Pustaka		51
Lampiran.....		53

Daftar Gambar

Gambar 2.1 USB Glove Kit.....	5
Gambar 2.2 Visualisasi Glove Kit.....	6
Gambar 2. 3 Sensor Bend	7
Gambar 2. 4 Sensor Akselerometer KMX62 (Kionix, 2015).....	8
Gambar 2.5 Isyarat Angka SIBI	10
Gambar 2. 6 Panduan Isyarat Angka SIBI (kemdikbud, 2020).....	10
Gambar 2. 7 Diagram KNN (Christopher, 2021).	12
Gambar 2. 8 Geometri Euclidean (Christopher, 2021).....	12
Gambar 2. 9 Bentuk Dimensi Array (Tris, 2019).....	14
Gambar 2. 10 Confussion Matrix (Aras, 2020)	16
Gambar 2. 11 Confusion Matrix untuk Accuracy (Nugroho, 2019)	17
Gambar 2. 12 Confusion Matrix untuk Precision (Nugroho, 2019).....	18
Gambar 2. 13 Confusion Matrix untuk Recall atau Sensitivity (Nugroho, 2019).....	18
Gambar 2. 14 Tampilan Google Colaboratory	19
Gambar 2. 15 Website Anaconda	20
Gambar 2. 16 Tampilan Anaconda.....	20
Gambar 2. 17 Penginstallan Jupyter Notebook	21
Gambar 2. 18 Tampilan Jupyter Notebook	22
Gambar 3. 1 Alur Penelitian	24
Gambar 3. 2 Visualisasi Glove Kit.....	28
Gambar 3. 3 Ukuran tangan berdasarkan berat (Chaerunnisa, 2017).....	32
Gambar 3. 4 Proses pengambilan data dari subjek.....	32
Gambar 3. 5 Alur pengambilan data.....	33
Gambar 3. 6 Ilustrasi text to speech	34
Gambar 4. 1 Pengambilan data.....	35
Gambar 4. 2 Tampilan pengambilan data.....	36
Gambar 4. 3 Tampilan file yang tersimpan	36
Gambar 4. 4 Penggabungan data pertama	38
Gambar 4. 5 Penggabungan data kedua	38
Gambar 4. 6 Tampilan file yang sudah diolah.....	39
Gambar 4. 7 Import library.....	39
Gambar 4. 8 Import dan menampilkan data	40
Gambar 4. 9 Membagi data set.....	40
Gambar 4. 10 Tranformasi menggunakan MinMaxScaler	40
Gambar 4. 11 Grafik error rate nilai K	41
Gambar 4. 12 Rincian error rate	41
Gambar 4. 13 K dengan error rate terkecil	41
Gambar 4. 14 Classification report KNN	42
Gambar 4. 15 Konversi data.....	43

Gambar 4. 16 Hasil konversi	43
Gambar 4. 17 Memasukan 2 data csv	44
Gambar 4. 18 Pemilihan indeks.....	44
Gambar 4. 19 Prediksi KNN.....	44
Gambar 4. 20 Mengubah nama variabel.....	44
Gambar 4. 21 Proses konversi	45
Gambar 4. 22 Hasil penggabungan data	45
Gambar 4. 23 Menginput 4 data	46
Gambar 4. 24 Mengklasifikasikan data dengan menggunakan model KNN	46
Gambar 4. 25 Konversi data	47
Gambar 4. 26 Tampilan hasil konversi.....	47
Gambar 4. 27 Hasil	47
Gambar 4. 28 Proses instal pyttx3	48
Gambar 4. 29 Proses install gtts.....	48
Gambar 4. 30 Rincian suara dan bahasa.....	48

DAFTAR TABEL

Tabel 3. 1 Spesifikasi Komputer.....	26
Tabel 3. 2 Gambar dan skema microcontroller.....	29
Tabel 3. 3 Spesifikasi Komputer.....	32

Bab I

Pendahuluan

1.1 Latar Belakang

Komunikasi merupakan suatu aktivitas saling bertukar informasi pada dua orang atau lebih yang pasti dilakukan setiap saat dengan berbagai cara, sehingga komunikasi menjadi kebutuhan bagi setiap Masyarakat, tidak luput bagi penyandang difabel khususnya tunarungu dan tunawicara.

Salah satu cara untuk berkomunikasi dengan tunarungu dan tunawicara adalah dengan menggunakan bahasa isyarat, namun bahasa isyarat yang bersifat universal tidak ada. Di Indonesia terdapat 2 jenis bahasa isyarat yaitu Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO).

SIBI merupakan bahasa isyarat yang telah diresmikan oleh Presiden Republik Indonesia sehingga bahasa isyarat SIBI lebih sering digunakan dan diajarkan di sekolah-sekolah SLB/B, “Pihak sekolah dan juga para guru menggunakan SIBI sebagai bahasa pengantar materi pembelajaran pada siswa Tuli” (Winarsih, 2007).

Bahasa Isyarat SIBI merupakan bahasa nasional yang digunakan di Indonesia dan harus dikuasai namun bahasa isyarat SIBI cukup sulit untuk dipelajari baik itu untuk penyandang disabilitas ataupun bukan penyandang disabilitas, karena bahasa yang digunakan dalam SIBI merupakan bahasa Indonesia yang baku, sehingga cukup sulit untuk dimengerti dan dipelajari.

Teknologi *Text To Speech* (TTS) merupakan suatu sistem yang mengkonversi teks menjadi ucapan menyerupai ucapan manusia dalam berbagai bahasa, dengan menggunakan *Text To Speech* (TTS) dapat memudahkan pengenalan hasil output pada klasifikasi.

Penelitian ini bertujuan untuk mengembangkan penelitian sebelumnya penerjemah bahasa isyarat SIBI dengan menggunakan alat USB Glove Kit, yang terdapat sensor tekuk di setiap ruas jari yang datanya dapat diambil dan diolah menggunakan komputer.

Pada penelitian sebelumnya bahasa isyarat SIBI yang dapat diterjemahkan adalah angka sampai 30, dilanjutkan dengan penelitian abjad a-z dengan menggunakan metode

KNN dengan output berupa teks. Untuk pengembangan yang ingin dilakukan penulis adalah untuk mengembangkan penelitian sebelumnya agar dapat menerjemahkan angka hingga jutaan serta menampilkan hasil klasifikasi dalam bentuk teks dan suara.

Berdasarkan penjelasan latar belakang di atas, diharapkan penelitian ini dapat menjawab permasalahan utama dan memenuhi harapan.

1.2 Identifikasi Masalah

Berdasarkan latar belakang, maka identifikasi masalah adalah

1. Bahasa isyarat SIBI yang bersifat baku cukup sulit untuk dipelajari.
2. Penelitian terdahulu hanya dapat mengklasifikasikan angka hingga 30.
3. Tidak adanya fitur teks to speech pada penelitian sebelumnya dan hanya mengeluarkan output *single*.

1.3 Batasan Masalah

Batasan masalah pada praktik kerja lapangan ini sebagai berikut,

1. Sensor tekuk/ sensor bend yang terdapat dalam sarung tangan hanya membaca gerakan tangan atau jari.
2. Identifikasi isyarat hanya angka hingga jutaan dengan batas maksimal yakni 999.999.999 dalam Bahasa Isyarat SIBI yang diambil dari kamus resmi SIBI.

1.4 Rumusan Masalah

Bagaimana menambahkan kemampuan alat USB glove kit untuk mengklasifikasikan angka dalam bahasa isyarat hingga jutaan serta mengeluarkan output dalam bentuk teks dan suara dalam Bahasa Indonesia.

1.5 Tujuan Penelitian

Menemukan cara untuk mengklasifikasikan angka hingga jutaan dengan metode machine learning serta menampilkan hasil dalam bentuk teks dan suara Bahasa Indonesia.

1.6 Luaran

Luaran dari tugas akhir ini adalah untuk klasifikasi angka isyarat SIBI hingga jutaan yang dapat disuarakan dalam Bahasa Indonesia.

1.7 Manfaat

Manfaat yang diperoleh dari penelitian adalah sebagai berikut:

- a. Bagi Ma Chung Human Machine Interaction Research Center (HMI)
Dengan dibuatnya penelitian ini, diharapkan dapat menambah bahan bacaan mahasiswa yang melakukan penelitian di Ma Chung Human Machine Interaction Research Center.
- b. Bagi Mahasiswa
Menambah wawasan dan pengalaman dalam menggunakan USB Glove kit serta dapat menarik minat untuk mempelajari bahasa isyarat.
- c. Bagi Universitas Ma Chung
Dapat menambah bahan belajar bagi mahasiswa.

1.8 Sistematika Penulisan

Sistematika penulisan penelitian ini adalah sebagai berikut :

Bab I : Pendahuluan

Pada bab ini akan dijelaskan latar belakang dari penelitian ini, berupa rumusan masalah, indentifikasi masalah, tujuan, sistematika penulisan, luaran, serta manfaat dari penelitian yang dilakukan.

Bab II : Tinjauan Pustaka

Pada bab ini akan menjelaskan teori-teori pendukung penelitian yang dilakukan, Seperti menjelaskan apa itu *Flexpoint USB Glove kit*, *Sensor Bend*, *Sensor Magnetometer*, *Sensor Akselerometer*, *Bahasa Isyarat*, *Artificial Intelligence*, *Machine Learning*, *K-Nearest Neighbors (KNN)*, *Python*, *Scikit Learn*, *Numerical Python (NumPy)*, *Pandas*, *Pytsx3*, *Confusion Matrix*, *Google Colaboratory*, *gTTS (Google Text-to-Speech)*, *Anaconda*, *Jupyter Notebook*.

Bab III : Analisis dan Rancangan Sistem

Pada bab ini akan menjelaskan lebih rinci lagi mengenai Alur penelitian, Analisis Kebutuhan, Desain Sistem, sejumlah refrensi yang berguna untuk membantu proses penelitian, Gerakan Gestur Tangan, Pembacaan Data oleh Sensor, USB Glove Kit, Pengumpulan Data, Pembelajaran Data Menggunakan K-Nearest Neighbor, Implementasi Sistem, Rancangan Pengujian, Text to Speech

Bab IV : Hasil dan Pembahasan

Pada bab ini akan dijelaskan proses dan hasil yang didapatkan dari pembuatan model hingga program berhasil berjalan, dimulai dari pembahasan mengenai Pengumpulan Data, Proses Pengolahan Data, Proses Pembuatan Model KNN, Hasil Pengujian KNN , Pengujian Konversi, Pengujian Menggabungkan 2 data, Pengujian Menggabungkan 2 data lebih, Pengujian Text to Speech.

Bab V : Kesimpulan dan Saran

Pada bab ini mengandung kesimpulan dari hasil yang diperoleh dari penelitian yang dilakukan serta saran bagi peneliti lain agar dapat meningkatkan tingkat keberhasilan dalam membuat alat bantu dalam menerjemahkan bahasa isyarat.

Bab II

Tinjauan Pustaka

2.1 Flexpoint USB Glove kit

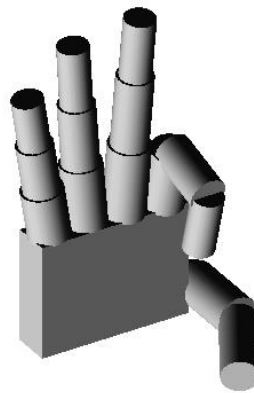
Flexpoint Glove kit adalah sebuah sarung tangan yang dilengkapi dengan sensor tekuk (bend sensor) disetiap ruas jari dan mikrokontroler. USB Glove kit merupakan produk hasil dari Flexpoint Sensor System (FLXT) yang merupakan pemasok terkemuka dalam hal teknologi sensor panca indra dalam bidang industry, termasuk otomotif, medis, control industri, dan produk konsumen, dalam pembelian Flexpoint Glove kit sudah termasuk dengan sarung tangan, sensor tekuk, dan mikrokontroler. Untuk tampilan dari USB Glove Kit dapat dilihat pada Gambar 2.1.



Gambar 2.1 USB Glove Kit

Flexpoint Glove kit memiliki lima buah sensor tekuk atau sensor bend yang berisi tiga buah sensor panjang dan dua buah sensor pendek. Tiga buah sensor panjang digunakan pada jari telunjuk, jari tengah, dan jari manis. Sedangkan untuk dua buah sensor pendek digunakan pada jari jempol dan jari kelingking. Fungsi dari sensor bend ini adalah mendeteksi apakah jari – jari pada tangan tegak lurus atau tertekuk. sebuah mikrokontroler yang sudah dipasang Kionix KMX62 yang merupakan sebuah sensor yang berisi sensor akselerometer, dan sensor magnetometer.

Selain itu Glove Kit, juga diberikan software (perangkat lunak) untuk menjalankan programnya, software tersebut adalah program C# yang membaca data serial port dari virtual COM port dan ditampilkan menggunakan OpenGL untuk menunjukkan pergerakan sensor secara 3D virtual dalam layar. Selain menunjukkan virtualisasi pergerakan sensornya, kita juga dapat melihat data-data numerik yang terbaca pada sensor dengan membuka sebuah emulator serial. Bentuk data-data yang dikeluarkan adalah dua buah tekukan pada masing-masing jari, tiga buah sumbu dari akselerometer dan magnetometer. Untuk visualisasi dari Glove kit dapat dilihat pada gambar 2.2



Gambar 2.2 Visualisasi Glove Kit

2.2 Sensor Bend

Sensor Bend merupakan sebuah sensor yang mengukur tikungan (tekukan), yang resistansinya dapat berubah saat ditekuk. Pada sarung tangan USB Glove Kit sensor bend berfungsi untuk

- a. Mengukur gerakan jari: Sensor bend yang terpasang pada setiap jari sarung tangan USB Glove Kit dapat digunakan untuk mengukur pergerakan setiap jari. Hal ini memungkinkan pengguna untuk mengontrol perangkat
- b. Pengenalan gestur : dengan menggabungkan data dari sensor bend yang terpasang pada setiap jari, USB Glove Kit dapat digunakan untuk mengenali Gerakan/gestur tangan. Gestur ini dapat dipahami sebagai perintah dalam aplikasi desain 3D, atau kontrol komputer secara umum.

terdapat lima buah sensor bend yang terdiri dari tiga sensor berukuran panjang yang terdapat pada jari telunjuk, jari tengah, jari manis dan dua sensor berukuran pendek yang terdapat pada ibu jari dan jari kelingking, untuk mendapatkan hasil yang baik maka pemasangan sensor harus dilakukan dengan benar, tidak terbalik. (Flexpoint Sensor Systems, 2016) seperti pada Gambar 2.3.



Gambar 2. 3 Sensor Bend

2.3 Sensor Magnetometer

Magnetometer merupakan sensor yang digunakan untuk mengukur kekuatan dan arah medan magnet. Sensor Magnetometer diperkenalkan pada tahun 1833 oleh Carl Friedrich Gauss dengan fungsi untuk mengukur medan magnet bumi. Satuan internasional medan magnet adalah Tesla. Untuk pengukuran geomagnetik, satuannya adalah nanotesla (nT) (Bangkit, Harry; Ruhimat, Mamat, 2015), Satuan yang digunakan adalah Gauss dengan rincian $1 \text{ Gauss} = 100.000 \text{ nT}$, atau $1 \text{ Gauss} = 100.000 \text{ gamma}$.

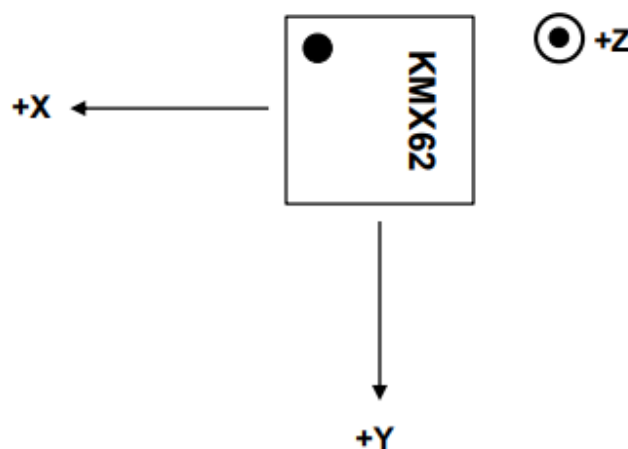
Terdapat 2 tipe Magnetometer, tipe pertama adalah magnetometer skalar, dengan fungsi hanya untuk mengukur kekuatan total medan magnet, dan tipe kedua adalah magnetometer vector dengan fungsi yang dapat mengukur besar dan arah medan magnet pada 3 koordinat yang dikenal dengan X,Y,Z atau komponen H,D,Z (Buletin, 2012)

2.4 Sensor Akselerometer

Akselerometer merupakan sensor yang digunakan untuk mengukur percepatan suatu benda, Akselerometer dapat mengukur percepatan dinamis dan percepatan statis.

Pengukuran dinamis adalah pengukuran percepatan benda bergerak, sedangkan pengukuran statis adalah pengukuran gravitasi bumi (Lab Immersa, 2018).

Sensor Akselerometer yang digunakan pada USB Glove kit adalah Kionix KMX62 milik perusahaan flexpoint sensor ini merupakan gabungan dari sensor akselerometer dan magnetometer yang memiliki tiga arah yaitu X, Y, dan Z. Nilai keluaran positif atau bertambah ketika perangkat digerakkan ke arah +X, +Y, atau +Z, sedangkan nilai keluaran negatif atau berkurang bila perangkat digerakkan ke -X, -Y, dan -Z (Kionix, 2015).



