BAB III
METODE PENELITIAN

3.1 Subjek dan Objek Penelitian

Subjek pada penelitian ini adalah serangan DDoS sebagai ancaman keamanan jaringan komputer yang dibahas dalam penelitian ini.

Objek pada penelitian ini adalah deteksi serangan DDoS, penelitian ini difokuskan dengan pengembangan dari dua *Machine Learning* yaitu *Random Forestc* dan KNN untuk secara efektif mengidentifikasi serangan DDoS.

3.2 Alat dan Bahan

3.2.1 Alat

Penelitian ini menggunakan perangkat keras dan perangkat lunak yang diharapkan dapat membantu dalam menyelasaikan penelitian ini.

1. Perangkat Keras (*Hardware)*

Device	Lenovo LOQ
Processor	Intel(R) Core(TM) i7-13650HX 2.60 GHz
RAM	8,0 GB

2. Perangkat Lunak (*Software*)

Sistem Operasi	*Windows 10 Pro 64-bit*
Bahasa Pemrograman	*Python*
Aplikasi	*Visual Studio Code*

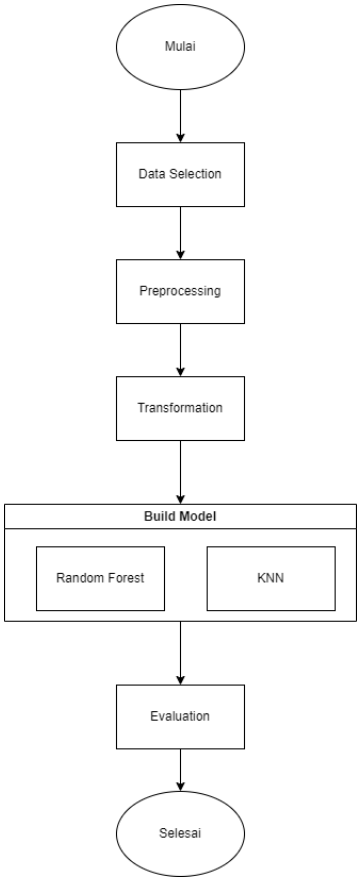
3.2.2 Bahan

Bahan yang digunakan penelitian ini merupakan *dataset* dengan nama CICDDoS2019 adalah penyerangan DDoS menggunakan protkcol TCP/UDP dimana serangan dibagi menjadi 2 serangan DDoS berbasis Refleksi dan Eksploitasi, *dataset* diperoleh dari website *university of new Brunswick. Dataset* yang didapatkan berformat CSV dengan features lebih dari 80 dan

macam-macam jenis serangan dari DDoS, yang digunakan pada penelitian jenis serangan DDoS TFTP dengan kapasitas file sebesar 8.66GB.

3.3 Diagram Alir Penelitian

Penelitian ini dilakukan untuk membandingkan hasil akurasi dari *Machine Learning Random Forest* dan *K-Neurest Network* (KNN) dalam mendeteksi serangan DDoS. Proses penelitian ini digambarkan pada diagram alir yang ada pada Gambar 3.1.



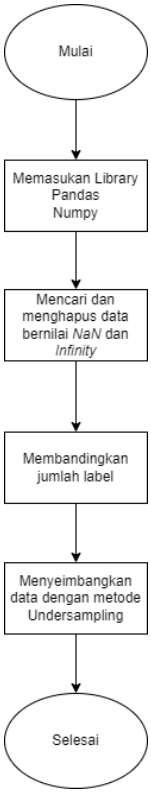
Gambar 3. 1 Diagram Alir Penelitian

3.3.1 Data Selectiom

Penelitian ini menggunakan data yang didapat dari *website universitas of new burnswick* dengan 2 data yang diunduh dengan masing-masing memiliki kapasitas 2.2 Gb dan 890 MB. Data bernama CICDDoS2019 berformat *.csv* berisi macam-macam serangan DDoS yang dilakukan dan memiliki kurang lebih 88 *features* dari masing-masing file.

