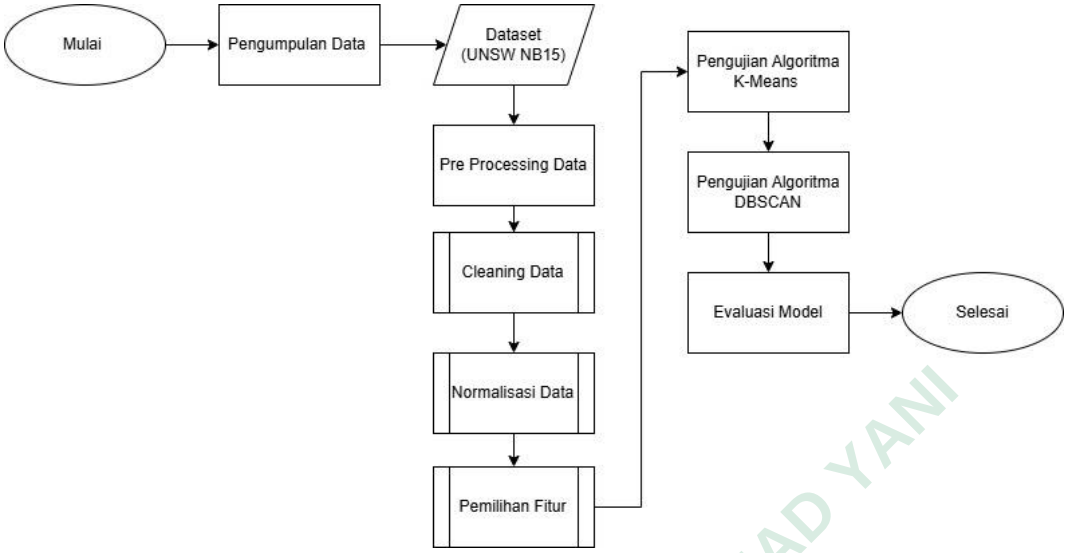
BAB 3
METODE PENELITIAN

Penelitian ini menggunakan metode kombinasi algoritma yang bertujuan untuk mengevaluasi efektivitas algoritma K-Means dan DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) dalam mendeteksi anomali pada sistem *Intrusion Detection System* (IDS). Metode kombinasi yang digunakan bersifat sekuensial, yaitu algoritma diterapkan secara berurutan untuk meningkatkan akurasi dan efisiensi deteksi. Dalam hal ini, proses diawali dengan algoritma K-Means untuk mengelompokkan data berdasarkan pola perilaku jaringan. Hasil klasterisasi dari K-Means kemudian digunakan sebagai input untuk algoritma DBSCAN, yang bertugas mengidentifikasi data yang menyimpang sebagai anomali berdasarkan kepadatan data.

Untuk menentukan konfigurasi *cluster*, penelitian ini menggunakan matrik evaluasi *Silhouette Score*. Metrik ini digunakan untuk mengukur seberapa baik suatu objek cocok dengan klasternya sendiri dibandingkan dengan klaster lain, serta membantu dalam menentukan jumlah *cluster* (k) yang paling sesuai pada algoritma K-Means. Evaluasi kinerja model dilakukan dengan menghitung nilai *false positive* untuk mengukur efektivitas klasterisasi yang dilakukan oleh kedua algoritma.

Dataset yang digunakan dalam penelitian ini adalah UNSW-NB15, yang diperoleh dari situs resmi *Kaggle.com*. Dataset ini menyajikan data serangan siber terkini yang *representatif* untuk pengujian sistem IDS. Adapun tahapan penelitian meliputi: pengumpulan data, pre processing data, pengujian algoritma, evaluasi model dalam mendeteksi anomali jaringan.Tahapan dalam penelitian ini yang akan dilakukan oleh penulis terdiri dari beberapa langkah. Berikut adalah flowchart tahapan penelitian pada Gambar 3. 1.



