BAB III
METODE PENELITIAN

Bab ini menyajikan langkah-langkah penelitian untuk melakukan analisis *big data* sinyal sEMG wajah menggunakan Apache Spark pada komputer terdistribusi. Metode penelitian dirancang secara sistematis untuk memastikan proses penelitian dilakukan dengan langkah-langkah terstruktur dalam rangka mencapai tujuan mengidentifikasi pola emosi secara lebih akurat.

3.1 Jenis Penelitian

Penelitian ini adalah jenis eksperimental dengan pendekatan *mixed-method* yang dikembangkan menggunakan model *iterative-incremental development* (IID). Pemilihan metodologi ini didasarkan pada karakteristik penelitian yang membutuhkan eksperimen berulang dan pengembangan bertahap dalam implementasi sistem analisis *big data* [68]. Model IID memberikan fleksibilitas dalam mengoptimasi parameter *clustering* dan infrastruktur komputasi terdistribusi, serta memungkinkan evaluasi dan penyempurnaan berkelanjutan pada setiap komponen sistem [68], [69]. Jenis penelitian eksperimental dengan model IID telah terbukti efektif dalam analisis data berskala besar. Hal ini dibuktikan dengan pencapaian performa MSE yang lebih baik (5,1%-7,8%) dibandingkan model konvensional (11,6%-16,9%) dalam pembelajaran fitur data dinamis, sekaligus mempertahankan pengetahuan yang telah dipelajari sebelumnya [70]. Selain itu, pendekatan *mixed-method* bertujuan untuk menggabungkan kekuatan analisis kuantitatif dalam evaluasi performa sistem dengan analisis kualitatif terhadap proses pengembangan dan interpretasi hasil, sehingga mampu memberikan pemahaman yang utuh terhadap sistem yang dikembangkan [71].

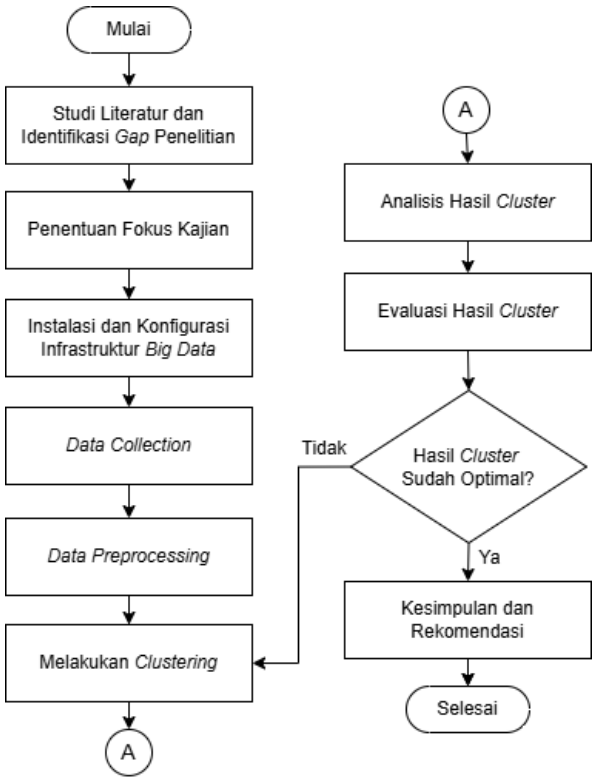
Jenis penelitian eksperimental dengan model IID ini dipilih karena memiliki beberapa keunggulan spesifik dalam konteks penelitian analisis *big data* sinyal sEMG [72], [73]:

1. Mendukung penyesuaian dan optimasi sistem secara berkelanjutan.
2. Memungkinkan proses validasi bertahap untuk memastikan kualitas hasil.
3. Memberikan fleksibilitas terhadap perubahan kebutuhan selama pengembangan.
4. Memfasilitasi eksperimen dengan berbagai parameter dan konfigurasi.
5. Memfasilitasi deteksi dan penanganan masalah lebih awal dalam proses pengembangan.

Keunggulan lainnya dari pendekatan eksperimental dengan model IID adalah dukungan terhadap dokumentasi yang terstruktur dan evaluasi yang sistematis, sehingga penelitian menghasilkan kontribusi yang signifikan dalam pengembangan sistem analisis emosi berbasis sinyal sEMG menggunakan *big data* [74].

3.2 Alur Penelitian

Alur penelitian dibuat untuk mengarahkan penelitian dengan struktur yang sistematis serta mempermudah dalam memahami proses penelitian dari awal hingga akhir. Penelitian ini dilakukan dengan tahapan yang ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Alur Penelitian Analisis *Big Data* Sinyal sEMG

1. Studi literatur dan identifikasi *gap* penelitian

   Pada tahap ini dilakukan pengumpulan dan analisis literatur terkait analisis sinyal sEMG wajah dan metode *clustering* pada pemrosesan *big data*.

