

**TUGAS 1**  
**KLASIFIKASI PADA DATASET CIFAR10 DENGAN CNN**

**SEBAGAI TUGAS EKSPLORASI PADA**  
**MATA KULIAH PEMBELARAN MESIN LANJUT**

Oleh  
**RAHMAN INDRA KESUMA**  
**NIM: 33221026**  
**(Program Studi Doktor Teknik Elektro dan Informatika)**



**INSTITUT TEKNOLOGI BANDUNG**  
**Maret 2022**

## Beberapa Tahapan Utama pada Ekplorasi dan Percobaan

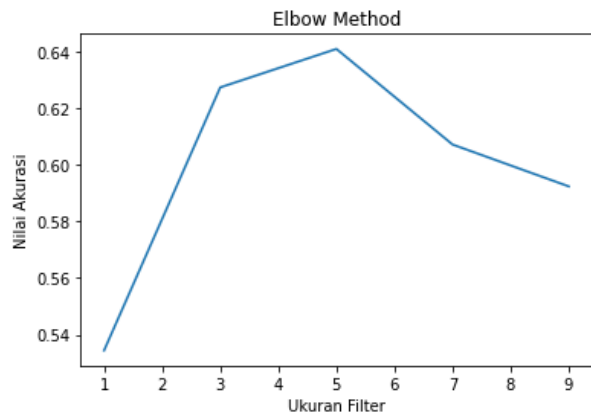
1. Melakukan pembagian dataset dalam tiga kelompok data, (i) data training untuk menghasilkan model pembelajaran; (ii) data validation digunakan untuk memiliki model dengan setting hyperparameter terbaik; serta (iii) data testing untuk menguji kualitas kinerja dari model dengan setting hyperparameter terbaik. Pembagian dataset tersebut seperti terlihat pada gambar berikut.

```
# Melakukan pembagian dari data testing menjadi dua kelompok data, yaitu validation data dan testing data
valid_images, test_images, valid_labels, test_labels = train_test_split(remaining_images, remaining_labels, test_size=0.5)

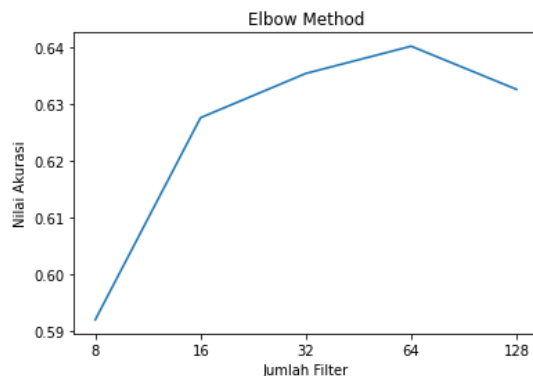
print("Ukuran Training Data -> ", train_images.shape)
print("Ukuran Validation Data -> ", valid_images.shape)
print("Ukuran Testing Data -> ", test_images.shape)
```

Ukuran Training Data -> (50000, 32, 32, 3, 1)  
Ukuran Validation Data -> (5000, 32, 32, 3, 1)  
Ukuran Testing Data -> (5000, 32, 32, 3, 1)

2. Proses eksplorasi dimulai dengan mencari ukuran filter terbaik pada convolutional layer yang digunakan. Arsitektur awal yang digunakan adalah jumlah conv layer adalah 1, jumlah filter sebanyak 32, dan jumlah hidden unit pada 1 hidden layer adalah 128. Berikut hasil percobaan pencarian ukuran filter yang meliputi percobaan ukuran 1x1, 3x3, 5x5, dan 9x9. Berdasarkan grafik tersebut dapat terlihat bahwa ukuran filter 5x5 memiliki nilai akurasi tertinggi.

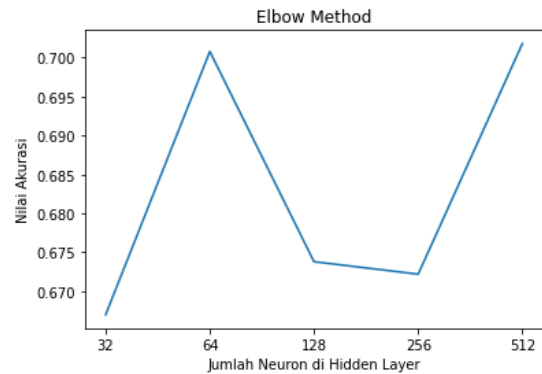


3. Proses eksplorasi dilanjutkan untuk mencari nilai jumlah filter dalam arsitektur. Yang mana dilakukan pencarian hasil terbaik dari beberapa nilai jumlah filter yaitu 8, 16, 32, 64, dan 128. Dengan menggunakan arsitektur 1 conv layer, 5x5 filter size, dan 128 hidden neuron pada 1 hidden layer, maka diperoleh hasil eksplorasi sebagai berikut. Dapat diamati bahwa jumlah filter 64 memiliki nilai akurasi terbaik.