Pemilihan data yang dilakukan agar tidak semua data yang diunduh digunakan hanya satu macam serangan DDoS yang bernama DrDoS_DNS dengan ukuran file sebesar 1.98GB dan memiliki sebanyak 88 features. Alasan menggunakan file ini dikarenakan menurut *website Cloud Flare* jenis penyerangan yang paling umum digunakan pada tahun 2023 adalah DNS. *Dataset* CICDDOS2109 memiliki label yang terkait dengan penelitian ini berupa label Benign dan DrDoS_DNS. Label Benign mengartikan lalu lintas jaringan normal sedangkan DrDoS_DNS mengartikan lalu lintas jaringan sedang diserang.

3.3.2 Preprocessing

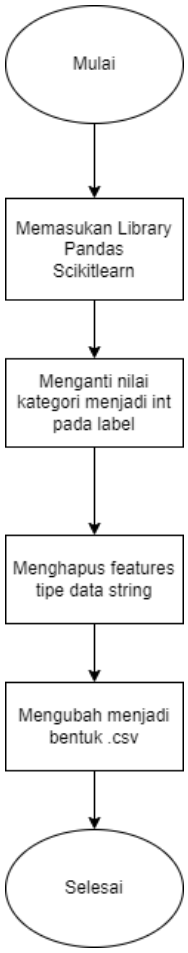


Gambar 3. 2 Diagram Alir Preprocessing

Menganalisis kekosongan data , data yang kosong atau hilang akan dihapus dari *dataset* guna untuk menghindari eror pada saat pemodelan. Menghapus nilai *NaN* dan *Infinity* pada data, agar akurasi dari model dapat meningkat. Untuk mendeteksi apakah ada nilai *NaN* dan *Infinity* dengan fungsi *isna.any()* untuk nilai *NaN* serta *isin.any()* untuk nilai *infinity*. Data dicek

apakah data serangan dan data normal memiliki jumlah data yang sama jika tidak sama maka menggunakan *Undersampling* dengan memanggil library *imbalance learn* dengan memisahkan label dan *features* pada *dataset* untuk menyeimbangkan data serangan dan data normal.

3.3.3 Transformation



Gambar 3. 3 Diagram Alir Transformation

Pada tahap *transformation* akan menghapus beberapa *features* yang tidak dapat digunakan dalam pemodelan dikarenakan tipe data yang tidak bisa diolah oleh *Machine Learning* yaitu tipe data *string*. Dikarenakan label merupakan tipe data *string* maka sebelum menghapus kolom *features* yang tidak diperlukan dengan menganti nilai nya menjadi 0 dan 1. Nilai 0 mewakili *Benign* atau bukan penyerangan dan nilai 1 mewakili *DrDoS_DNS* atau penyerangan.

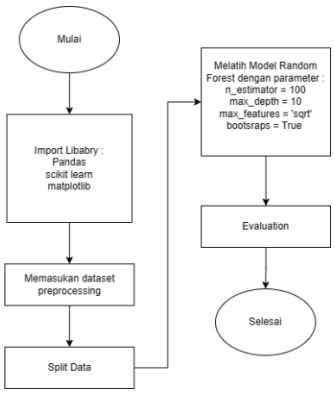
Pada saat menghapus *features* ada dua metode pada penelitian ini yang pertama tanpa menggunakan *selection features* dan kedua menggunakan *selection features* menggunakan metode *Information Gain* dengan menampilkan *score*. Menghapus *features* untuk tipe data *string* dan simpan *dataset* yang telah di *preprocessing* dan *transformation*.

Selanjutnya melakukan *selection features* menggunakan metode *Information Gain* dan melakukan *selection features* sendiri dengan referensi jurnal yang berlandaskan dari *information gain score*, lalu simpan *dataset* dengna nama yang berbeda. Alasan mengapa meyeleksi *features* agar dapat meringankan beban dari perangkat sehingga diharapkan pada penelitian selanjutnya bisa melakukan pendeteksian tanpa harus menggunakan semua *features* yang bersumber dari CICDDOS2019 khususnya *dataset* bernama *DrDoS_DNS*.