Gambar 2. 4 Sensor Akselerometer KMX62 (Kionix, 2015)

Pada USB Glove Kit Sensor Akselerometer berfungsi untuk :

- a. Pengukuran percepatan/akselerasi: Akselerometer digunakan untuk mengukur percepatan gerakan tangan atau perubahan kecepatan. Hal ini memungkinkan USB Glove Kit dapat mendeteksi gerakan tangan secara akurat. Misalnya, saat menggerakkan tangannya, akselerometer akan mengukur derajat percepatan ke berbagai arah.
- b. Deteksi Gerakan: Akselerometer digunakan untuk mendeteksi berbagai jenis gerakan tangan, termasuk gerakan atas, bawah, horizontal, dan rotasi.
- c. Gestur dan kontrol: Akselerometer dapat digunakan untuk mengenali gestur tangan. Misalnya, gestur tertentu dapat digunakan untuk mengontrol aplikasi,

seperti mengambil dan melepaskan objek virtual, menggeser jari melintasi layar, atau memilih item dari daftar.

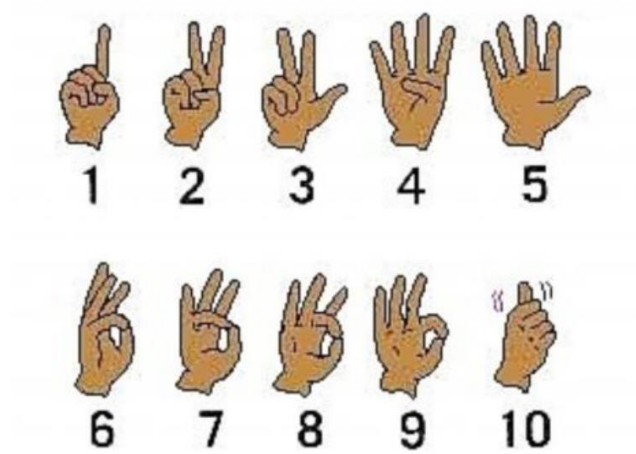
- d. Orientasi tangan: Akselerometer juga dapat membantu menentukan orientasi/arahan tangan, sensor dapat mendeteksi apakah tangan menghadap ke atas, ke bawah, ke samping, atau pada posisi tertentu. Informasi ini dapat digunakan untuk mengontrol objek virtual atau memicu tindakan berdasarkan arah tangan

2.5 Bahasa Isyarat

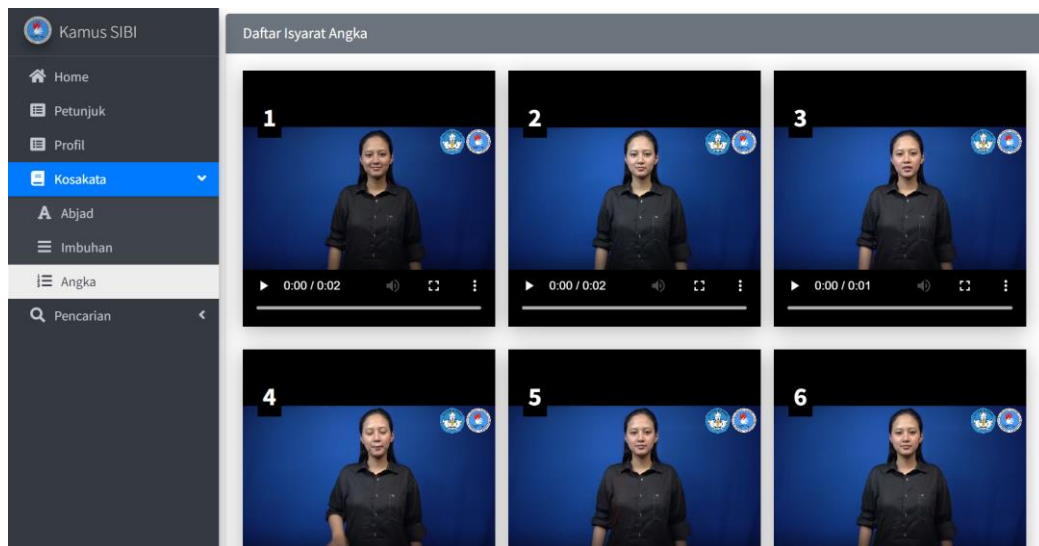
Bahasa adalah sistem komunikasi yang digunakan untuk menyampaikan pikiran, ide, perasaan, dan informasi kepada orang lain. Dalam hal ini, bahasa isyarat adalah jenis bahasa yang lebih mengandalkan ekspresi tubuh, pergerakan bibir, dan gerakan tangan atau jari (Krisnan, 2017). Bahasa isyarat ini mempunyai struktur yang berbeda dengan bahasa lisan, sama seperti bahasa lisan, bahasa isyarat secara alami terbentuk sesuai dengan budaya masyarakat masing-masing (Hadi, 2021).

Pada saat ini, diketahui ada sekitar 300 bahasa isyarat yang berbeda di seluruh dunia. di Indonesia, terdapat dua kategori bahasa gerak yang paling sering dipakai dalam berkomunikasi yaitu Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (Bisindo) (Murabagja, 2022).

Dalam penelitian ini Bahasa isyarat yang digunakan adalah Sistem Bahasa Isyarat Indonesia (SIBI) hal ini dikarenakan SIBI merupakan bahasa isyarat formal wajib yang sudah ditetapkan oleh pemerintah dan diajarkan di sekolah luar biasa (SLB) sehingga memiliki tata bahasa yang konsisten sehingga penggunaan pada setiap daerah tetap sama serta memiliki sumber yang terjamin yakni kamus resmi bahasa isyarat SIBI, sementara BISINDO merupakan bahasa isyarat yang berkembang secara alami pada masyarakat sehingga bahasa isyarat BISINDO memiliki variasi yang berbeda pada setiap daerahnya dan memiliki tata bahasa yang berbeda dengan bahasa lisan yang digunakan oleh orang normal pada umumnya sehingga untuk penelitian menggunakan bahasa isyarat BISINDO tidak bisa dilakukan karena bahasa isyarat BISINDO kurang konsisten. Untuk bahasa isyarat dan panduan bahasa isyarat yang digunakan dapat dilihat pada Gambar 2.5 dan Gambar 2.6.



Gambar 2.5 Isyarat angka SIBI



Gambar 2. 6 Panduan isyarat angka SIBI (*kemdikbud, 2020*).

2.6 Artificial Intelligence

Kecerdasan buatan (AI) merupakan cabang pengetahuan komputer yang berkonsentrasi pada perbaikan sistem komputer yang mampu menjalankan aktivitas yang pada umumnya terkait dengan kecerdasan manusia. Kecerdasan buatan (AI) bertujuan untuk menciptakan program atau mesin komputer yang dapat memahami, belajar, berpikir, dan berperilaku seperti manusia. Pada saat yang sama, Menurut McLeod dan Schell, kecerdasan buatan terdiri dari kemampuan mesin, seperti komputer, untuk menunjukkan perilaku yang dianggap cerdas dengan cara yang dilakukan manusia untuk

menunjukkan kemampuan tersebut. (Dicoding, 2020). Dengan menggunakan data untuk dijadikan sumber pengetahuan AI dengan cara data yang telah dikumpulkan akan dianalisa untuk menemukan pola dan korelasinya, sehingga AI dapat memprediksi keadaan di masa depan. Contoh fitur AI yang berkembang saat ini antara lain: algoritma deep face (pendeteksi wajah) yang saat ini dikembangkan oleh facebook untuk mengenali wajah pada sebuah postingan, data mining (pengumpulan informasi), beberapa Rekomendasi *E-Commerce* yang akan menampilkan produk-produk yang sering dibutuhkan atau berkaitan dengan produk yang sebelumnya pernah dibeli, dan asisten virtual seperti Google Assistant atau Siri dari Apple.

2.7 Machine Learning

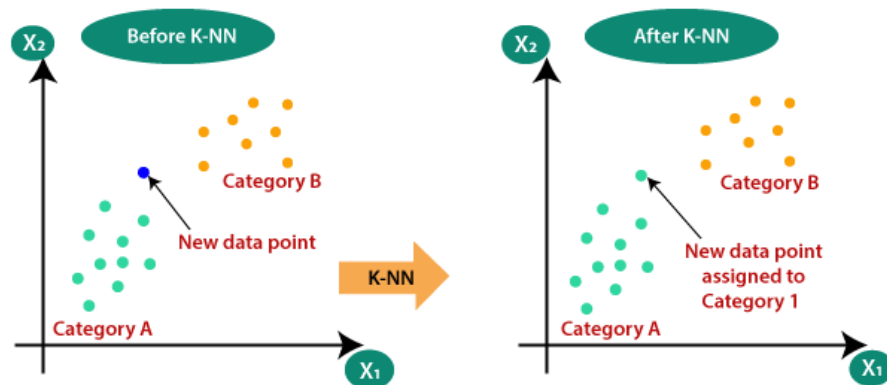
Machine learning adalah cabang kecerdasan buatan yang berfokus pada pengembangan algoritma dan model komputer yang mampu mempelajari pola dari data dan mengambil keputusan, menentukan atau melakukan tugas tanpa pemrograman eksplisit. Machine Learning memungkinkan sistem komputer untuk belajar dari pengalaman (data) dan meningkatkan kinerjanya seiring waktu tanpa harus diprogram ulang secara khusus untuk tugas tertentu. Sistem komputer menggunakan algoritma pembelajaran mesin untuk memproses data dalam jumlah besar dan mengidentifikasi pola dalam data. Hal ini memungkinkannya untuk memprediksi hasil dengan lebih akurat dari sekumpulan data masukan tertentu (Andhika, 2019).

Supervised learning, dan Unsupervised learning merupakan 2 jenis teknik pada Machine learning. supervised learning data dan label yang akan di train atau test sudah diketahui, namun pada unsupervised learning label tidak diketahui sehingga pada unsupervised learning melabel-label akan diberikan berdasarkan persamaan dari data-datanya. Pada penelitian ini jenis teknik yang digunakan adalah Supervised learning.

2.8 K-Nearest Neighbors (KNN)

K-nearest neighbor (KNN) adalah sebuah algoritma dalam machine learning yang mengklasifikasikan data berdasarkan data pelatihan (train record) yang diambil dari k tetangga terdekat (nearest neighbor), juga merupakan salah satu algoritma pembelajaran berbasis instansi (instance-based learning) yang sederhana dan umum digunakan dalam pengenalan pola dan analisis data, dengan menggunakan KNN untuk mengidentifikasi

katagori atau kelas dari suatu data set akan lebih mudah, sebagai contoh dapat dilihat pada Gambar 2.7.

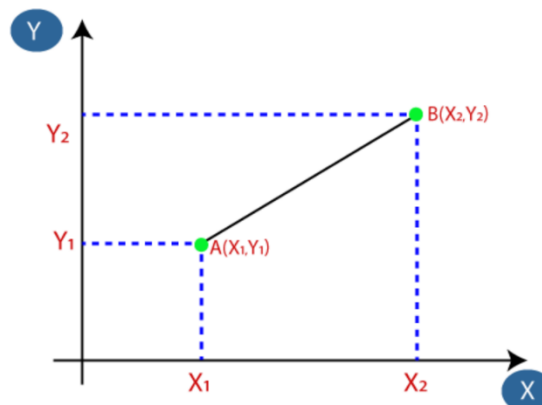


Gambar 2. 7 Diagram KNN (Christopher, 2021).

Algoritma dari KNN juga cukup mudah yaitu:

1. Menentukan banyaknya tetangga (K) yang akan digunakan untuk mempertimbangkan penentuan kelas.
2. Hitung jarak antara data baru dan setiap titik data dalam kumpulan data..
3. Ambil beberapa nilai K data dengan jarak terdekat, lalu menentukan kelas dari data baru tersebut.

Teknik pencarian tetangga terdekat dapat menggunakan persamaan jarak Euclidean dan setelah selesai dalam hal menentukan k, maka k akan digunakan untuk menidentifikasi data dan model dari KNN pun sudah siap Untuk persamaan jarak Euclidean menggunakan persamaan 2.1 dan Gambar 2.8.



Gambar 2. 8 Geometri Euclidean (Christopher, 2021)

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.1)$$

Keterangan:

x : Data Uji

y : Data Latih

2.9 Python

Python merupakan bahasa pemrograman yang populer serta sering digunakan untuk analisis data. Selain itu, Python didukung oleh banyak platform berbeda sehingga Python dapat digunakan di banyak sistem operasi berbeda seperti Windows, macOS, dan Linux. Python dirancang dengan sintaks yang sederhana dan mudah dibaca, sehingga cocok untuk pemula pemrograman. Hal ini memungkinkan pengembang untuk mengekspresikan ide-ide mereka dengan jelas dan efektif, selain itu python memiliki sifat yang Fleksibel Python digunakan dalam berbagai bidang termasuk pengembangan web, ilmu data, kecerdasan buatan, pengembangan perangkat lunak desktop, pemrograman jaringan, dan banyak lagi. Ada banyak perpustakaan dan modul Python yang tersedia untuk berbagai tugas, membuatnya sangat fleksibel.

Pada tahun 1991, sebuah bahasa pemrograman baru pertama kali dikembangkan oleh seorang individu bernama Guido van Rossum. Python tetap dalam tahap pengembangan oleh Python Software Foundation sampai sekarang. Bahasa Python kompatibel dengan hampir semua sistem operasi, termasuk sistem operasi Linux. Sebagian besar distribusi juga telah menyertakan Python (Jose, 2021).

2.10 Scikit Learn

Scikit-Learn atau *Sklearn* adalah pustaka bahasa pemrograman Python yang dikembangkan menggunakan *NumPy*, *SciPy*, dan *Matplotlib*. Pustaka ini berperan penting dalam pemrosesan data dan pelatihan data untuk keperluan pembelajaran machine-learning (Andhika, 2019).

Scikit-Learn atau *sklearn* memiliki banyak fitur dalam memudahkan pengolahan data yaitu: *Classification*, *Regression*, *Clustering*, *Dimensionality reduction*, *Model selection*, dan *Preprocessing data*. *Module* ini sangat sering digunakan terutama dalam

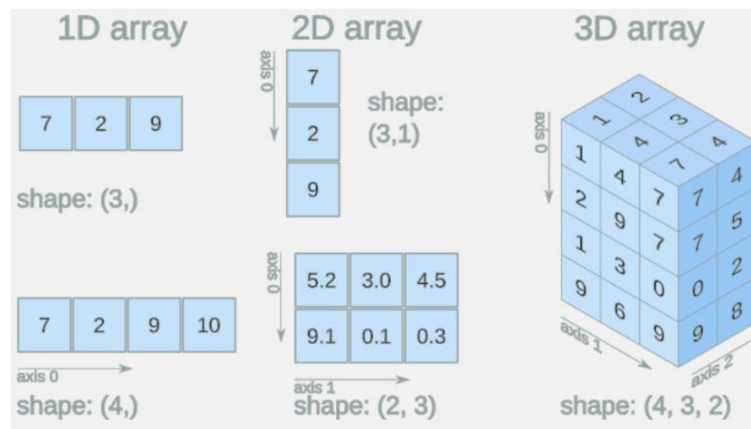
machine learning, pada penelitian ini Scikit-Learn atau sklearn digunakan pada tahap preprocessing baik untuk metode KNN.

2.11 Numerical Python (NumPy)

Numerical Python, atau yang biasa disebut sebagai *Numpy*, merupakan salah satu pustaka dalam bahasa Python yang digunakan untuk melakukan perhitungan numerik. *Numpy* menyediakan berbagai jenis struktur data dan fungsi yang kuat untuk bekerja dengan array dan matriks multidimensi, serta operasi matematika yang efisien untuk menganalisis data numerik. *NumPy* memiliki kemampuan untuk membuat objek array dengan dimensi N, yang menyerupai daftar pada Python (Atmaja, 2020).

NumPy juga dapat digunakan untuk memanipulasi Array dalam Python sehingga penggunaan *NumPy* dapat memudahkan proses komputasi data, hal tersebut dapat dilakukan dengan beberapa metode yaitu : *Indexing*, *Slicing* dan *Reshaping*.

Pada penelitian ini *NumPy* digunakan untuk menentukan dan melihat nilai input dan output pada data dengan menggunakan metode Indexing dan dengan mengubah beberapa dimensi array, Untuk melihat bentuk array dalam dimensi, bisa dilihat di ilustrasi Gambar 2.9



Gambar 2. 9 Bentuk Dimensi Array (Tris, 2019)

Bentuk array pada gambar diatas dapat dibagi menjadi berikut :

1. Array 0D, misalnya skalar $a = \text{np.array}(2)$.
2. Array 1D, misalnya vektor dengan bentuk (5),(6),(7).
3. Array 2D, misalnya matriks berbentuk (5,1), (2, 5), (3, 5).
4. Array 3D, misalnya matriks berbentuk (2,3,4), (5,2,3), (6,5,4).
5. Array ND, misalnya matriks berukuran (n, 5,4,3), (n, n, 4,3,2).

