

**PERANCANGAN DAN IMPLEMENTASI APLIKASI
MOBILE IDENTIFIKASI AUTISM SPECTRUM
DISORDER (ASD) BERBASIS CONVOLUTIONAL
NEURAL NETWORK (CNN)**

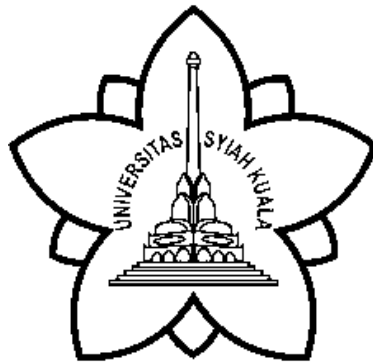
PROPOSAL TUGAS AKHIR

Program Studi
Teknik Komputer

Diajukan Oleh :

ALYAFI

1904111010051



**JURUSAN TEKNIK ELEKTRO DAN KOMPUTER
FAKULTAS TEKNIK UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH**

2022

PERNYATAAN KEASLIAN PROPOSAL

Saya menyatakan dengan sesungguhnya proposal tugas akhir dengan judul **”Perancangan dan Implementasi Aplikasi Mobile Identifikasi Autism Spectrum Disorder (ASD) berbasis Convolutional Neural Network (CNN)”** bukan merupakan tiruan atau duplikasi dari tugas akhir atau karya ilmiah yang telah dipublikasi oleh orang lain, kecuali bagian sumber informasinya dicantumkan sebagaimana mestinya. Bila ditemukan hal-hal yang tidak sesuai dengan isi pernyataan ini, saya menerima dan menghormati segala konsekuensi akademis yang diberikan Program Teknik Komputer Universitas Syiah Kuala.

Banda Aceh, September 2022

Al Yafi

NIM : 1904111010051

PENGESAHAN PROPOSAL

APLIKASI MOBILE IDENTIFIKASI AUTISM SPECTRUM DISORDER (ASD) DENGAN BERBASIS CONVOLUTIONAL NEURAL NETWORK

Diajukan Oleh :

Al Yafi

1904111010051

Telah disetujui oleh
Komisi Pembimbing

Banda Aceh, 2022

Pembimbing I

Pembimbing II

Dr. Ir. Melinda, S.T., M.Sc., IPU., ASEAN Eng.
NIP. 19790610200212201

Prof. Dr.Ir. Fitri Arnia, S.T., M.Eng.Sc., IPU
NIP. 199010252020072101

Mengetahui,
Koordinator Program Studi Teknik Komputer

Dr. Kahlil Muchtar, S.T., M.Eng
NIP. 198512022017091101

ABSTRAK

Autism Spectrum Disorder (ASD) merupakan sebuah kondisi neurodevelopmental yang dicirikan oleh kesulitan dalam interaksi sosial, komunikasi, dan perilaku yang repetitif. Ini merupakan spektrum kondisi, dengan tingkat keparahan yang bervariasi dari individu ke individu. Beberapa orang dengan ASD mungkin memiliki keterbatasan yang signifikan dalam berkomunikasi dan bersosialisasi, sementara yang lain mungkin memiliki keterbatasan yang lebih ringan dan mampu bekerja dan berfungsi secara independen. Diagnosis ASD dibuat oleh profesional kesehatan mental berdasarkan evaluasi perilaku dan gejala yang dicatat dalam waktu yang cukup lama sehingga dibutuhkan diagnosis terhadap anak penderita ASD dengan perlakuan medis yang cepat dan tepat sasaran. EEG atau Electroencephalography adalah sebuah metode pemeriksaan yang digunakan untuk memantau aktivitas listrik di dalam otak. Dalam konteks *Autism Spectrum Disorder* (ASD), data EEG dapat digunakan untuk membantu dalam diagnosis atau evaluasi kondisi ini. Saat ini fasilitas diagnosis dini penderita ASD dengan sinyal *Electroencephalogram* (EEG) masih terbatas, sehingga dibutuhkan biaya yang besar untuk mengakses fasilitas tersebut. Penelitian ini mengusulkan sebuah Aplikasi *Mobile* yang diharapkan dapat meningkatkan akses bagi penderita ASD untuk mendiagnosis data EEG secara mandiri. Aplikasi *Mobile* yang dibangun dapat membedakan penderita ASD dan normal berdasarkan data sinyal otak EEG secara *asynchronous*. Selain itu, penelitian ini menghasilkan algoritma *preprocessing* dan data sinyal EEG BCI2000 sehingga dapat diotomasi menggunakan *python*. Penelitian ini juga menghasilkan luaran model terlatih *Deep Learning Convolutional Neural Network* yang di-deploy menggunakan Python-Flask sehingga diagnosis sinyal EEG penderita ASD dan normal dapat digunakan diberbagai *platform* melalui restAPI.

DAFTAR ISI

PERNYATAAN KEASLIAN PROPOSAL	i
PENGESAHAN PROPOSAL	ii
ABSTRAK	iii
DAFTAR ISI.....	iv
DAFTAR TABEL	vii
DAFTAR SINGKATAN.....	viii
BAB 1 PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 RUMUSAN MASALAH.....	3
1.3 RUANG LINGKUP	3
1.4 TUJUAN	4
1.5 URGENSI DAN MANFAAT PENELITIAN	4
1.6 SISTEMATIKA PENULISAN	4
BAB 2 TINJAUAN PUSTAKA.....	6
2.1 <i>AUTISM SPECTRUM DISORDERS (ASD)</i>	6
2.2 <i>ELECTROENCEPHALOGRAPHY (EEG)</i>	6
2.3 KARAKTERISTIK GELOMBANG OTAK.....	9
2.4 <i>BRAIN COMPUTER INTERFACE (BCI)</i>	11
2.5 BCI2KREADER.....	12
2.6 <i>PRE-PROCESSING DATASET</i>	13
2.6.1 <i>EEG Signal to Matrix</i>	13
2.6.2 <i>Discrete Wavelet Transform</i>	14
2.6.3 <i>EEG to Frequency Spectrum Image (FSI)</i>	15
2.7 <i>DEEP LEARNING</i>	17
2.7.1 TensorFlow	18
2.7.2 Convolutional Neural Network (CNN)	18
2.7.3 Dropout	21
2.7.4 Loss dan Akurasi	21
2.8 ARSITEKTUR SISTEM APLIKASI MOBILE	22

2.8.1 Layanan Firebase.....	23
2.8.2 Python Flask.....	24
2.8.3 Flutter	24
2.9 UJI PEFORMA SISTEM.....	24
2.9.1 Convusion Matrix	25
2.9.2 Uji Perfroma Mobile Apps.....	26
2.10 PENELITIAN TERKAIT	27
BAB 3 METODELOGI DAN JADWAL PENELITIAN	31
3.1 METODE PENELITIAN	31
3.1.1 Bahan Penelitian	31
3.1.2 Alat Penelitian	32
3.1.3 Alur Penelitian	33
3.2 JADWAL PENELITIAN	45
BAB 4 LUARAN	47
DAFTAR PUSTAKA	48

DAFTAR GAMBAR

Gambar 2.1 Headset EEG: (a) headset EEG dengan jumlah Channel sebanyak 16 dan (b) posisi Channel EEG diperoleh menurut sistem internasional 10-20 [17].	9
Gambar 2.2 Contoh sinyal EEG pada penderita ASD [18].	10
Gambar 2.3 Contoh hasil pembacaan sinyal EEG BCI2000 dengan modul Python yakni BCI2kReader [20].	13
Gambar 2.4 Data EEG yang ditampung dalam Matrix 2 Dimensi [16].	14
Gambar 2.5 Contoh merubah sinyal EEG ke bentuk Frequency Spectrum Image [26].	17
Gambar 2.6 Struktur Convolutional Neural Network (CNN) [23].	19
Gambar 2.7 Lapisan konvolusi [24], [32].	20
Gambar 2.8 Lapisan Pooling [32].	20
Gambar 2.9 lapisan softmax [24].	20
Gambar 2.10 Tampilan loss dan accuracy [32].	22
Gambar 2. 11 Contoh arsitektur aplikasi mobile menggunakan	23
Gambar 2.12 Confusion Matrix [31].	25
Gambar 3.1 Grafik sinyal EEG menggunakan python.	31
Gambar 3.2 Alur Penelitian.	33
Gambar 3.3 Proses Preprocessing pada sinyal EEG	34
Gambar 3.4 Klasifikasi Data EEG menggunakan CNN	35
Gambar 3.5 Tahapan Deploy Trained Model	37
Gambar 3.6 Tahapan Pembangunan Mobile Apps	39
Gambar 3.7 Skema arsitektur umum aplikasi mobile yang ditawarkan untuk klasifikasi penderita ASD menggunakan sinyal EEG	40
Gambar 3 8 Use case dan Komponen Diagram pada aplikasi yang akan dibuat..	41

DAFTAR TABEL

Tabel 2.1 Studi literatur terkait Aplikasi klasifikasi sinyal EEG	27
Tabel 2.2 Studi literatur terkait klasifikasi sinyal EEG.....	28
Tabel 3.1 Alat Penelitian.....	32
Tabel 3.2 Jadwal Penelitian.....	45

DAFTAR SINGKATAN

Autism Spectrum Disorder	: (ASD)
Electroencephalograph	: (EEG)
Convolutional Neural Network	: (CNN)
Brain Computer Interface	: (BCI)
King Abdulaziz University	: (KAU)
Application Programming Interface	: (API)
False Positive	: (FP)
False Negative	: (FN)
True Positive	: (TP)
True Negative	: (TN)

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Diagnosis penderita *Autism Spectrum Disorder* (ASD) pada anak umumnya hanya mengandalkan beberapa parameter pemeriksaan fisik, perilaku, dan pemeriksaan tumbuh kembang anak, pemeriksaan ini meliputi dengan memeriksa keluhan utama seperti keterlambatan dalam perkembangan bahasa, regresi kemampuan berbahasa, keterlambatan dalam kemampuan komunikasi, perilaku abnormal, bentuk fisik, perilaku dan sebagainya [1]. Model pemeriksaan ini akan membutuhkan waktu lama karena harus memperhatikan tumbuh kembang anak dan harus memerlukan usaha yang lebih untuk memastikan bahwa seorang anak menderita ASD atau tidak. Interpretasi sinyal EEG secara manual dapat menjadi sulit karena beberapa hal berikut:

1. Keanekaragaman pola aktivitas otak: Sinyal EEG dapat menunjukkan pola aktivitas otak yang sangat beragam, dan ini bisa membuat sulit untuk menentukan apa yang normal dan apa yang abnormal [1].
2. Variasi individual: Setiap individu memiliki pola aktivitas otak yang unik dan berbeda, sehingga sulit untuk membandingkan sinyal EEG antar individu dan menentukan apa yang normal dan apa yang abnormal [2].
3. Kebisingan: Sinyal EEG dapat terkontaminasi oleh kebisingan dari luar seperti gerakan otot, elektromagnetik, atau sumber-sumber eksternal lainnya. Ini membuat sulit untuk membedakan antara sinyal EEG yang benar-benar mencerminkan aktivitas otak dan sinyal yang terkontaminasi [3].
4. Kemampuan interpretasi yang berbeda: Interpretasi sinyal EEG sangat bergantung pada keahlian dan pengalaman individu yang melakukannya. Beberapa orang mungkin memiliki kemampuan interpretasi yang lebih baik dibandingkan orang lain, dan ini bisa mempengaruhi kualitas interpretasi yang diperoleh [4].

Karena alasan-alasan ini, interpretasi sinyal EEG secara manual membutuhkan keterampilan dan pengalaman yang tinggi, dan seringkali harus dikonfirmasi oleh analisis komputer untuk memastikan akurasi dan validitas hasil.

Beberapa teknologi canggih saat ini digunakan untuk membantu dalam mendiagnosis Autism Spectrum Disorder (ASD) melalui EEG (Electroencephalography), di antaranya:

1. Analisis EEG otomatis: Ini adalah teknologi yang menggunakan algoritma komputer untuk menganalisis sinyal EEG dan mengidentifikasi pola aktivitas otak yang mungkin terkait dengan ASD. Analisis otomatis ini dapat membantu dalam mengurangi variasi individual dan meningkatkan akurasi dalam interpretasi sinyal EEG [3].
2. Brain-Computer Interface (BCI): BCI adalah teknologi yang menghubungkan otak dan komputer, memungkinkan sinyal otak untuk dipantau dan dianalisis secara real-time. Dalam konteks ASD, BCI dapat digunakan untuk memantau dan menganalisis respon otak terhadap rangsangan sosial dan lingkungan, membantu dalam mengidentifikasi pola aktivitas otak yang anormal [5], [6].
3. Artificial Intelligence (AI): AI adalah teknologi yang menggunakan algoritma komputer untuk menganalisis dan mempelajari data secara otomatis. Dalam konteks ASD, AI dapat digunakan untuk membantu dalam analisis EEG, meningkatkan akurasi dalam mengidentifikasi pola aktivitas otak yang mungkin terkait dengan kondisi ini [5], [7], [8].

Teknologi-teknologi ini membantu dalam mempercepat dan mempermudah proses diagnostik dan membantu dalam membuat diagnosis yang lebih akurat dan cepat. Namun, harus diingat bahwa teknologi ini hanya memberikan informasi tambahan dan bahwa diagnosis akhir harus dibuat oleh profesional kesehatan mental berdasarkan evaluasi perilaku dan gejala yang dicatat dalam waktu yang cukup lama. Namun hingga saat ini, masih sedikit teknologi yang dapat menganalisis sinyal EEG dan mendiagnosis penderita ASD secara otomatis sehingga membutuhkan banyak biaya dan waktu dan membuat banyak anak penderita ASD tidak menyadari penyakit ini secara dini dan menimbulkan dampak buruk bagi perkembangannya [4], [5], [7].

