Berdasarkan gambar 4.2, menunjukkan hasil ekstraksi fitur yang dilakukan dengan metode PCA dan dapat dilihat metode ini berhasil mereduksi dimensi fitur dari 46 menjadi 10 fitur dengan cara mengambil fitur dengan varians yang paling tinggi. Hasil ekstraksi fitur ini digunakan untuk mengolah *data training* IDS menggunakan CNN. Pada proses ini, PCA berhasil mengekstraksi fitur sehingga menghasilkan range nilai 0 sampai 4.

4.2 ANALISIS DATA DENGAN CNN

Pada penelitian ini menggunakan parameter *epoch* 10 dan 50 untuk model dan pelatihan model. Berikut adalah model CNN yang dapat dilihat pada gambar 4.3:

```
Model: "Cnn-Lstm_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_1 (Conv1D)           (None, 10, 32)            320

 max_pooling1d_1 (MaxPoolin  (None, 5, 32)             0
 g1D)

 lstm_1 (LSTM)               (None, 512)               1116160

 dense_1 (Dense)             (None, 2)                 1026

=================================================================
Total params: 1117506 (4.26 MB)
Trainable params: 1117506 (4.26 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Gambar 4. 3 Membangun Model CNN

Berdasarkan gambar 4.3, dapat dilihat bahwa hasil dari model "Cnn-Lstm_model" terdiri dari 4 *layer* atau lapisan. Lapisan pertama adalah lapisan *Conv1D*. Lapisan ini menggunakan filter 32 untuk mengonversi data input menjadi data output dengan bentuk (None, 10, 32). Lapisan kedua adalah lapisan *MaxPooling1D*. Lapisan ini menggunakan jendela *pooling* 2 untuk mengurangi

ukuran data *output* dari lapisan pertama menjadi (None, 5, 32). Lapisan ketiga adalah lapisan LSTM. Lapisan ini menggunakan 512 unit LSTM untuk memproses data output dari lapisan kedua. Lapisan keempat adalah lapisan *Dense*. Lapisan ini menggunakan 2 unit *neuron* untuk menghasilkan output akhir dari model. Dan total dari keseluruhan parameter *Cnn-lstm*_model berjumlah 1117506 parameter.

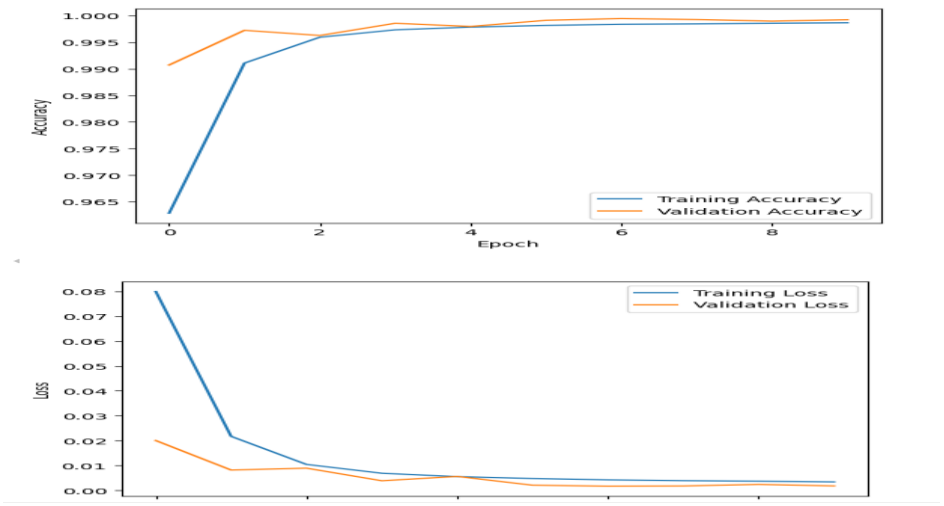
4.2.1 Deteksi Menggunakan *Epoch* 10

Pelatihan model yang pertama menggunakan *epoch* (iterasi) 10 sebagai parameter. Berikut adalah pelatihan model menggunakan parameter *epoch* 10 yang dapat dilihat pada gambar 4.4:

Gambar 4. 4 Pelatihan Model CNN Epoch 10

Berdasarkan gambar 4.4, model "*Cnn-Lstm_model*" dilatih selama 10 *epoch*. Pada *epoch* pertama, *loss model* adalah 0.1683 dan *accuracy* adalah 0.9141. Selama pelatihan akurasi mengalami peningkatan. Dan dapat disimpulkan bahwa model "*Cnn-Lstm_model*" mengalami peningkatan performa selama pelatihan. Hal ini terlihat dari penurunan nilai *loss* dan peningkatan nilai *accuracy*. Tujuan pelatihan adalah untuk mengurangi nilai fungsi *loss* dan meningkatkan akurasi pada kedua jenis data.

