# **ASL Alphabeth Classification**

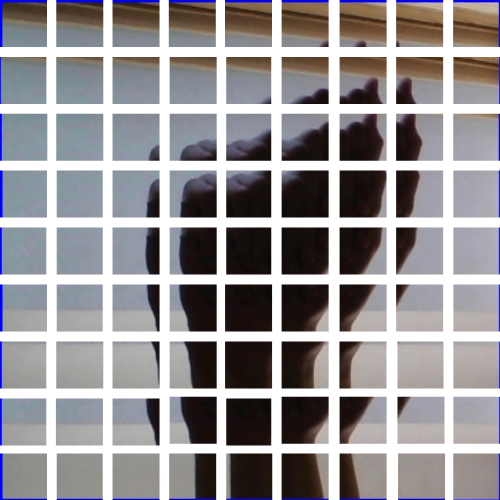
# **Anggota :**

* Kezia Angeline - C14190048 → model\_filter1 + model\_kaggle(1-3)
* Alycia Jane Sidarta - C14190056 → model\_filter3 + model\_layer + model\_epoch
* Michael Alexander - C14190048 → model\_filter2 + model\_kaggle(4-6)

# **Teori, model yang diujikan, dataset :**

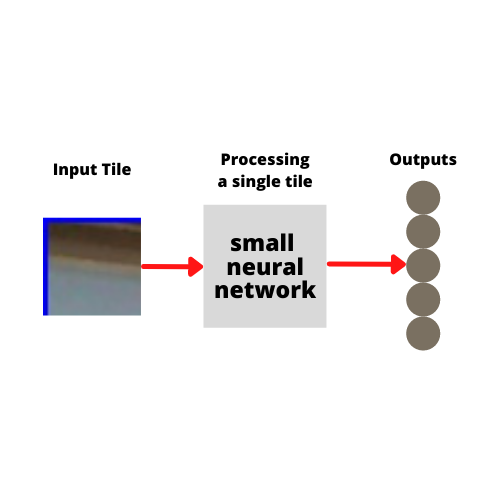
Implementasi yang kami gunakan pada percobaan ini adalah **deep learning** dengan menggunakan **CNN**.

* Kami menggunakan CNN karena pada project yang kami buat mengharuskan memiliki banyak layer yang digunakan untuk memproses dan mengekstrak fitur dari sebuah data yang dimana dataset dan test yang kami gunakan dalam bentuk gambar 2D untuk mendeteksi gambar **Sign Language Alphabet**.
* Cara kerja :
  + Langkah 1



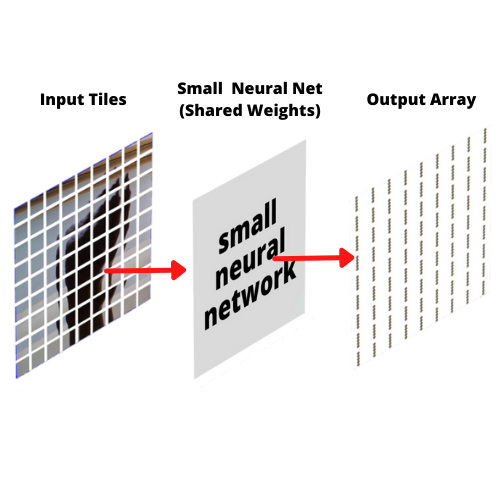
Dari gambar tangan yang menggenggam diatas ini, merupakan hasil proses konvolusi yang dimana menjadi 81 gambar yang lebih kecil