2.12 Pandas

Pandas merupakan perpustakaan Python yang khusus ditujukan untuk mengerjakan analisis data, termasuk pemrosesan data, pengolahan data, dan penghapusan data. Pandas memberikan format dan metode data tingkat tinggi untuk meningkatkan kecepatan, kemudahan, dan ekspresivitas dalam bekerja dengan data terstruktur/tabel (Rohman, 2019). Pandas memiliki banyak sekali kegunaan dan keunggulan terutama dalam bidang analisis data yang biasanya memakan banyak waktu diantaranya :

1. Data Cleansing
2. Data Fill
3. Normalisasi Data
4. Visualisasi data
5. Analisis Statistik
6. Inspeksi Data
7. Penggabungan dan penyatuan data
8. Memuat dan menyimpan data
9. Membersihkan datasets pada machine learning.
10. Pembuatan DataFrame sekaligus melakukan analisis serta manipulasi dan menyimpan database

Kelebihan pandas adalah sebagai berikut:

1. Pandas dapat digunakan dengan perpustakaan lain karena dibangun menggunakan NumPy, yang berarti banyak struktur perpustakaan yang digunakan atau diduplikasi di pandas.
2. Program perpustakaan Pandas sendiri dapat dijalankan menggunakan berbagai editor teks seperti Google Colab dan Jupyter Notebook.
3. Data yang dihasilkan oleh pandas dapat digunakan sebagai masukan untuk fungsi plot di Matplotlib, analisis statistik di SciPy, serta untuk algoritma pembelajaran mesin di Scikit-learn.
4. Pandas dapat mengimpor data dari berbagai format seperti CSV, Excel, database SQL, dan data dengan sumber web seperti HTML dan JSON. Hal ini memudahkan untuk memuat dan mengakses data dari berbagai sumber

2.13 Pyttsx3

Pyttsx3 merupakan library yang ada pada python yang digunakan untuk mengkonversikan *text-to-speech* (TTS). Sehingga program Python yang dapat mengonversi teks menjadi audio dan memutarnya melalui speaker atau headphone tanpa perlu menggunakan koneksi internet namun pilihan bahasa yang dapat digunakan sangat terbatas karena bahasa atau suara yang dapat digunakan adalah bahasa dari perangkat itu sendiri.

2.14 Confusion Matrix

Confusion matrix adalah tabel yang menyajikan informasi perbandingan antara hasil klasifikasi oleh sistem (model) dengan hasil klasifikasi yang sebenarnya. *Confusion matrix* adalah matriks yang mengilustrasikan capaian suatu model pemisahan pada sekelompok data uji yang angka sebenarnya sudah diketahui. Ilustrasi dari confusion matrix dapat dilihat pada gambar 2.10.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Gambar 2. 10 Confussion Matrix (Aras, 2020)

Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada confusion matrix yaitu:

- True Positives (TP) Jumlah kasus di mana model memprediksi kelas positif dengan benar. Artinya, model memprediksi bahwa sebuah contoh adalah positif dan bahwa contoh tersebut sebenarnya positif.

- b. True Negatives (TN) jumlah kasus di mana model memprediksi kelas negatif dengan benar. Model memprediksi bahwa sebuah contoh negatif dan, pada kenyataannya, contoh itu negatif.
- c. False Positives (FP) dikenal sebagai *type 1 error*. Ini adalah berapa kali model salah memprediksi kelas positif padahal seharusnya negatif. Model memprediksi bahwa sebuah contoh adalah positif, tetapi pada kenyataannya contoh tersebut adalah negatif.
- d. False Negatives (FN) dikenal sebagai *type 2 error*. Ini adalah jumlah kasus di mana model salah memprediksi kelas negatif padahal seharusnya positif. Model memprediksi bahwa sebuah contoh adalah negatif, tetapi pada kenyataannya contoh tersebut adalah positif.

Terdapat beberapa hasil perhitungan yang didapat dari confusion matrix yaitu :

a. Accuracy

Akurasi adalah ukuran yang menjelaskan tingkat keakuratan model dalam melakukan klasifikasi. Dengan demikian, keakuratan dapat dihitung sebagai perbandingan prediksi yang benar (baik dalam hal positif maupun negatif) terhadap seluruh sekumpulan data. Jadi, dapat diinterpretasikan bahwa tingkat keakuratan mencerminkan sejauh mana prediksi nilainya mendekati nilai yang sebenarnya. Ilustrasi persamaan yang digunakan untuk mencari tingkat akurasi dapat dilihat pada Gambar 2.11.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Confusion matrix yang menggambarkan nilai accuracy.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Gambar 2. 11 Confusion Matrix untuk Accuracy (Nugroho, 2019)

b. Precision

Precision menjelaskan tingkat akurasi antara data yang diminta dan hasil prediksi yang diberikan oleh model. Maka dapat disimpulkan bahwa Precision adalah rasio prediksi positif yang benar terhadap keseluruhan hasil prediksi positif. Ilustrasi persamaan yang digunakan untuk mencari Precision dapat dilihat pada gambar 2.12.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Confusion matrix yang menggambarkan nilai precision.

$$precision = \frac{TP}{TP + FP} \quad (2)$$

Gambar 2. 12 Confusion Matrix untuk Precision (Nugroho, 2019)

c. Recall atau Sensitivity

Recall atau Sensitivity menjelaskan keberhasilan model dalam mengambil informasi. Maka dapat disimpulkan bahwa Recall atau Sensitivity adalah tingkat prediksi positif yang benar untuk semua data nyata yang positif. Ilustrasi persamaan yang digunakan untuk mencari tingkat Recall atau Sensitivity dapat dilihat pada gambar 2.13.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Confusion matrix yang menggambarkan nilai recall.

$$recall = \frac{TP}{TP + FN} \quad (3)$$

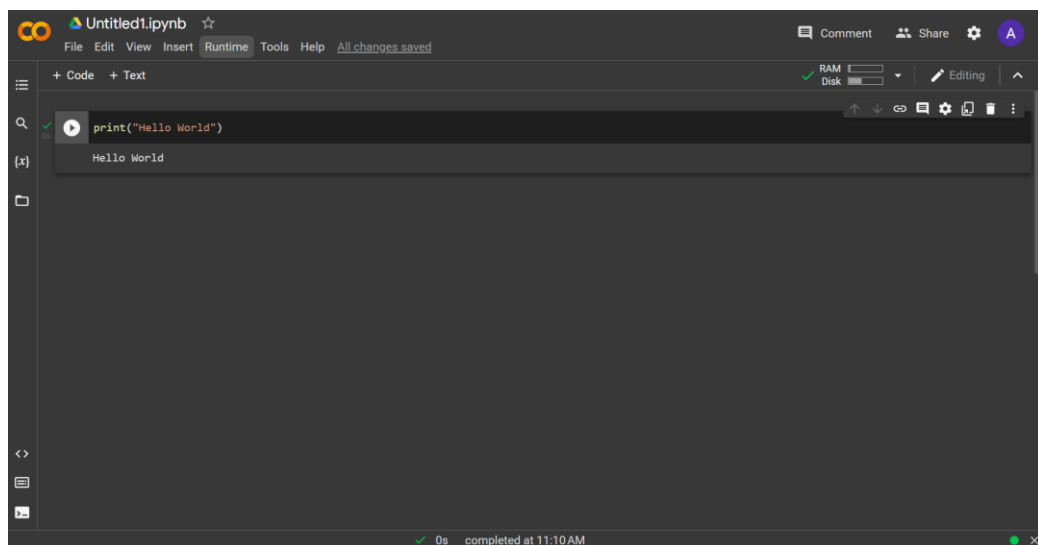
Gambar 2. 13 Confusion Matrix untuk Recall atau Sensitivity (Nugroho, 2019)

2.15 Google Colaboratory

Google Colaboratory atau biasanya dikenal dengan sebutan google colab merupakan salah satu fasilitas yang disediakan oleh google yang dapat membantu user dalam melakukan pengolahan data secara online dengan hanya menggunakan akun google saja tanpa menginstal software tambahan yang dapat diakses secara gratis namun pada versi lebih tinggi perlu biaya untuk mengupgradenya. Terdapat 3 jenis prosesor yang digunakan untuk keperluan pengolahan data yaitu machine learning yang telah disediakan oleh Google Colaboratory yaitu :

1. Central Processing Unit (CPU),
2. Graphic Processing Unit (GPU)
3. Tensor Processing Unit (TPU).

Pada penelitian ini mulanya menggunakan Google Colaboratory namun karena banyaknya data yang diolah sehingga Google Colaboratory versi tidak berbayar tidak dapat mengolah data yang akan digunakan pada penelitian ini. Untuk Tampilan Google Colaboratory dapat dilihat pada Gambar 2.14.



Gambar 2. 14 Tampilan Google Colaboratory

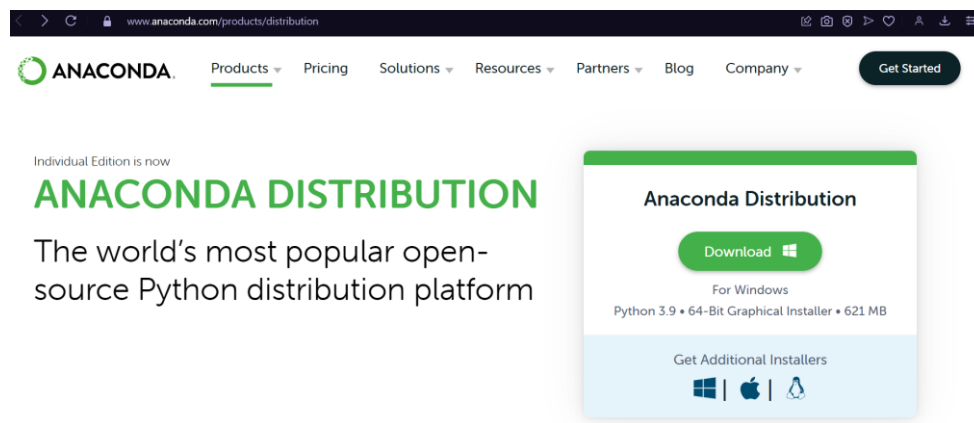
2.16 gTTS (Google Text-to-Speech)

gTTS (Google Text-to-Speech) adalah library Python yang disediakan oleh Google yang digunakan untuk konversi *text-to-speech* dengan menggunakan layanan TTS Google Terjemahan untuk mengonversi teks menjadi ucapan sehingga memerlukan koneksi internet untuk menggunakan gTTS karena data teks harus dikirim ke server

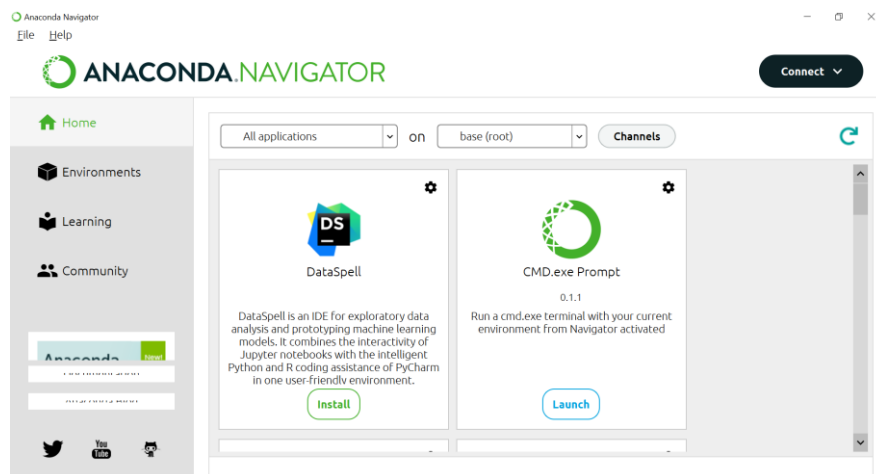
Google dan audio yang dihasilkan dikirim kembali ke program. gTTS digunakan karena memiliki keunggulan dalam mendukung banyak bahasa yang ditawarkan oleh Google Terjemahan

2.17 Anaconda

Anaconda merupakan sebuah paket distribusi Python yang dikembangkan oleh Continuum Analytics yang berisi kumpulan paket Python beserta beberapa paket tambahan yang secara khusus digunakan untuk pemrograman ilmu data, matematika, dan teknik dalam satu distribusi tunggal. Paket ini dirancang untuk memudahkan dalam melakukan instalasi, serta dapat digunakan dengan mudah di berbagai platform. Pemasangan platform ini juga sangat mudah dan didapatkan secara gratis pada website official anaconda seperti pada Gambar 2.15 dan Gambar 2.16 tampilan Anaconda setelah diinstal dan dijalankan.



Gambar 2. 15 Website Anaconda



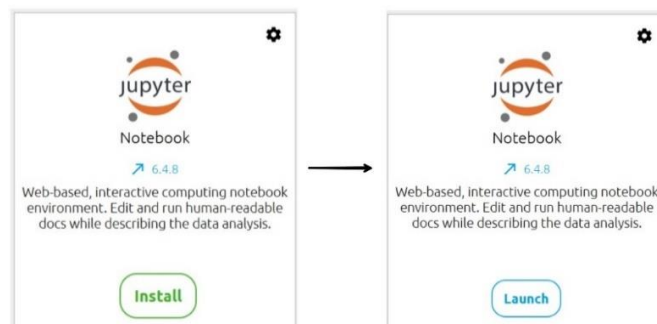
Gambar 2. 16 Tampilan Anaconda

Selain itu, Anaconda juga memiliki keunggulan dalam kesederhanaan penggunaannya dan kejelasan dalam pemahaman oleh pengguna..(Saputro, 2018). Terdapat fitur dan komponen utama dari anaconda yakni : Anaconda Navigator : Anaconda Navigator adalah GUI yang berfungsi untuk menemukan, menginstal, dan mengelola paket dan lingkungan dengan mudah menggunakan antarmuka pengguna yang sederhana dan intuitif.

- a. Library : Anaconda sudah menyertakan banyak Library populer seperti NumPy, pandas, scikit-learn, Matplotlib, dll. Hal ini sangat berguna untuk melakukan analisis data dan proyek ilmu data tanpa perlu melakukan installasi terlebih dahulu.
- b. Konsol : Anaconda menyertakan konsol berbasis teks yang disebut "Anaconda Prompt" yang berfungsi untuk menjalankan skrip Python, dan melakukan tugas pengembangan lainnya.
- c. Platform : Anaconda tersedia untuk berbagai platform, termasuk Windows, macOS, dan Linux, sehingga anaconda dapat menjalankan proyek di berbagai sistem operasi.

2.18 Jupyter Notebook

Jupyter Notebook merupakan singkatan dari tiga bahasa pemrograman, yakni Julia (Ju), Python (Py), dan R. Aplikasi ini dipakai untuk membuat dan membagikan dokumen yang memiliki kode, hasil hitungan, visualisasi, dan teks. Jupyter Notebook dapat langsung dijalankan melalui platform Anaconda dengan cara menginstal didalam Anaconda navigator hingga tombol “Install” berubah menjadi “Launch” seperti pada Gambar 2.17



Gambar 2. 17 Penginstallan Jupyter Notebook

dan tampilan setelah Jupyter Notebook dijalankan pada Gambar 2.18



Gambar 2. 18 Tampilan Jupyter Notebook

Jupyter Notebook memiliki beberapa fitur yang dapat mempermudah penggunaannya yaitu :

1. File

Fungsi kolom ini adalah untuk membuat buku catatan baru atau membuka buku catatan yang telah dibuat sebelumnya. Perubahan nama buku catatan yang dibuat dapat dilakukan jika diperlukan.

2. Save and Checkpoint

untuk membuat checkpoint yang akan kembali ke titik terakhir jika terjadi sesuatu yang tidak terduga.

3. View

Fungsi kolom ini adalah untuk mengaktifkan atau menonaktifkan tampilan judul dan toolbar.

4. Edit

Untuk memotong, menyalin, atau menempel dari sel yang ada, fungsi Edit Notebook Jupyter juga dapat berfungsi untuk memisahkan, menggabungkan, atau menghapus sel.

5. Insert

Untuk memasukkan atau menyisipkan cell di atas atau di bawah cell yang diinginkan.

6. Cell

Untuk menjalankan satu, beberapa, atau bahkan semua sel yang tersedia, ubah jenis sel yang dipilih dan hapus hasil dari sel yang telah Anda kerjakan.

7. Kernel

Untuk mengerjakan program yang sedang berjalan seperti melakukan restart, reconnect, mematikan, atau mengganti program yang akan digunakan dalam notebook pada kolom ini.

8. Widget

Untuk menambah atau menghapus widget yang ada. Widgetnya sendiri adalah JavaScript yang digunakan untuk membuat konten sel menjadi dinamis.

9. Help

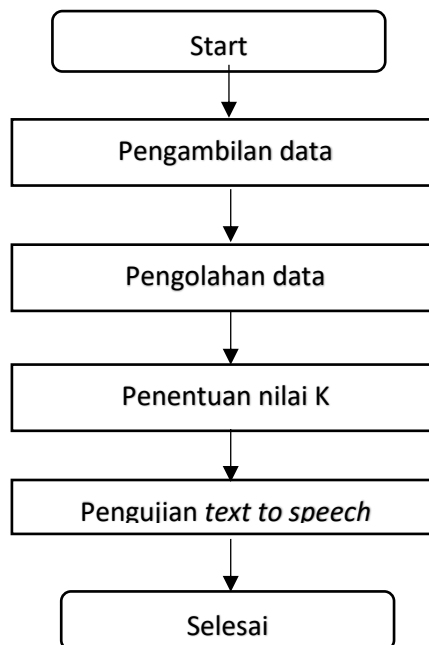
Untuk bantuan atau ingin mempelajari notebook seperti penggunaan keyboard shortcut, UI, dll.