Penelitian ini mengusulkan sebuah aplikasi mobile yang dapat membedakan penderita ASD dan normal berdasarkan data sinyal otak EEG secara *asynchronous* artinya pengguna hanya perlu memiliki data EEG yang sudah direkam kemudian

meng-*input* data tersebut menggunakan *Smartphone* dan informasi hasil diagnosis akan ditampilkan di *Smartphone*. Penelitian ini menghasilkan algoritma *preprocessing* dan data sinyal EEG BCI2000 menggunakan *library python* BCI2K yang memungkinkan *preprocessing* data EEG diautomasi menggunakan *python* [6], [9]. Selain itu, penelitian ini menghasilkan luaran model terlatih *Deep Learning* dengan menggunakan metode *Convolutional Neural Network* (CNN) yang di-*deploy* menggunakan Python-Flask sehingga diagnosis penderita ASD dan normal berdasarkan sinyal EEG dapat digunakan diberbagai *platform* melalui restAPI [10]–[12], tetapi penelitian ini hanya difokuskan pada penggunaan *platform Android* menggunakan multi-*platform Framework* yakni Flutter [13]. Penelitian ini juga diharapkan membantu masyarakat sehingga mempercepat diagnosa penderita ASD agar dapat ditangani secara tepat sasaran. Data penderita ASD dan data normal diklasifikasikan dengan benar ke kelas yang sesuai. Penanganan penyakit ini membutuhkan kerja sama yang erat dan terpadu sehingga klasifikasi ASD ini akan sangat membantu penderita dan dapat membuat diagnosis dalam waktu yang lebih cepat memungkinkan penderita memperoleh penanganan dan terapi secara tepat sasaran.

1.2 RUMUSAN MASALAH

Rumusan masalah pada penelitian ini adalah:

1. Tenaga kesehatan/dokter kesulitan menganalisis data EEG secara manual dan analisis dilakukan berdasarkan evaluasi perilaku dan gejala yang dicatat dalam waktu yang cukup lama.
2. Untuk beberapa format data EEG dibutuhkan metode *preprocessing* khusus dan tidak otomatis.
3. Implementasi model *Deep Learning* terlatih yang telah diautomasi pada Aplikasi *Mobile*.

1.3 RUANG LINGKUP

Adapun ruang lingkup pada penelitian ini adalah membuat aplikasi *mobile* yang dapat mengidentifikasi penderita ASD secara *asynchronous*, membangun *Deep Learning* model untuk klasifikasi penderita ASD dan membuat arsitektur

Cloud untuk memungkinkan menggunakan model *Deep Learning* pada aplikasi *mobile*.

1.4 TUJUAN

Tujuan pada Penelitian ini adalah :

1. Merancang dan membangun Aplikasi *Mobile* berbasis android yang dapat membedakan data sinyal EEG normal dan ASD secara *Asynchronous*.
2. Membangun algoritma *preprocessing* sinyal EEG BCI200 menggunakan pemrograman python.
3. Membangun sistem automasi klasifikasi sinyal EEG menggunakan model *Deep Learning* dengan metode *Convolutional Neural Network* (CNN) berbasis *Framework Tensorflow*.

1.5 URGENSI DAN MANFAAT PENELITIAN

Adapun manfaat yang diharapkan dari penelitian ini yaitu dapat mendiagnosis sinyal EEG antara ASD dan Normal menggunakan aplikasi mobile secara *asynchronous* yang memungkinkan penggunaan aplikasi dan alat EEG tidak perlu digunakan secara bersamaan sehingga menurunkan biaya operasional untuk melakukan diagnosis. Penelitian ini juga diharapkan membantu masyarakat dan mempercepat diagnosis ASD dan dapat ditangani secara tepat sasaran.

1.6 SISTEMATIKA PENULISAN

Penelitian Tugas Akhir ini terdiri dari komponen-komponen beserta tahapan dan penjelasan pada setiap bab dan sub-babnya. Adapun sistematika penulisan pada tugas akhir ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Pada bab ini berisi tentang latar belakang, rumusan masalah, tujuan, urgensi dan manfaat penelitian penelitian dan sistematika penulisan penelitian tugas akhir membahas pengimplementasian model *Deep Learning* dan ekstraksi model yang

akan digunakan restAPI agar memungkinkan dilakukannya identifikasi ASD dan Normal menggunakan aplikasi *mobile*.

BAB 2 TINJAUAN PUSTAKA

Pada bab ini berisi tentang teori yang berkaitan dengan ASD, arsitektur aplikasi secara keseluruhan seperti sistem klasifikasi menggunakan *Deep Learning* dengan metode CNN (Convolutional Neural Network), dan Cloud Service untuk penerapan model *Deep Learning* pada aplikasi *mobile*.

BAB 3 METODOLOGI DAN JADWAL PENELITIAN

Bab ini berisi penjabaran metode penelitian yang digunakan secara lebih rinci. Bab ini juga menjelaskan tentang tahapan penelitian, model yang digunakan untuk klasifikasi citra yang menggunakan *Convolutional Neural Network* (CNN), metode pembuatan aplikasi, bahan dan alat, dan jadwal penelitian.

BAB 4 LUARAN PENELITIAN

Pada bab ini berisi tentang luaran yang diharapkan pada penelitian ini.

DAFTAR PUSTAKA

Bab ini berisi referensi yang digunakan pada penelitian tugas akhir ini. Referensi ini didapat dari berbagai sumber seperti buku, dan jurnal-jurnal penelitian, referensi online.

BAB 2

TINJAUAN PUSTAKA

2.1 *AUTISM SPECTRUM DISORDERS (ASD)*

Autism Spectrum Disorder (ASD) adalah kondisi neurodiversitas yang dicirikan oleh pola perilaku, komunikasi, dan interaksi sosial yang abnormal. Kondisi ini biasanya ditemukan sejak masa kecil dan dapat mempengaruhi banyak aspek dari kehidupan seseorang, termasuk keterampilan berkomunikasi, kemampuan untuk membangun dan mempertahankan hubungan sosial, dan kemampuan untuk belajar dan beradaptasi dengan lingkungan baru [4].

ASD berada di spektrum, artinya bahwa gejala dan tingkat keparahan dapat bervariasi dari individu ke individu. Beberapa individu dengan ASD mungkin memiliki keterampilan berkomunikasi yang baik dan mampu belajar dan beradaptasi dengan lingkungan dengan baik, sementara yang lain mungkin memiliki keterbatasan yang signifikan dalam hal ini [14].

Faktor yang mempengaruhi pengembangan ASD belum sepenuhnya diketahui, namun penelitian menunjukkan bahwa kombinasi faktor genetik dan lingkungan mungkin memainkan peran. Beberapa tipe terapi dan intervensi, seperti terapi bahasa, terapi perilaku, dan terapi sosial, dapat membantu individu dengan ASD untuk meningkatkan keterampilan dan memfasilitasi interaksi sosial yang lebih baik [14].

Diagnosis ASD dapat dibuat oleh profesional kesehatan mental melalui evaluasi perilaku dan gejala yang dicatat dalam waktu yang cukup lama, dan dapat memerlukan beberapa tes tambahan, seperti tes neuropsikologis dan tes EEG. Diagnosis dini dan intervensi yang tepat dapat membantu individu dengan ASD untuk mencapai potensinya dan meningkatkan kualitas hidup mereka [4].

2.2 *ELECTROENCEPHALOGRAPHY (EEG)*

Electroencephalogram (EEG) adalah teknik pengukuran yang digunakan untuk memantau aktivitas listrik dalam otak. Ini dilakukan dengan menempatkan

elektroda pada kepala individu dan merekam sinyal elektrik yang dihasilkan oleh sel-sel saraf di dalam otak [4].

EEG merupakan salah satu alat yang paling penting dalam neurodiagnostik, dan sering digunakan untuk membantu menentukan diagnosa kondisi medis seperti epilepsy, insomnia, dan cedera otak. Dalam konteks autisme, EEG dapat digunakan untuk memantau aktivitas listrik dalam otak individu dengan diagnosa ASD dan membandingkannya dengan individu tanpa diagnosa untuk mencari perbedaan yang signifikan [14].

Proses EEG sangat non-invasif dan tidak menyakitkan, dan umumnya memakan waktu sekitar satu jam. Elektroda yang digunakan dalam proses ini biasanya ditempelkan pada kepala dengan menggunakan gel elektrolitik, dan sinyal yang dihasilkan diterjemahkan menjadi pola gelombang yang dapat dianalisis oleh profesional kesehatan untuk menentukan apakah ada perbedaan dalam aktivitas listrik dalam otak [4].

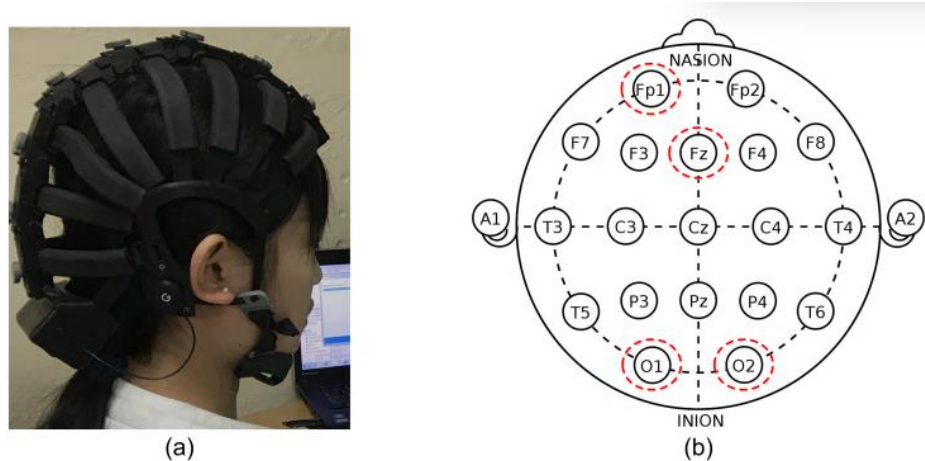
Meskipun EEG adalah alat yang berguna dan penting dalam proses diagnostik, interpretasi sinyal EEG tidak selalu mudah dan memerlukan interpretasi yang akurat dari profesional kesehatan yang berpengalaman. Oleh karena itu, hasil EEG harus diinterpretasikan dengan hati-hati dan dalam konteks yang lebih luas dari gejala dan riwayat medis individu.

Channel EEG merujuk pada elektroda atau titik perekaman di kepala manusia untuk merekam aktivitas listrik yang terjadi di otak. Channel EEG umumnya digunakan dalam perekaman elektroensefalogram (EEG), sebuah teknik diagnostik medis yang mengukur dan merekam aktivitas listrik di otak dengan mengamati pola gelombang otak yang terjadi pada waktu yang berbeda [15]. Dalam perekaman EEG, beberapa elektroda dipasang pada kepala dan ditempatkan pada lokasi yang ditentukan untuk merekam aktivitas listrik di berbagai bagian otak. Channel EEG sering digunakan dalam studi neurofisiologi, psikofisiologi, dan neurologi untuk memahami fungsi dan aktivitas otak dalam berbagai kondisi dan situasi. Ada beberapa jenis channel EEG yang tersedia, termasuk 16-channel, 32-channel, 64-channel, dan 128-channel EEG, tergantung pada jumlah elektroda yang digunakan dalam perekaman aktivitas otak [15], [16] .

Pada penelitian ini EEG yang digunakan adalah sebanyak 16 channel yang berarti perekaman aktivitas listrik pada 16 titik di kepala manusia. Setiap channel memiliki nama dan posisi penempatan yang spesifik pada kepala. Berikut adalah daftar nama channel dan posisi penempatannya pada 16 channel EEG:

1. Fp1: Forehead, kiri atas
2. Fp2: Forehead, kanan atas
3. F3: Frontal, kiri atas
4. F4: Frontal, kanan atas
5. C3: Central, kiri tengah
6. C4: Central, kanan tengah
7. P3: Parietal, kiri atas
8. P4: Parietal, kanan atas
9. O1: Occipital, kiri atas
10. O2: Occipital, kanan atas
11. F7: Frontal, kiri lebih atas dari F3
12. F8: Frontal, kanan lebih atas dari F4
13. T3: Temporal, kiri atas dari C3
14. T4: Temporal, kanan atas dari C4
15. T5: Temporal, kiri atas dari P3
16. T6: Temporal, kanan atas dari P4

Perlu dicatat bahwa ada sistem perekaman EEG lainnya yang dapat menggunakan nama channel dan posisi penempatan yang berbeda. Oleh karena itu, penting untuk memeriksa sistem perekaman EEG yang digunakan untuk memastikan nama channel dan posisi penempatan yang tepat.



Gambar 2.1 Headset EEG: (a) headset EEG dengan jumlah Channel sebanyak 16 dan (b) posisi Channel EEG diperoleh menurut sistem internasional 10-20 [17].

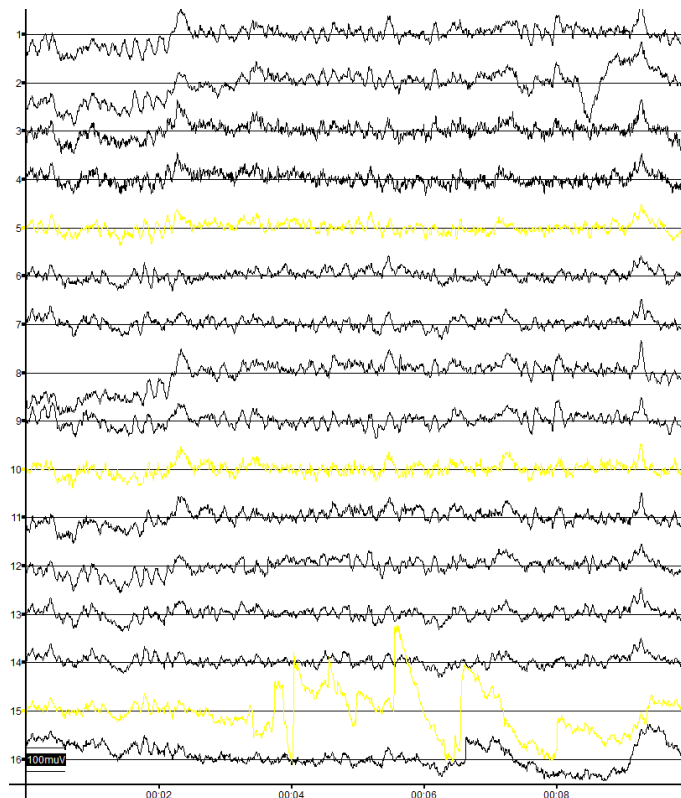
Pada Gambar 2.1 (a) menunjukkan headset yang dipakai untuk merekam sinyal EEG dari berbagai area kulit kepala. Pada Gambar 2.1 (b) merupakan gambaran posisi Channel EEG ditempatkan. Secara khusus, saluran bipolar yang terdiri dari elektroda Fp1 dan Fz digunakan untuk merekam sinyal EEG dari korteks frontal, dan saluran bipolar lain yang terdiri dari elektroda O1 dan O2 digunakan untuk menangkap sinyal EEG dari korteks oksipital [16], [18].