* Langkah 2



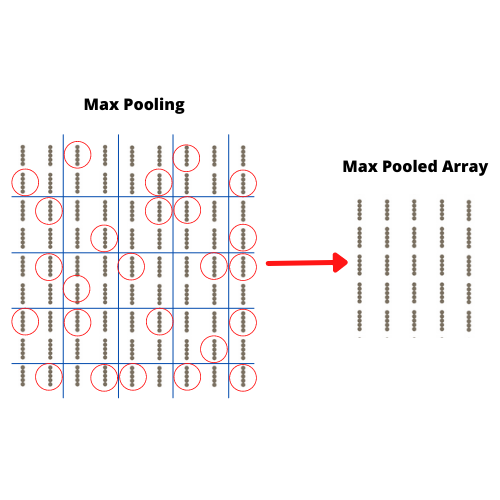
Dari gambar yang telah di konvolusi menjadi gambar yang lebih kecil, akan dimasukkan small neural network yang dijadikan input untuk menghasilkan representasi fitur. Hal ini berguna untuk mengenali sebuah object, dengan menggunakan filter yang sama pada bagian masing-masing gambar kecilnya.

* Langkah 3



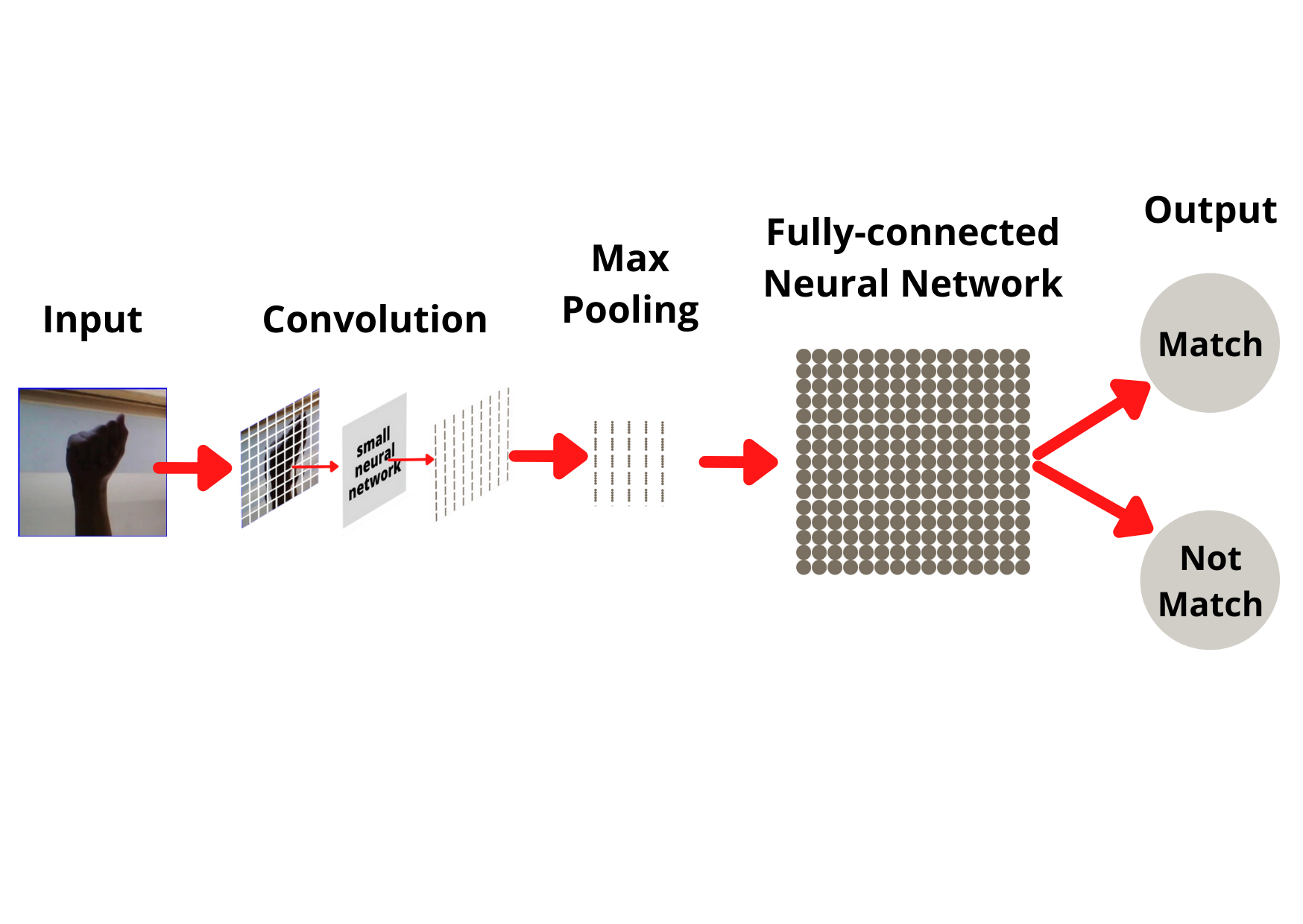
Dari masing-masing gambar kecil tersebut di simpan ke dalam array baru