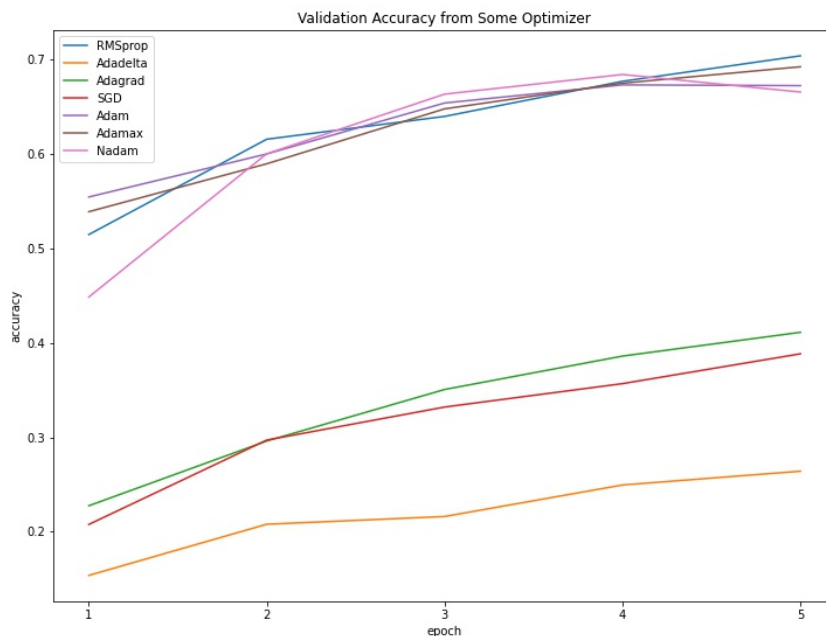


4. Selanjutnya proses eksplorasi berfokus pada pencarian jumlah conv layer untuk beberapa nilai jumlah filter. Dari eksplorasi tersebut diperoleh hasil bahwa kombinasi 2 conv layer yang memiliki jumlah neuron 128 di conv1 dan 256 di conv2 menjadi yang terbaik.

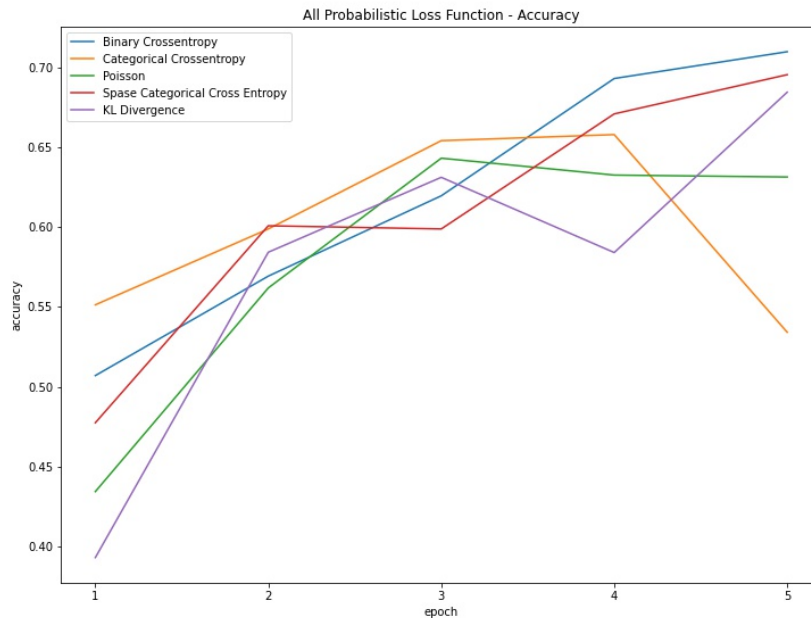
5. Proses berlanjut untuk mencari nilai jumlah neuron pada 1 hidden layer yang didefinisikan pada arsitektur CNN. Dilakukan percobaan untuk mengecek beberapa nilai jumlah neuron, meliputi 32, 64, 128, 256, dan 512. Hasil dari proses percobaan dapat terlihat pada gambar berikut, yang mana jumlah neuron 512 pada hidden layer menghasilkan nilai akurasi terbaik terhadap validation data yaitu sebesar 0,7018 (70,18%).