2.3 KARAKTERISTIK GELOMBANG OTAK

Gelombang otak adalah sinyal elektrik yang dihasilkan oleh aktivitas sel saraf di dalam otak. Ada beberapa jenis gelombang otak yang dikenali, masing-masing memiliki frekuensi dan amplitudo yang berbeda. Gelombang delta adalah gelombang otak dengan frekuensi terendah dan amplitudo tertinggi. Ini biasanya terjadi pada saat tidur dan menunjukkan aktivitas listrik yang sangat rendah dalam otak. Gelombang theta adalah gelombang otak dengan frekuensi sedang dan amplitudo sedang. Ini biasanya terjadi pada saat tidur lelap atau meditasi dan menunjukkan sedikit aktivitas listrik dalam otak. Gelombang alfa adalah gelombang otak dengan frekuensi sedang dan amplitudo rendah. Ini biasanya terjadi pada saat relaksasi dan menunjukkan sedikit aktivitas listrik dalam otak. Gelombang beta adalah gelombang otak dengan frekuensi tertinggi dan amplitudo rendah. Ini biasanya terjadi pada saat konsentrasi dan menunjukkan aktivitas listrik

yang sangat tinggi dalam otak. Gelombang gamma adalah gelombang otak dengan frekuensi yang sangat tinggi dan amplitudo yang sangat rendah. Ini menunjukkan aktivitas listrik yang sangat kompleks dan intens dalam otak. Masing-masing jenis gelombang otak memiliki karakteristik yang berbeda dan dapat memberikan informasi tentang bagaimana otak berfungsi. Oleh karena itu, analisis gelombang otak dapat digunakan untuk memahami kondisi medis seperti epilepsy, insomnia, dan cedera otak [18].

Aktivitas otak manusia dapat diperoleh dari perekam sinyal listrik di otak yang dikenal sebagai electroencephalography (EEG). Ketika sel saraf diaktifkan secara elektrik atau neurologis, EEG mendeteksi potensi dalam tubuh yang dihasilkan oleh neuron di otak melalui jaringan sel saraf [15].



Gambar 2.2 Contoh sinyal EEG pada penderita ASD [18]

Gambar 2.2 menunjukkan hasil percobaan sinyal EEG yang pernah dilakukan untuk mengetahui fungsionalitas otak maupun gangguan yang terjadi pada sistem neurologis otak anak autis.

2.4 BRAIN COMPUTER INTERFACE (BCI)

Brain-Computer Interface (BCI) adalah sistem teknologi yang memungkinkan manusia untuk berinteraksi dengan komputer dengan menggunakan aktivitas otak mereka sebagai input. BCI mengambil sinyal *electroencephalography* (EEG) dari otak dan dikonversi menjadi perintah untuk komputer [6].

BCI dapat digunakan untuk berbagai tujuan, seperti membantu penderita kelumpuhan atau gangguan bicara untuk berinteraksi dengan lingkungan mereka, atau membantu atlet profesional meningkatkan kinerja mereka dengan memanfaatkan sinyal otak mereka. BCI juga memiliki potensi untuk digunakan dalam bidang militer, kedokteran, dan hiburan [19].

BCI mengandalkan pada prinsip dasar dari neuroplastisitas, yaitu kemampuan otak untuk memodifikasi dan memperluas jaringan saraf seiring dengan latihan dan pengalaman. Dengan melatih otak untuk mengirimkan sinyal yang spesifik dan terfokus, seseorang dapat membangun kemampuan untuk mengendalikan perangkat BCI dengan aktivitas otak mereka [6].

Meskipun BCI memiliki potensi yang sangat besar, teknologi ini masih dalam tahap awal dan memiliki beberapa kendala, seperti kurangnya akurasi dan keandalan, dan masalah teknis yang masih perlu diselesaikan [6]. Namun, dengan terus berkembangnya penelitian dan pengembangan, diharapkan BCI akan menjadi lebih efektif dan dapat digunakan secara luas dalam berbagai aplikasi.

Data EEG yang digunakan pada penelitian ini merupakan data yang berformat BCI2000 dengan ekstensi “.dat” sehingga membutuhkan modul khusus untuk membaca data tersebut menggunakan bahasa pemrograman tertentu [18]. Pada matlab data ini dapat dibaca menggunakan modul EEG Lab namun sulit diimplementasi kedalam system android. Namun pada pemrograman python, sejauh ini hanya ada satu modul yang dapat membaca data ini yakni BCI2kReader yang masih dalam tahap pengembangan lebih lanjut sehingga pemakaian modul ini diperlukan kehati-hatian dalam penggunaanya [20]. Maka dari itu penelitian ini menggunakan pemrograman python yang lebih fleksibel dan dapat diimplementasikan di berbagai platform dan mempermudah automasi seluruh proses seperti *preprocessing* hingga klasifikasi data EEG.

2.5 BCI2KREADER

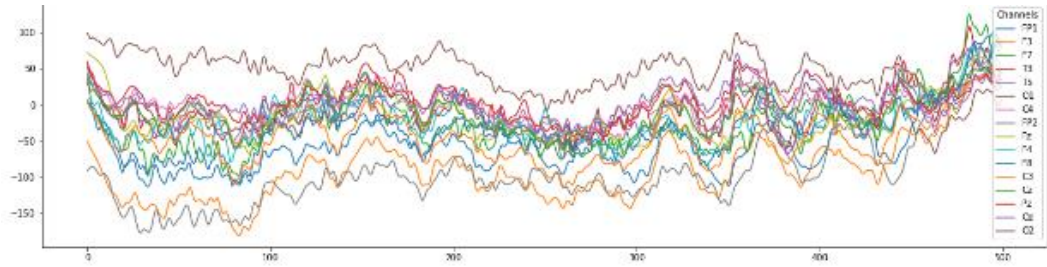
BCI2000Reader adalah sebuah paket Python yang digunakan untuk membaca dan menganalisis data BCI2000. BCI2000 adalah sebuah sistem open-source untuk pengambilan dan pemrosesan data BCI yang digunakan oleh banyak peneliti dalam bidang neuroteknologi. Dengan menggunakan BCI2000Reader, peneliti dapat dengan mudah mengimpor data BCI2000 ke dalam lingkungan Python mereka dan menganalisisnya menggunakan berbagai library analisis data populer, seperti NumPy, Pandas, dan Matplotlib. BCI2000Reader memiliki antarmuka yang mudah digunakan dan memungkinkan peneliti untuk membaca data EEG, stimulus, dan respon dari file BCI2000 [6].

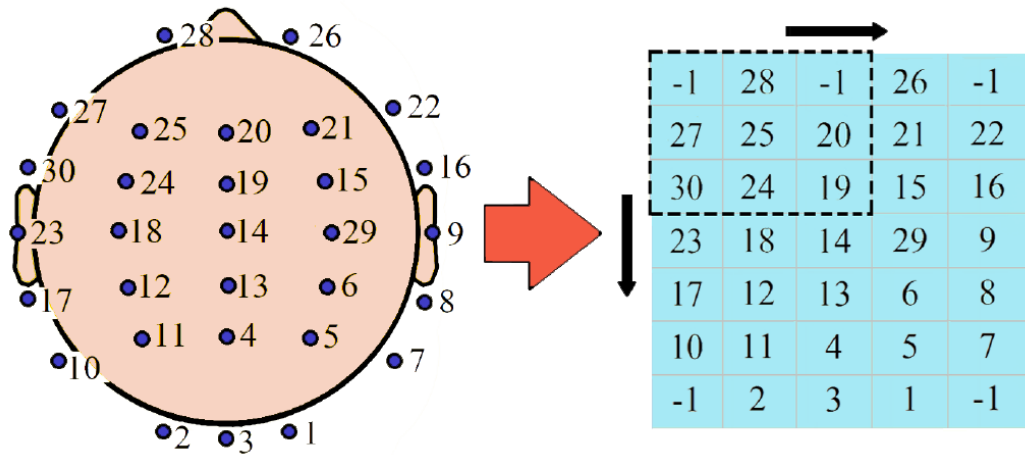
BCI2000Reader juga memiliki beberapa fitur canggih, seperti pemrosesan data online dan offline, dukungan untuk format data yang berbeda, dan dukungan untuk beberapa jenis peralatan BCI. Hal ini membuat BCI2000Reader menjadi pilihan yang baik bagi peneliti yang ingin memproses dan menganalisis data BCI2000 mereka dengan efisien [19].

BCI2000Reader sangat berguna bagi peneliti yang menggunakan data BCI2000 dalam penelitian mereka, karena memungkinkan mereka untuk memproses dan menganalisis data secara cepat dan efisien. Ini juga membantu memastikan konsistensi dan akurasi data, sehingga membuat hasil penelitian lebih valid dan dapat dipercaya. Oleh karena itu, BCI2000Reader merupakan paket yang sangat berguna bagi para peneliti dalam bidang neuroteknologi [19].

Untuk membaca data BCI2000 yang berformat “.dat”. dalam bahasa pemrograman Python, digunakan modul BCI2KReader. Karena modul ini masih dalam pengembangan, fungsi-fungsi yang tersedia pada modul ini masih sedikit yakni hanya sebatas membaca, dan memotong sinyal BCI2000 [20].

Pada gambar 2.3, pembacaan sinyal EEG BCI2000 menggunakan modul Python Yakni BCI2kReader mengembalikan data dalam bentuk *Channel* dan *Data Point* dalam format Numpy matrix [20].





Gambar 2.4 Data EEG yang ditampung dalam *Matrix* 2 Dimensi [16]

Pada Gambar 2.4 diperlihatkan bagaimana data sinyal dari setiap *channel* EEG disimpan dalam bentuk *matrix* atau *array* 2 dimensi. Sumbu x pada *Array* 2 dimensi ini merepresentasikan banyak nya *channel* yang digunakan pada alat EEG dan sumbu y merepresentasikan lamanya perekaman dalam satuan *millisecond* [16]. jika *channel* yang digunakan pada saat perekaman EEG sebanyak 16, maka tinggi matriks (sumbu y) akan sebanyak 16. Sedangkan Panjang sumbu x akan tergantung berapa lama perekaman dilakukan dan setiap *time-step* pada sumbu x adalah 1 *millisecond*.

2.6.2 Discrete Wavelete Transform

Pada penelitian ini, Discrete Wavelet Transform (DWT) digunakan untuk mengekstraksi atau mentapis sinyal ke dalam frekuensi tertentu melalui proses low dan high pass filter. Ekstraksi sinyal berguna untuk memperkecil panjang dan ukuran data [5].

$$W_j(n) = \sum_k h_{k-2n} x_{j-1}(k) \quad (2.1)$$

$$D_j(n) = \sum_k g_{k-2n} x_{j-1}(k) \quad (2.2)$$

di mana:

- $W_j(n)$ dan $D_j(n)$ adalah koefisien DWT untuk aproksimasi (approximation) dan detail pada level j dan lokasi n pada sinyal.
- $x_{j-1}(k)$ adalah subsinyal pada level $j-1$ pada sinyal.
- h_k dan g_k adalah filter-wavelet yang digunakan untuk aproksimasi dan detail, masing-masing.

Proses DWT dilakukan dengan mengulang langkah-langkah DWT untuk setiap subsinyal pada level $j-1$ pada sinyal, di mana setiap subsinyal dipecah menjadi subsinyal aproksimasi dan detail pada level j pada waktu diskrit n [21]. Proses ini dapat diulang-ulang untuk mendapatkan level frekuensi dan resolusi yang lebih tinggi pada subsinyal.

Hasil DWT dapat digunakan untuk berbagai aplikasi, seperti kompresi data, pengolahan sinyal, penghilangan noise, dan analisis pola [25].

2.6.3 EEG to Frequency Spectrum Image (FSI)

Sinyal dalam bentuk spektrogram adalah representasi grafis dari sinyal suara yang menunjukkan bagaimana energi dari sinyal tersebut tersebar dalam domain waktu dan frekuensi. Secara umum, spektrogram adalah representasi grafis dari transformasi Fourier dari sinyal suara yang dihitung pada setiap jendela waktu kecil dalam sinyal tersebut.

Dalam spektrogram, sumbu x biasanya menunjukkan waktu, sedangkan sumbu y menunjukkan frekuensi. Intensitas warna atau kegelapan pada titik-titik di spektrogram menunjukkan seberapa besar energi yang dimiliki oleh sinyal pada frekuensi tertentu dan pada waktu tertentu.

Spektrogram dapat memberikan informasi yang sangat berguna dalam analisis sinyal suara, seperti mengetahui frekuensi tertinggi atau terendah dari sinyal, menemukan keberadaan suara atau pola suara tertentu, serta membantu dalam pengolahan sinyal seperti penghilangan noise atau isolasi frekuensi tertentu dari sinyal.

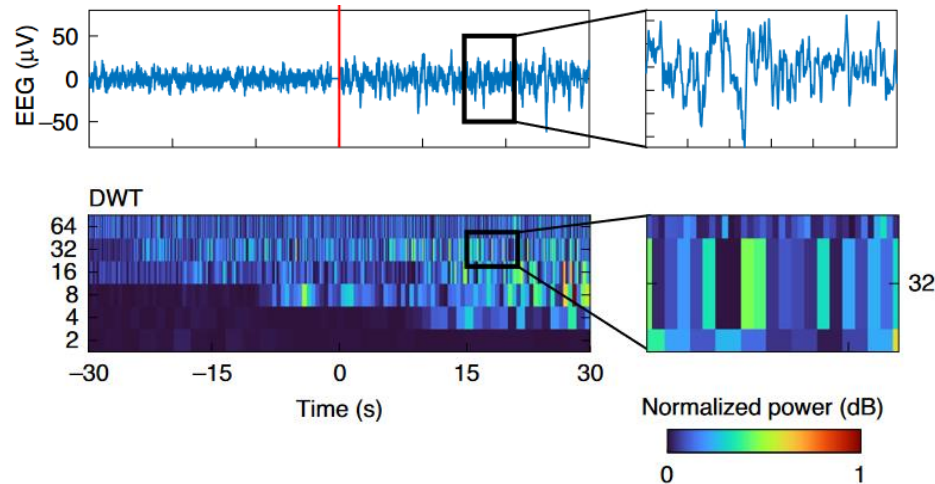
Agar data dapat diklasifikasi menggunakan Convolutional Neural Network (CNN), maka data *matrix* EEG ini akan dirubah kedalam bentuk citra digital. Untuk mengubah *matrix* menjadi citra digital, hal yang pertama dilakukan adalah dengan

memahami dasar-dasar citra digital. Citra digital adalah representasi visual dari data numerik yang disimpan sebagai matrix (juga dikenal sebagai grid) dari nilai-nilai piksel. Piksel adalah singkatan dari "picture element " dan merepresentasikan unit terkecil dari sebuah citra [14], [26].