Bab III

Analisis dan Perancangan Sistem

3.1 Alur penelitian

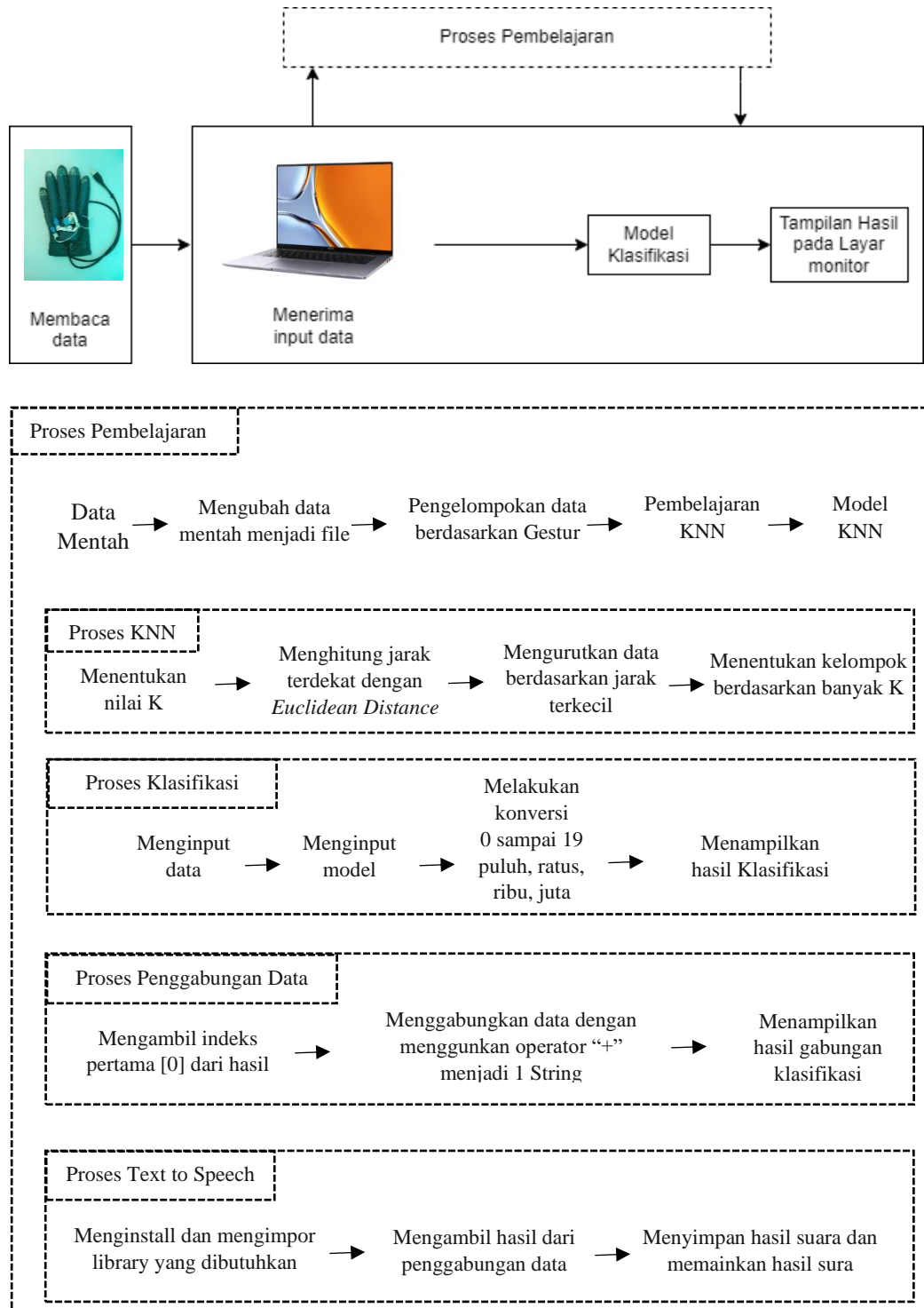
Tahapan awal penelitian ini dimulai dengan mengumpulkan data gestur tangan yang dibutuhkan untuk proses klasifikasi dilanjutkan dengan pengolahan data dimana semua data dari responden akan disatukan dan dinormaslisasi selanjutnya dari sekumpulan data tersebut akan ditentukan nilai K dengan melihat error rate terkecil kemudian model dari proses klasifikasi akan diuji akurasi, setelah mendapatkan akurasi yang baik model klasifikasi akan siap digunakan untuk proses selanjutnya, pada proses selanjutnya adalah dengan menguji data klasifikasi lain dengan menguji satu per satu data yang telah diambil kemudian proses akan dilakukan hingga mengeluarkan hasil klasifikasi dan proses selanjutnya adalah mengubah hasil klasifikasi yang berupa teks menjadi suara. Semua langkah akan dijelaskan secara detail dari awal hingga akhir sehingga data dapat digunakan sebagai data latih untuk pengujian secara real time kepada responden. Susunan tahapan pengerjaan dapat dilihat pada gambar 3.1.



Gambar 3. 1 Alur Penelitian

3.2 Desain Sistem

Desain kinerja sistem dapat dijabarkan pada bagian alur yang dimuat pada ilustrasi yang terdapat pada Gambar 3.2.



Gambar 3. 2 Desain sistem (Santoso, 2022)

Spesifikasi computer yang digunakan dalam penelitian ini dapat dilihat pada tabel 3.1.

Tabel 3. 1 Spesifikasi Komputer

Komponen	Keterangan
Sistem Operasi	Windows 11 Home Single Language 64-bit (Version 22H2)
Prosesor	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz
RAM	8,00 GB (7,88 GB usable)
Resolusi Layar	1920 x 1080
GPU	NVIDIA GeForce MX130

3.3 Penelitian Terdahulu

Penelitian sebelumnya telah dilakukan oleh (Santoso, 2022) dalam penelitiannya yang berjudul “Rancang Bangun Aplikasi Klasifikasi Gestur Tangan dengan Menggunakan Metode K-Nearest Neighbors” dan penelitian yang dilakukan oleh (Wugow, 2022) dalam penelitiannya yang berjudul “Pengembangan Klasifikasi Angka dan Alfabet Bahasa Isyarat dengan Perbandingan Metode KNN dan ANN”

Adapun persamaan penelitian terdahulu dengan penelitian ini adalah sebagai berikut:

- Bahasa isyarat yang digunakan adalah Bahasa isyarat SIBI
- Bahasa isyarat yang diklasifikasikan meliputi angka
- Metode yang digunakan untuk klasifikasi yakni KNN

Sedangkan perbedaan penelitian terdahulu dengan penelitian ini adalah sebagai berikut:

- Batasan klasifikasi Bahasa isyarat khususnya dalam angka pada penelitian terdahulu yakni 1 sampai 10, namun pada penelitian ini Batasan angka yang akan diklasifikasikan menjadi 0 sampai 999.999.999.
- Output yang dikeluarkan pada penelitian terdahulu yakni hanya 1 hasil klasifikasi, namun pada penelitian ini output yang dikeluarkan adalah gabungan dari banyak data yang telah diklasifikasi.

- c. Jumlah data yang digunakan untuk penelitian terdahulu ialah sesuai dengan jumlah hasil output, yakni jika hasil output meliputi angka 1 sampai 10 maka jumlah data gestur yang digunakan dalam proses klasifikasi juga ada 10, namun pada penelitian ini jumlah output yang bisa dikeluarkan dapat mencapai 0 sampai 999.999.99 dengan jumlah gestur yang digunakan untuk proses klasifikasi sebanyak 24 gerakan.
- d. Pada penelitian terdahulu hasil output yang dikeluarkan hanyalah berupa teks saja, namun pada penelitian ini terdapat 2 hasil output yakni berupa teks dan suara dalam Bahasa Indonesia.

3.4 Analisis Kebutuhan

Analisis kebutuhan pada tugas akhir ini adalah diperlukannya penambahan fitur klasifikasi pada angka yang mencakup ribuan hingga jutaan serta penambahan fitur dalam mengeluarkan hasil output yakni *text to speech*, penelitian ini bertujuan untuk menambah fitur klasifikasi sensor bend glove kit pada penelitian terdahulu yang mencakup banyak angka tanpa terlalu banyak data gestur yang digunakan dengan menggunakan metode klasifikasi yang efektif dan sudah di uji yaitu KNN.

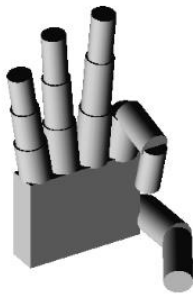
3.5 Studi Pustaka

Studi Pustaka pada penelitian ini dilakukan untuk mempelajari dan mencari referensi pada penelitian terdahulu dan teori dasar yang digunakan pada sensor bend glove kit. Dasar teori yang dibutuhkan pada penelitian ini seperti KNN, Bahasa Isyarat Sibi dan Sensor Bend Glove Kit.

3.6 Gerakan Gestur Tangan

Seperti yang sudah dijelaskan pada Bab 2, pergerakan tangan manusia untuk memperagakan bahasa isyarat memerlukan gerakan pada jari-jari, telapak tangan, lengan bawah, serta lengan atas. Banyak bahasa isyarat yang memerlukan dua tangan untuk memperagakan bahasa isyarat, namun pada penelitian ini khusus hanya mengambil angka yang mencakup jutaan dari Bahasa isyarat SIBI yang hanya menggunakan gestur satu tangan yakni dengan menggunakan tangan kanan. Pembacaan sensor atas pergerakan tangan khususnya pada Gerakan tekukan pada jari

dan telapak tangan yang nantinya akan menjadi data masukan bagi pembelajaran KNN. Setiap pergerakan tangan memiliki 16 buah data yang dapat dihasilkan namun pada penelitian ini data yang akan diambil akan diproses untuk mengambil data yang akan digunakan yakni hanya dengan mengambil 13 data sensor dari 16 data yang dapat dihasilkan. Untuk visualisasi pergerakan jari dan telapak tangan manusia dapat diperlihatkan menggunakan aplikasi bawaan dari flexpoint glove kit dan dapat dilihat pada Gambar 3.3 berikut.



Gambar 3. 2 Visualisasi Glove Kit

3.7 Pembacaan Data oleh Sensor

Perangkat USB Glove Kit yang terpasang Sensor Bend di dalamnya memiliki banyak fitur dan berikut adalah penjelasan mengenai data dan fitur yang ada pada USB Glove kit :

- MCP-Pinky : lipatan metacarpophalangeal jari kecil/kelingking.
- PIP-Pinky : lipatan proximal interphalangeal jari kecil/kelingking.
- MCP-Ring : lipatan metacarpophalangeal jari cincin/manis.
- PIP-Ring : lipatan proximal interphalangeal jari cincin/manis
- MCP-Mid : lipatan metacarpophalangeal jari tengah
- PIP-Mid : lipatan proximal interphalangeal jari tengah
- MCP-Index : lipatan metacarpophalangeal jari telunjuk
- PIP-Index : lipatan proximal interphalangeal jari telunjuk
- MCP-Thumb : lipatan metacarpophalangeal ibu jari
- PIP-Thumb : lipatan proximal interphalangeal ibu jari
- ACC-X : sudut X pembacaan sensor akselerometer
- ACC-Y : sudut Y pembacaan sensor akselerometer

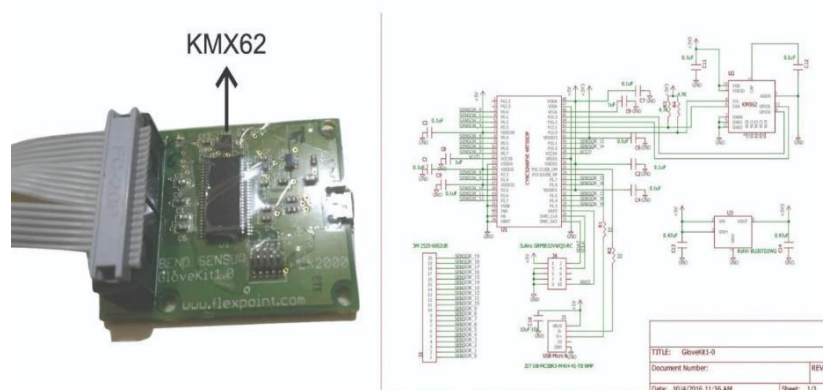
ACC-Z : sudut Z pembacaan sensor akselerometer

Metacarpophalangeal: tekukan jari yang terdapat pada telapak tangan

Proximal interphalangeal: tekukan jari setelah *metacarpo-phalangeal*

3.8 USB Glove Kit

USB Glove Kit terdiri dari tiga buah komponen utama yang terdiri atas 1) microcontroller (komputer mini) yaitu PSoC CY8C3246PVI-147. Tujuan penggunaan microcontroller ini adalah karena memiliki arus DAC (alat pengubah sinyal digital menjadi analog) yang sudah terintegrasi. Hal ini memungkinkan sensor tekukan akan dihubungkan dengan ADC secara langsung tanpa memerlukan komponen eksternal seperti resistor pembagi tegangan atau op-AMPS. Microcontroller dan skemanya dapat dilihat pada Gambar 3.2 Lima buah sensor tekukan yang berguna untuk membaca pergerakan jari-jari tangan seperti yang sudah dipaparkan pada bab 2 dan bab 3 sebuah alat gabungan sensor akselerometer dan sensor magnetometer Kionix KMX62 yang berguna untuk membaca pergerakan tangan, hal ini juga sudah dipaparkan pada bab dua (Santoso, 2022).



Tabel 3. 2 Gambar dan skema microcontroller

3.9 Pengumpulan Data

Pengumpulan data pada penelitian Tugas Akhir ini menggunakan 5 sensor bend yang terpasang pada sarung tangan sebagai input data. Subjek diperlukan pada penelitian ini untuk mengambil nilai input data dengan cara subjek menggunakan sarung tangan yang sudah terpasang sensor bend pada tangan kanan kemudian menggerakkan tangan sesuai dengan gestur bahasa isyarat SIBI yang diperlukan yakni

angka 0 sampai 19, Gestur yang menunjukkan puluhan, Gestur yang menunjukkan ratusan, Gestur yang menunjukkan ribuan dan Gestur yang menunjukkan jutaan. Data yang hendak dikumpulkan pada penelitian ini adalah data angka 0 sampai 19 serta gestur khusus dalam Bahasa Isyarat Sibi yakni : puluhan, ratusan, ribuan dan jutaan, dengan total gestur yakni sebanyak 24 gerakan yang nantinya digunakan untuk proses klasifikasi Bahasa isyarat angka, kelima sensor yang terpasang memiliki sepuluh fitur dominan serta tiga fitur tambahan dari accelerometer (Widodo, et al., 2021), yaitu:

- y1: MCP (metacarpophalangeal) kelingking
- y2: PIP (proximal interphalangeal) kelingking
- y3: MCP (metacarpophalangeal) manis
- y4: PIP (proximal interphalangeal) manis
- y5: MCP (metacarpophalangeal) tengah
- y6: PIP (proximal interphalangeal) tengah
- y7: MCP (metacarpophalangeal) telunjuk
- y8: PIP (proximal interphalangeal) telunjuk
- y9: MCP (metacarpophalangeal) ibu jari
- y10: PIP (proximal interphalangeal) jari
- y11: Accelerometer sumbu x
- y12: Accelerometer sumbu y
- y13: Accelerometer sumbu z

3.10 Pembelajaran Data Menggunakan K-Nearest Neighbor

Pada penelitian ini untuk mengklasifikasikan hasil pergerakan gestur tangan huruf dan angka menggunakan metode K-Nearest Neighbor (KNN). Metode pembelajaran KNN yang digunakan pada penelitian ini dimulai dari menginisialisasi nilai k sebagai penentu data masukan dan akan diklasifikasikan pada kelas yang sudah dipaparkan pada bab dua, nilai k harus bernilai angka ganjil. Setelah itu akan

dilakukan perhitungan untuk mencari jarak Euclidean (Euclidean Distance). Setelah itu nilai K dengan tingkat akurasi yang paling tinggi akan digunakan sebagai model pengujian.

3.11 Implementasi Sistem

Pada tahap implementasi sistem merupakan tahap lanjutan setelah pembelajaran data menggunakan KNN. Implementasi yang dilakukan adalah membuat sistem pembelajaran dengan data latih berjumlah 24 gestur (20 angka dari 0-19, Gestur Puluh, Gestur Ratus, Gestur Ribuan dan Gestur Juta). Masing-masing data yang dilatih memiliki jumlah data 200 data dalam satu kali rekam yang direkam selama satu detik. Setiap data akan dilakukan 1 kali rekam untuk mendapatkan sampel yang cukup demi mencapai akurasi yang tinggi.

Seperti yang dijelaskan sebelumnya, pengambilan data akan dilakukan dengan menggerakkan perangkat sesuai dengan gestur yang diperagakan dan sesuai dengan angka dalam kamus SIBI. Data tersebut akan dibaca dan kemudian akan dijadikan nilai masukan dalam pembelajaran metode yang telah dipaparkan sebelumnya. Hasil pembelajaran dan pengujian akan ditampilkan dalam bentuk kata, angka dan suara.