2. Penentuan fokus kajian

   Berdasarkan hasil studi literatur dan identifikasi *gap* penelitian, ditentukan fokus kajian yang spesifik untuk pengembangan proposal penelitian.

3. Instalasi dan konfigurasi infrastruktur *big data*

   Tahap ini dilakukan *setup* infrastruktur *big data* pada *cluster* komputer yang terhubung dalam jaringan lokal dengan sistem operasi Ubuntu, konfigurasi HDFS sebagai sistem penyimpanan terdistribusi, dan Apache Spark untuk pemrosesan data. Manajemen *cluster* dilakukan melalui koneksi SSH dari perangkat (user client).

4. *Data Collection*

   Tahap ini mencakup proses pengumpulan data yang diperoleh dari *platform Kaggle*. *Dataset* yang digunakan adalah EMGDataVR dari EmteqLabs, yang berisi rekaman sinyal sEMG wajah.

5. *Data preprocessing*

   Pada tahap ini dilakukan pembersihan awal seperti seleksi fitur (feature selection), pembersihan baris data (row filtering), serta deteksi dan penanganan *outlier* (outlier detection), untuk meningkatkan kualitas dan memastikan data yang akan dianalisis tersebut bersih, relevan, dan sesuai konteks penelitian.

6. Melakukan *clustering*

   Tahap ini dilakukan teknik *clustering* yang telah ditentukan untuk memperoleh hasil yang akan dianalisis dari *dataset* menggunakan pemrosesan *big data*.

7. Analisis hasil *cluster*

   Tahap ini dilakukan evaluasi performa sistem yang telah ditetapkan dan analisis hasil *cluster*. Analisis juga mencakup perbandingan hasil dengan penelitian-penelitian relevan sebelumnya untuk mengidentifikasi kontribusi dan kebaruan.

8. Evaluasi hasil *cluster*

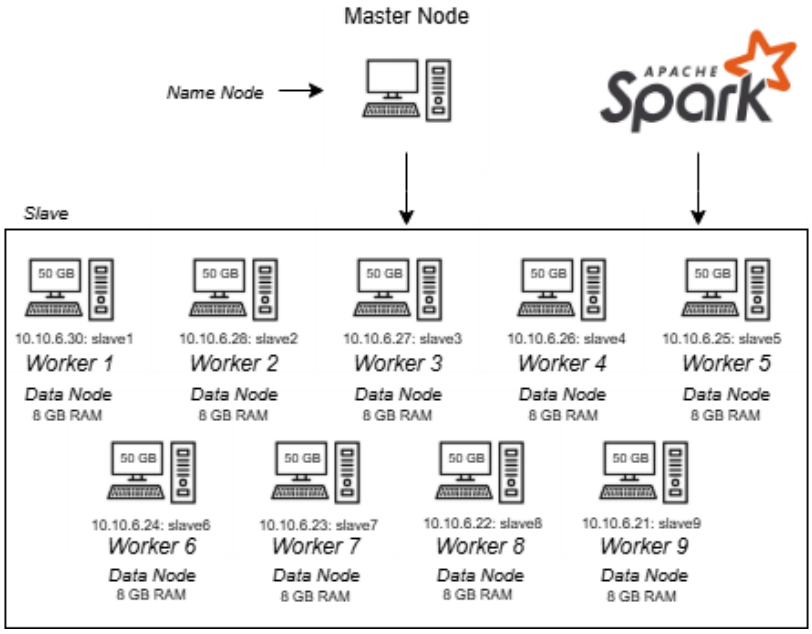
Tahap evaluasi dilakukan untuk menilai kesesuaian hasil *cluster* dengan kategori stimulus emosi berdasarkan teori *Facial Action Coding System* (FACS) oleh Paul Ekman. Evaluasi difokuskan pada kesesuaian pola *cluster* yang terbentuk dengan *Action Units* (AU) yang merepresentasikan kontraksi otot tertentu. Jika hasil *clustering* belum menunjukkan korespondensi yang optimal dengan pola AU yang seharusnya muncul pada masing-masing kategori emosi, maka dilakukan perbaikan, baik dari segi parameter maupun metode *clustering*, dan tahap percobaan diulang.

9. Kesimpulan dan rekomendasi

   Tahap akhir penelitian adalah penyusunan kesimpulan untuk menjawab rumusan penelitian dan rekomendasi pengembangan penelitian selanjutnya berdasarkan hasil dan keterbatasan penelitian saat ini.