Analisis sinyal dalam bentuk frequency spectrum image (FSI) atau sering juga disebut spektrogram, dapat memberikan beberapa manfaat, di antaranya:

1. Menampilkan karakteristik frekuensi dari sinyal suara: FSI dapat membantu kita melihat karakteristik frekuensi dari sinyal suara, seperti frekuensi tertinggi atau terendah yang ada pada sinyal tersebut. Dengan demikian, kita dapat menentukan apakah sinyal tersebut cocok untuk keperluan tertentu, misalnya untuk analisis akustik atau untuk kompresi data.
2. Menemukan keberadaan suara atau pola suara tertentu: FSI dapat membantu kita menemukan keberadaan suara atau pola suara tertentu pada sinyal, bahkan ketika sinyal tersebut tidak mudah didengar dengan telinga manusia. Hal ini sangat berguna dalam analisis suara dalam konteks keamanan, seperti pengawasan kualitas produk atau deteksi suara-suara yang mencurigakan.
3. Membantu dalam pengolahan sinyal: FSI dapat membantu dalam pengolahan sinyal, seperti penghilangan noise atau isolasi frekuensi tertentu dari sinyal. Dengan melihat FSI, kita dapat mengetahui bagian frekuensi mana dari sinyal yang memerlukan perbaikan atau penghilangan.
4. Visualisasi sinyal suara yang lebih mudah dimengerti: FSI memberikan representasi grafis dari sinyal suara, yang dapat membantu kita memahami sinyal tersebut dengan lebih mudah. Hal ini sangat berguna dalam pengajaran atau pembelajaran tentang analisis sinyal suara, karena dapat membantu siswa atau pengguna untuk melihat karakteristik sinyal suara dengan lebih jelas.

Dalam keseluruhan, analisis sinyal dalam bentuk FSI memberikan banyak manfaat bagi penggunaan sinyal suara, dan dapat membantu kita dalam memahami, mengelola, dan memanfaatkan sinyal suara dengan lebih efektif.



Gambar 2.5 Contoh merubah sinyal EEG ke bentuk *Frequency Spectrum Image* [26]

Pada gambar 3.4 merupakan visualisasi bagaimana sinyal EEG diubah kedalam bentuk *Frequency Spectrum Image*. Sebelum sinyal dirubah kedalam bentuk *Frequency Spectrum Image*, dilakukan beberapa tahapan seperti pembersihan sinyal dan merubah sinyal menggunakan *Discrete Wavelet Transform* (DWT). Setelah sinyal melewati tahapan tersebut, barulah diubah ke bentuk *Frequency Spectrum Image* [20], [26], [27].

2.7 DEEP LEARNING

Deep learning adalah subbidang pembelajaran mesin yang algoritmanya terinspirasi oleh struktur otak manusia. Struktur ini disebut jaringan saraf tiruan atau *Artificial Neural Networks* (ANN). Pada dasarnya *Deep Learning* adalah jaringan saraf dengan tiga atau lebih lapisan ANN. Itu dapat belajar dan beradaptasi dengan sejumlah besar data dan memecahkan masalah yang sulit dipecahkan dengan algoritma pembelajaran mesin lainnya [5], [7], [24].

Penelitian ini menggunakan metode Deep learning untuk mengklasifikasikan data sinyal EEG dari subjek dan memberikan diagnosis apakah subjek memiliki gejala ASD atau tidak. Penelitian ini juga menggunakan modul *TensorFlow* dalam menggunakan deep learning agar mudah di automasi dan diterapkan ke berbagai platform [28].

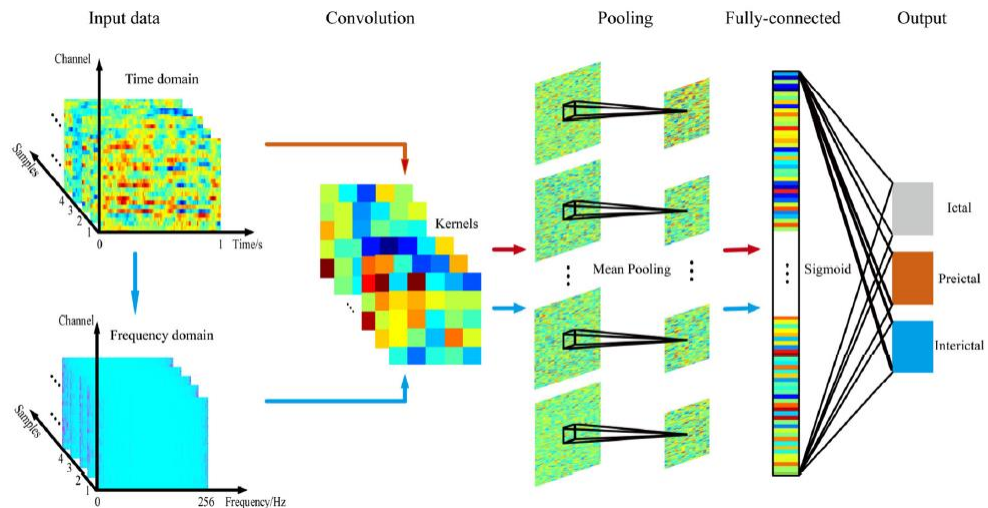
2.7.1 TensorFlow

TensorFlow adalah platform open source end-to-end untuk machine learning. Ini memiliki ekosistem alat, pustaka, dan sumber daya komunitas yang komprehensif dan fleksibel yang memungkinkan peneliti mendorong Machine Learning mutakhir dan pengembang dengan mudah membangun dan menerapkan aplikasi yang didukung *Machine Learning* [29]. TensorFlow menawarkan beberapa tingkat abstraksi sehingga tool ini merupakan tool yang tepat untuk kebutuhan penelitian ini. Membangun dan melatih model dengan menggunakan *Keras* API tingkat tinggi yang memudahkan penggunaan machine learning untuk penelitian agar dapat di automasi, dan diterapkan diberbagai platform [30].

Penelitian ini menggunakan modul tensorflow untuk mendesain, membangun, menjalankan, dan mengimplementasikan model *Deep Learning* untuk mencapai tujuan utama pada penelitian ini yakni mengklasifikasikan sinyal EEG dan memberikan diagnosis dini untuk mengetahui subjek menderita ASD atau tidak secara otomatis.

2.7.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah bagian dari jaringan saraf dalam, yang merupakan jaringan saraf tiruan yang banyak digunakan dalam pengenalan dan pemrosesan gambar. Algoritma ini dibuat terutama untuk menangani data piksel dan gambar visual. Neuron dari algoritma *Convolutional Neural Network* dimaksudkan untuk berfungsi mirip dengan *lobus frontal*, terutama daerah korteks visual, pada otak manusia dan hewan. Korteks visual adalah area otak yang bertanggung jawab untuk memproses informasi dalam bentuk *input* visual. Inilah yang membedakan CNN dengan teknik *neural network* lainnya dalam pengolahan citra. Contohnya termasuk pengenalan wajah, kategorisasi gambar, dan aplikasi CNN lainnya dalam visi komputer. Seperti jaringan saraf sederhana, CNN berisi parameter yang dapat dipelajari, seperti jaringan saraf, yaitu bobot, *bias*, dan lain-lain [24], [31].



Gambar 2.6 Struktur Convolutional Neural Network (CNN) [23]

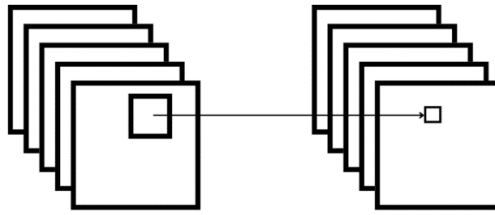
Berikut adalah urutan proses dari Gambar 2.6 yaitu:

1. Lapisan *Input*

Lapisan *input* CNN harus berisi data gambar. Data gambar, seperti yang ditetapkan sebelumnya, diwakili oleh matriks tiga dimensi. Itu harus dibentuk kembali menjadi satu kolom [32].

2. Lapisan Konvolusi

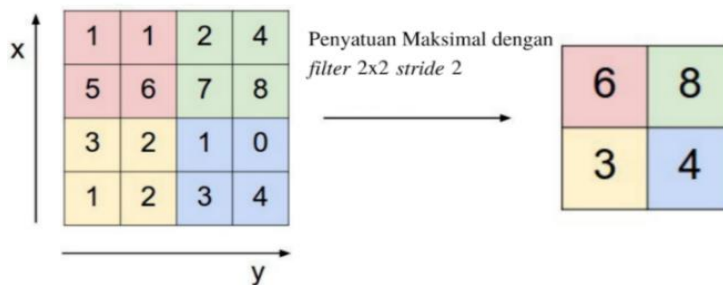
Karena mengekstraksi properti gambar, lapisan konvo juga disebut lapisan ekstraktor fitur. Untuk memulai, bagian dari gambar ditautkan ke lapisan konvoi, yang melakukan operasi konvolusi dan menghitung perkalian titik antara bidang reseptif dan filter. Metode ini menghasilkan bilangan bulat tunggal dari volume keluaran. Filter selanjutnya diteruskan ke bidang reseptif berikutnya dari gambar masukan yang identik dan prosesnya diulang. Pendekatan yang sama diulang sampai seluruh gambar telah diproses. Luaran akan dikirim ke tingkat berikutnya [32].



Gambar 2.7 Lapisan konvolusi [24], [32].

3. Lapisan Pooling

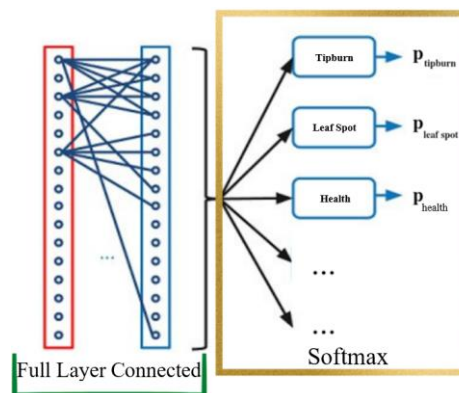
Selama konvolusi, lapisan Pooling digunakan untuk membantu mengurangi *volume* spasial pada gambar *input*.



Gambar 2.8 Lapisan *Pooling* [32].

4. *Softmax*

Lapisan terakhir CNN adalah *softmax*, umumnya dikenal sebagai lapisan logistik. Itu ada di bagian paling bawah dari lapisan yang terhubung sepenuhnya. *Softmax* digunakan untuk multi-klasifikasi, sedangkan logistik digunakan untuk klasifikasi biner [24], [32].



Gambar 2.9 lapisan *softmax* [24].

5. Lapisan *output*

Lapisan *output* adalah semacam regularisasi yang berupaya meminimalkan *overfitting* dengan meningkatkan akurasi pengujian, terkadang dengan mengorbankan akurasi pelatihan. Lapisan *output* dalam arsitektur jaringan kami secara acak mengirimkan *input* dari lapisan sebelumnya ke lapisan berikutnya dengan probabilitas “p” untuk setiap *mini-batch* di *set* pelatihan kami [30]. Lapisan *output* memanfaatkan fungsi kerugian seperti entropi silang kategori untuk menghitung kesalahan prediksi. Setelah *forward pass*, *backpropagation* memperbarui bobot dan bias untuk mengurangi kesalahan dan kerugian.

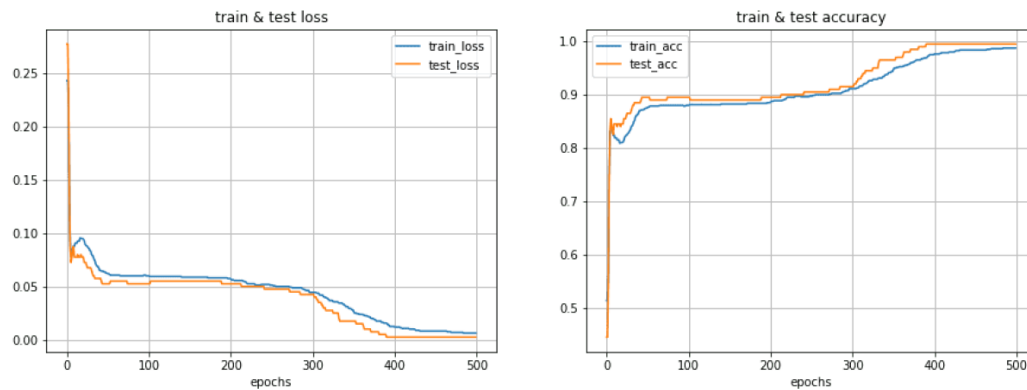
2.7.3 Dropout

Dropout adalah pendekatan regularisasi jaringan saraf yang melibatkan pemilihan neuron secara acak. Putus sekolah mengacu pada penghapusan neuron dari lapisan tersembunyi dan terlihat jaringan [11]. Menghapus neuron melibatkan penghapusan sementara dari jaringan yang ada. Neuron yang akan dihapus akan dipilih secara acak. Setiap neuron akan diberi probabilitas antara 0 dan 1. Ini mencegah *overfitting* dan mempercepat proses pelatihan [11], [32].

2.7.4 Loss dan Akurasi

Loss atau kerugian menunjukkan tingkat kerugian pada klasifikasi citra, dimana semakin rendah loss maka model yang dibangun semakin baik. Namun hal tersebut tidak berlaku jika model telah mengalami *overfitting*. Nilai loss didapatkan dari perhitungan pada training dan validation dan hasilnya menginterpretasikan seberapa bagus model ketika melakukan training dan validation [31].

Untuk nilai loss tidak dinyatakan dalam persen seperti accuracy, hal tersebut karena berasal dari penjumlahan error dari setiap contoh epoch atau validation sets. Pada dasarnya tujuan model pembelajaran digunakan untuk mengurangi nilai loss sesuai dengan parameter model dengan mengganti nilai weight vector melalui metode optimasi yang berbeda seperti *backpropagation* pada Neural Network. Jika disimpulkan maka nilai loss menggambarkan seberapa bagus atau buruknya suatu model setelah dilakukan iterasi dari optimasi [31], [32].