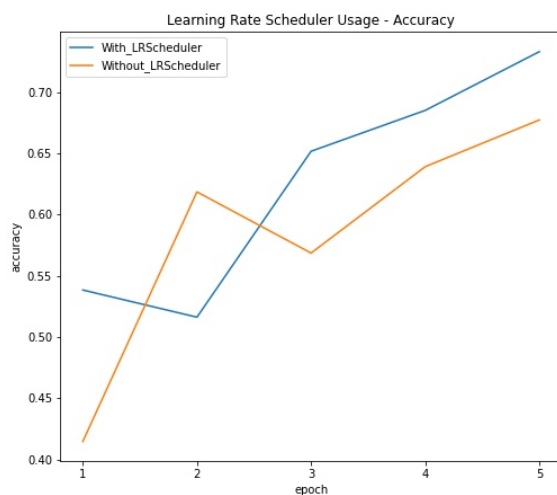
6. Percobaan dilakukan dengan melakukan pencarian metode optimizer dalam pencarian evaluasi bobot terbaik setiap akhir dari satu batch training. Dilakukan percobaan dengan beberapa optimizer yang disediakan meliputi, RMSProp, Adadelata, Adagrad, SGD with Momentum, Adam, Adamax, dan Nadam. Berdasarkan hasil yang diperoleh dan terlihat pada grafik, dapat diambil keputusan bahwa penggunaan RMSProp dinilai lebih stabil untuk menghasilkan akurasi dalam 5 epoch training.



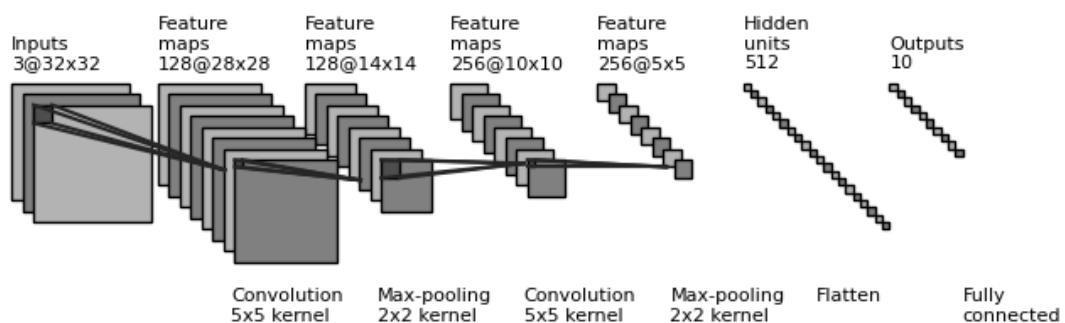
7. Percobaan dilanjutkan dengan penentuan Loss Function yang dapat meningkatkan kinerja dari arsitektur CNN yang telah diperoleh hingga percobaan terakhir. Percobaan beberapa loss function meliputi binary crossentropy, categorical crossentropy, sparse categorical crossentropy, poisson dan kl divergence. Berdasarkan hasil percobaan yang bisa diamati juga pada gambar berikut, terlihat bahwa loss function dengan binary crossentropy dinilai lebih stabil mengalami peningkatan kinerja dengan nilai terbaik untuk akurasi pada validation data sebesar 0,7102.



8. Percobaan terakhir dilakukan untuk melihat pengaruh ketika digunakannya learning rate scheduler pada proses training yang dilakukan. Percobaan membandingkan antara learning rate yang menggunakan nilai statis sebesar 0,001 dengan learning rate yang menerapkan scheduler dengan nilai awal 0,001 dan mengalami penurunan hingga mendekati nilai 0. Dari perbandingan yang dilakukan, diperoleh hasil bahwa learning rate dengan scheduler memperoleh akurasi yang stabil dan lebih tinggi dibandingkan yang tanpa scheduler.



9. Sehingga diperoleh hasil Arsitektur terbaik adalah sebagai berikut.



10. Hasil model dengan arsitektur tersebut dan proses pelatihan yang melibatkan RMSProp optimizer, Binary Crossentropy loss function, dan penggunaan learning rate scheduler dengan nilai learning rate awal 0,001 yang mengalami penurunan secara linier, diperoleh kualitasnya terhadap data testing adalah sebagai berikut.

```
# Menguji model terbaik ke testing data yang telah disediakan
hasil_predict_test = best_model.predict(test_images)
loss, acc = best_model.evaluate(test_images, onehot_test_label, verbose=1)
print('Restored model, accuracy: {:.5.2f}%'.format(100 * acc))

157/157 [=====] - 17s 106ms/step - loss: 0.1357 - accuracy: 0.7304
Restored model, accuracy: 73.04%
```

11. Berikut juga ditampilkan contoh testing data yang berhasil diklasifikasikan dengan benar dan yang tidak benar oleh model terbaik.