3.3 Arsitektur Sistem

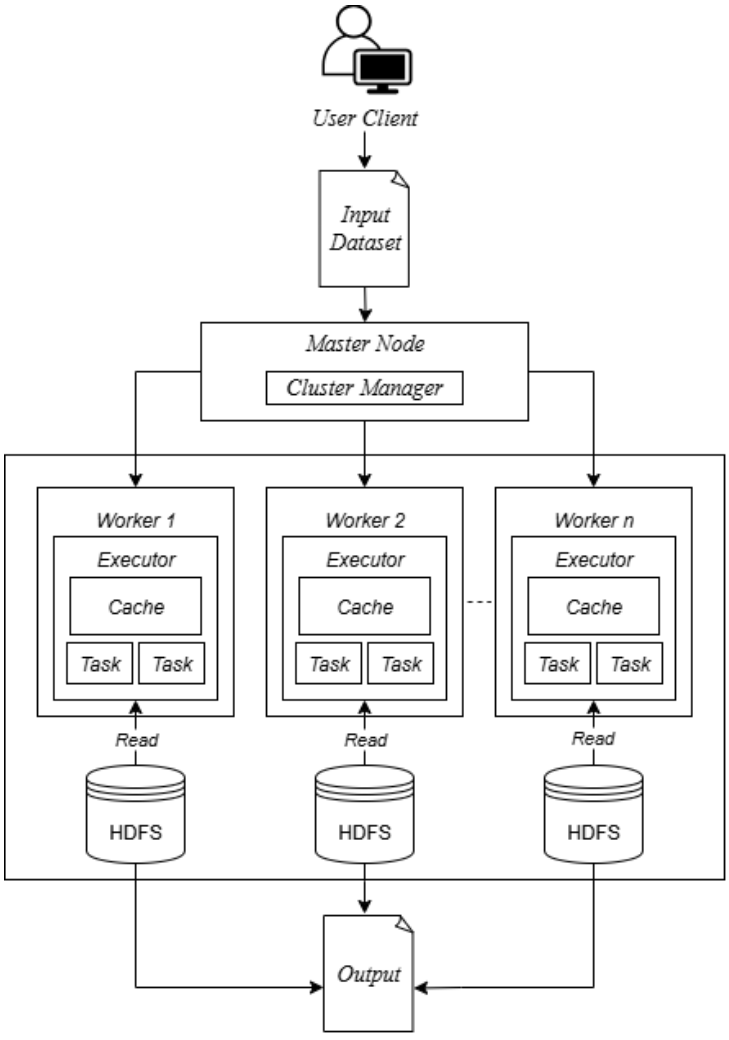
Arsitektur sistem dirancang untuk mengoptimalkan proses analisis *big data* sinyal sEMG dari awal pengambilan data hingga hasil identifikasi pola emosi. Sistem ini dibangun di atas komputasi terdistribusi dengan Apache Spark untuk memastikan efisiensi dan skalabilitas pemrosesan data berskala besar.



Gambar 3. 2 Arsitektur Infrastruktur *Cluster* Spark

Infrastruktur Spark penelitian ini terdiri dari satu *master node* dan 9 *worker nodes* yang terhubung dalam satu jaringan. Sistem penyimpanan menggunakan HDFS untuk mengakses data secara paralel dan efisien. Skema arsitektur infrastruktur *cluster* Spark ditampilkan pada Gambar 3.2.

Gambar 3.3 menunjukkan arsitektur sistem yang diterapkan dalam penelitian ini untuk memproses sinyal sEMG secara terdistribusi menggunakan Apache Spark. Arsitektur ini mendeskripsikan alur pemrosesan mulai dari proses *input* data sampai keluaran hasil identifikasi pola emosi, termasuk komponen dalam eksekusi paralel di lingkungan Spark.



Gambar 3. 3 Arsitektur Sistem Pemrosesan Terdistribusi Sinyal sEMG Menggunakan Apache Spark

Berikut penjelasan setiap komponen pada Gambar 3.3:

1. *User Client*

   Perangkat pengguna (laptop penulis) yang berfungsi untuk mengontrol dan menjalankan proses eksperimen.

2. *Input Dataset*

   Dataset sinyal sEMG wajah dari 38 partisipan dengan total lebih dari 30 juta baris, disimpan di HDFS dalam format partisi agar dapat diakses secara paralel oleh semua *node* di dalam *cluster*.

3. *Master Node* (Cluster Manager)

   Node utama yang menjalankan Spark Master dan HDFS NameNode, bertanggung jawab untuk mengkoordinasikan seluruh proses komputasi dan distribusi tugas ke *worker nodes*.

4. *Worker Nodes*

   Terdapat 9 *worker nodes* yang masing-masing berperan sebagai DataNode (HDFS) sekaligus *worker* dalam Spark. *Worker nodes* menjalankan *executor*, yaitu unit eksekusi yang memproses *task* komputasi secara paralel sesuai instruksi dari *master node*. Konfigurasi dengan 9 *worker nodes* dipilih untuk mendistribusikan beban komputasi paralel untuk mengurangi waktu eksekusi. Pemilihan ini mempertimbangkan pada kebutuhan pemrosesan data sinyal sEMG berskala besar, dimana penambahan jumlah *node* berkontribusi dalam peningkatan efisiensi, kapasitas pemrosesan, dan skalabilitas sistem secara keseluruhan.