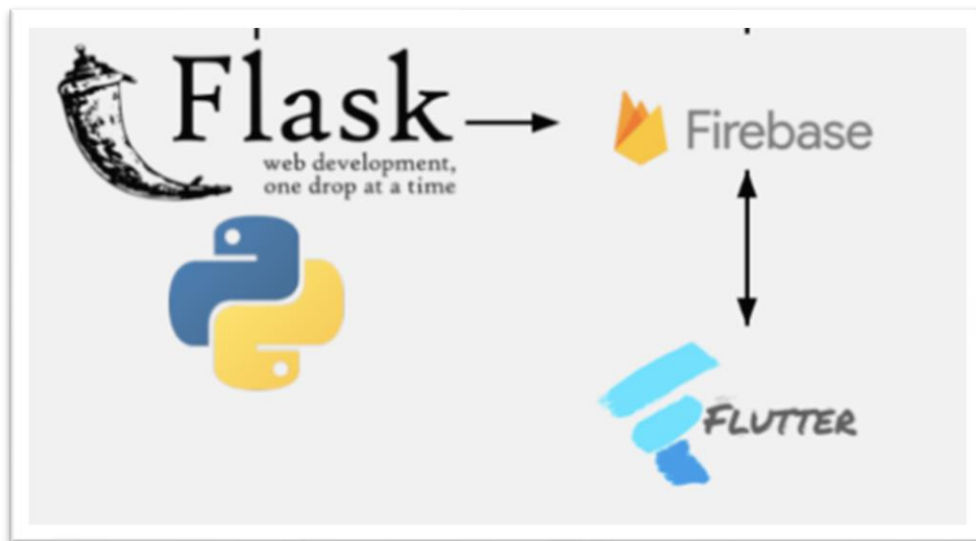


Gambar 2.10 Tampilan *loss* dan *accuracy* [32]

Pada Gambar 2.9 menunjukkan grafik hubungan antara *loss*, *accuracy* dan *epoch*. Nilai akurasi dapat diperoleh setelah mempelajari parameter model dan pelatihan lebih lanjut (tidak ada pelatihan yang dilakukan). Sampel uji kemudian dimasukkan ke dalam model, jumlah kesalahan dalam model (kehilangan lingkaran nol) dicatat, dan kemudian dibandingkan dengan target yang sebenarnya. Kemudian tingkat kesalahan klasifikasi dihitung. Misalnya, jika jumlah sampel uji 1000 dan model yang dihasilkan memberikan nilai benar 952, maka nilai akurasi yang diperoleh adalah 95,2%.

2.8 ARSITEKTUR SISTEM APLIKASI MOBILE

Flutter adalah toolkit UI yang dibuat oleh Google. Ini digunakan untuk membangun aplikasi seluler lintas platform untuk Android dan iOS dan aplikasi desktop untuk Windows, Mac, dan Linux. UI yang dibuat dengan Flutter selalu bergantung pada tumpukan teknologi backend untuk fungsionalitas inti seperti proses klasifikasi sinyal menggunakan *deep learning*. Salah satu teknologi backend tersebut adalah framework Flask dan firebase [28]. Flask adalah kerangka kerja web Python untuk membangun aplikasi web [12].



Gambar 2. 11 Contoh arsitektur aplikasi mobile menggunakan Flutter, firebase, Flask [28].

Pada gambar 2.10 diperlihatkan arsitektur aplikasi *mobile* yang menggunakan *framework* Flutter dengan fungsi inti Flask dan Firebase [12], [13], [33]. Flask digunakan untuk menjalankan fungsi-fungsi inti dari aplikasi mobile tersebut dan digunakan *Cloud service* Firebase untuk menyimpan database.

2.8.1 Layanan Firebase

Firebase adalah layanan milik google yang memiliki beberapa fitur yang dapat membantu developer untuk pengembangan fungsi inti dari sebuah aplikasi. Dengan adanya layanan ini, pengembangan aplikasi dapat lebih difokuskan pada bagian tampilan [33].

Firebase memiliki beberapa layanan yang sangat berguna bagi pengembangan aplikasi *mobile*. Layanan pada firebase seperti *authentication* [16] membuat aplikasi menjadi lebih aman bagi pengguna. Tidak hanya itu, firebase juga menyediakan *Cloud Firestore* dan *Cloud Storage* untuk menyimpan *database* dan juga memungkinkan pengguna untuk meng-*upload* semua jenis *file* kedalam penyimpanan berbasis cloud tersebut [16]. Dengan ini, aplikasi mobile yang diintegrasikan dengan firebase akan tetap berjalan dengan kinerja tinggi tanpa harus memakan penyimpanan local tambahan.

2.8.2 Python Flask

Flask merupakan web framework dari bahasa pemrograman python. Flask digunakan sebagai inti dan kerangka kerja aplikasi. Dengan menggunakan Flask dan bahasa python, pengembangan dapat membuat API yang terstruktur dengan mudah dan tidak memerlukan suatu alat atau Pustaka tertentu di dalam penggunaannya [12].

Pada penelitian ini, python flask digunakan untuk preprocessing data EEG dan menjalankan model deep learning terlatih untuk klasifikasi sinyal EEG penderita ASD dan normal. Flask juga dapat mengembalikan data berupa restAPI yang dapat diakses berbagai platform. Pada penelitian ini, platform yang digunakan adalah android yang dibangun dengan framework flutter [13].

2.8.3 Flutter

Flutter merupakan *framework* open source yang diciptakan oleh google untuk membangun aplikasi yang indah, berjalan dengan *native* dan *multi-platform* seperti Android, iOS, web, maupun desktop hanya dengan satu *codebase* [13]. Flutter memiliki dua komponen yakni Software development kit (SDK) yang merupakan tools yang berguna untuk menjalankan aplikasi berbagai platform, dan Framework UI yang merupakan komponen tampilan seperti tombol, text, dan lainnya sesuai kustomisasi kebutuhan. Kelebihan Flutter lainnya adalah memiliki kustomisasi yang tidak terlalu rumit sehingga dapat menghemat waktu pengembangan [13].

Pada penelitian ini, flutter akan diintegrasikan dengan *Cloud Service Firebase* [33] dan digunakan untuk membangun sebuah aplikasi mobile dengan kinerja tinggi yang memungkinkan pengguna untuk menginput data file EEG dengan mudah dan langsung mendapat hasil diagnosis secara cepat.

2.9 UJI PERFORMA SISTEM

Pada penelitian ini akan dilakukan dua jenis uji dan evaluasi performa yakni *Confusion Matrix* untuk evaluasi performa model Deep learning yang dilatih

sebelumnya dan performa Mobile apps yang meliputi usability test, performa komunikasi data, dan lainnya.

2.9.1 Confusion Matrix

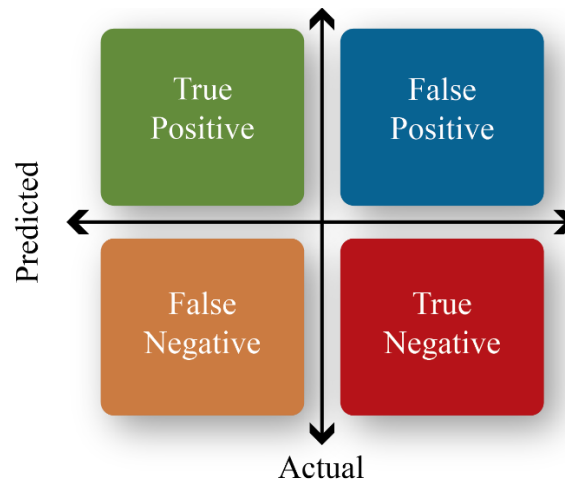
Confusion Matrix adalah matriks yang menampilkan data Klasifikasi dan prediksi. Matriks konfusi berukuran $n \times n$. dimana n adalah jumlah kelas yang berbeda [22]. Untuk mengevaluasi kinerja metode yang diusulkan, Validasi silang digunakan [34]. Kinerja klasifikasi telah dievaluasi sebagai Parameter seperti benar Positif (TP), benar Negatif (TN), positif palsu (FP), pilepsy palsu (FN). Sensitivitas, spesifisitas, skor F1, dan akurasi keseluruhan menggunakan rumus :

$$\text{Sensitivitas} = \frac{TP}{TP+FN} \times 100 \quad (2.3)$$

$$\text{Spesitivitas} = \frac{TN}{TN+FP} \times 100 \quad (2.4)$$

$$\text{Akurasi} = TP + \frac{TN}{TP+FP+TN+FN} \times 100 \quad (2.5)$$

$$\text{F1 Score} = 2 \frac{TP}{2 TP+FP+FN} \quad (2.6)$$



Gambar 2.12 *Confusion Matrix* [31]

Keterangan rumus (Pers 2.3) - (Pers 2.6) dan Gambar 2.12 sebagai berikut:

1. True Positive (TP) adalah ketika model memprediksi hasil positif dan kenyataannya memang positif. Dalam kata lain, model berhasil mengidentifikasi suatu kejadian sebagai benar terjadi.
2. True Negative (TN) adalah ketika model memprediksi hasil negatif dan kenyataannya memang negatif. Dalam kata lain, model berhasil mengidentifikasi suatu kejadian sebagai tidak terjadi.
3. False Positive (FP) adalah ketika model memprediksi hasil positif, tetapi kenyataannya negatif. Dalam kata lain, model salah mengidentifikasi suatu kejadian sebagai terjadi, padahal sebenarnya tidak terjadi.
4. False Negative (FN) adalah ketika model memprediksi hasil negatif, tetapi kenyataannya positif. Dalam kata lain, model salah mengidentifikasi suatu kejadian sebagai tidak terjadi, padahal sebenarnya terjadi.

2.9.2 Uji Perfroma Mobile Apps

Tahap-tahap pengujian performa aplikasi mobile yang memiliki 26pilep Deep Learning meliputi beberapa 26pileps berikut:

1. Verifikasi Data Latih: Langkah ini melibatkan memastikan bahwa data latih yang digunakan untuk membentuk model Deep Learning adalah benar dan valid [35].
2. Uji Validitas Model: Langkah ini melibatkan menguji model Deep Learning untuk memastikan bahwa model tersebut memiliki akurasi dan kinerja yang baik. Ini bisa dilakukan dengan membandingkan hasil prediksi model dengan data latih dan menghitung metrik seperti presisi, recall, dan F1 score [32].
3. Uji Performa: Langkah ini melibatkan menguji performa aplikasi mobile yang menggunakan model Deep Learning dalam kondisi normal dan ekstrem. Ini termasuk melakukan pengujian pada berbagai ukuran 26pile, resolusi, dan jaringan seluler [36].
4. Uji Keamanan: Langkah ini melibatkan menguji keamanan aplikasi mobile yang menggunakan model Deep Learning, termasuk memastikan bahwa data pribadi pengguna tidak dapat dicuri atau dicuri oleh pihak ketiga [36], [37].

5. Uji Scalability: Langkah ini melibatkan menguji skalabilitas aplikasi mobile yang menggunakan model Deep Learning, termasuk memastikan bahwa aplikasi tersebut dapat berfungsi dengan baik pada skala besar dan dapat diperluas untuk mengatasi peningkatan beban [36].
6. Uji Kompatibilitas: Langkah ini melibatkan menguji kompatibilitas aplikasi mobile yang menggunakan model Deep Learning dengan perangkat dan 27pilep operasi yang berbeda [36], [38].
7. Uji Usabilitas: Langkah ini melibatkan menguji usabilitas aplikasi mobile yang menggunakan model Deep Learning, termasuk memastikan bahwa aplikasi mudah digunakan dan memiliki antarmuka yang intuitif [36].

Semua 27pileps-langkah ini penting untuk dilakukan untuk memastikan bahwa aplikasi mobile yang menggunakan model Deep Learning bekerja dengan baik dan memenuhi standar kualitas yang diharapkan.

2.10 PENELITIAN TERKAIT

B. Studi literatur terkait IoT klasifikasi sinyal EEG

Tabel 2.1 Studi literatur terkait Aplikasi klasifikasi sinyal EEG

No	Referensi	Latar Belakang	Metode	Hasil
1	Ref [3]	Tujuan : <ul style="list-style-type: none"> • mengembangkan aplikasi smartphone berbasis Android untuk monitoring deteksi tidur berdasarkan hasil klasifikasi sinyal EEG. 	<ul style="list-style-type: none"> • SVM, • ANN, • KNN 	<ul style="list-style-type: none"> • Aplikasi berbasis android yang dapat mengidentifikasi 27pilepsy melalui sinyal EEG yang datanya diambil langsung dari alat penangkap sinyal EEG
2	Ref [12]	<ul style="list-style-type: none"> • Prototype untuk melakukan Mind 	<ul style="list-style-type: none"> • Cloud Computing 	<ul style="list-style-type: none"> • Aplikasi yang dapat memproses sisnyal

		Wave EEG dan aplikasi Android yang berkomunikasi melalui Bluetooth.		EEG untuk berbagai aplikasi seperti BCI, Neurofeedback, prediksi serangan 28pilepsy, dll.
--	--	---	--	---

B. Studi literatur terkait klasifikasi sinyal EEG

Tabel 2.2 Studi literatur terkait klasifikasi sinyal EEG

1	Ref [9]	<p>Tujuan :</p> <ul style="list-style-type: none"> • menyajikan perbandingan klasifikasi sinyal Electroencephal EEG dan BCI • mengekstrak pola sinyal EEG berkualitas buruk dan secara bersamaan dengan adanya suara artefak 	<ul style="list-style-type: none"> • SVM • KNN • MLP-ANN • LR 	<ul style="list-style-type: none"> • LR mendapat akurasi klasifikasi EEG sebesar 73,03% • SVM mendapat akurasi sebesar 68,97%.
2	Ref [10]	<ul style="list-style-type: none"> • Mengatasi masalah implementasi 28pilepsy elisitasi emosi, ekstraksi fitur, • klasifikasi. Rekaman EEG dari database eENTERFACE tujuan simulasi. 	<ul style="list-style-type: none"> • Nearest Neighbor classifier (kNN) 	<ul style="list-style-type: none"> • tingkat akurasi klasifikasi 90,77%

3	Ref [4]	<ul style="list-style-type: none"> • menggunakan sinyal saraf untuk mengontrol aplikasi ponsel 	<ul style="list-style-type: none"> • IoT 	<ul style="list-style-type: none"> • headset elektroensefalografi nirkabel (EEG) untuk penggerak aplikasi ponsel
4	Ref [11]	<p>Tujuan :</p> <ul style="list-style-type: none"> • memperoleh scalogram sinyal EEG • mengekstrak fitur dari 29 pilepsy29 dari sinyal EEG subjek normal dan ASD. 	<ul style="list-style-type: none"> • CNN 	<ul style="list-style-type: none"> • CNN GoogLeNet, AlexNet, MobileNet dan SqueezeNet mencapai akurasi validasi 75%, 75,84%, 79,45%, dan 82,98%
5	Ref [5]	<p>Tujuan :</p> <ul style="list-style-type: none"> • Diagnosis kejang 29pilepsy sinyal Electroencephalgraphy (EEG). 	<ul style="list-style-type: none"> • Deep Learning Convolutional Neural Network (CNN) 	<ul style="list-style-type: none"> • Akurasi klasifikasi bernilai 72.49%, • Nilai loss bernilai 0.576.
6	Ref [6]	<p>Tujuan :</p> <ul style="list-style-type: none"> • mengklasifikasikan berbagai pola sampel rekaman EEG yaitu autis, 29pilepsy dan normal menggunakan algoritma deep learning. • Untuk menghilangkan artifak dari dataset EEG 	<ul style="list-style-type: none"> • Preprocessing: Independent Component Analysis (ICA) • Klasifikasi: Deep Convolutional Neural Network (DCNN) 	<ul style="list-style-type: none"> • Akurasi klasifikasi yang dibuat oleh model DCNN yang didapatkan mencapai 80% untuk validasi kumpulan data pelatihan.