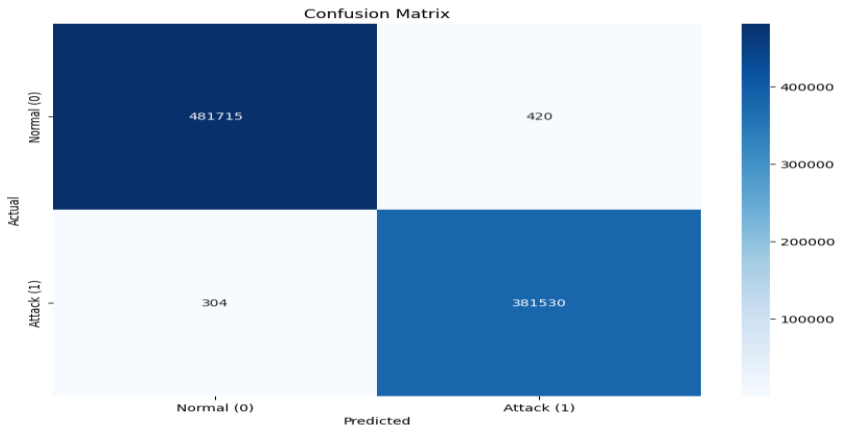
Berikut adalah visualisasi dari hasil *loss* dan *accuracy* dataset 1 pada pelatihan model CNN menggunakan *epoch* 10 dapat dilihat pada gambar 4.5:



Gambar 4. 5 Visualisasi Loss dan Akurasi Epoch 10

Berdasarkan gambar 4.5, dapat dilihat bahwa *loss model* dan *validation loss* mengalami penuruan secara signifikan yang artinya *epoch* 10 mampu mengurangi *loss* (kerugian) pada model dan akurasi meningkat dari 0.0035 menjadi 0.9986.

Berikut adalah gambar *confusion matrix* yang diperoleh dari hasil perhitungan dataset 1 yang dapat dilihat pada gambar 4.6:



Gambar 4. 6 Confusion Matrix Epoch 10

Berdasarkan gambar 4.6, dapat dilihat bahwa hasil dari *confusion matrix* pada *epoch* 10 mendapatkan hasil *True Positive* (TP) dengan nilai 481715, hasil *True Negative* (TN) dengan nilai 381530 hasil *False Positive* (FP) dengan nilai 420 dan hasil *False Negative* (FN) dengan nilai 304.

Kemudian hasil *confusion matrix* dari dataset dikumpulkan ke dalam tabel yang dapat dilihat pada tabel 4.3:

Tabel 4. 3 Hasil Confusion Matrix Seluruh Data (Epoch 10)

No	File	Jumlah Paket
1	phase1_NetworkData.csv	12062397
2	phase1_NetworkData_sampled.csv	2412481
3	phase2NetworkData.csv	9536824
4	phase2_NetworkData_sampled.csv	1907366

Berdasarkan tabel 4.3, dapat simpulkan bahwa nilai *confusion matrix* pada *True Positive* (TP) yang tertinggi adalah dataset pertama dengan nilai 12062397. TP disini mewakili kasus di mana sistem dengan benar memprediksi bahwa *traffic* tersebut adalah serangan DDoS.

Setelah mendapatkan hasil *confusion matrix*, maka didapatkan juga hasil performa model CNN dari dataset yang telah dikumpulkan dalam tabel 4.4:

Tabel 4. 4 Hasil Performa Model CNN (Epoch 10)

Metric	Value
Accuracy	0.999162007
Precision	0.9991620699
Recall	0.999162007
F1 Score	0.9991620202