Gambar 3.1 *Flowchart* Alur Penelitian

1) Tahap Persiapan

a) Pengumpulan Data

Tahap awal dalam penelitian ini adalah proses pengumpulan data menggunakan dataset sekunder yang diambil dari situs resmi *Kaggle.com*. Dataset yang digunakan adalah UNSW-NB15, yang merupakan salah satu dataset modern dan komprehensif dalam bidang keamanan jaringan. Dataset ini dikembangkan oleh *Australian Centre for Cyber Security* (ACCS) dan berisi rekaman lalu lintas jaringan yang terdiri dari aktivitas normal serta sembilan jenis serangan siber, salah satunya adalah *Denial of Service* (DoS), yang menjadi fokus utama dalam penelitian ini. Pemilihan dataset sekunder dilakukan karena data tersebut telah tersusun secara terstruktur. Keunggulan lainnya adalah cakupan datanya yang luas dan realistis, sehingga sangat mendukung pengembangan model deteksi intrusi. Setelah proses pengunduhan selesai, bagian data pelatihan (*training*) dan data pengujian (*testing*) dari dataset ini digabungkan menjadi satu kesatuan untuk memastikan bahwa seluruh variasi lalu lintas jaringan dapat dianalisis secara menyeluruh. Proses penggabungan ini penting dalam pendekatan

unsupervised learning, karena metode ini tidak memerlukan label kelas dalam pelatihan model untuk mendeteksi anomali.

2) Tahap *Pre Processing* Data

Tahap *preprocessing* yang dilakukan setelah mendapatkan data yang relevan dengan pengolahan data yang bertujuan untuk memastikan bahwa data yang digunakan dalam proses pengujian memiliki kualitas yang baik. Proses ini dilakukan untuk meningkatkan akurasi model serta mengurangi potensi dalam data. Tahapan preprocessing mencakup beberapa langkah, diantaranya:

a) *Cleaning* Data

Proses cleaning data dilakukan sebagai tahap awal untuk memastikan bahwa data yang digunakan dalam penelitian ini berada dalam kondisi yang layak untuk dianalisis. *Cleaning* data merupakan proses identifikasi dan perbaikan terhadap ketidaksesuaian atau kesalahan yang terdapat dalam dataset. Langkah-langkah yang dilakukan meliputi:

1. Menghapus data yang duplikat, untuk menghindari bias dalam proses klasterisasi.
2. Mengisi atau menghapus data yang hilang (*missing values*), terutama pada fitur numerik yang akan digunakan dalam algoritma K-Means dan DBSCAN.
3. Mengubah tipe data menjadi numerik, karena algoritma yang hanya dapat memproses data numerik.

b) Normalisasi Data

Normalisasi data yang bertujuan untuk menyamakan skala antar fitur sehingga tidak ada satu fitur pun yang memiliki pengaruh dominan dalam perhitungan jarak pada algoritma klasterisasi. Hal ini penting karena perbedaan skala nilai antar fitur dapat menyebabkan bias dalam proses pengelompokan data, di mana fitur dengan nilai yang lebih besar cenderung memiliki bobot lebih tinggi. Dalam penelitian ini, proses normalisasi dilakukan menggunakan

metode *Min-Max Scaling*, yaitu teknik yang nilai setiap fitur ke dalam rentang [0, 1].

c) Pemilihan Fitur

Pemilihan fitur merupakan tahap penting dalam proses deteksi anomali, karena tidak semua fitur dalam dataset memiliki kontribusi yang signifikan terhadap identifikasi serangan. Dalam penelitian ini, dilakukan seleksi fitur berdasarkan analisis korelasi antar variabel serta relevansinya terhadap pola serangan *Denial of Service* (DoS). Fokus diarahkan pada serangan DoS karena jenis serangan ini memiliki pola perilaku jaringan yang cukup unik dan berbeda secara signifikan dari lalu lintas normal, seperti volume data yang tinggi dan aktivitas yang berulang dalam waktu singkat. Oleh karena itu, dipilih lima fitur numerik utama yang dianggap paling representatif dalam menggambarkan aktivitas jaringan saat terjadi serangan DoS.