* Langkah 4



Pada array tersebut masih terlalu besar, sehingga harus dilakukan max-pooled yang berguna untuk mengambil nilai pixel terbesar di setiap max-pooled sehingga dapat mengecilkan ukuran array nya.

* Langkah 5



Karena array merupakan sekelompok angka, dengan menggunakan array yang kecil maka dapat dilanjutkan dengan Fully-connected neural network yang akan memutuskan apakah gambar itu cocok atau tidak.

* Penjelasan Dataset
  + Dataset yang kelompok kami gunakan terdiri dari 29 folder (A-Z, nothing, del, space)
  + Dataset kami terdiri dari 2 macam, yaitu data training dan data test
  + Untuk data training kami mengambil dari kaggle, yang mana di dalam 1 folder terdapat 3000 gambar yang terdiri dari berbagai macam letak posisi sign tangan yang berbeda, berguna untuk membuat kalkulasi (keakuratan) lebih tinggi jika memiliki training yang lebih banyak.
  + Dataset yang kami gunakan dalam percobaan ini kami bedakan menjadi 3 yaitu, dataset 500, daset 1000, dataset 2000

# **Hasil Pengujian, analisa :**

hasil percobaan filter dengan menggunakan kernel

|  | test yang displit | | | test foto sendiri | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | dataset 500 | dataset 1000 | dataset 2000 | dataset 500 | dataset 1000 | dataset 2000 |
| filter | 93.66% | 97.76% | 98.94% | 18.45% | 13.62% | 17.24% |
| filter1 | 55.69% | 92.69% | 99.30% | 17.56% | 15.35% | 18.97% |
| filter2 | 87.13% | 97.93% | 99.50% | 12.93% | 14.48% | 19.66% |
| filter3 | 78.00% | 95.52% | 99.50% | 11.40% | 16.03% | 15.00% |
| Tertinggi | 93.66% | 97.93% | 99.50% | 18.45% | 16.03% | 19.66% |
|  | filter | filter3 | filter2, filter3 | filter | filter3 | filter2 |
| Terendah | 55.69% | 92.69% | 98.94% | 11.40% | 13.62% | 15.00% |
|  | filter1 | filter1 | filter | filter3 | filter | filter3 |

hasil percobaan filter dengan menggunakan kernel1

|  | test yang displit | | | test foto sendiri | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | dataset 500 | dataset 1000 | dataset 2000 | dataset 500 | dataset 1000 | dataset 2000 |
| filter\_1 | 91.21% | 98.03% | 99.59% | 7.41% | 12.93% | 11.90% |
| filter1\_1 | 85.31% | 93.37% | 99.20% | 9.48% | 8.28% | 13.45% |
| filter2\_1 | 79.27% | 94.69% | 97.49% | 7.93% | 10.52% | 10.17% |
| filter3\_1 | 76.48% | 97.31% | 99.44% | 7.24% | 8.27% | 11.21% |
| Tertinggi | 91.21% | 98.03% | 99.59% | 9.48% | 12.93% | 13.45% |
|  | filter\_1 | filter\_1 | filter\_1 | filter1\_1 | filter\_1 | filter1\_1 |
| Terendah | 76.48% | 93.37% | 97.49% | 7.24% | 8.27% | 10.17% |
|  | filter3\_1 | filter1\_1 | filter2\_1 | filter3\_1 | filter3\_1 | filter2\_1 |