7	Ref [8]	<p>Tujuan:</p> <ul style="list-style-type: none"> • mengembangkan kerangka kerja diagnostik yang efisien sinyal EEG untuk mengidentifikasi ASD secara otomatis. 	<ul style="list-style-type: none"> • Deep Learning Convolutional Neural Network (CNN) • Machine Learning (ML) Classifier Model 	<ul style="list-style-type: none"> • model berbasis Deep Learning CNN mencapai akurasi yang lebih tinggi (99,15%) • dengan model berbasis ML (95,25%)
---	------------	--	--	---

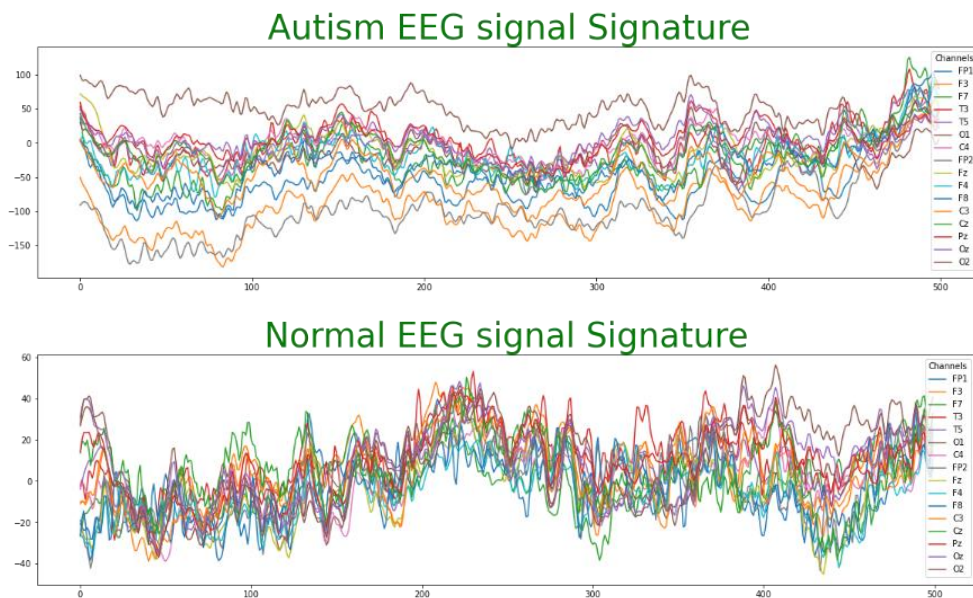
BAB 3

METODELOGI DAN JADWAL PENELITIAN

3.1 METODE PENELITIAN

3.1.1 Bahan Penelitian

Penelitian ini menggunakan data sinyal EEG yang didapatkan dari penelitian yang dilakukan pada Universitas King Abdulaziz (KAU), Jeddah Arab Saudi. KAU), Jeddah, Arab Saudi. Dataset yang didapat sebanyak 17 subjek yang terdiri dari 4 sinyal EEG normal dan 13 sinyal EEG untuk penderita ASD dengan extensi *dat*. Data ini diperoleh dari website malhaddad <https://malhaddad.kau.edu.sa/Pages-BCI-Datasets-En.aspx> [18]. Pada penelitian ini data dibagi dan dialokasikan ke dua folder yang masing-masing berisi sinyal EEG normal dan penderita ASD serta menghapus data-data yang tidak layak untuk diproses. Untuk memvisualikan data EEG pada penelitian ini dapat digunakan dua metode yakni menggunakan python dan menggunakan BCI2000 *viewer*, berikut adalah visualisasi data EEG:



Gambar 3.1 Grafik sinyal EEG menggunakan python

Gambar 3.1 merupakan gambar visualisasi sinyal EEG menggunakan *library* python BCI2k *reader*. Pada gambar 3.1 diperlihatkan dua sinyal yakni pada gambar atas merupakan sinyal EEG penderita ASD dan gambar kedua yang

merupakan sinyal EEG normal. Pada sinyal ini terdapat 16 kanal (Fp1, Fp2, F7, F3, Fz, F4, F8, T3, C3, Cz, C4, T5, Pz, O1, Oz dan O2) yang masing-masing kanal ditandai dengan warna yang berbeda (legenda pada grafik). Sinyal ini nantinya akan diproses dalam beberapa tahap seperti membaca sinyal kedalam program, mengolah sinyal, dan pengklasifikasian sinyal menggunakan *Convolutional Neural Network*.

3.1.2 Alat Penelitian

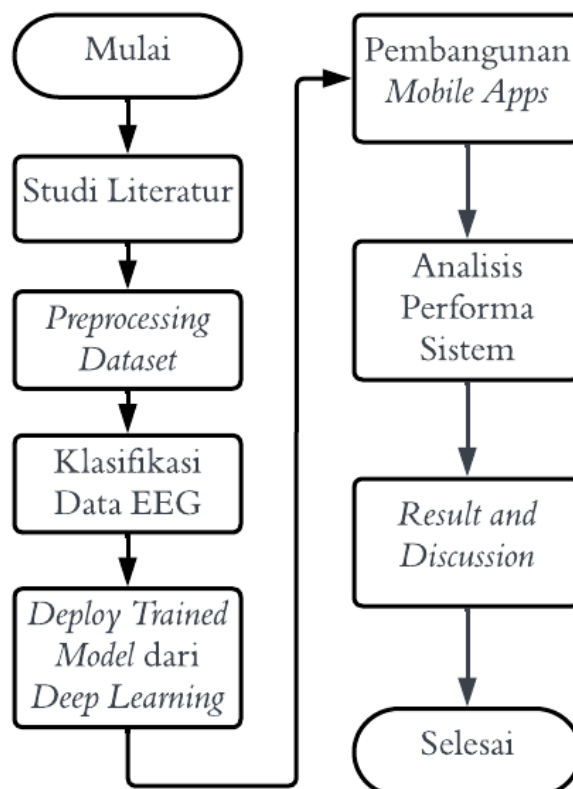
Tabel 3.1 Alat Penelitian

No.	Alat	Spesifikasi	Jumlah	Justifikasi
1	Laptop	<ul style="list-style-type: none"> • RAM 12 GB • Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz (8 CPUs), ~1.8GHz • NVIDIA GeForce MX230 	1	Perangkat Eksekusi program
2	Operating System	Windows 11 Home Single Language 64-bit (10.0, Build 22000) (22000.co_release.210604-1628)	1	Sistem operasi untuk pengembangan penelitian
3	Deep Learning IDE	Jupyter Notebook	1	Platform lingkungan pengembangan <i>deep learning</i>
4	Library Deep Learning	TensorFlow	1	Framework pengembangan deep learning
5	Python Preprocessing Library	<ul style="list-style-type: none"> • BCI2K • Numpy • Pandas • Matplotlib 	1	Modul pembaca data EEG dan <i>Preprocessing</i>

		<ul style="list-style-type: none"> • Flask 		
6	Cloud Service	<ul style="list-style-type: none"> • Fire Base • Flask 	1	Database API dll
7	Android SDK	Flutter	1	Pengembangan aplikasi <i>mobile</i>
8	Smartphone	Samsung Galaxy M22	1	Emulator output pengembangan aplikasi <i>mobile</i>

3.1.3 Alur Penelitian

Bagian ini menjelaskan tahapan-tahapan dalam pembuatan aplikasi diagnosis ASD menggunakan sinyal EEG. Berikut adalah alur tahapan pada penelitian:



Gambar 3.2 Alur Penelitian

A. Studi Literatur

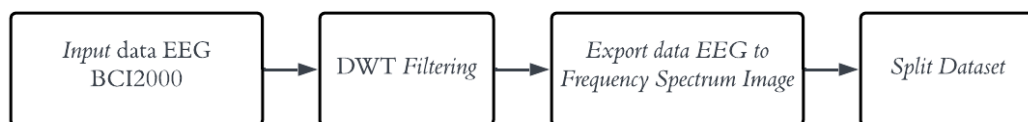
Tahap selanjutnya adalah studi literatur. Pada tahapan ini dilakukan pencarian referensi yang dibutuhkan untuk pada penelitian ini. Tahap studi literatur ini bersifat teori yang mencakup teori mengenai sifat Autism Spectrum Disorder (ASD), sifat gelombang otak, struktur data sinyal EEG, hingga mempelajari teori-teori yang dibutuhkan untuk membangun model yang dapat mengestimasi objek melalui sinyal EEG.

B. Sumber Dataset

Dataset sinyal EEG berasal dari penelitian pada universitas King Abdulaziz (KAU), Jeddah, Arab Saudi. Dataset yang didapat sebanyak 17 subjek yang terdiri dari 4 sinyal EEG normal dan 13 sinyal EEG untuk penderita ASD dengan ekstensi *dat*. Data ini diperoleh dari website malhaddad <https://malhaddad.kau.edu.sa/Pages-BCI-Datasets-En.aspx>. Pada penelitian ini data dibagi dan dialokasikan ke dua folder yang masing-masing berisi sinyal EEG normal dan penderita ASD serta menghapus data-data yang tidak layak untuk diproses.

Pada pemrograman Python, sinyal ini hanya bisa diinput menggunakan package BCI2kReader yang nantinya datanya akan terdiri dari channel dan datapoint. Nilai channel dan datapoint yang akan diproses sebelum melakukan training data pada model deep learning.

C. *Preprocessing Dataset*

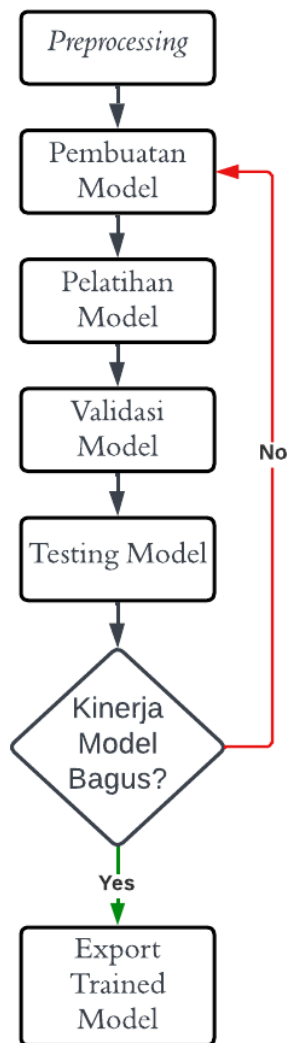


Gambar 3.3 Proses Preprocessing pada sinyal EEG

Pada tahapan ini dilakukan dengan membaca dataset satu persatu yang nantinya datanya akan terdiri dari channel dan datapoint. Proses akan mengambil nilai channel dan datapoint dari sinyal EEG dilakukan dengan menggunakan library Numpy dan membentuk matriks dua dimensi dan data sinyal EEG diubah kedalam

bentuk citra digital. Citra digital EEG akan dipisahkan menjadi beberapa bagian yakni data latih yang terdiri dari 80% dari keseluruhan dataset dan data validasi yang berisi 20% data dari keseluruhan dataset. Data latih dan validasi ini dibagi lagi sehingga terdapat empat bagian yakni data latih, label latih, data validasi, serta label validasi dan data siap untuk dilatih oleh model *Neural Network*.

D. Klasifikasi sinyal EEG menggunakan Convolutional Neural Network (CNN)



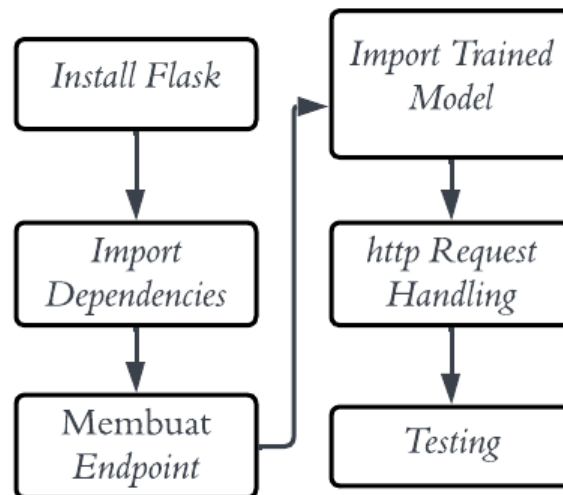
Gambar 3.4 Klasifikasi Data EEG menggunakan CNN

Pengklasifikasian sinyal EEG menggunakan *Convolutional Neural Network* (CNN) dilakukan dengan beberapa tahap yakni:

1. **Preprocessing Data:** Data yang digunakan untuk pelatihan model deep learning biasanya perlu dipreproses agar lebih mudah dikelola dan digunakan. Ini bisa termasuk konversi format data, normalisasi nilai, augmentasi data, dan lain-lain. Tujuan dari preprocessing data adalah untuk meningkatkan kualitas data dan akurasi model.
2. **Pembuatan Model:** Selanjutnya, model CNN dibuat dengan menggunakan library atau framework deep learning yang ada. Tahapan ini melibatkan penentuan jumlah layer, jumlah filter pada tiap layer, fungsi aktivasi, dan lain sebagainya.
3. **Pelatihan Model:** Setelah pembuatan model, model dilatih dengan menggunakan data pelatihan. Pada tahap ini, model mempelajari pola-pola dalam data dan menghasilkan model yang dapat digunakan untuk melakukan prediksi. Pelatihan model dilakukan dengan menggunakan metode backpropagation dan optimisasi pada loss function.
4. **Validasi Model:** Setelah pelatihan model, tahap validasi dilakukan untuk mengevaluasi performa model dengan menggunakan data validasi. Performa model dinilai berdasarkan metrik-metrik seperti *akurasi*, *sensitivity*, dan *precision*. Jika performa model belum optimal, maka model dapat diulang kembali pelatihannya dengan menggunakan konfigurasi yang berbeda.
5. **Testing Model:** Setelah model berhasil dilatih, tahap terakhir adalah pengujian model pada data uji. Pada tahap ini, model diuji untuk melihat apakah dapat mengklasifikasikan data uji dengan akurat atau tidak.
6. **Ekspor Model:** Jika model telah dilatih dan diuji dengan baik, model dapat diekspor ke dalam format yang dapat digunakan oleh aplikasi. Salah satu format yang umum digunakan adalah format berkas protobuf, yang bisa digunakan oleh berbagai jenis aplikasi.