3) Tahap Pengujian Algoritma

a) Algoritma K-Means

Proses pembentukan *cluster* dalam penelitian ini dimulai dengan penerapan algoritma K-Means. Algoritma K-Means merupakan salah satu metode *unsupervised learning* yang digunakan untuk mengelompokkan data berdasarkan kemiripan. Pada penelitian ini, algoritma tersebut dimanfaatkan untuk mengelompokkan data serangan jaringan dengan tujuan mengidentifikasi pola-pola perilaku yang mirip. Hasil pengelompokan bergantung pada pemilihan jumlah *cluster* (nilai k) yang tepat. Berikut ini cara kerja algoritma K-Means yang digunakan dalam penelitian ini:

1. Menentukan jumlah *cluster* (nilai k):

   Langkah pertama adalah menetapkan berapa banyak kelompok (*cluster*) yang ingin dibentuk. Dalam penelitian ini, dicoba beberapa pilihan jumlah *cluster*, yaitu dari k = 2 hingga k = 8, untuk mencari jumlah yang paling tepat.

2. Menentukan titik pusat awal (*centroid*) secara acak:
   Setelah jumlah *cluster* ditentukan, algoritma akan memilih titik pusat awal untuk masing-masing *cluster*. Titik ini disebut *centroid*, dan pada tahap awal, titik-titik ini dipilih secara acak dari data yang tersedia.
3. Mengelompokkan data ke *centroid* terdekat:
   Selanjutnya, setiap data akan dihitung jaraknya ke semua *centroid* yang sudah ditentukan. Data kemudian dimasukkan ke dalam kelompok (c*luster*) yang memiliki pusat terdekat dengannya.
4. Menghitung ulang posisi *centroid*:
   Setelah semua data dikelompokkan, posisi setiap *centroid* akan diperbarui. Caranya adalah dengan menghitung rata-rata dari seluruh titik data yang ada dalam satu *cluster*. Posisi *centroid* baru ini mewakili pusat yang lebih akurat untuk *cluster* tersebut.
5. Mengulang proses hingga stabil:
   Langkah pengelompokan dan pembaruan posisi *centroid* akan terus diulang hingga posisi semua *centroid* tidak berubah lagi secara signifikan. Ini menandakan bahwa proses sudah mencapai kondisi stabil.
6. Menilai kualitas *cluster* dengan *Silhouette Score*:
   Setelah proses stabil, dilakukan evaluasi terhadap hasil pengelompokan dengan menggunakan *Silhouette Score*. Skor ini mengukur seberapa mirip data dalam satu *cluster* dan seberapa jauh perbedaannya dengan data dari *cluster* lain. Semakin tinggi nilainya, semakin baik hasil pengelompokannya.
7. Memilih nilai k terbaik berdasarkan skor tertinggi:
   Dari beberapa nilai k yang diuji, dipilih nilai k dengan *Silhouette Score* paling tinggi. Nilai ini dianggap sebagai jumlah *cluster* terbaik untuk membagi data.

Hasil dari *cluster* yang dibentuk oleh K-Means ini kemudian digunakan sebagai dasar awal sebelum masuk ke tahap pendeteksian anomali menggunakan algoritma DBSCAN.

b) Algoritma DBSCAN

Setelah proses pengelompokan awal dilakukan menggunakan algoritma K-Means, langkah selanjutnya dalam penelitian ini adalah menerapkan DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*). Algoritma ini digunakan untuk mendeteksi anomali secara lebih spesifik, terutama untuk mengenali serangan jaringan seperti DoS (*Denial of Service*) yang muncul sebagai penyimpangan dari pola data utama. DBSCAN bekerja dengan melihat kepadatan data, bukan jarak rata-rata, sehingga lebih efektif dalam mengenali data yang tidak sesuai pola. Berikut ini cara kerja algoritma DBSCAN yang digunakan dalam penelitian ini:

1. Menentukan parameter *eps* (ε):

   Nilai *epsilon* (ε) adalah jarak maksimum antar dua titik yang masih dianggap berada dalam satu lingkungan. Nilai ini ditentukan menggunakan metode *elbow* dari grafik *k-distance* (jarak ke tetangga ke-k), dengan mengambil rata-rata dari nilai k terbaik yang diperoleh pada tahap K-Means. Tujuannya agar nilai ε sesuai dengan distribusi data yang telah dinormalisasi.

2. Menentukan parameter *minPts* (minimum points):

   Nilai *minPts* adalah jumlah minimum titik dalam radius ε agar suatu titik bisa dianggap sebagai pusat (inti) dari *cluster*. Dalam penelitian ini, nilai *minPts* = 5 digunakan karena merupakan nilai yang umum.