3.12 Rancangan Pengujian

Pengujian penelitian ini dilakukan dengan menguji gestur angka SIBI 0 sampai 19 dan gestur khusus (puluhan, ratusan, ribuan dan jutaan), dengan menggunakan subjek sebanyak 5 yang dilakukan oleh peneliti sendiri dan 4 orang lainnya, jumlah subjek sebanyak 5 dikarenakan cukup representative, serta jika terlalu banyak maka data juga yang diambil akan lebih banyak yang nantinya akan mengacu pada spesifikasi dari device yang digunakan dalam penelitian, subjek yang diambil berdasarkan jenis kelamin yakni laki-laki dan Perempuan serta berdasarkan berat badan yakni 50 Kg sampai 70 Kg, untuk bentuk tangan dapat dilihat pada gambar hal ini dikarenakan ukuran tangan dan jari khususnya pada laki-laki dan Perempuan yang berbeda serta berat badan yang juga mempengaruhi bentuk tubuh khususnya pada tangan dan jari, proses pengambilan data dilakukan di beberapa lokasi yang berbeda yaitu Lab HMI dan Rumah peneliti, untuk ukuran tangan data dilihat pada gambar 3.3 dan untuk gambar proses pengambilan data dapat dilihat pada gambar 3.4.



Gambar 3. 3 Ukuran tangan berdasarkan berat (Chaerunnisa, 2017)



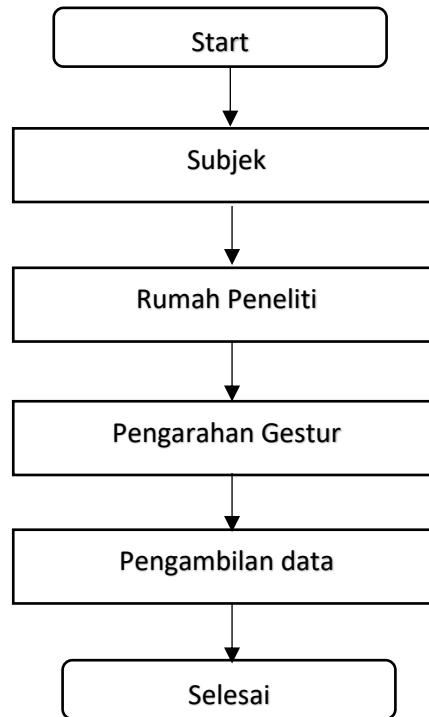
Gambar 3. 4 Proses pengambilan data dari subjek

Untuk spesifikasi komputer dan software yang digunakan dapat melihat tabel 3.3

Komponen	Keterangan
Sistem Operasi	Windows 11 Home Single Language 64-bit (Version 22H2)
Prosesor	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz
RAM	8,00 GB (7,88 GB usable)
Resolusi Layar	1920 x 1080
GPU	NVIDIA GeForce MX130

Tabel 3. 3 Spesifikasi komputer

Untuk prosedur yang dilakukan dalam pengambilan data dapat dilihat pada Gambar 3.5.



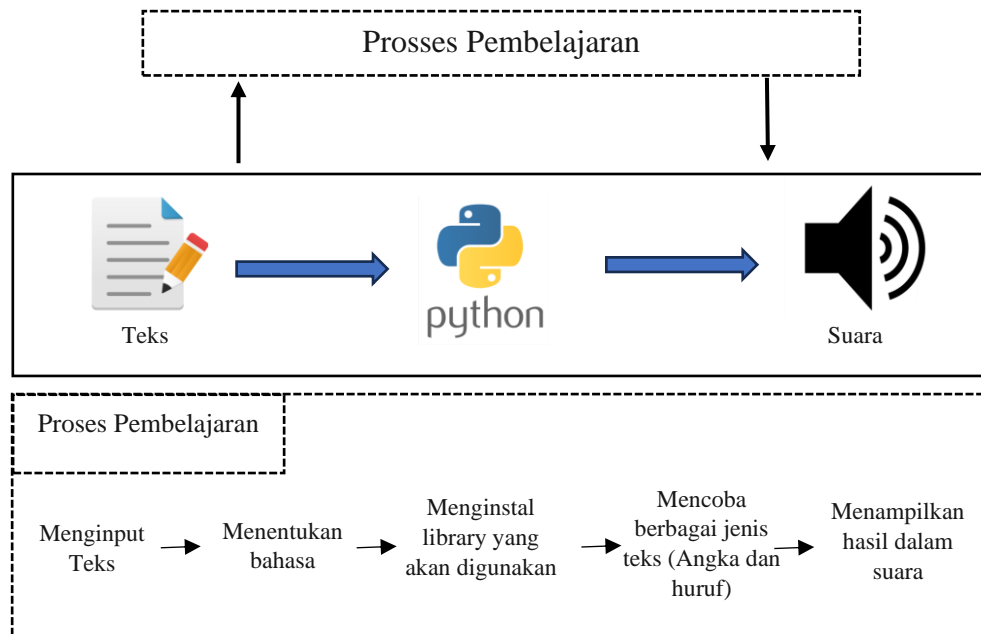
Gambar 3. 5 Alur pengambilan data

Prosedur pengambilan data dimulai dari subjek yang akan datang ke lokasi pengambilan data (rumah peneliti), kemudian peneliti akan memberikan contoh dan mengarahkan gestur yang akan dilakukan oleh subjek selanjutnya pengambilan data akan dilakukan, pengambilan dilakukan dengan menggunakan kode yang sudah ada sebelumnya yang telah dirubah untuk memenuhi kebutuhan yaitu mengaktifkan sensor yang sebelumnya belum diaktifkan, mengubah port sensor agar terhubung dengan perangkat yang dimiliki peneliti serta memperbanyak data yang akan disimpan dengan jumlah awal 100 menjadi 200 data, data dari sensor yang diambil akan langsung tersimpan dalam bentuk file *.csv*

Data yang diambil adalah Gerakan gestur angka sebanyak 20 dan 4 tambahan gestur satuan yaitu : puluh, ratus, ribu dan juta sehingga total data yang akan diambil setiap subjeknya adalah sejumlah 24 file data, dengan total data 4800 data untuk tiap subjeknya.

3.13 Text to Speech

Konversi dari text ke suara dijabarkan pada bagian alur yang dimuat pada ilustrasi yang terdapat pada Gambar 3.6.



Gambar 3. 6 Ilustrasi text to speech

Bab IV

Hasil dan Pembahasan

4.1 Pengumpulan Data

Pengambilan data dilakukan di laboratorium HMI dan di lokasi lain yaitu rumah peneliti, Jumlah partisipan pada penelitian ini ada 5 orang termasuk peneliti sendiri, pengambilan data akan tersimpan dalam bentuk file .csv yang berisikan 24 data Gerakan angka bahasa isyarat SIBI yang dilakukan oleh partisipan. Gerakan yang akan diambil adalah gerakan angka isyarat SIBI 0 sampai 19 serta beberapa gestur khusus yaitu puluh, ribu dan juta. Setiap gerakan akan memiliki 13 data sensor yang nantinya akan digunakan yaitu 10 data terdiri dari 2 data pada setiap jari tangan dan 3 data dari *akselerometer*, berikut data yang berasal dari sensor bend :

- a. Metacarpo-phalangeal: tekukan jari yang terdapat pada telapak tangan
- b. Proximal interphalangeal: tekukan jari setelah metacarpo-phalangeal

Berikut data yang diambil dari *akselerometer* :

- a. accX : sudut X pembacaan sensor akselerometer
- b. accY : sudut Y pembacaan sensor akselerometer
- c. accZ : sudut Z pembacaan sensor akselerometer

pengambilan data dilakukan dengan menggunakan sebuah sarung tangan Glove-kit yang telah dihubungkan dengan laptop dan data nantinya akan tersimpan langsung dalam bentuk file .csv. Untuk tampilan pengambilan data dapat dilihat pada gambar 4.1, 4.2 dan 4.3.



Gambar 4. 1 Pengambilan data

```

C:\Windows\py.exe
Enter Gesture Name :
cth
Press Enter to Start Recording..
['5866', '1249', '3216', '5504', '15888']
['514', '932', '704', '3656', '4549', '5417', '12472', '1519', '5218', '1249', '3104', '5664', '15952']
['565', '932', '704', '3684', '4641', '5388', '12925', '1547', '5245', '1249', '3104', '5664', '15952']
['514', '999', '732', '3656', '4549', '5388', '13258', '1519', '5218', '1249', '3056', '5360', '16048']
['565', '999', '732', '3684', '4641', '5388', '13258', '1547', '5245', '1249', '3056', '5360', '16048']
['514', '932', '704', '3628', '4549', '5388', '13560', '1491', '5190', '1249', '3056', '5904', '15568']
['565', '999', '732', '3684', '4641', '5388', '13258', '1547', '5245', '1249', '2976', '5904', '15568']
['514', '999', '732', '3600', '4549', '5388', '14798', '1491', '5218', '1249', '2976', '5904', '15568']
['565', '999', '704', '3628', '4610', '5359', '13258', '1547', '5218', '1249', '2960', '5568', '15728']
['514', '999', '704', '3572', '4549', '5359', '13560', '1491', '5218', '1249', '2960', '5568', '15728']
['565', '999', '732', '3628', '4641', '5359', '13107', '1547', '5245', '1306', '3056', '5904', '15120']
['514', '932', '704', '3544', '4518', '5330', '12925', '1464', '5190', '1249', '3056', '5904', '15120']
['514', '932', '704', '3600', '4610', '5330', '12925', '1519', '5245', '1249', '3280', '5728', '15248']
['514', '932', '704', '3544', '4518', '5330', '12925', '1464', '5218', '1249', '3280', '5728', '15248']
['514', '999', '732', '3600', '4610', '5330', '12925', '1519', '5245', '1249', '3136', '5712', '15712']
['462', '932', '704', '3515', '4518', '5330', '12925', '1464', '5190', '1249', '3136', '5712', '15712']
['514', '932', '704', '3572', '4580', '5330', '13560', '1519', '5218', '1249', '3008', '5584', '15264']
['462', '932', '704', '3515', '4518', '5301', '13107', '1436', '5218', '1249', '3008', '5584', '15264']
['514', '999', '732', '3572', '4580', '5330', '12774', '1519', '5245', '1306', '2880', '5712', '15680']
['514', '999', '704', '3487', '4487', '5301', '13560', '1436', '5218', '1249', '2880', '5712', '15680']
['514', '932', '704', '3544', '4549', '5301', '13258', '1491', '5218', '1249', '2704', '5616', '16000']
['514', '999', '704', '3487', '4487', '5301', '13711', '1436', '5190', '1249', '2704', '5616', '16000']
['514', '999', '704', '3515', '4549', '5301', '12774', '1491', '5245', '1306', '2976', '5792', '15008']
['462', '932', '704', '3459', '4457', '5273', '13862', '1436', '5190', '1249', '2976', '5792', '15008']

```

Gambar 4. 2 Tampilan pengambilan data

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	12325	1179	3766	32739	16504	8740	7001	32740	2572	256	16608	-304	1536
2	12216	1151	3766	32739	16655	8768	7230	32740	2601	330	16640	-352	1664
3	12407	1179	3738	32739	16504	8740	6829	32740	2572	256	16640	-352	1664
4	12216	1179	3738	32739	16655	8740	7603	32740	2601	330	16672	-432	1568
5	12489	1179	3738	32739	16504	8740	6570	32740	2572	256	16672	-432	1568
6	12298	1179	3710	32739	16655	8740	6054	32740	2601	330	16624	-112	1888
7	12571	1179	3681	32739	16655	8711	8894	32740	2572	256	16624	-112	1888
8	12298	1124	3625	32739	16655	8711	8780	32740	2601	293	16560	-256	1904
9	12653	1151	3625	32739	16806	8682	9583	32740	2601	256	16560	-256	1904
10	12352	1124	3568	32739	16806	8653	10817	32740	2601	293	16560	-384	1392
11	12653	1151	3540	32739	16806	8538	14949	32740	2601	256	16560	-384	1392
12	12380	1151	3455	32739	16987	8451	15408	32740	2629	330	15968	464	2160
13	12680	1151	3398	32739	17289	8336	14518	32740	2629	256	15968	464	2160
14	12434	1124	3285	32739	17590	8192	13916	32740	2657	293	16000	-656	1552
15	12762	1124	3172	32739	18677	8047	14662	32740	2685	256	16000	-656	1552
16	12434	1124	3030	32739	19763	7817	17302	32740	2742	293	16256	-304	2160
17	12571	1124	2860	32739	21754	7615	17302	32740	2798	293	16256	-304	2160
18	12079	1096	2662	32739	24711	7355	19081	32740	2855	330	17248	112	2528
19	12134	1124	2463	32739	26401	7124	17302	32740	2912	330	17248	112	2528
20	11533	1124	2265	32739	27638	6865	15551	32740	2883	330	18048	1136	3664
21	11560	1151	2124	32739	28875	6720	12596	32740	2912	330	18048	1136	3664

Gambar 4. 3 Tampilan file yang tersimpan

File yang tersimpan dalam bentuk file.csv, memiliki jumlah baris pada setiap file sebanyak 200 data tersebut merupakan hasil murni dari pengambilan data sensor maka pemberian nama file saat menyimpan file. csv harus sesuai dengan gestur yang akan diambil, karena hasil dari file yang ditampilkan dari pengambilan data masih belum memiliki nilai output serta label sehingga nantinya file harus diberikan nilai outout dan label secara manual ataupun secara kode,

4.2 Proses Pengolahan Data

Data sensor diambil dengan menggunakan kode yang telah dibuat oleh peneliti sebelumnya (Santoso, 2022), kode yang sudah ada tersebut akan diubah sesuai kebutuhan yaitu membuka 13 fitur dengan cara merubah *regular expression* untuk menampilkan 13 data tersebut. Data yang diambil akan menjadi sebuah file *.csv* yang kemudian akan diolah secara manual yaitu dengan memberikan output pada setiap filenya dan memberikan keterangan tabelnya(memberikan label), label yang digunakan ada 13 yaitu :

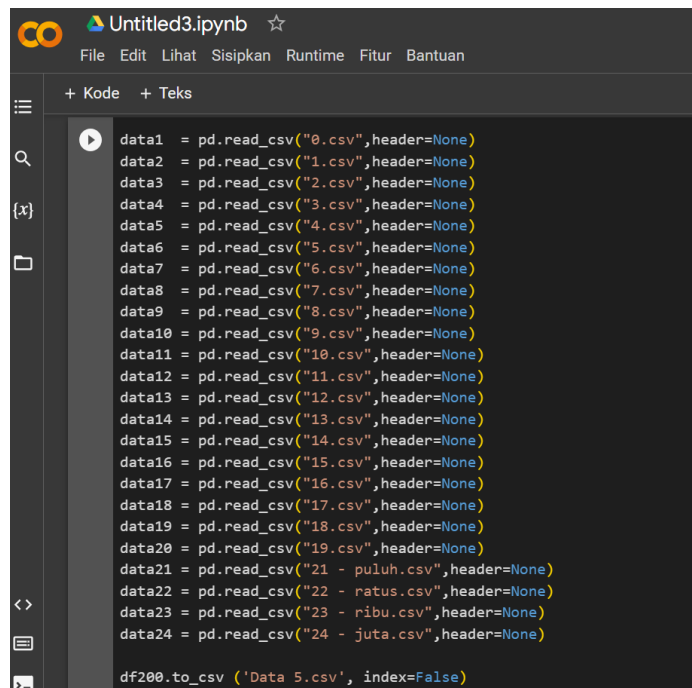
MCP-Pinky	: lipatan metacarpophalangeal jari kecil/kelingking.
PIP-Pinky	: lipatan proximal interphalangeal jari kecil/kelingking.
MCP-Ring	: lipatan metacarpophalangeal jari cincin/manis.
PIP-Ring	: lipatan proximal interphalangeal jari cincin/manis
MCP-Mid	: lipatan metacarpophalangeal jari tengah
PIP-Mid	: lipatan proximal interphalangeal jari tengah
MCP-Index	: lipatan metacarpophalangeal jari telunjuk
PIP-Index	: lipatan proximal interphalangeal jari telunjuk
MCP-Thumb	: lipatan metacarpophalangeal ibu jari
PIP-Thumb	: lipatan proximal interphalangeal ibu jari
ACC-X	: sudut X pembacaan sensor akselerometer
ACC-Y	: sudut Y pembacaan sensor akselerometer
ACC-Z	: sudut Z pembacaan sensor akselerometer
Output	: Klasifikasi output sensor (ditambahkan secara manual)

Untuk klasifikasi output pada gestur tertentu dilakukan *encoding* untuk mempermudah proses pada kode dan pengolahan data untuk gestur tertentu yang di encoding adalah sebagai berikut :

21	: Gestur “Puluh”
22	: Gestur “Ratus”
23	: Gestur “Ribu”
24	: Gestur “Juta”

Kemudian setelah pemberian nilai output, data akan digabungkan, proses penggabungan data dilakukan dengan menggunakan code dengan beberapa tahapan

tahapan pertama adalah menggabungkan data satu subjek yang berjumlah 24 menjadi 1 file, untuk proses penggabungan data tahap pertama dapat dilihat pada gambar 4.4.