Dalam kesimpulannya, untuk melatih sebuah model CNN, perlu melakukan beberapa tahap, yaitu preprocessing data, pembuatan model, pelatihan model, validasi model, pengujian model, dan ekspor model. Tahap-tahap ini penting untuk mendapatkan model yang akurat dan dapat digunakan pada aplikasi yang diinginkan.

E. Deploy Trained Model



Gambar 3.5 Tahapan *Deploy Trained Model*

Deploy model yang telah dilatih menggunakan *framework deep learning* seperti TensorFlow adalah tahapan penting dalam pengembangan aplikasi berbasis AI. Salah satu cara untuk melakukan *deploy model* tersebut adalah dengan menggunakan Flask, yaitu *framework* web Python yang mudah digunakan dan fleksibel [12].

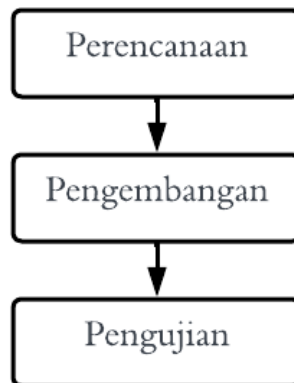
Pada Penelitian ini Flask memungkinkan untuk membuat API sederhana yang dapat mengintegrasikan model ke dalam aplikasi web atau aplikasi *mobile*. Dalam penggunaannya, Flask memberikan kemudahan dalam mengelola permintaan HTTP dari klien dan memberikan respons yang sesuai berdasarkan prediksi yang dihasilkan oleh model [12], [28]. Dengan demikian, penggunaan Flask sangat cocok untuk *deploy model* kecil atau menengah yang memerlukan skalabilitas yang lebih ringan dan cepat.

Merujuk pada Gambar 3.6, berikut ini adalah tahapan-tahapan yang harus dilakukan untuk *deploy trained model* menggunakan Flask:

1. Install Flask: Pertama-tama, pastikan Anda telah menginstal Flask pada komputer Anda. Anda dapat menginstalnya dengan menjalankan perintah "pip install Flask" pada terminal atau command prompt.
2. Import dependencies: Setelah menginstal Flask, Anda perlu mengimpor semua dependensi yang diperlukan untuk menjalankan aplikasi Flask. Ini termasuk numpy, pandas, dan lain-lain.
3. Membuat endpoint: Setelah semua dependensi diimpor, buat endpoint di aplikasi Flask. Endpoint adalah URL tempat model akan dijalankan. Misalnya, jika Anda ingin membuat model yang dapat mengklasifikasikan gambar, Anda dapat membuat endpoint seperti "/predict".
4. Mengimpor model: Setelah endpoint dibuat, Anda perlu mengimpor model yang telah dilatih sebelumnya ke dalam aplikasi Flask.
5. Menangani permintaan HTTP: Setiap kali permintaan HTTP diterima oleh endpoint, aplikasi Flask harus menangani permintaan tersebut dan memberikan respons yang sesuai. Dalam kasus ini, respons adalah prediksi yang dihasilkan oleh model.
6. Menjalankan aplikasi: Setelah semuanya siap, jalankan aplikasi Flask dengan menjalankan perintah "flask run" pada terminal atau command prompt.
7. *Testing*: Setelah aplikasi dijalankan, Anda dapat menguji model dengan menggunakan data yang sama dengan data yang digunakan pada saat pelatihan model. Pastikan model dapat memberikan hasil yang akurat.

Setelah model berhasil di-deploy menggunakan Flask, API akan diintegrasikan dengan aplikasi mobile untuk kemudahan dalam mendiagnosa ASD menggunakan data EEG pada penelitian ini.

F. Membangun *Mobile Apps*



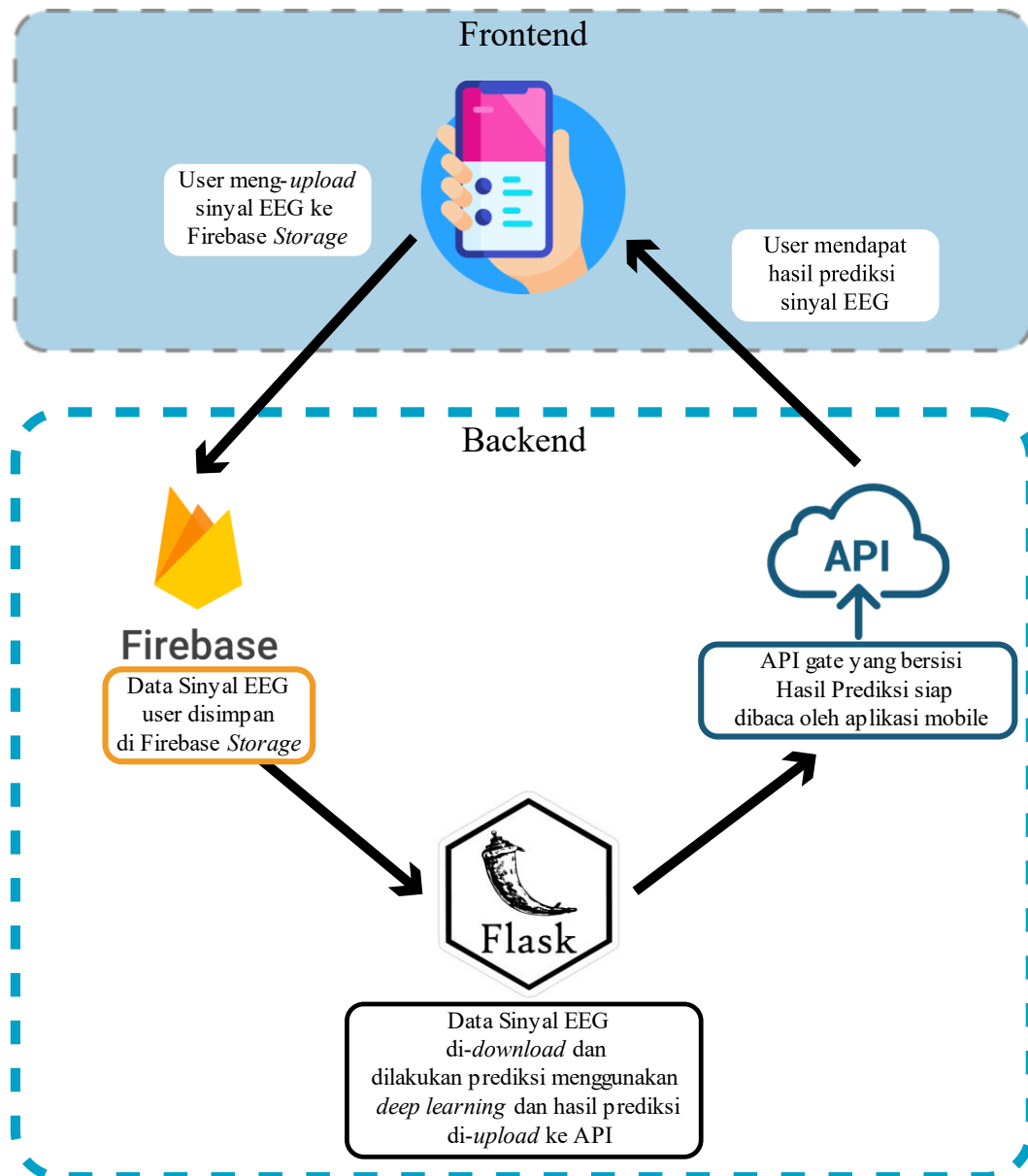
Gambar 3.6 Tahapan Pembangunan *Mobile Apps*

Pada tahap Pembangunan *Mobile Apps* ini dilakukan dengan tiga tahap yang diawali dengan tahap perencanaan yang mencakup desain *use case*, arsitektur umum, dan desain UI/UX. Selanjutnya dilakukan tahap pengembangan seperti slicing, pembuatan fitur, dan integrasi dengan cloud firebase [13], [33]. Tahap yang terakhir adalah pengujian yang mencakup skenario pengujian dan *usability test* [36].

a. Tahap Perencanaan

Tahap ini diawali dengan menganalisis persyaratan teknis dengan melibatkan analisis persyaratan teknis seperti arsitektur aplikasi, sistem operasi, bahasa pemrograman, platform, dan fitur yang dibutuhkan. Selanjutnya melakukan tahapan desain seperti:

1. Arsitektur umum: Tahapan desain yang paling awal adalah dengan membuat desain arsitektur umum yang merepresentasikan struktur umum dari aplikasi yang akan dibangun. Berikut adalah arsitektur umum yang merepresentasikan aplikasi mobile pada penelitian ini:



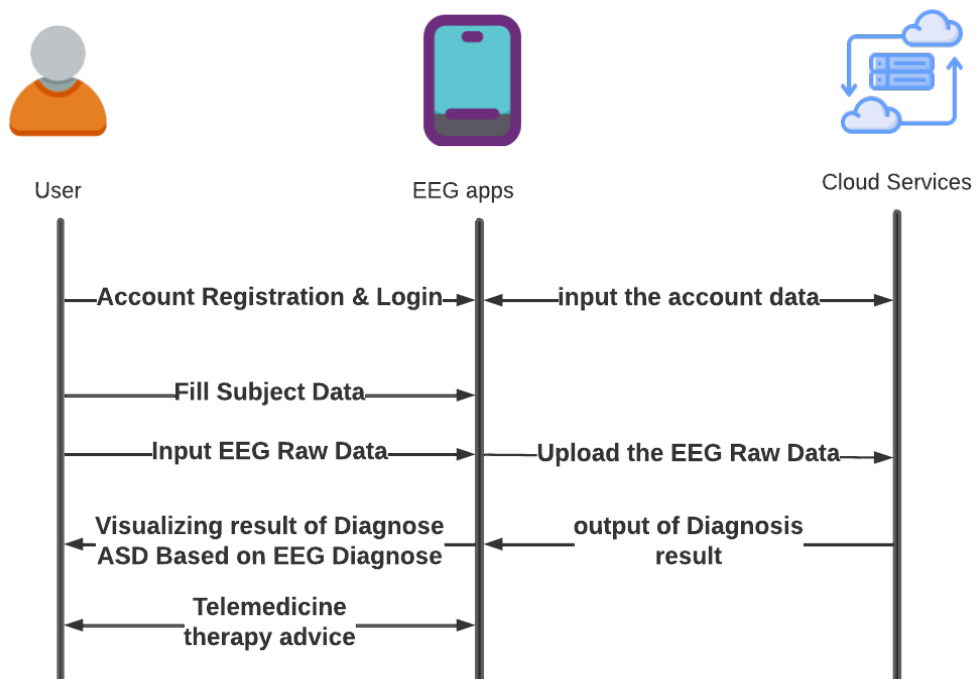
Gambar 3.7 Skema arsitektur umum aplikasi mobile yang ditawarkan untuk klasifikasi penderita ASD menggunakan sinyal EEG

Pada gambar 3.7 dapat dilihat bahwa sistem dari aplikasi mobile klasifikasi penderita ASD menggunakan sinyal EEG dibagi menjadi dua layer yakni *Front-end* dan *Back-end*. Pada *layer front-end* merupakan *layer* yang berisi tentang aktivitas yang berhubungan dengan antar pengguna. Dalam kasus penelitian ini *layer front-end* ini digunakan untuk meng-upload data Raw EEG dan visualisasi output dari hasil diagnosis data EEG serta beberapa fitur lainnya.

Pada Back-end layer merupakan bagian yang berisi sistem untuk diagnosis data EEG dan penyimpanan data. *Back-end layer* ini memiliki dua komponen yakni firebase yang berguna untuk penyimpanan data dan Flask digunakan untuk untuk framework yang mendiagnosis data EEG yang disimpan di database firebase.

2. Pembuatan *Use Case*: Tahap selanjutnya merupakan pembuatan *use case* pada pengembangan aplikasi Android merupakan tahap penting dalam mengidentifikasi kebutuhan dan fungsionalitas yang diperlukan oleh pengguna. *Use case* merupakan representasi *visual* dari interaksi antara pengguna dan aplikasi.

Gambar dibawah ini merupakan detail dari komponen & *use case relationship* diagram dari aplikasi yang ditawarkan:



Gambar 3 8 *Use case* dan Komponen Diagram pada aplikasi yang akan dibuat

Pada Gambar 3.8 merupakan gambaran umum *usecase* dan hubungan antar komponen dari aplikasi yang akan dibuat. Untuk registrasi awal, *user* hanya perlu melakukan registrasi akun dan mengisi beberapa data diri lainnya. Selanjutnya untuk bagian diagnosis, *user* hanya perlu meng-*input* data mentah EEG berjenis BCI2000 dan menunggu beberapa saat untuk melihat hasil diagnosis tersebut.

3. Pembuatan *Wireframe*: Tahap ini melibatkan pembuatan sketsa awal dari tampilan dan struktur aplikasi. Wireframe ini berfungsi sebagai dasar dalam merancang tampilan aplikasi.
4. Desain UI/UX: Tahap ini melibatkan desain tampilan antarmuka pengguna (UI) dan pengalaman pengguna (UX) yang menarik dan mudah digunakan. Hal ini mencakup pemilihan warna, ikon, font, dan layout yang tepat.
5. Pembuatan *Mockup*: Tahap ini melibatkan pembuatan tampilan aplikasi yang lebih realistis. Mockup ini membantu dalam memvisualisasikan bagaimana aplikasi akan terlihat pada perangkat Android yang berbeda.
6. Pengujian Desain: Tahap ini melibatkan pengujian desain aplikasi oleh tim pengembang maupun oleh pengguna untuk mengetahui apakah tampilan dan pengalaman pengguna sudah sesuai dengan kebutuhan dan keinginan pengguna.

Dengan melakukan tahap desain dengan baik, maka aplikasi *Android* yang dihasilkan akan memiliki tampilan yang menarik dan mudah digunakan. Tampilan yang menarik dan mudah digunakan akan meningkatkan pengalaman pengguna dan membuat pengguna lebih tertarik untuk menggunakan aplikasi.

b. Tahap Pengembangan Mobile Apps

Tahap ini merupakan tahap dimana desain-desain yang dirancang pada tahap sebelumnya diimplementasikan. Pada tahap ini dimulai dengan Implementasi UI dan UX dengan mulai membuat antarmuka pengguna (UI) dan mengatur interaksi dengan pengguna (UX). Pengembang dapat menggunakan teknologi seperti Android Studio atau Xcode untuk membuat layout, menambahkan widget, atau mengatur animasi dan transisi.