hasil percobaan filter dengan menggunakan kernel2

|  | test yang displit | | | test foto sendiri | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | dataset 500 | dataset 1000 | dataset 2000 | dataset 500 | dataset 1000 | dataset 2000 |
| filtter\_2 | 26.83% | 33.90% | 41.79% | 3.62% | 4.48% | 4.18% |
| filter1\_2 | 14.00% | 27.93% | 47.96% | 2.07% | 2.24% | 3.79% |
| filter2\_2 | 4.13% | 26.56% | 38.21% | 3.45% | 2.24% | 5.35% |
| filter3\_2 | 19.35% | 38.00% | 25.47% | 2.07% | 2.59% | 2.41% |
| Tertinggi | 26.83% | 38.00% | 47.96% | 3.62% | 4.48% | 5.35% |
|  | filter\_2 | filter3\_2 | filter1\_2 | filter\_2 | filter\_2 | filter2\_2 |
| Terendah | 4.13% | 26.56% | 25.47% | 2.07% | 2.24% | 2.41% |
|  | filter2\_2 | filter2\_2 | filter3\_2 | filter1\_2, filter3\_2 | filter1\_2, filter2\_2 | filter3\_2 |

hasil percobaan filter dengan menggunakan kernel3

|  | test yang displit | | | test foto sendiri | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | dataset 500 | dataset 1000 | dataset 2000 | dataset 500 | dataset 1000 | dataset 2000 |
| filter\_3 | 51.66% | 98.95% | 99.55% | 11.21% | 18.62% | 15.86% |
| filter1\_3 | 73.03% | 94.17% | 99.16% | 15.00% | 17.24% | 13.45% |
| filter2\_3 | 71.55% | 89.44% | 98.36% | 15.86% | 16.21% | 12.59% |
| filter3\_3 | 75.28% | 91.76% | 98.91% | 17.07% | 16.21% | 11.90% |
| Tertinggi | 75.28% | 98.95% | 99.55% | 17.07% | 18.62% | 15.86% |
|  | filter3\_3 | filter\_3 | filter\_3 | filter3\_3 | filter\_3 | filter\_3 |
| Terendah | 51.66% | 89.44% | 98.36% | 11.21% | 16.21% | 11.90% |
|  | filter\_3 | filter2\_3 | filter2\_3 | filter\_3 | filter2\_3, filter3\_3 | filter3\_3 |

hasil percobaan filter dengan menggunakan kernel4

|  | test yang displit | | | test foto sendiri | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | dataset 500 | dataset 1000 | dataset 2000 | dataset 500 | dataset 1000 | dataset 2000 |
| filter\_4 | 82.35% | 94.45% | 99.34% | 9.31% | 14.66% | 19.14% |
| filter1\_4 | 55.48% | 88.98% | 98.75% | 10.35% | 15.52% | 19.31% |
| filter2\_4 | 66.93% | 96.26% | 99.27% | 9.48% | 17.24% | 21.03% |
| filter3\_4 | 88.45% | 98.24% | 99.03% | 13.62% | 21.38% | 14.31% |
| Tertinggi | 88.45% | 98.24% | 99.34% | 13.62% | 21.38% | 21.03% |
|  | filter3\_4 | filter3\_4 | filter\_4 | filter3\_4 | filter3\_4 | filter2\_4 |
| Terendah | 55.48% | 88.98% | 98.75% | 9.31% | 14.66% | 14.31% |
|  | filter1\_4 | filter1\_4 | filter1\_4 | filter\_4 | filter\_4 | filter3\_4 |