3.3.4 Build Model

Pada tahap *build model* untuk *Machine Learning* memiliki tahap yang berbeda dalam persiapan sebelum melatih model yang harus menyesuaikan dengan model agar model dapat memperdalam dalam mempelajari *dataset* yang diberikan. Model yang dibangun memnggunakan beberepa parameter yang disesuaikan agar mendapatkan akurasi yang tinggi yang akan dijelaskan pada Pembangunan dari masing-masing model.

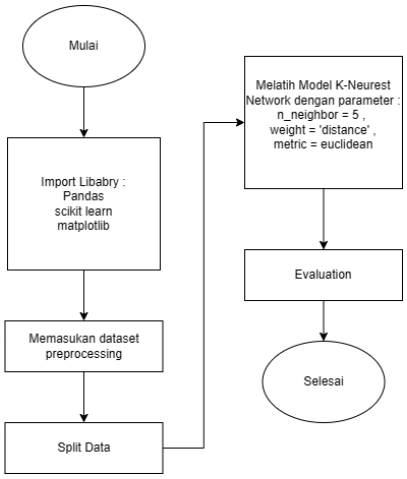
3.3.4.1 Random Forest



Gambar 3. 4 Diagram Alir *Random Forest*

Pembangunan model *Random Forest* dengan memanggil *library* yang dibutuhkan untuk membangun model yaitu *library pandas*, *scikitlearn*, dan *matplotlib*. Memasukan *dataset* yang sudah di *preprocessing* dan *transformation*. Memisahkan data latih dan data uji dengan perbandingan 80% untuk data latih dan 20% untuk data uji. Membangun model dengan memanggil model terlebih dahulu dari *library scikit learn*. Lalu menentukan parameter nya yaitu *n_estimators*, *max_depth*, *max_features,* dan *bootstraps. n_estimators* memiliki fungsi menentukan jumlah pohon, max_depth memiliki fungsi kedalaman pohon dalam mempelajari data, *max_features* memiliki fungsi sebagai jumlah fitur yang digunakan pada setaip node, dan *bootstraps* untuk berfungsi untuk menanggani *overfitting*.

3.3.4.2 K-Neurest Neighbor



Gambar 3. 5 Diagram Alir K-Neurest Network

Pembangunan model KNN menggunakan *library pandas*, *scikit-learn*, dan *matplotlib*, setelah itu memasukan *dataset* dengan *libarary pandas* dan lakukan *split* data dengan perbandingan data uji 80% dan data latih 20%. Memanggil model *K-Neurest-Network* dengan library dari *scikit learning*, setelah itu menentukan parameter nilai K, *weight*, dan parameter jarak (*metric*). Menentukan nilai K pada pembangunan model memiliki fungsi agar model dapat mengerti data yang diberikan dan dan seberapa responsif model saat data diberikan. *Weight* pada penelitian ini menggunakan *distance* dikarenakan jarak

antar titik dari *dataset* berbeda-beda. Parameter jarak menggunakan *Euclidean* berfungsi agar dapat menghitung jarak antar titik.

3.3.5 Evaluation

Tahap *evaluation* adalah tahapan untuk melakukan pengukuran data mining yaitu pengukuran kinerja dari KNN dan *Random Forest* yang sudah melalui proses training maupun proses testing menggunakan *dataset* serangan DrDoS_DNS. Evaluasi yang dilakukan berasal dari perhitungan confusion matrix yang dihitung berupa nilai *accuracy*, *precision*, *recall*, dan *f1-score*. Pada tahap evaluation memiliki 2 pelatihan yaitu pelatihan menggunakan *selection features Information Gain* dan pelatihan tanpa menggunakan *selection features Information Gain*.

3.3.5.1 Random Forest

Tahap evaluasi dengan memprediksikan nilai dari data uji setelah model *random forest* dilatih, yang akan menghasilkan nilai *Accuracy*, *Precision*, *Recall*, dan *F-1 Score*.

3.3.5.2 K-Neurest Network

Tahap evaluasi dengan memprediksikan nilai data uji setelah model telah dilatih dengan parameter yang ditentukan. Model yang telah dilatih akan di uji dengan data uji yang telah di *split*. Hasil dari evaluasi memberikan nilai *Accuracy*, *Precision*, *Recall*, dan *F-1 Score*.