3. Membentuk *cluster* berdasarkan kepadatan:

   DBSCAN memulai prosesnya dengan mengambil satu titik data, lalu menghitung berapa banyak titik tetangga yang berada dalam radius ε (*epsilon*). Jika jumlah titik tetangga tersebut sama dengan atau lebih dari *minPts*, maka titik tersebut dianggap sebagai titik inti dan mulai membentuk sebuah *cluster*. Sebaliknya, jika jumlah titik

tetangga kurang dari *minPts*, maka titik tersebut dianggap sebagai noise atau anomali, dan diberi label -1. Proses ini kemudian dilanjutkan ke titik-titik tetangga lainnya dan diulang hingga semua titik dalam dataset telah diperiksa.

4. Mendeteksi dan memberi label anomali:

   Titik-titik yang tidak masuk ke dalam *cluster* mana pun dianggap sebagai anomali. Dalam kasus ini, biasanya serangan DoS muncul sebagai titik yang menyimpang dan tidak mengikuti pola utama data.

5. Evaluasi hasil deteksi anomali:

   Hasil akhir menunjukkan pemisahan antara data normal (yang tergabung dalam *cluster* padat) dan data anomali (label -1). Hal ini membantu dalam analisis lebih lanjut terhadap pola serangan jaringan.

Dengan parameter ini, DBSCAN dapat secara efektif membedakan antara *cluster* padat dan titik-titik yang tidak sesuai pola umum, sehingga mampu memisahkan data normal dari anomali secara akurat, khususnya dalam serangan DoS yang kerap muncul sebagai penyimpangan dari distribusi data utama.

4) Tahap Evaluasi Model

Tahap evaluasi model ini bertujuan untuk mengukur efektivitas algoritma K-Means dan DBSCAN dalam mendeteksi anomali. Dengan ini *unsupervised learning* yang digunakan dalam matrik evaluasi diantaranya:

1. *Silhouette score*

   Silhouette score digunakan untuk mengevaluasi model K-Means dan DBSCAN dengan menggunakan rumus pada persamaan (1) berikut :

$$S(i) = \frac{b(i) - a(i)}{max\big(a(i), b(i)\big)} \quad (1)$$

   $a(i)$ = rata-rata jarak antara suatu titik dan semua titik dalam *cluster*nya.

$b(i)$ = rata-rata jarak antara suatu titik dan semua titik dalam *cluster* terdekat.

2. Tahap Perhitungan *False Positive Rate*

Setelah proses klasterisasi dan deteksi anomali selesai dilakukan, tahap selanjutnya adalah mengevaluasi kinerja model dalam mengidentifikasi data yang menyimpang. Evaluasi ini dilakukan dengan menghitung *False Positive Rate* (FPR), yaitu persentase data normal yang secara keliru diklasifikasikan sebagai anomali. Pengukuran FPR sangat penting dalam konteks sistem deteksi intrusi, karena tingkat kesalahan yang tinggi dalam mendeteksi aktivitas normal sebagai ancaman (*false alarm*) dapat menurunkan efektivitas dan efisiensi sistem secara keseluruhan. Kondisi ini tidak hanya membebani sistem dengan peringatan yang tidak relevan, tetapi juga berpotensi mengabaikan deteksi terhadap ancaman yang sebenarnya.

Rumus *False Positive Rate* pada persamaan (2) berikut:

$$FPR = \frac{FP}{FP + TN} \tag{2}$$

FP (*False Positive*) = Jumlah data normal yang salah diklasifikasikan sebagai anomali.

TN (*True Negative*) = Jumlah data normal yang diklasifikasikan dengan benar sebagai normal.

3.1 Bahan Penelitian

Dalam penelitian ini, data yang digunakan yaitu data sekunder dari dataset UNSW-NB15, yang diperoleh dari situs resmi *Kaggle.com.* dataset ini dikembangkan oleh tim peneliti IXIA *PerfectStorm di Cyber Range Lab*, UNSW Canberra untuk mensimulasikan lalu lintas jaringan yang mencakup trafik normal maupun berbahaya. Selain itu, dataset ini terdapat sembilan jenis serangan siber yang berbeda, yang digunakan dalam proses deteksi dan analisis anomali pada sistem keamanan jaringan.