5. HDFS

   Sistem penyimpanan data terdistribusi yang memungkinkan akses dan pemrosesan data secara paralel di seluruh *worker nodes*.

6. Proses Pemrosesan (Preprocessing dan Clustering)

   Tahap pemrosesan data terdiri dari:

   - *Preprocessing*: mencakup pemilihan fitur (feature selection), pembersihan baris data (row filtering), dan deteksi *outlier* (outlier detection) untuk meningkatkan kualitas data sebelum masuk ke proses analisis.

- *Clustering*: dilakukan menggunakan algoritma K-Means dan Bisecting K-Means yang dijalankan pada *cluster* Spark secara paralel dengan evaluasi menggunakan *silhouette coefficient.*

7. *Output*

   Hasil akhir dari proses *clustering* dianalisis untuk mengidentifikasi pola sinyal otot wajah yang berkaitan dengan kondisi emosi tertentu.

3.3.1 Alur Kerja Sistem

Alur kerja pada sistem ini menggambarkan tahapan yang dilakukan dalam proses analisis *big data* sinyal sEMG, mulai dari pemuatan data hingga interpretasi pola emosi. Tahapan alur kerja sistem secara rinci dapat dijelaskan sebagai berikut:

1. Sumber Data

   Dataset EmgDataVR tersedia dari Emteq Labs berisi rekaman sinyal sEMG wajah yang diperoleh dari 38 responden sehat menggunakan sistem emteqPRO dengan Pico Neo 2 Eye VR *headset*.

2. Penyimpanan Data

   Dataset disimpan ke dalam HDFS pada *cluster*,yang menyediakan penyimpanan terdistribusi untuk akses data paralel oleh semua *node*.

3. Infrastruktur Pemrosesan

   Penggunaan Apache Spark yang berjalan di atas sistem Hadoop dengan arsitektur *distributed computing* yang diimplementasikan pada jaringan komputer. *Cluster* terdiri dari satu *master node* yang menjalankan HDFS NameNode serta 9 *worker nodes* yang menjalankan HDFS DataNode.

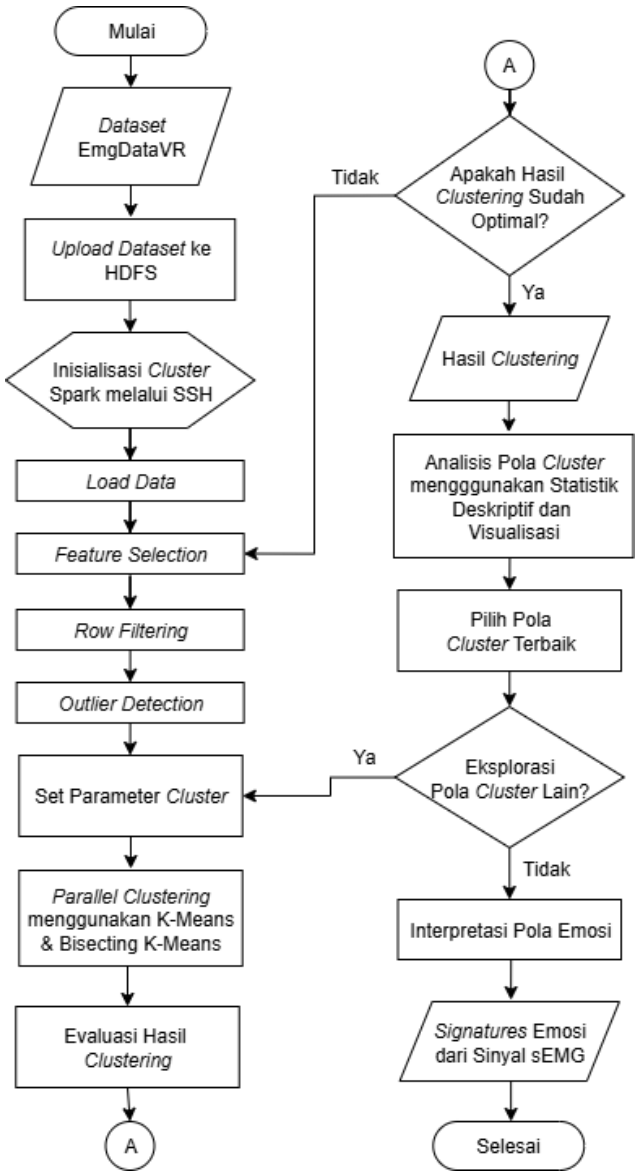
4. Pemrosesan Data

   Proses *loading* data menggunakan PySpark, dilanjutkan dengan *preprocessing* yang meliputi:

   - Pemilihan fitur relevan (feature selection)
   - Pembersihan baris data (row filtering)
   - Penanganan *outlier* (outlier detection)

5. Analisis *Clustering* dan Evaluasi

   Pemrosesan paralel dengan dua algoritma *clustering* (K-Means dan Bisecting K-Means). Proses ini dimulai dengan pengaturan parameter *cluster* yang sesuai untuk masing-masing algoritma. Hasil *clustering* kemudian dievaluasi menggunakan metrik *silhouette coefficient* untuk memilih konfigurasi terbaik. Jika hasil evaluasi belum optimal, proses akan kembali ke tahap pengaturan parameter *cluster*.