Selanjutnya pengembangan dilakukan dengan integrasi dengan API karena aplikasi *mobile* membutuhkan koneksi ke *server* atau sumber data eksternal untuk mengklasifikasikan data EEG, maka penelitian ini membutuhkan penulisan kode

untuk mengakses API tersebut. Hal ini melibatkan penggunaan teknologi seperti *http request* dan Firebase untuk mengakses REST API dari framework flask [12].

c. Pengujian

Tahapan pengujian pada pengembangan aplikasi android meliputi:

1. Pengujian Unit: Pengujian unit dilakukan pada level terkecil yaitu pada level kode dan fungsi-fungsi yang terdapat pada aplikasi. Pengujian unit bertujuan untuk memastikan bahwa setiap fungsi yang dibuat dapat berjalan dengan baik dan sesuai dengan spesifikasi yang telah ditentukan.
2. Pengujian Integrasi: Pengujian integrasi dilakukan pada level yang lebih tinggi yaitu pada integrasi antara berbagai komponen atau bagian aplikasi. Tujuannya adalah untuk memastikan bahwa setiap bagian aplikasi dapat berfungsi dengan baik ketika diintegrasikan dengan bagian yang lain.
3. Pengujian Fungsional: Pengujian fungsional bertujuan untuk memastikan bahwa aplikasi berfungsi dengan baik sesuai dengan kebutuhan pengguna dan spesifikasi yang telah ditentukan. Pengujian ini meliputi pengujian fitur, pengujian interaksi pengguna, pengujian integrasi dengan API, dan pengujian UI/UX.
4. Pengujian Performa: Pengujian performa dilakukan untuk memastikan bahwa aplikasi dapat berjalan dengan baik dan responsif di berbagai kondisi, seperti pada berbagai jenis perangkat, jaringan internet yang berbeda, dan beban pengguna yang berbeda.
5. Pengujian Keamanan: Pengujian keamanan dilakukan untuk memastikan bahwa aplikasi terlindungi dari berbagai jenis ancaman keamanan, seperti serangan hacking dan peretasan. Pengujian keamanan meliputi pengujian enkripsi data, pengujian proteksi jaringan, pengujian proteksi sandi, dan pengujian integrasi dengan protokol keamanan seperti HTTPS.

Tahapan pengujian mobile apps ini sangat penting untuk memastikan bahwa aplikasi dapat berfungsi dengan baik dan memenuhi kebutuhan pengguna serta standar kualitas yang telah ditetapkan. Pengujian yang baik dan sistematis dapat mengurangi risiko kegagalan aplikasi dan meningkatkan pengalaman pengguna.

G. Analisis Peforma Sistem

Analisis dilakukan dengan beberapa tahap yakni menganalisis hasil training pada model deep learning, dan kinerja keseluruhan arsitektur pada aplikasi *mobile* pada penelitian ini.

1. Analisis pada pelatihan *Deep Learning*

Analisis pada pelatihan *Deep Learning* dilakukan dengan melihat grafik yang dihasilkan dari proses *training* data. Grafik ini menampilkan informasi akurasi dari setiap *epochs* (hyperparameter yang menentukan berapa kali algoritma pembelajaran akan bekerja mengolah seluruh dataset training) yang dilakukan pada *training*. Analisis ini dilakukan dengan melihat apakah grafik *training* dan *validation* terjadi *fit* atau *overfitting*. Jika terjadi *overfitting*, maka akan dilakukan evaluasi pada *preprocessing* data sinyal EEG dan perbaikan model. Hasil percobaan pengolahan sinyal EEG tersebut juga dianalisis menggunakan *confusion matrix* berdasarkan parameter-parameter performa metode dalam analisa sinyal yaitu Sensitivitas (Pers 2.3), Spesitivitas (Pers 2.4), Akurasi (Pers 2.5), dan F1 *Score* (Pers 2.6). Dari analisis pada confusion matrix yang dihasilkan tersebut diharapkan memperoleh hasil yang sesuai dengan tujuan penelitian.

2. Analisis kinerja keseluruhan arsitektur pada aplikasi

Tahap analisis ini dilakukan dengan metode analisis *Black Box* dengan memperhatikan kinerja pada sistem arsitektur secara keseluruhan baik itu pada sistem Cloud, dan Aplikasi Mobile itu sendiri. Pada sistem cloud dilakukan beberapa analisis seperti koneksi untuk upload dan download data pada flask dan memastikan seluruh sistem bekerja dengan baik tanpa error. Pada Aplikasi mobile dilakukan analisis kinerja sistem seperti berjalan dengan baik atau terdapat bug atau error. Jika terdapat kesalahan akan dilakukan evaluasi dan perbaikan kinerja.

Pengujian ini bertujuan untuk menganalisa kinerja aplikasi android yang telah dibangun. Pengujian akan dilakukan sebanyak 30 kali dengan memeriksa bebrapa kriteria-kriteria pengujian. Kriteria pengujian kinerja adalah sebagai berikut:

- Pengguna *Register* akun dan *Login*.
- Aplikasi Android dapat terhubung dengan Firebase
- Aplikasi Android dapat terhubung dengan Cloud Classifier Engine
- Pengguna dapat mengupload file Raw data EEG ke cloud storage
- Aplikasi Android dapat menampilkan hasil klasifikasi.
- Aplikasi Android dapat menampilkan rekomendasi terapi sesuai dengan hasil yang ditampilkan

3.2 JADWAL PENELITIAN

Penelitian ini akan dilaksanakan sesuai dengan jadwal yang ditampilkan pada tabel berikut:

Tabel 3.2 Jadwal Penelitian

No	Kegiatan	Bulan																							
		1				2				3				4				5				6			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Persiapan Penelitian																								
	Identifikasi Masalah																								
	Studi Literatur																								
	Pengumpulan dataset																								
2	Pelaksanaan																								
	Klasifikasi sinyal EEG menggunakan CNN																								
	Evaluasi dan Analisa Hasil																								
	merancang dan membuat																								

[illegible]

BAB 4

LUARAN

Luaran utama pada penelitian ini adalah sebuah aplikasi mobile yang dapat membedakan penderita ASD dan normal berdasarkan data sinyal otak EEG secara *asynchronous*. Penelitian ini juga menghasilkan algoritma *preprocessing* dan data *pipeline* sinyal EEG BCI2000 sehingga dapat di automasi menggunakan pemrograman python. Penelitian ini juga menghasilkan luaran model terlatih *Deep Learning Convolutional Neural Network* yang di-*deploy* menggunakan Python-Flask sehingga diagnosis sinyal EEG penderita ASD dan normal dapat digunakan diberbagai *platform* melalui restAPI. Penelitian ini juga diharapkan membantu masyarakat dan mempercepat diagnosa penderita ASD agar dapat ditangani secara tepat sasaran.

DAFTAR PUSTAKA

- [1] “1.biomedical_signal_and Image_processing_najarian_splinter (1).pdf.”
- [2] “[John_G._Webster]_Medical_Instrumentation_Applicat(BookFi.org) (1).pdf.”
- [3] E. H. Shortliffe and J. J. Cimino, Eds., *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*. London: Springer London, 2014. doi: 10.1007/978-1-4471-4474-8.
- [4] “1.biomedical_signal_and Image_processing_najarian_splinter (1).pdf.”
- [5] T. M. Deserno, Ed., *Biomedical Image Processing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-15816-2.
- [6] “BCPy2000: BCPy2000.”
<http://bci2000.org/downloads/BCPy2000/BCPy2000.html> (accessed Mar. 07, 2023).
- [7] F. Shi *et al.*, “Review of Artificial Intelligence Techniques in Imaging Data Acquisition, Segmentation, and Diagnosis for COVID-19,” *IEEE Rev. Biomed. Eng.*, vol. 14, pp. 4–15, 2021, doi: 10.1109/RBME.2020.2987975.
- [8] M. Jamshidi *et al.*, “Artificial Intelligence and COVID-19: Deep Learning Approaches for Diagnosis and Treatment,” *IEEE Access*, vol. 8, pp. 109581–109595, 2020, doi: 10.1109/ACCESS.2020.3001973.
- [9] “3.11.2 Documentation.” <https://docs.python.org/3/> (accessed Mar. 08, 2023).
- [10] “What is a REST API? | IBM.” <https://www.ibm.com/topics/rest-apis> (accessed Mar. 08, 2023).
- [11] S. Cheng, “Convolutional Neural Networks”.
- [12] “Welcome to Flask — Flask Documentation (2.2.x).”
<https://flask.palletsprojects.com/en/2.2.x/> (accessed Mar. 08, 2023).
- [13] “Flutter documentation | Flutter.” <https://docs.flutter.dev/> (accessed Mar. 07, 2023).
- [14] M. N. A. Tawhid, S. Siuly, and H. Wang, “Diagnosis of autism spectrum disorder from EEG using a time–frequency spectrogram image-based approach,” *Electron. Lett.*, vol. 56, no. 25, pp. 1372–1375, Dec. 2020, doi: 10.1049/el.2020.2646.

- [15] A. B. Usakli, “Improvement of EEG Signal Acquisition: An Electrical Aspect for State of the Art of Front End,” *Comput. Intell. Neurosci.*, vol. 2010, pp. 1–7, 2010, doi: 10.1155/2010/630649.
- [16] M. S. N. Chowdhury, A. Dutta, M. K. Robison, C. Blais, G. A. Brewer, and D. W. Bliss, “Deep Neural Network for Visual Stimulus-Based Reaction Time Estimation Using the Periodogram of Single-Trial EEG,” *Sensors*, vol. 20, no. 21, p. 6090, Oct. 2020, doi: 10.3390/s20216090.
- [17] T.-H. Nguyen, D.-L. Yang, and W.-Y. Chung, “A High-Rate BCI Speller Based on Eye-Closed EEG Signal,” *IEEE Access*, vol. 6, pp. 33995–34003, 2018, doi: 10.1109/ACCESS.2018.2849358.
- [18] “Dr. Mohammed Jaffer Alhaddad - BCI Datasets.” <https://malhaddad.kau.edu.sa/Pages-BCI-Datasets-En.aspx> (accessed Mar. 07, 2023).
- [19] “Dissemination | National Center for Adaptive Neurotechnologies.” <https://www.neurotechcenter.org/research/bci2000/dissemination> (accessed Mar. 07, 2023).
- [20] “BCI2kReader · PyPI.” <https://pypi.org/project/BCI2kReader/#description> (accessed Mar. 07, 2023).
- [21] B. Xu *et al.*, “Wavelet Transform Time-Frequency Image and Convolutional Network-Based Motor Imagery EEG Classification,” *IEEE Access*, vol. 7, pp. 6084–6093, 2019, doi: 10.1109/ACCESS.2018.2889093.
- [22] X. Lun, Z. Yu, T. Chen, F. Wang, and Y. Hou, “A Simplified CNN Classification Method for MI-EEG via the Electrode Pairs Signals,” *Front. Hum. Neurosci.*, vol. 14, p. 338, Sep. 2020, doi: 10.3389/fnhum.2020.00338.
- [23] M. Zhou *et al.*, “Epileptic Seizure Detection Based on EEG Signals and CNN,” *Front. Neuroinformatics*, vol. 12, p. 95, Dec. 2018, doi: 10.3389/fninf.2018.00095.
- [24] W.-L. Mao, H. I. K. Fathurrahman, Y. Lee, and T. W. Chang, “EEG dataset classification using CNN method,” *J. Phys. Conf. Ser.*, vol. 1456, no. 1, p. 012017, Jan. 2020, doi: 10.1088/1742-6596/1456/1/012017.
- [25] “SPECTRAL AUDIO SIGNAL PROCESSING.” <https://ccrma.stanford.edu/~jos/sasp/> (accessed Mar. 08, 2023).

- [26] L. P. A. Arts and Egon. L. van den Broek, “The fast continuous wavelet transformation (fCWT) for real-time, high-quality, noise-resistant time–frequency analysis,” *Nat. Comput. Sci.*, vol. 2, no. 1, pp. 47–58, Jan. 2022, doi: 10.1038/s43588-021-00183-z.
- [27] Q. Liu *et al.*, “Spectrum Analysis of EEG Signals Using CNN to Model Patient’s Consciousness Level Based on Anesthesiologists’ Experience,” *IEEE Access*, vol. 7, pp. 53731–53742, 2019, doi: 10.1109/ACCESS.2019.2912273.
- [28] B. Yanti, A. Z. A. Ramadhani, Z. F. Muhamad, and A. Yafi, “Application for COVID-19 severity diagnosis and asynchronous telehealth: development and prototype of G-COV,” *Bali Med. J.*.
- [29] “API Documentation | TensorFlow v2.11.0.” https://www.tensorflow.org/api_docs (accessed Mar. 08, 2023).
- [30] “Keras: the Python deep learning API.” <https://keras.io/> (accessed Mar. 08, 2023).
- [31] F. Chollet, *Deep learning with Python*. Shelter Island, New York: Manning Publications Co, 2018.
- [32] M. Sahu and R. Dash, “A Survey on Deep Learning: Convolution Neural Network (CNN),” in *Intelligent and Cloud Computing*, vol. 153, D. Mishra, R. Buyya, P. Mohapatra, and S. Patnaik, Eds. Singapore: Springer Singapore, 2021, pp. 317–325. doi: 10.1007/978-981-15-6202-0_32.
- [33] “Dokumentasi Firebase.” <https://firebase.google.com/docs?hl=id> (accessed Mar. 08, 2023).
- [34] S. Muhammad Usman, S. Khalid, and M. H. Aslam, “Epileptic Seizures Prediction Using Deep Learning Techniques,” *IEEE Access*, vol. 8, pp. 39998–40007, 2020, doi: 10.1109/ACCESS.2020.2976866.
- [35] F. Chollet, *Deep learning with Python*. Shelter Island, New York: Manning Publications Co, 2018.
- [36] “Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions,” *Choice Rev. Online*, vol. 49, no. 10, pp. 49-5403-49–5403, Jun. 2012, doi: 10.5860/CHOICE.49-5403.

- [37] P. Deemer, G. Benefield, C. Larman, and B. Vodde, “A Lightweight Guide to the Theory and Practice of Scrum”.
- [38] J. Loeliger and M. McCullough, *Version control with Git*, Second edition. Beijing: O’Reilly, 2012.