3.2 Alat Penelitian

Untuk mendukung pelaksanaan penelitian, diperlukan alat yang digunakan dalam penelitian ini adalah laptop dengan spesifikasi yang cukup untuk menjalankan sistem operasi dan perangkat lunak pengembangan serta koneksitas Internet. Berikut sistem operasi dan software yang digunakan untuk menunjang penelitian sebagai berikut :

1. Sistem Operasi: Operasi Windows 11 64-bit.
2. Processor: AMD A6 M509BA-HD621
3. RAM: 12 GB
4. Storage: SSD 256 GB
5. Software: Google Colaboratory

3.3 Jalan Penelitian

Jalan penelitian dalam penelitian ini mencakup penjelasan singkat dalam proses yang diambil selama pelaksanaan penelitian. Berikut ini adalah penjelasan dari setiap tahapan jalannya penelitian ini :

1) Tahap Persiapan
   a) Pengumpulan Data

      Menggunakan dataset sekunder UNSW-NB15 yang diunduh dari situs Kaggle. Mencakup lalu lintas jaringan normal serta sembilan jenis serangan siber, termasuk serangan *Denial of Service* (DoS) yang menjadi fokus penelitian. Dataset ini memiliki cakupan data yang luas, memungkinkan analisis yang menyeluruh terhadap lalu lintas jaringan. Setelah data diunduh, bagian data training dan testing digabung menjadi satu untuk mendukung pendekatan *unsupervised learning*, yang tidak memerlukan label kelas saat membangun model deteksi anomali.

2) *Preprocessing* Data

   Setelah mendapatkan data, dilakukan *preprocessing* untuk meningkatkan kualitas data sebelum dianalisis. Tujuan dari tahap ini adalah memastikan data bersih, relevan, dan sesuai format yang

dibutuhkan algoritma. Tahap preprocessing mencakup tiga langkah penting: *cleaning data*, normalisasi data, dan pemilihan fitur. Proses ini sangat penting untuk meningkatkan akurasi model.

3) Tahap Pengujian Algoritma

a) Algoritma K-Means, dilakukan untuk mengelompokkan data berdasarkan kemiripan pola. Penentuan jumlah *cluster* terbaik (nilai *k*) dilakukan menggunakan metrik *Silhouette Score* yang mengukur seberapa baik setiap titik berada di dalam *cluster* yang tepat. Nilai *k* dengan *Silhouette Score* tertinggi dipilih karena menunjukkan struktur *cluster* yang paling sesuai. Hasil klasterisasi ini menjadi dasar untuk pengujian lanjutan menggunakan DBSCAN.

b) Algoritma DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*), Setelah menggunakan K-Means, dilakukan pengujian lanjutan dengan algoritma DBSCAN yang lebih unggul dalam mendeteksi anomali karena berbasis kepadatan. DBSCAN menggunakan dua parameter utama, yaitu *eps* (radius jarak maksimum antar titik) dan *minPts* (jumlah minimum titik untuk membentuk *cluster*). DBSCAN efektif mendeteksi data *noise* atau outlier yang tidak dapat ditangkap oleh K-Means, sehingga kedua metode saling melengkapi.

4) Tahap Evaluasi Model

Evaluasi dilakukan untuk mengukur efektivitas model dalam mendeteksi anomali menggunakan pendekatan *unsupervised learning*. Dua matrik evaluasi utama digunakan, yaitu *False Positive Rate* (FPR) dan *Silhouette Score*. Evaluasi ini penting untuk memastikan bahwa model mampu membedakan aktivitas normal dan anomali dengan baik.

a) Silhouette Score

Digunakan untuk mengevaluasi kualitas pembentukan *cluster* pada algoritma K-Means dan DBSCAN. Nilai *Silhouette Score* yang tinggi menunjukkan bahwa data dikelompokkan dengan baik dan masing-masing *cluster* memiliki kejelasan batas yang tinggi.

b) Perhitungan *False Positive Rate*

False Positive Rate dihitung untuk menilai tingkat kesalahan model dalam mengklasifikasikan data normal sebagai anomali. Nilai FPR yang tinggi menunjukkan banyaknya false alarm, yang dapat mengurangi keandalan sistem deteksi intrusi. Evaluasi ini penting agar sistem IDS tidak memberikan peringatan yang tidak akurat.