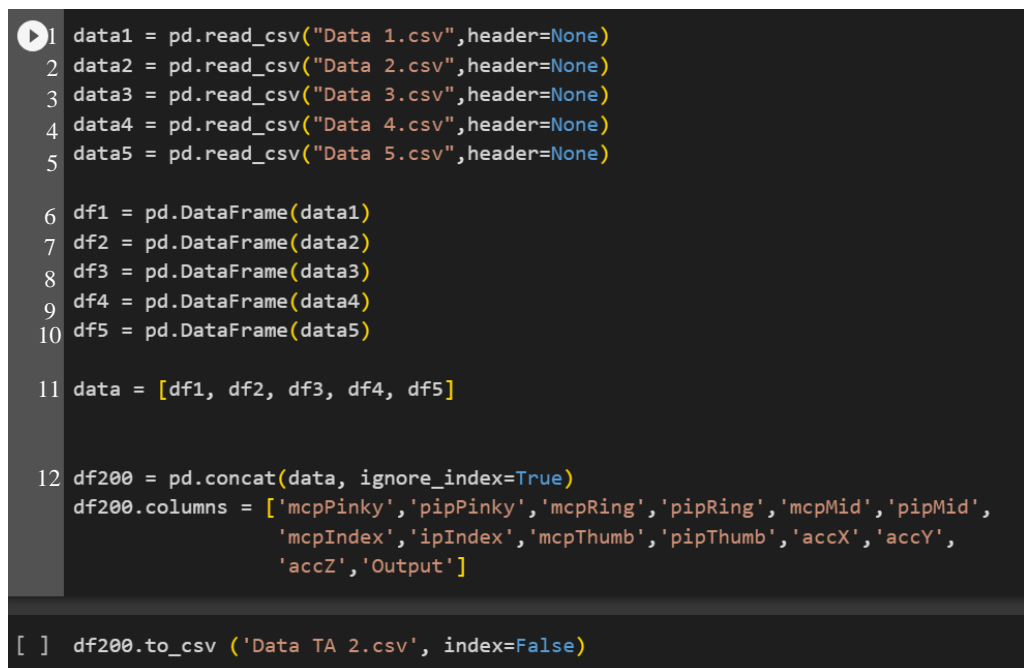
```

data1 = pd.read_csv("0.csv",header=None)
data2 = pd.read_csv("1.csv",header=None)
data3 = pd.read_csv("2.csv",header=None)
data4 = pd.read_csv("3.csv",header=None)
data5 = pd.read_csv("4.csv",header=None)
data6 = pd.read_csv("5.csv",header=None)
data7 = pd.read_csv("6.csv",header=None)
data8 = pd.read_csv("7.csv",header=None)
data9 = pd.read_csv("8.csv",header=None)
data10 = pd.read_csv("9.csv",header=None)
data11 = pd.read_csv("10.csv",header=None)
data12 = pd.read_csv("11.csv",header=None)
data13 = pd.read_csv("12.csv",header=None)
data14 = pd.read_csv("13.csv",header=None)
data15 = pd.read_csv("14.csv",header=None)
data16 = pd.read_csv("15.csv",header=None)
data17 = pd.read_csv("16.csv",header=None)
data18 = pd.read_csv("17.csv",header=None)
data19 = pd.read_csv("18.csv",header=None)
data20 = pd.read_csv("19.csv",header=None)
data21 = pd.read_csv("21 - puluh.csv",header=None)
data22 = pd.read_csv("22 - ratus.csv",header=None)
data23 = pd.read_csv("23 - ribu.csv",header=None)
data24 = pd.read_csv("24 - juta.csv",header=None)

df200.to_csv ('Data 5.csv', index=False)

```

Gambar 4. 4 Penggabungan data pertama



```

1 data1 = pd.read_csv("Data 1.csv",header=None)
2 data2 = pd.read_csv("Data 2.csv",header=None)
3 data3 = pd.read_csv("Data 3.csv",header=None)
4 data4 = pd.read_csv("Data 4.csv",header=None)
5 data5 = pd.read_csv("Data 5.csv",header=None)

6 df1 = pd.DataFrame(data1)
7 df2 = pd.DataFrame(data2)
8 df3 = pd.DataFrame(data3)
9 df4 = pd.DataFrame(data4)
10 df5 = pd.DataFrame(data5)

11 data = [df1, df2, df3, df4, df5]

12 df200 = pd.concat(data, ignore_index=True)
   df200.columns = ['mcpPinky','pipPinky','mcpRing','pipRing','mcpMid','pipMid',
   'mcpIndex','ipIndex','mcpThumb','pipThumb','accX','accY',
   'accZ','Output']

[ ] df200.to_csv ('Data TA 2.csv', index=False)

```

Gambar 4. 5 Penggabungan data kedua

Kemudian setelah menggabungkan data satu subjek yang berjumlah 24 menjadi 1 file, Langkah berikutnya adalah menggabungkan data keseluruhan subjek serta pemberian

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	mcPinky	pipPinky	mcPRing	pipRing	mcPMid	pipMid	mcPIndex	pipIndex	mcPThum	pipThum	mcPaccX	accY	accZ	Output
2	2445	28613	32739	32739	32737	8025	20995	32740	1466	3575	15120	688	8096	0
3	2445	28613	32739	32739	32737	8025	20995	32740	1433	3575	14960	-144	7680	0
4	2445	28613	32739	32739	32737	8025	20995	32740	1466	3603	14960	-144	7680	0
5	2445	28613	32739	32739	32737	8025	20995	32740	1466	3575	14992	608	8320	0
6	2424	28613	32739	32739	32737	7996	20995	32740	1433	3575	14992	608	8320	0
7	2445	28613	32739	32739	32737	7996	20995	32740	1433	3575	15056	416	8080	0
8	2445	28613	32739	32739	32737	7996	20995	32740	1466	3575	15056	416	8080	0
9	2445	28613	32739	32739	32737	8025	20995	32740	1466	3575	15136	560	8288	0
10	2424	28613	32739	32739	32737	7996	20995	32740	1433	3575	15136	560	8288	0
11	2424	28613	32739	32739	32737	7967	20995	32740	1433	3575	15120	704	8096	0
23995	10965	12389	207	141	386	636	5342	137	4978	301	-4240	3520	2848	24
23996	11223	12389	207	154	386	636	5314	137	4948	273	-4240	3520	2848	24
23997	11017	12703	207	141	386	636	5286	137	4978	301	-5952	5136	4128	24
23998	11223	13122	207	141	415	607	5257	137	4948	273	-5952	5136	4128	24
23999	11043	12808	207	128	386	607	5229	110	4948	273	-8048	6000	3824	24
24000	11301	12808	207	141	415	636	5229	137	4948	273	-8048	6000	3824	24
24001	11146	12441	207	128	386	636	5229	110	4948	301	-10048	6576	5120	24
24002	11430	13306	207	141	445	636	5229	165	4978	301	-10048	6576	5120	24
24003	11275	13830	207	128	415	636	5229	137	4948	273	-11776	6528	5808	24
24004	11584	14092	207	154	445	636	5229	193	4978	301	-11776	6528	5808	24
24005	11430	13830	207	128	415	636	5257	110	4948	273	-13520	7280	7552	24

4.3 Proses Pembuatan Model KNN

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import classification_report, confusion_matrix
```

Untuk dataset yang akan digunakan adalah 1 file .csv berisikan keseluruhan data partisipan dataset kemudian akan diupload ke *jupyter notebook* dan bila dataset sudah

terbaca/bisa ditampilkan di *jupyter notebook*, dapat dilihat pada gambar 4.8.

```
In [2]: df1 = pd.read_csv("Data TA.csv")
df1
```

Out[2]:

	mcpPinky	pipPinky	mcpRing	pipRing	mcpMid	pipMid	mcpIndex	ipIndex	mcpThumb	pipThumb	accX	accY	accZ	Output
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2445	28613	32739	32739	32737	8025	20995	32740	1466	3575	15120	688	8096	0
2	2445	28613	32739	32739	32737	8025	20995	32740	1433	3575	14960	-144	7680	0
3	2445	28613	32739	32739	32737	8025	20995	32740	1466	3603	14960	-144	7680	0
4	2445	28613	32739	32739	32737	8025	20995	32740	1466	3575	14992	608	8320	0
...
24000	11146	12441	207	128	386	636	5229	110	4948	301	-10048	6576	5120	24
24001	11430	13306	207	141	445	636	5229	165	4978	301	-10048	6576	5120	24
24002	11275	13830	207	128	415	636	5229	137	4948	273	-11776	6528	5808	24
24003	11584	14092	207	154	445	636	5229	193	4978	301	-11776	6528	5808	24
24004	11430	13830	207	128	415	636	5257	110	4948	273	-13520	7280	7552	24

24005 rows × 14 columns

Gambar 4. 8 Import dan menampilkan data

Selanjutnya dataset akan dibagi menjadi 4 yaitu X_train, X_test, y_train, y_test dengan perbandingan X : Y = 70 : 30, dapat dilihat pada gambar 4.9.

```
X_train, X_test, y_train, y_test = train_test_split (X, y, test_size = 0.3)

print("X Train:", X_train.shape)
print("y Train:", y_train.shape)
print("X Test:", X_test.shape)
print("y Test:", y_test.shape)
```

X Train: (16943, 13)
y Train: (16943,)
X Test: (7262, 13)
y Test: (7262,)

Gambar 4. 9 Membagi data set

Kemudian akan dilakukan proses transformasi data menggunakan *MinMaxScaler*, dapat dilihat pada gambar 4.10.

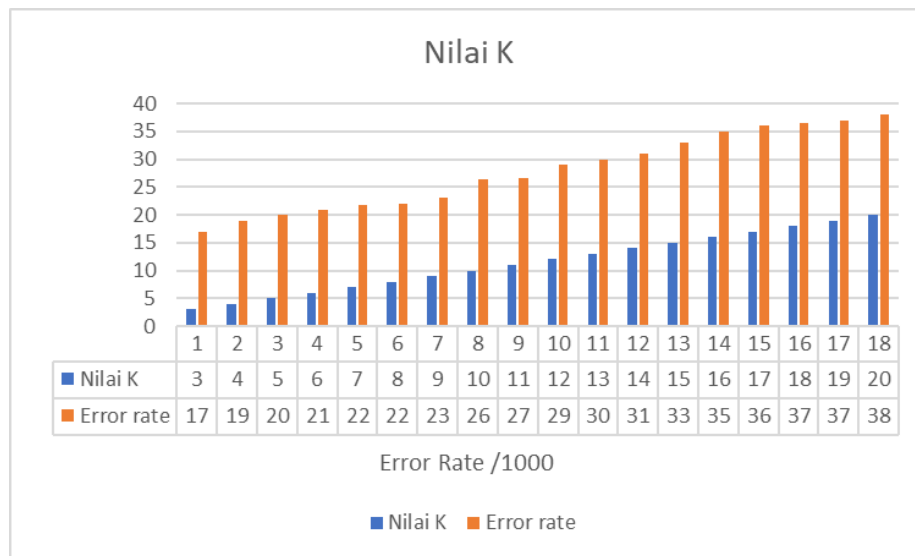
```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
scaler.fit(X_train)

x_train = scaler.transform(X_train)
x_test = scaler.transform(X_test)
```

Gambar 4. 10 Tranformasi menggunakan MinMaxScaler

Kemudian menentukan nilai K untuk KNN dengan memilih k yang memiliki *error rate* terkecil, untuk mendapatkan nilai K dapat melihat gambar 4.11, 4.12 dan 4.13.



Gambar 4. 11 Grafik error rate nilai K

```
error_rate = []
for i in range(3,20):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_train,y_train)
    pred_i = knn.predict(x_test)
    error_rate.append(np.mean(pred_i != y_test))

print(error_rate)
```

[0.017495140238822548, 0.019161344071091362, 0.020688697584004444, 0.02124409886142738, 0.02179950013885032, 0.022771452374340462, 0.023743404609830602, 0.02638156067758956, 0.026659261316301027, 0.029019716745348513, 0.029991668980838656, 0.03193557345181894, 0.0330463760666482, 0.035129130797000835, 0.036239933351846706, 0.036378783671202446, 0.037073035267981114]

Gambar 4. 12 Rincian error rate

```
1 error_rate = []
2 for i in range(3,20):
3     knn = KNeighborsClassifier(n_neighbors=i)
4     knn.fit(x_train,y_train)
5     pred_i = knn.predict(x_test)
6     error_rate.append(np.mean(pred_i != y_test))
7 print("Minimum Error:-",min(error_rate),"at K =", error_rate.index(min(error_rate)))
8 Minimum Error:- 0.017495140238822548 at K = 0
```

Gambar 4. 13 K dengan error rate terkecil

Dengan menggunakan kode tersebut K dengan error rate terkecil dapat ditentukan dengan cepat, pada line kedua menjelaskan range K yang akan digunakan adalah 3 sampai 20, kemudian dilanjutkan di line ke-3 yaitu dengan menggunakan klasifikasi KNN dengan K yang digunakan adalah dari 3 sampai 20, kemudian pada line ke-4 dan 5 adalah data klasifikasi yang digunakan, kemudian pada line ke-6 adalah untuk hasil

error rate nya dan pada line ke-7 bertujuan untuk mengeluarkan output berupa hasil perhitungan K dan pada K berapa (perhitungan K dimulai dari 0 sehingga jika input dimulai dari 3 dan hasil K-nya adalah 3 maka yang akan ditampilkan adalah 0, jika 1 maka yang ditampilkan adalah 4, dst.), dengan menggunakan kode tersebut dapat diketahui bahwa K dengan error rate terkecil dari 3 sampai 20 adalah $k = 3$

4.4 Hasil Pengujian KNN

Hasil model KNN kemudian akan diuji dengan menggunakan keseluruhan file data yang telah disatukan untuk mendapatkan hasil dari Classification Report, untuk hasil Classification Report mendapatkan hasil sebesar 0.98 atau 98% pada akurasi hal ini disebabkan beberapa gestur memiliki kesamaan seperti angka satuan dan belasan yang berbeda pada akselometer saja untuk Classification Report dapat dilihat pada gambar 4.14.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	296
1	0.99	0.99	0.99	305
2	1.00	0.99	0.99	282
3	1.00	1.00	1.00	279
4	0.98	1.00	0.99	307
5	1.00	1.00	1.00	310
6	1.00	1.00	1.00	301
7	0.99	0.99	0.99	296
8	1.00	1.00	1.00	312
9	0.99	1.00	1.00	281
10	0.85	0.84	0.84	301
11	1.00	1.00	1.00	279
12	1.00	0.99	0.99	290
13	0.99	1.00	0.99	313
14	1.00	0.97	0.99	310
15	1.00	0.99	0.99	301
16	0.95	0.98	0.97	302
17	0.99	0.97	0.98	315
18	1.00	0.98	0.99	301
19	0.99	1.00	0.99	312
21	0.84	0.85	0.84	292
22	1.00	1.00	1.00	304
23	1.00	1.00	1.00	309
24	1.00	1.00	1.00	304
accuracy			0.98	7202
macro avg	0.98	0.98	0.98	7202
weighted avg	0.98	0.98	0.98	7202

Gambar 4. 14 Classification report KNN

4.5 Pengujian Konversi

Konversi dilakukan pada gestur-gestur khusus yaitu puluh, ratus, ribu dan juta yang sebelumnya sudah *encoding* menjadi angka yaitu :

- 21 : Gestur “Puluh”
- 22 : Gestur “Ratus”
- 23 : Gestur “Ribu”
- 24 : Gestur “Juta”

Konversi dilakukan untuk mengubah output angka menjadi karakter pada gestur khusus yaitu puluh, ratus, ribu dan juta sehingga output akan mudah dikenali serta dapat dengan mudah diubah menjadi suara (*Text to speech*) nantinya, untuk mengkonversikan dapat dilihat pada gambar 4.15.

```
1 def konv(hasil_prediksi):
2     if hasil_prediksi == 21:
3         return "puluh"
4     if hasil_prediksi == 22:
5         return "ratus"
6     if hasil_prediksi == 23:
7         return "ribu"
8     if hasil_prediksi == 24:
9         return "juta"
10
11     else:
12         return str(hasil_prediksi)
```

Gambar 4. 15 Konversi data

Dengan menggunakan kode tersebut data yang sebelumnya *encoding* akan diubah menjadi kata, dapat dilihat pada line ke 2 sampai 5 menunjukkan bahwa bila hasil output yang dihasilkan adalah angka 21,22,23,dan 24 maka output yang dihasilkan akan dirubah menjadi karakter, kemudian setelah hasil output sudah dikonversi Langkah selanjutnya adalah ditampilkan apakah sudah terkonversi dengan benar, untuk hasil output dapat dilihat pada gambar 4.16.

```
hasil_konversi = konversi_hasil_prediksi(y_pred[0])
print(hasil_konversi)

ratus
```

Gambar 4. 16 Hasil konversi

4.6 Pengujian Menggabungkan 2 data

Penggabungan dilakukan dengan menggabungkan 2 data klasifikasi yang ada sehingga menghasilkan output tertentu, sebagai contoh menggabungkan klasifikasi pada gestur angka 2 dan klasifikasi gestur khusus ratus, “ratus merupakan” hasil yang sudah dikonversikan dari 22 menjadi “ratus” sehingga menghasilkan output “2ratus”. Langkah pertama adalah memasukan 2 data yang akan dilakukan klasifikasi, dapat dilihat pada gambar 4.17.

```
df_new1 = pd.read_csv("2.csv")  
df_new2 = pd.read_csv("22 - ratus.csv")
```

Gambar 4. 17 Memasukan 2 data csv

Kemudian akan dipilih indeks untuk melakukan klasifikasi, pada penelitian ini lokasi/indeks yang dipilih adalah seluruh kolom kecuali pada bagian yang terakhir(output), dapat dilihat pada gambar 4.18.