Gambar 3. 4 *Flowchart* Sistem Pemrosesan dan Analisis Sinyal sEMG Menggunakan Apache Spark

6. Analisis dan Visualisasi

   Penggunaan statistik deskriptif dan teknik visualisasi untuk mengidentifikasi pola-pola *cluster* yang terbentuk. Sistem mengevaluasi apakah distribusi pola *cluster* sudah optimal. Jika belum optimal, dilakukan pemilihan pola *cluster* terbaik sebelum melanjutkan ke tahap interpretasi pola emosi.

7. Interpretasi Hasil

   Penentuan pola (signatures) emosi dari sinyal sEMG yang dapat digunakan sebagai dasar untuk sistem pengenalan emosi.

Keseluruhan arsitektur pada Gambar 3.4 dirancang untuk memaksimalkan efisiensi pemrosesan *distributed computing* dari Apache Spark untuk memproses data secara efisien dan *scalable* pada infrastruktur *multi-node*.

Selanjutnya, Gambar 3.5 menjelaskan proses *distributed computing* yang diimplementasikan menggunakan Apache Spark pada infrastruktur *cluster*. Alur tahapan proses tersebut meliputi:

1. *Upload* kode PySpark

   Proses mengunggah kode PySpark yang berisi instruksi *preprocessing*, *clustering*, atau analisis lain terhadap *dataset* yang tersimpan di HDFS.

2. Inisialisasi *SparkSession*

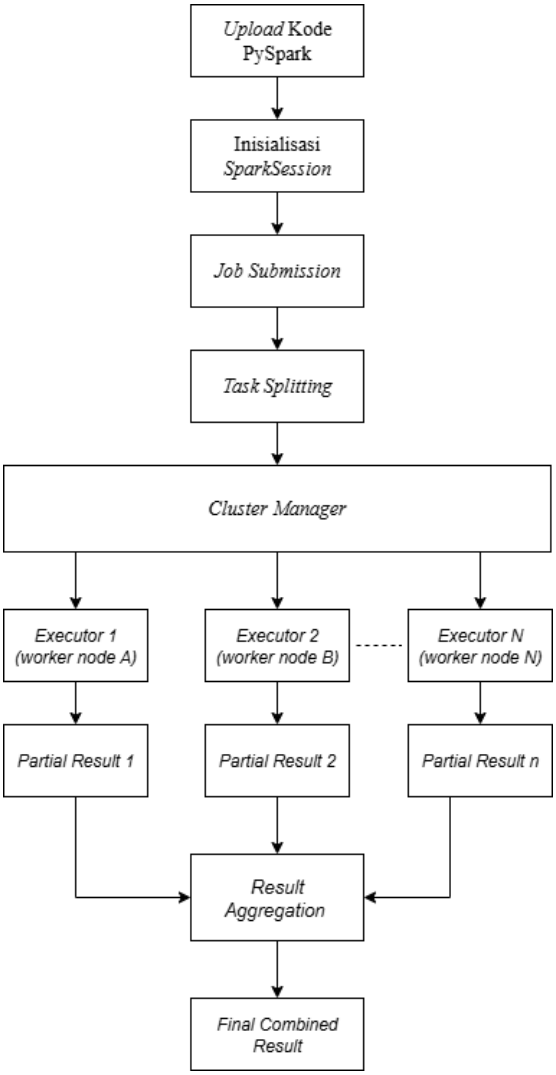
   SparkSession adalah titik awal dalam menjalankan semua operasi Spark. Setelah kode berhasil diunggah, *SparkSession* diinisialisasi untuk mengatur konteks eksekusi Spark, mengelola koneksi ke cluster, dan mengakses data dari HDFS.

3. *Job Submission*

   Spark membagi instruksi dalam kode menjadi satu atau lebih *job*. Setiap *job* adalah unit kerja yang berisi sejumlah *stage* (tahapan eksekusi) yang nantinya akan diproses lebih lanjut dalam sistem terdistribusi.

4. *Task Splitting*

   Setiap *stage* dipecah menjadi beberapa *task* yang lebih kecil. *Task* merupakan unit terkecil dari komputasi dalam Spark yang akan dijalankan secara paralel oleh *executor* di masing-masing *node*.



Gambar 3. 5 Proses *Distributed Computing* untuk Pemrosesan Sinyal sEMG Menggunakan Apache Spark

5. *Cluster Manager*

   Cluster Manager bertanggung jawab mengatur penjadwalan *task* dan alokasi *resource* pada masing-masing *node*. *Cluster Manager* menentukan *executor* mana yang menjalankan *task* tertentu berdasarkan ketersediaan dan efisiensi.