```
X_new1 = df_new1.iloc[:, :-1].values  
X_new2 = df_new2.iloc[:, :-1].values
```

Gambar 4. 18 Pemilihan indeks

Kemudian akan dilakukan prediksi KNN dengan model KNN yang sudah dibuat pada pembuatan model KNN dengan nilai K=1, Untuk prediksi KNN dapat dilihat pada Gambar 4.19.

```
y_pred1 = knn.predict(X_new1)  
y_pred2 = knn.predict(X_new2)
```

Gambar 4. 19 Prediksi KNN

Kemudian akan dilakukan proses konversi dan penggabungan data, namun sebelumnya variabel “y_pred” akan diubah menjadi “data” agar memudahkan penulisan kode, dapat dilihat pada gambar 4,19, untuk konversi data dapat dilihat pada gambar 4.20 dan Untuk hasil penggabungan data dapat dilihat pada gambar 4.21.

```
data1 = y_pred1  
data2 = y_pred2
```

Gambar 4. 20 Mengubah nama variabel

```

1 def konv(hasil_prediksi):
2     if hasil_prediksi == 21:
3         return "puluh"
4     if hasil_prediksi == 22:
5         return "ratus"
6     if hasil_prediksi == 23:
7         return "ribu"
8     if hasil_prediksi == 24:
9         return "juta"
10
11 else:
12     return str(hasil_prediksi)

```

Gambar 4. 21 Proses konversi

Sama seperti proses konversi sebelumnya, dengan menggunakan kode tersebut data yang sebelumnya *encoding* akan diubah menjadi kata pada line ke 2 sampai 5 menunjukkan bahwa bila hasil output yang dihasilkan adalah angka 21,22,23,dan 24 maka akan dirubah, kemudian setelah hasil output sudah dikonversi Langkah selanjutnya adalah ditampilkan apakah sudah terkonversi dengan benar, untuk hasil output dapat dilihat pada gambar 4.22.

```

1 hasil_gabungan = konv(data1[0]) + konv(data2[0])
2 print(hasil_gabungan)
3 2ratus

```

Gambar 4. 22 Hasil penggabungan data

Dengan menggunakan kode tersebut dua data hasil prediksi akan digabungkan pada, line pertama digunakan untuk menggabungkan data yang sebelumnya telah melalui proses konversi, kemudian hasil gabungan akan ditampilkan menggunakan kode pada line ke-2 dan hasil output dapat dilihat pada line ke-3.

4.7 Pengujian Menggabungkan 2 data lebih

Pengujian berikutnya adalah menguji apakah data klasifikasi dapat digabungkan jika jumlah data lebih dari 2, hal ini bertujuan untuk mendapatkan hasil klasifikasi angka yang besar sekaligus menguji apakah program tetap dapat berjalan bila data yang dimasukan banyak, pada pengujian ini memiliki proses yang sama

dengan pengujian 2 data namun pada pengujian ini peneliti akan menggunakan 4 buah data yaitu gestur angka 2 dan 5 serta gestur khusus yaitu ribu dan ratus “ratus dan ribu merupakan” hasil yang sudah dikonversikan dari 22 dan 23 menjadi “ratus” sehingga output yang akan keluar seperti pada pengujian 2 data sebelumnya akan berupa angka dan karakter yang nantinya akan dilakukan proses *text to speech*, untuk proses dan hasil dari penggabungan data dapat dilihat pada gambar 4.23, gambar 4.24, gambar 4.25, gambar 4.26 dan gambar 4.27.

```
df_new1 = pd.read_csv("2.csv")
df_new2 = pd.read_csv("5.csv")
df_new3 = pd.read_csv("22 - ratus.csv")
df_new4 = pd.read_csv("23 - ribu.csv")
```

Gambar 4. 23 Menginput 4 data

```
1 X_new1 = df_new1.iloc[:, :-1].values
  X_new2 = df_new2.iloc[:, :-1].values
  X_new3 = df_new3.iloc[:, :-1].values
  X_new4 = df_new4.iloc[:, :-1].values
  X_new5 = df_new5.iloc[:, :-1].values
```

```
2 y_pred1 = knn.predict(X_new1)
  y_pred2 = knn.predict(X_new2)
  y_pred3 = knn.predict(X_new3)
  y_pred4 = knn.predict(X_new4)
  y_pred5 = knn.predict(X_new5)
```

```
3 data1 = y_pred1
  data2 = y_pred2
  data3 = y_pred3
  data4 = y_pred4
  data5 = y_pred5
```

```
4 print(data4[0])
```

23

Gambar 4. 24 Mengklasifikasikan data dengan menggunakan model KNN

pada kotak/line ke-1 menentukan data yang akan dikalsifikasikan [:, :-1] artinya data yang akan diklasifikasikan adalah semua kolom kecuali kolom terakhir, pada kotak/line ke-2 meenggunakan model KNN untuk mengklasifikasikan data pada kotak/line ke-3 menyederhanakan nama agar lebih mudah digunakan proses pada kotak/line ke-3 bisa

diabaikan dan pada kotak/line ke-4 menunjukkan hasil output pada setiap data, proses selanjutnya yakni proses konversi pada gambar 4. 25

```
1 def konv(hasil_prediksi):
2     if hasil_prediksi == 21:
3         return "puluh"
4     if hasil_prediksi == 22:
5         return "ratus"
6     if hasil_prediksi == 23:
7         return "ribu"
8     if hasil_prediksi == 24:
9         return "juta"
10
11     else:
12         return str(hasil_prediksi)
```

Gambar 4. 25 Konversi data

Sama seperti proses konversi sebelumnya, dengan menggunakan kode tersebut data yang sebelumnya *encoding* akan diubah menjadi kata pada line ke 2 sampai 5 menunjukkan bahwa bila hasil output yang dihasilkan adalah angka 21,22,23,dan 24 maka akan dirubah sekaligus untuk menguji apakah jika data yang konversi lebih dari 1 hasil konversi tetap berjalan dengan baik, untuk hasil konversi dari tiap data yang hasil klasifikasi dapat dilihat pada gambar 4.26

```
a = konv(data2[0])
print (a)
ribu
```

Gambar 4. 26 Tampilan hasil konversi

kemudian setelah hasil output sudah berhasil dikonversi Langkah selanjutnya adalah menampilkan hasil gabungan dari keseluruhan data yang telah diklasifikasikan apakah dapat tergabung dengan benar, untuk hasil output gabungan dapat dilihat pada gambar 4.27.

```
hasil_gabungan = konv(data1[0]) + konv(data4[0]) + konv(data2[0]) + konv(data3[0])
print(hasil_gabungan)
2ribu5ratus
```

Gambar 4. 27 Hasil

4.8 Pengujian Text to Speech

Pengujian berikutnya adalah menguji apakah data klasifikasi dapat disuarakan dengan benar menggunakan Bahasa Indonesia, pada pengujian ini peneliti menggunakan 2 library *text to speech* yaitu Pyttsx3 dan gTTS yang sebelumnya sudah terinstall pada perangkat peneliti, untuk proses install ataupun melihat apakah library sudah terinstall sebelumnya dapat dilihat pada gambar 4.28 dan 4.29.

```
!pip install pyttsx3
```

Requirement already satisfied:

Gambar 4. 28 Proses instal pyttsx3

```
pip install gtts
```

Requirement already satisfied:

Gambar 4. 29 Proses install gtts

Perbedaan antara kedua library ini yaitu library Pyttsx3 dapat digunakan tanpa menggunakan koneksi internet sementara library gTTS harus menggunakan koneksi internet.

```
for voice in voices:
    print("Voice:")
    print(" - ID: {}".format(voice.id))
    print(" - Name: {}".format(voice.name))
    print(" - Languages: {}".format(voice.languages))
    print(" - Gender: {}".format(voice.gender))
    print(" - Age: {}".format(voice.age))