6. *Executor* N (worker node N)

   Task yang telah dibagi oleh *driver* dieksekusi secara paralel oleh beberapa *executor,* yang ditempatkan pada *worker nodes*/*slave*. Jumlah *executor* serta penempatannya ditentukan oleh *cluster manager* berdasarkan ketersediaan

sumber daya. Setiap *executor* pada *worker node* berbeda memproses sebagian data dan menghasilkan hasil parsial (partial result).

7. Penggabungan Hasil (Result Aggregation)
   Setelah semua *executor* menyelesaikan tugasnya, hasil parsial dikembalikan ke *driver* atau digabung melalui operasi agregasi tertentu (seperti reduce, groupBy, atau collect). Proses ini menghasilkan *output* akhir yang merepresentasikan keseluruhan hasil komputasi terdistribusi.
8. Hasil Akhir Terpadu (Final Combined Result)
   Hasil gabungan seluruh proses komputasi terdistribusi. Data telah melalui pemrosesan atau analisis sesuai instruksi awal dan siap untuk dievaluasi lebih lanjut.

Alur proses pada Gambar 3.5 ini menggambarkan mekanisme eksekusi terdistribusi pada Apache Spark, di mana setiap tugas dibagi ke dalam beberapa unit kerja dan dijalankan secara paralel oleh *executor* pada masing-masing *worker node*. Hasil parsial dari setiap *worker node* kemudian digabungkan untuk membentuk *output* akhir yang utuh. Pendekatan ini mendukung efisiensi pemrosesan data berukuran besar, seperti sinyal sEMG, dalam konteks analisis berbasis komputasi terdistribusi.

3.3.2 Algoritma Sistem

Bagian ini menjelaskan *pseudocode* dari tahapan utama sistem yang terdiri dari proses *preprocessing* dan *clustering*. *Pseudocode* ini bertujuan untuk memformulasikan alur logika sistem sebelum direalisasikan dalam implementasi program di lingkungan Apache Spark.

3.3.2.1 *Preprocessing*

Proses *preprocessing* dilakukan menggunakan Apache Spark dengan bahasa pemrograman PySpark untuk menangani volume data sebesar 30.327.000 baris dengan ukuran ±13,9 GB dalam format .csv yang dihasilkan dari sinyal sEMG 38 responden. Tahapan utama yang dilakukan meliputi *feature selection*, penghapusan

baris data dengan *missing value* dan data yang keliru atau tidak sesuai (row filtering), serta *outlier detection*. Proses *filtering* biosinyal tidak dilakukan karena pada *dataset* asli (mentah) sudah melalui tahap *filtering*.

1. *Feature selection* & *row filtering*

Feature selection pada penelitian ini bertujuan menyaring atribut tidak relevan dan membersihkan baris data tidak sesuai untuk analisis lebih lanjut. *Feature selection* dilakukan berdasarkan penelitian pendahuluan serta karakteristik sinyal sEMG. Kolom-kolom yang dihapus tersebut, yaitu:

- Kolom metadata: *Frame, Time, arousal,* dan *valence*
- Kolom sinyal sEMG terfilter:
  Emg/Filtered/zygo/weighted
  Emg/Filtered/orbi/weighted
  Emg/Filtered/front/weighted
  Emg/Filtered/corr/weighted
- Kolom sinyal sEMG kontak dan sinyal sEMG terfilter yang tidak diperlukan dalam proses *clustering*:
  Emg/Contact[0], *Emg/Filtered*[0], *Emg/Contact*[1], *Emg/Filtered*[1], *Emg/Contact*[2], *Emg/Filtered*[2], *Emg/Contact*[3], *Emg/Filtered*[3], *Emg/Contact*[4], *Emg/Filtered*[4], *Emg/Contact*[5], *Emg/Filtered*[5], *Emg/Contact*[6], *Emg/Filtered*[6]

Setelah penghapusan kolom-kolom tersebut, tersisa 7 kolom sinyal sEMG amplitudo dan 1 kolom *event* yang dilakukan *row filtering* pada data berikut:

- Baris data dengan nilai *event* "*break*" yang tidak mewakili respons emosional terhadap stimulus.
- Semua baris data kolom yang tidak bernilai (NULL) agar tidak mengganggu proses *clustering*.

Tahapan-tahapan ini dirancang agar data yang digunakan pada tahap analisis memiliki kualitas yang baik dan hanya memuat fitur-fitur yang relevan.

2. *Outlier detection*

Penanganan *outlier* dilakukan menggunakan metode IQR (Interquartile Range) untuk semua kolom numerik. Proses ini bertujuan untuk menghilangkan *noise* dan nilai ekstrem yang dapat mengganggu pola asli pada sinyal sEMG. Algoritma 3 menunjukkan rangkaian proses deteksi dan penghapusan *outlier*.