Voice:
- ID: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_DAVID_11.0
- Name: Microsoft David Desktop - English (United States)
- Languages: []
- Gender: None
- Age: None
Voice:
- ID: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-GB_HAZEL_11.0
- Name: Microsoft Hazel Desktop - English (Great Britain)
- Languages: []
- Gender: None
- Age: None
Voice:
- ID: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_ZIRA_11.0
- Name: Microsoft Zira Desktop - English (United States)
- Languages: []
- Gender: None
- Age: None
```

Gambar 4. 30 Rincian suara dan bahasa

Namun pada penggunaan Pyttsx3 pilihan bahasa maupun suara sangat sedikit karena bahasa dan suara di library ini diambil dari perangkat sehingga untuk menggunakan bahasa tertentu perlu melihat rincian tersebut jika pilihan bahasa yang diinginkan tidak

tersedia maka bahasa dan suara akan terlafalkan dengan bahasa yang tersedia, untuk melihat rincian suara dapat dilihat pada gambar 4.30.

Bab V

Kesimpulan dan Saran

5.1 Kesimpulan

- a. Berdasarkan pada hasil pengumpulan data pada penelitian ini adalah 5 subjek termasuk peneliti dengan melakukan 24 gerakan gestur tangan yaitu angka 0 – 19 dan beberapa gestur khusus yaitu satuan puluh, ratus, ribu dan juta, untuk 1 file data akan berisikan 200 baris data setiap subjek memiliki 4800 baris data sehingga untuk total datanya adalah 24.000 baris data.
- b. Berdasarkan hasil uji dengan menggunakan metode KNN mendapatkan akurasi sebesar 98% pada confusion matrix.
- c. Berdasarkan hasil pengujian secara offline dengan menggunakan model KNN dengan nilai $K = 3$ mendapatkan hasil yang baik dengan akurasi yang cukup tinggi namun terdapat kesalahan pada beberapa gestur khusus ribu dan ratus.
- d. Berdasarkan dari hasil data yang diuji output yang dikeluarkan dapat disuarakan dengan lancar (*text to speech*) dengan menggunakan Bahasa Indonesia dengan hasil output berupa angka dan kalimat, Serta untuk beberapa kasus dapat menterjemahkan beberapa gestur BISINDO/ASL yang memiliki kesamaan dengan SIBI serta bisa dirangkai dan disuarakan.

5.2 Saran

- a. Dalam melakukan pengambilan data saat partisipan akan melakukan Gerakan berikutnya memerlukan jeda sebentar agar data sebelumnya tidak ikut diambil.
- b. Melanjutkan penelitian secara real time dan dapat menggunakan cara yang lebih efektif.
- c. Melanjutkan penelitian hingga bisa menterjemahkan kalimat.
- d. Mengecek detail suara yang ada di dalam perangkat untuk bahasa tertentu sehingga dapat mempercepat proses text to speech.

Daftar Pustaka

- Andhika, W. (2019, 06 20). *Medium*. Retrieved from Medium :
<https://medium.com/@wahyuandhika/belajar-machine-learning-basic-of-scikit-learn-a1685db819a8>
- Aras, A. C. (2020). Explaining what learned models predict: In which cases can we trust machine learning models and when is caution required ? *Explaining what learned models predict: In which cases can we trust machine learning models and when is caution required ?*, 2.
- Atmaja, B. T. (2020, October 09). *Memahami Pengindeksan pada NumPy*. Retrieved from Sains Hack: <https://www.sainshack.com/2020/10/09/memahami-pengindeksan-pada-numpy/>
- Bangkit, Harry; Ruhimat, Mamat;. (2015). Kalibrasi Magnetometer Tipe 1540 Menggunakan Kalibrator Magnetometer. *Kalibrasi Magnetometer Tipe 1540 Menggunakan Kalibrator Magnetometer*, 56.
- Buletin, K. (2012). Magnetometer. In *Magnetometer* (p. vol 4 no 2).
- Chaerunnisa. (2017). *Berat Badan Turun Drastis*. Suara.com.
- Christopher, A. (2021, 02 02). *K-Nearest Neighbor*. Retrieved from Medium:
<https://medium.com>
- Dicoding. (2020, 07 15). *dicoding*. Retrieved from dicoding:
<https://www.dicoding.com/blog/kecerdasan-buatan-adalah/>
- Flexpoint Sensor Systems. (2016). *Flexpoint Sensor Systems*. Retrieved 11 30, 2022, from <http://www.flexpoint.com>
- Hadi, A. (2021). *Sejarah Hari Bahasa Isyarat Internasional 23 September & Tujuan*. Solo: tirta.id.
- Jose, A. G. (2021, 04 21). *All About Tech*. Retrieved from All About Tech:
<https://amalgjose.com/2021/04/04/how-to-do-text-to-speech-conversion-using-python/>
- kemdikbud. (2020, 06). *Kamus Sibi*. Retrieved from Kemdikbud:
<https://pmpk.kemdikbud.go.id/sibi/>
- Kionix. (2015). *Kionix*. Retrieved 11 30, 2022, from
<https://www.mouser.com/pdfdocs/Kionix-KMX62-1031-Datasheet.pdf>
- Krisnan. (2017). *Belajar Bahasa Isyarat Tingkat Dasar: Mengenal Huruf dan Angka*. meenta.

- Lab Immersa. (2018). *Immersa Lab*. Retrieved 11 30, 2022, from <https://www.immersa-lab.com/pengertian-accelerometer-dan-cara-kerjanya.htm>
- Muarabagja, M. H. (2022). *Hari Bahasa Isyarat Internasional: Kenali 2 Jenis Bahasa Isyarat di Indonesia*. Bandung: tempo.co.
- Murabagja, M. H. (2022, september 23). Hari Bahasa Isyarat Internasional: Kenali 2 Jenis Bahasa Isyarat di Indonesia. *difabel*.
- Nugroho, K. S. (2019, 11 13). *Medium Confusion Matrix untuk Evaluasi Model pada Supervised Learning*. Retrieved from Medium: <https://ksnugroho.medium.com>
- Santoso, D. A. (2022). *Rancang Bangun Aplikasi Klasifikasi Gestur Tangan dengan Menggunakan Metode K-Nearest Neighbors*. Malang: Skripsi Universitas Ma Chung.
- Saputro, T. T. (2018, 12 24). *Belajar Python Bersama Anaconda, Subrutin*. Retrieved from Subrutin: <https://subrutin.com>
- Tris, B. (2019, 09 15). *memahami-sizeshape-numpy-array*. Retrieved from <https://bagustris.blogspot.com>: <https://bagustris.blogspot.com/2019/09/memahami-sizeshape-numpy-array.html>
- Widodo, R. B. (2022). *Machine Learning Metode K-Nearesrt Neighbors Klasifikasi Angka Bahasa Isyarat*. Malang: Media Nusa Creative.
- Winarsih, M. (2007). *Intervensi Dini Bagi Anak Tunarungu dalam Pemerolehan Bahasa*. Jakarta.
- Wugow, K. C. (2022). *Pengembangan Klasifikasi Angka dan Alfabet Bahasa Isyarat dengan Perbandingan Metode KNN dan ANN*. Malang: Skripsi Universitas Ma Chung.

Lampiran

Profil partisipan dan Proses pengambilan data

Partisipan 1	
	<p>Nama : Amat Umur ; 38 tahun Berat badan : 55 Kg Jenis kelamin : Laki – laki</p>
Proses pengambilan data	
	

Partisipan 2



Nama : Turiani
Umur : 60 tahun
Berat badan : 53 Kg
Jenis kelamin : Perempuan

Proses pengambilan data



Partisipan 3



Nama : Ida Indra Trisnawati
Umur : 49 tahun
Berat badan : 61 Kg
Jenis kelamin : Perempuan

Proses pengambilan data



Partisipan 4



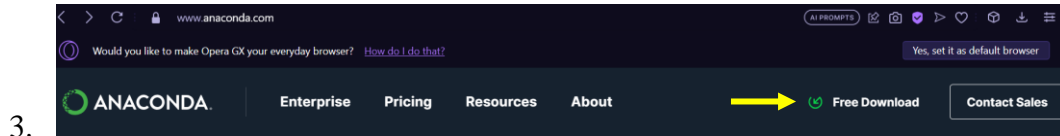
Nama : Harry Liman
Umur ; 50 tahun
Berat badan : 70 Kg
Jenis kelamin : Laki – laki

Proses pengambilan data

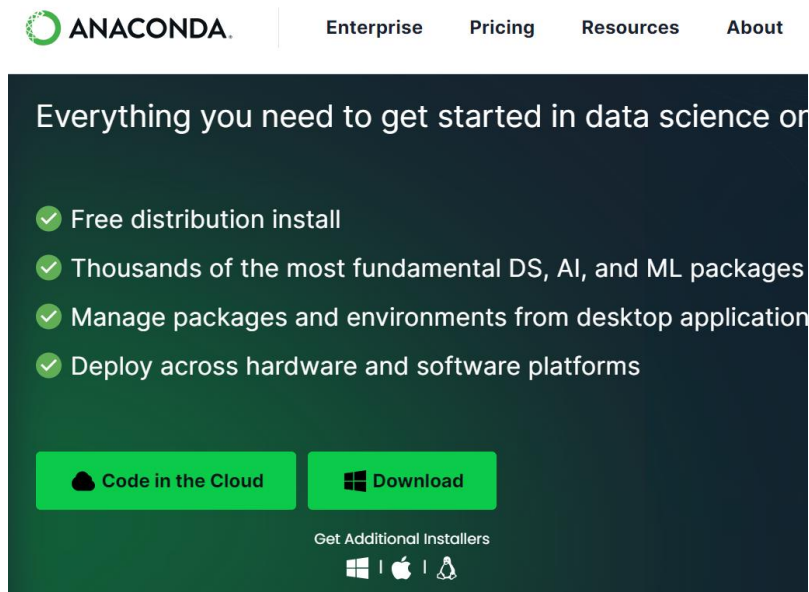


Panduan Instalasi Anaconda dan Jupyter Notebook :

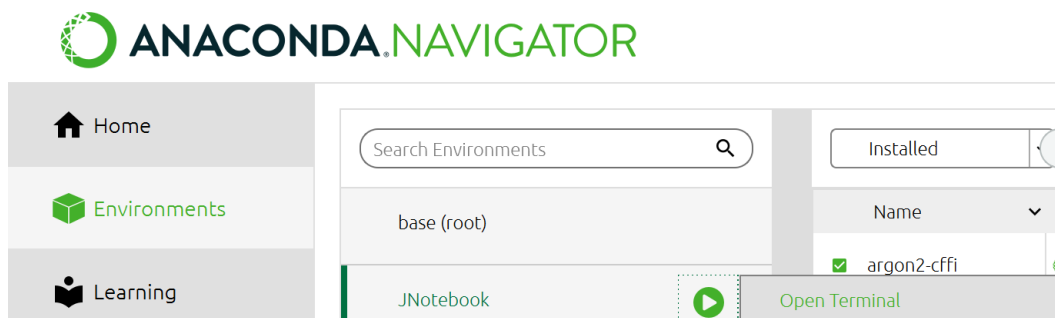
1. Buka website anaconda pada tautan berikut : <https://www.anaconda.com>
2. Pilih tap download



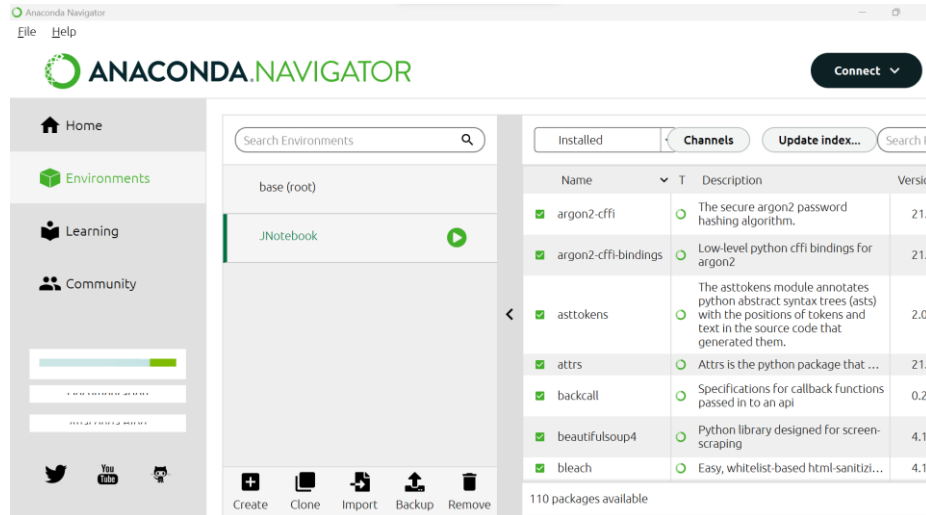
4. Download sesuai kebutuhan device (Windows/Linux/Ios)



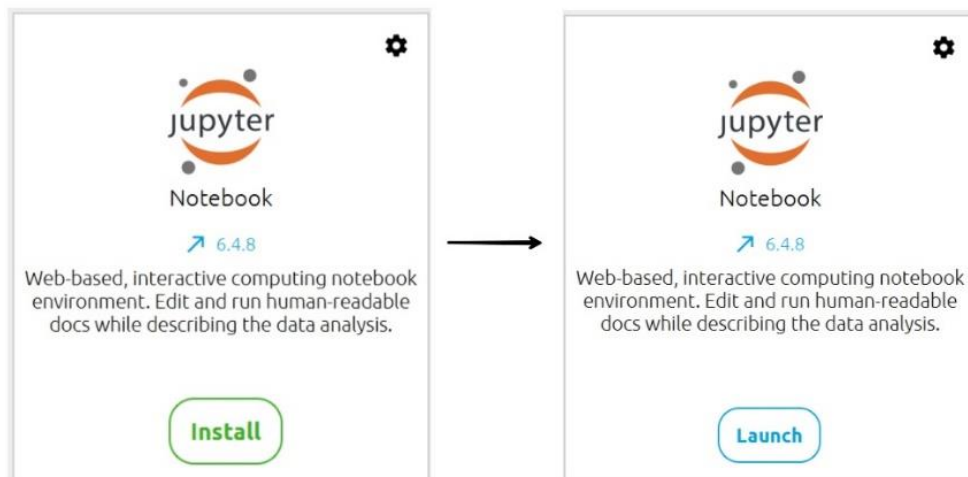
5. Install dan tunggu hingga proses penginstalan selesai
6. Buka anaconda yang sudah terinstall, maka akan masuk pada tampilan anaconda navigator.
7. Untuk mencari library python yang dibutuhkan dapat menggunakan tautan berikut: <https://pypi.org>
8. Untuk menambahkan library pada anaconda terdapat pada bagian “Enviroments”



9. Untuk melihat library yang terinstall dapat membuka pada bagian “Enviroments” dan juga dapat membuat pekerjaan baru sehingga library tidak akan bercampuran dengan menggunakan “Create” pada “Enviroments”



10. Menginstall Jupyter Notebook pada Anaconda Navigator, hingga keterangan “install” berubah menjadi “launch”

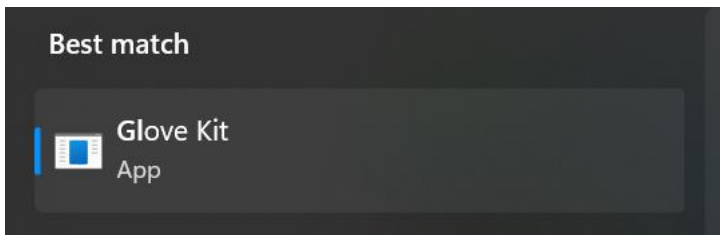


Panduan Penggunaan USB Glove Kit :

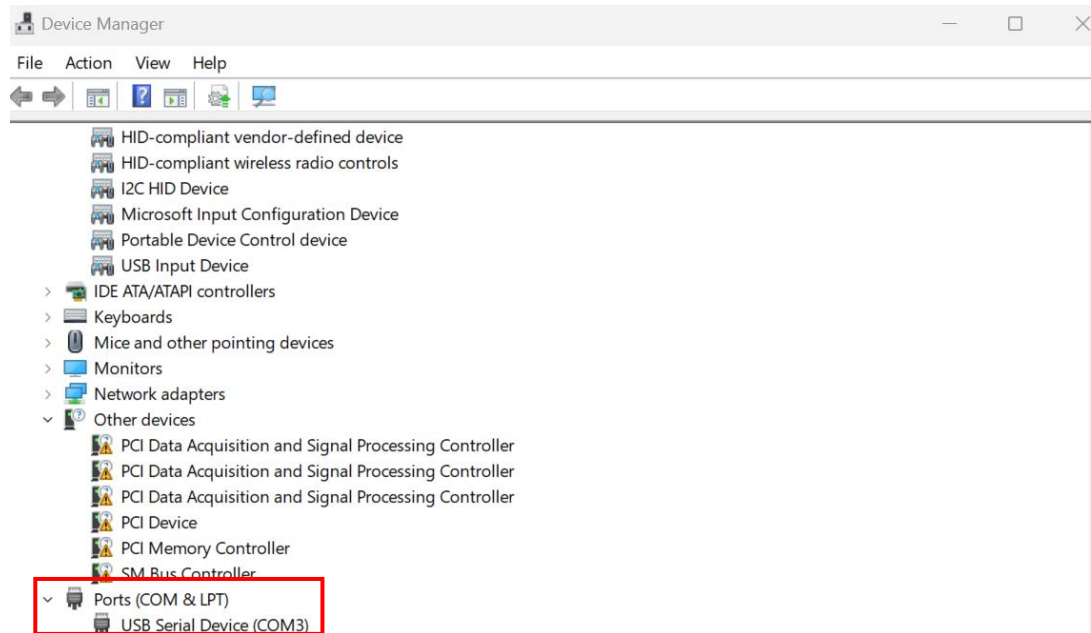
1. Menghubungkan USB Glove Kit dengan device



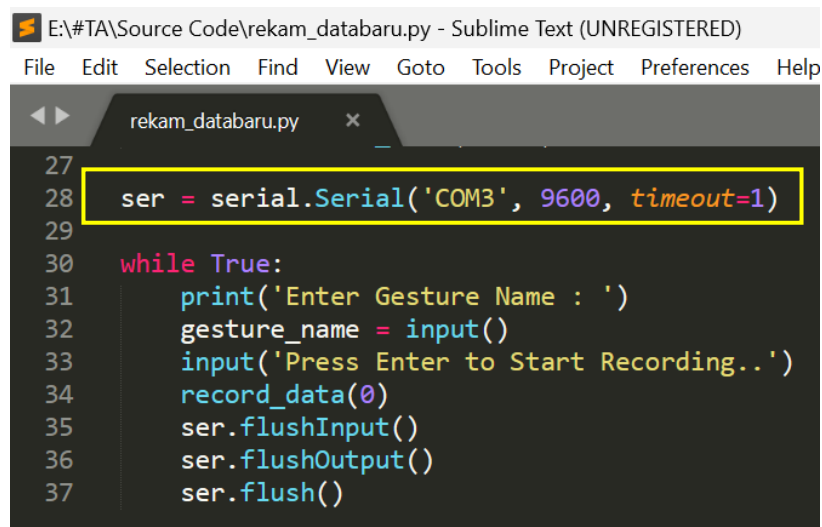
2. Membuka aplikasi glove kit untuk memastikan USB Glove kit sudah terhubung



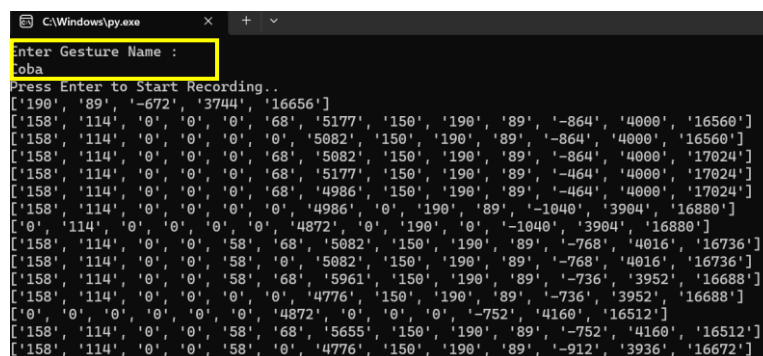
3. Membuka device manager untuk melihat port yang digunakan oleh USB Glove kit



4. Mengganti port pada code serial jika port tidak sama



5. Data siap untuk diambil dan menuliskan nama data untuk file csv nya nanti



Kode yang digunakan

1. Pengambilan data sensor

a. Import Library yang digunakan

```
import serial
import time
import csv
import re
```

b. Setting serial port yang akan digunakan

```
ser = serial.Serial('COM3', 9600, timeout=1)
```

c. Pengambilan data

```
while True:
    print('Enter Gesture Name : ')
    gesture_name = input()
    input('Press Enter to Start Recording..')
    record_data(0)
    ser.flushInput()
    ser.flushOutput()
    ser.flush()
```

d. Kode lengkap

```
import serial
import time
import csv
import re

def bacadata():
    data = ser.readline()
    return data

def record_data(index):
    if (index > 200) :
        print('Record End')
        return
    caught = bacadata()
    caught = re.findall(rb'(?<=:) (.*) (?=])',
    caught)
    caught = [x.decode() for x in caught][:-4]
```



```

        if (index > 0) :
            with open(gesture_name + '.csv', 'a',
encoding='iso-8859-1', newline='\n') as csvfile:
                writer          = csv.writer(csvfile)
                writer.writerow(catched)
            index+=1
            print(catched)

        return record_data(index)

ser = serial.Serial('COM3', 9600, timeout=1)

while True:
    print('Enter Gesture Name : ')
    gesture_name = input()
    input('Press Enter to Start Recording..')
    record_data(0)
    ser.flushInput()
    ser.flushOutput()
    ser.flush()

```

2. Penyatuan file sensor dan pemberian label

```
import numpy as np
import pandas as pd
import tensorflow as tf

data1 = pd.read_csv("Data 1.csv",header=None)
data2 = pd.read_csv("Data 2.csv",header=None)
data3 = pd.read_csv("Data 3.csv",header=None)
data4 = pd.read_csv("Data 4.csv",header=None)
data5 = pd.read_csv("Data 5.csv",header=None)

df1 = pd.DataFrame(data1)
df2 = pd.DataFrame(data2)
df3 = pd.DataFrame(data3)
df4 = pd.DataFrame(data4)
df5 = pd.DataFrame(data5)

data = [df1, df2, df3, df4, df5]

df200 = pd.concat(data, ignore_index=True)
df200.columns =
['mcpPinky','pipPinky','mcpRing','pipRing','mcpMid','p
ipMid',

'mcpIndex','ipIndex','mcpThumb','pipThumb','accX','acc
Y',

'accZ','Output']

df200.to_csv ('Data TA.csv', index=False)
```

3. Model KNN

a. Import Library yang digunakan

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import classification_report,
confusion_matrix
```

b. Penentuan nilai K

```
error_rate = []
for i in range(3,20):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_train,y_train)
    pred_i = knn.predict(x_test)
    error_rate.append(np.mean(pred_i != y_test))

print(error_rate)
```

c. Classification_report

```
print(classification_report(y_test, y_pred))
```

d. Kode lengkap

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import classification_report,
confusion_matrix

df1 = pd.read_csv("Data TA.csv")
df1

X = df1.iloc[:, :13].values
y = df1.iloc[:, 13].values

X_train, X_test, y_train, y_test = train_test_split
(X, y, test_size = 0.3)

print("X Train:", X_train.shape)
print("y Train:", y_train.shape)
print("X Test:", X_test.shape)
print("y Test:", y_test.shape)
```

```

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
scaler.fit(X_train)

x_train = scaler.transform(X_train)
x_test = scaler.transform(X_test)

from sklearn.neighbors import KNeighborsClassifier

error_rate = []
for i in range(3,20):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_train,y_train)
    pred_i = knn.predict(x_test)
    error_rate.append(np.mean(pred_i != y_test))

print(error_rate)

knn3 = KNeighborsClassifier (n_neighbors=3)

knn3.fit(X_train, y_train)

import joblib
joblib.dump(knn3, 'KNN.ta')

model = joblib.load('KNN.ta')
y_pred = model.predict(X_test).astype(int).flatten()
print(y_pred)

y_test = y_test.flatten()
print(y_test)
print("Y Test:", y_test.shape)
print("Y Pred:", y_pred.shape)

df_knn = pd.DataFrame({'Actual':y_test,
                       'Predicted':y_pred})
df_knn

data100 = pd.DataFrame(df_knn)
data100.to_csv ('Hasil klasifikasi KNN.csv',
index=False)
print(classification_report(y_test, y_pred))

```

4. Pengujian data

a. Import Library yang digunakan

```
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
```

b. Import Model KNN

```
knn = KNeighborsClassifier(n_neighbors=3)
```

c. Import file yang akan di uji

```
df_new1 = pd.read_csv("0.csv")
```

d. Konversi data

```
def konv(hasil_prediksi):
    if hasil_prediksi == 21:
        return "puluh"
    if hasil_prediksi == 22:
        return "ratus"
    if hasil_prediksi == 23:
        return "ribu"
    if hasil_prediksi == 24:
        return "juta"

    else:
        return str(hasil_prediksi)
```

e. Penggabungan data

```
hasil_gabungan = konv(data1[0]) + konv(data4[0]) +
konv(data2[0]) + konv(data3[0])

print(hasil_gabungan)
```

f. Kode lengkap

```
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier

df1 = pd.read_csv("Data TA.csv")
df1

X_train = df1.iloc[:, :13].values
y_train = df1.iloc[:, 13].values
```

```

knn = KNeighborsClassifier(n_neighbors=3)

knn.fit(X_train, y_train)

df_new1 = pd.read_csv("0.csv")

X_new1 = df_new1.iloc[:, :-1].values

y_pred1 = knn.predict(X_new1)

data1 = y_pred1

def konv(hasil_prediksi):
    if hasil_prediksi == 21:
        return "puluh"
    if hasil_prediksi == 22:
        return "ratus"
    if hasil_prediksi == 23:
        return "ribu"
    if hasil_prediksi == 24:
        return "juta"

    else:
        return str(hasil_prediksi)

print(data1[0])

hasil_gabungan = konv(data1[0]) + konv(data4[0]) +
konv(data2[0]) + konv(data3[0])

print(hasil_gabungan)

```

5. Text to speech

a. Instal dan Import Library yang digunakan

```
!pip install pyttsx3
pip install gtts
import pyttsx3
from gtts import gTTS
from io import BytesIO
import os
```

b. Kode lengkap

```
!pip install pyttsx3
pip install gtts
import pyttsx3
from gtts import gTTS
from io import BytesIO
import os

tts = gTTS(hasil_gabungan, lang='id')
tts.save("hasil_suara.mp3")

os.system("start hasil_suara.mp3")
```