Algoritma 3. Penghapusan *Outlier*

```
   INPUT: semua nama kolom numerik yang sudah di
   normalisasi
   OUTPUT: data bersih tanpa outlier

   ALGORITMA:
1  columns ← ambil semua nama kolom numerik
2  outlierMask ← DF baru dengan ukuran sama seperti
   data input, diisi False

3  FOR col IN columns DO:
4  Q1 ← kuartil ke-1 dari col
5  Q3 ← kuartil ke-3 dari col
6  IQR ← Q3 - Q1
7  lowerBound ← Q1 - 1.5 × IQR
8  upperBound ← Q3 + 1.5 × IQR

9  FOR i IN normalizedDF[col] DO:
        IF normalizedDF[i][col] < lowerBound OR
        normalizedDF[i][col] > upperBound THEN
             outlierMask[i][col] ← True

10 rowOutlierCount ← jumlah nilai True per baris
   dalam outlierMask

11 cleanDF ← ambil semua baris dari input di mana
   rowOutlierCount < 4

12 write("Outlier detection selesai: " +
   jumlah_baris(cleanDF) + " baris data bersih")
13 return cleanDF
```

3.3.2.2 *Clustering*

Clustering dilakukan untuk mengelompokkan data sinyal sEMG wajah berdasarkan kemiripan pola nilai sinyal sEMG amplitudo untuk mengungkap struktur alami yang merepresentasikan respons emosional. Setelah melalui tahapan

preprocessing, data kemudian dikelompokkan menggunakan dua algoritma berbeda, yaitu K-Means dan Bisecting K-Means. Kedua algoritma ini dijalankan pada *cluster* Spark agar mampu menangani data dalam skala besar secara paralel dan terdistribusi.

Nilai parameter K (jumlah cluster) dioptimalkan dengan mempertimbangkan *silhouette coefficient* sebagai metrik evaluasi, karena mampu memberikan penilaian kuantitatif yang objektif tanpa bergantung pada visualisasi seperti *elbow method*. Hal ini merupakan pendekatan terbaik untuk sinyal sEMG berskala besar, di mana kurva *elbow* tidak efektif divisualisasikan. Selain itu, *silhouette coefficient* memungkinkan evaluasi dilakukan secara otomatis dan efisien dalam lingkungan komputasi terdistribusi seperti Apache Spark. Berikut algoritma yang digunakan dalam penelitian:

1. K-Means

Proses *clustering* menggunakan algoritma K-Means yang diimplementasikan pada *cluster* Spark untuk mendukung pemrosesan data secara paralel dan terdistribusi. Pemilihan jumlah *cluster* (K) dioptimalkan menggunakan nilai *silhouette coefficient* untuk memperoleh struktur *cluster* yang paling representatif terhadap data. Rangkaian proses tersebut dijelaskan menggunakan *pseudocode* pada Algoritma 4.

Algoritma 4. K-Means (sEMG)

```
    INPUT: k, dataPaths, excludedSubjects (10,27,40)
    OUTPUT: centroid, assignments, evaluasi, statistik

    ALGORITMA:
1   validSubjects ← [1..41] tanpa excludedSubjects
2   dataPaths ← gabungkan semua file CSV dari
    validSubjects
3   rawDF ← baca data dari dataPaths

4   numericCols ← semua kolom numerik dari rawDF tanpa
    kolom ID
5   featureDF ← gabungkan numericCols menjadi vektor
    "features"
```

Algoritma 4. K-Means (sEMG)

```
model ← KMeans(k=k).fit(featureDF)
predictions ← model.transform(featureDF)

centroid ← model.clusterCenters
assignments ← predictions["prediction"]

evaluasi ← hitung silhouette score dari predictions

rawUnique ← hapus duplikat rawDF berdasarkan "ID"
predUnique ← hapus duplikat predictions[["ID",
   "prediction"]]
fullDF ← gabungkan rawUnique dan predUnique
   berdasarkan "ID"

FOR setiap fitur col dalam numericCols DO
    stats ← hitung count, min, max, avg, stddev per
     cluster
    median ← approxQuantile(fullDF where
     cluster=cid, col)
    mode ← nilai col dengan count terbanyak per
     cluster
END FOR

clusterCounts ← jumlah baris per cluster
simpan hasil predictions ke folder output

RETURN centroid, assignments, evaluasi, statistik
```

2. Bisecting K-Means

Proses *clustering* juga dilakukan menggunakan algoritma Bisecting K-Means yang diimplementasikan pada *cluster* Spark untuk pemrosesan terdistribusi. Pemilihan jumlah *cluster* (K) dioptimalkan berdasarkan nilai *silhouette coefficient* untuk memperoleh pembagian *cluster* yang optimal. Tahapan proses ini dijelaskan pada Algoritma 5.

Algoritma 5. Bisecting K-Means (sEMG)

```
INPUT: k, dataPaths, excludedSubjects (10,27,40)
OUTPUT: centroid, assignments, evaluasi, statistik

ALGORITMA:
```

Algoritma 5. Bisecting K-Means (sEMG)

```
validSubjects ← [1..41] tanpa excludedSubjects
dataPaths ← gabungkan semua file CSV dari
validSubjects
rawDF ← baca data dari dataPaths

numericCols ← semua kolom numerik dari rawDF tanpa
kolom ID
featureDF ← gabungkan numericCols menjadi vektor
"features"

clusterList ← [seluruh data dalam featureDF]
assignments ← DataFrame kosong
WHILE jumlah cluster dalam clusterList < k DO
    pilih cluster C dengan jumlah data terbesar
    dari clusterList
    splitResult ← KMeans(k=k).fit(C)
    assignC ← hasil prediksi splitResult
    ganti C dalam clusterList dengan dua cluster
    hasil split
    gabungkan assignC ke assignments
END WHILE

gabungkan semua centroid dari hasil split ke dalam
centroid
predictions ← hasil penggabungan semua assignC

evaluasi ← hitung silhouette score dari predictions

rawUnique ← hapus duplikat rawDF berdasarkan "ID"
predUnique ← hapus duplikat predictions[["ID",
"prediction"]]
fullDF ← gabungkan rawUnique dan predUnique
berdasarkan "ID"

FOR setiap fitur col dalam numericCols DO
    stats ← hitung count, min, max, avg, stddev per
    cluster
    median ← approxQuantile(fullDF where
    cluster=cid, col)
    mode ← nilai col dengan count terbanyak per
    cluster
END FOR

clusterCounts ← jumlah baris per cluster
simpan hasil predictions ke folder output

RETURN centroid, assignments, evaluasi, statistik
```

3.4 Alat dan Bahan

Dalam melakukan penelitian ini, dibutuhkan alat dan bahan yang menunjang kelancaran dan keberhasilan penelitian. Tabel 3.1 menyajikan informasi perangkat keras yang digunakan beserta spesifikasinya, sedangkan Tabel 3.2 memuat rincian perangkat lunak yang dimanfaatkan dalam proses penelitian ini serta Tabel 3.3 menampilkan deskripsi *dataset* yang digunakan untuk analisis data.

Tabel 3. 1 Informasi *Hardware* yang Digunakan dalam Penelitian untuk Memproses Sinyal sEMG pada Sistem Komputasi Terdistribusi

Komponen *Hardware*	**Spesifikasi**
Laptop client (penulis)	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz – 8,00 GB RAM
Master Node	8,00 GB RAM, 50GB *Harddisk*
Worker Nodes	8,00 GB RAM, 50GB *Harddisk*

Tabel 3. 2 Informasi *Software* yang Digunakan dalam Penelitian untuk Memproses Sinyal sEMG pada Sistem Komputasi Terdistribusi

Kategori *Software*	**Aplikasi/*Tool***	**Fungsi**
Sistem Operasi	Ubuntu Server 22.04 LTS	Sistem operasi pada *cluster nodes*
Big Data Storage	HDFS	Sistem penyimpanan terdistribusi
Big Data Processing	Apache Spark	*Framework* untuk *distributed computing*
Bahasa Pemrograman	PySpark	Bahasa pemrograman untuk implementasi sistem
Remote Access	PuTTY	SSH *client* untuk akses *remote* ke *cluster*
Remote Desktop	AnyDesk	*Remote access*
File Transfer	WinSCP	Transfer *file* antara komputer dan *cluster*

Kategori *Software*	Aplikasi/*Tool*	Fungsi
Machine Learning	Spark MLlib	*Library machine learning* untuk implementasi *clustering*

Tabel 3. 3 Informasi *Dataset* Mentah Sinyal sEMG dari EmteqLabs yang Digunakan dalam Penelitian

Aspek	Deskripsi
Nama *Dataset*	EmgDataVR dari EmteqLabs
Jumlah Partisipan	38 orang (14 wanita, 24 pria), usia 18–68 tahun (rata-rata 33,4 ± 13,6 tahun)
Perangkat Rekam	emteqPRO sensor + Pico Neo 2 Eye VR *headset*
Jenis Stimulus	25 video terstandarisasi (negatif, netral, dan positif)
Jenis Data	Sinyal sEMG dari 7 titik otot wajah
Format Data	*File* CSV (Comma-Separated Values)
Volume Data	±800.000 baris per partisipan × 38 = 30.327.000 baris total; ukuran ±300 MB per partisipan, total ±13,9 GB
Jumlah Kolom	31 kolom
Urutan Sensor	Sensor dikodekan dari [0] hingga [6] sesuai lokasi otot wajah sebagai berikut: [0] : *Right Frontalis* (dahi kanan) [1] : *Right Zygomaticus* (pipi kanan) [2] : *Right Orbicularis* (sekitar mata kanan) [3] : *Center Corrugator* (tengah alis) [4] : *Left Orbicularis* (sekitar mata kiri) [5] : *Left Zygomaticus* (pipi kiri) [6] : *Left Frontalis* (dahi kiri)

Aspek	Deskripsi