hasil percobaan filter dengan menggunakan kernel5

|  | test yang displit | | | test foto sendiri | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | dataset 500 | dataset 1000 | dataset 2000 | dataset 500 | dataset 1000 | dataset 2000 |
| filter\_5 | 55.79% | 91.50% | 99.60% | 12.59% | 18.28% | 18.79% |
| filter1\_5 | 30.03% | 88.03% | 99.58% | 11.90% | 13.62% | 18.10% |
| filter2\_5 | 71.41% | 92.09% | 97.78% | 14.48% | 15.35% | 15.00% |
| filter3\_5 | 78.07% | 92.98% | 99.46% | 10.00% | 14.83% | 12.07% |
| Tertinggi | 78.07% | 92.98% | 99.60% | 14.48% | 18.28% | 18.79% |
|  | filter3\_5 | filter3\_5 | filter\_5 | filter2\_5 | filter\_5 | filter\_5 |
| Terendah | 30.03% | 88.03% | 97.78% | 10.00% | 13.62% | 12.07% |
|  | filter1\_5 | filter1\_5 | filter2\_5 | filter3\_5 | filter1\_5 | filter3\_5 |

hasil percobaan filter dengan menggunakan kernel6

|  | test yang displit | | | test foto sendiri | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | dataset 500 | dataset 1000 | dataset 2000 | dataset 500 | dataset 1000 | dataset 2000 |
| filter\_6 | 94.55% | 98.67% | 99.58% | 17.41% | 13.28% | 16.21% |
| filter1\_6 | 98.93% | 93.70% | 98.93% | 14.66% | 15.00% | 14.66% |
| filter2\_6 | 69.97% | 96.02% | 99.31% | 8.79% | 19.66% | 15.35% |
| filter3\_6 | 86.28% | 97.28% | 98.03% | 16.89% | 21.21% | 14.14% |
| Tertinggi | 98.93% | 98.67% | 99.58% | 17.41% | 21.21% | 16.21% |
|  | filter1\_6 | filter\_6 | filter\_6 | filter\_6 | filter3\_6 | filter\_6 |
| Terendah | 69.97% | 93.70% | 98.03% | 8.79% | 13.28% | 14.14% |
|  | filter2\_6 | filter1\_6 | filter3\_6 | filter2\_6 | filter\_6 | filter3\_6 |

perhitungan hasil percobaan filter

|  | tertinggi | terendah |
| --- | --- | --- |
| filter | 21 | 7 |
| filter1 | 3 | 12 |
| filter2 | 5 | 8 |
| filter3 | 12 | 9 |

* Dari hasil percobaan mengubah filter kami mendapatkan bahwa akurasi tertinggi didapatkan ketika menggunakan **filter** yang mana model ini menggunakan filter yang meningkat secara bertahap.
* Sedangkan akurasi terendah didapatkan ketika menggunakan **filter1** yang mana model ini menggunakan filter yang makin lama makin menurun (bertolak belakang dengan filter)

hasil percobaan mengubah kernel

|  | test yang displit | | | test foto sendiri | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | dataset 500 | dataset 1000 | dataset 2000 | dataset 500 | dataset 1000 | dataset 2000 |
| kernel | 93.66% | 97.76% | 98.94% | 18.45% | 13.62% | 17.24% |
| kernel1 | 91.21% | 98.03% | 99.59% | 7.41% | 12.93% | 11.90% |
| kernel2 | 26.83% | 33.90% | 41.79% | 3.62% | 4.48% | 4.18% |
| kernel3 | 51.66% | 98.95% | 99.55% | 11.21% | 18.62% | 15.86% |
| kernel4 | 82.35% | 94.45% | 99.34% | 9.31% | 14.66% | 19.14% |
| kernel5 | 55.79% | 91.50% | 99.60% | 12.59% | 18.28% | 18.79% |
| kernel6 | 94.55% | 98.67% | 99.58% | 17.41% | 13.28% | 16.21% |
| Tertinggi | 93.66% | 98.95% | 99.60% | 18.45% | 18.62% | 19.14% |
|  | kernel | kernel3 | kernel5 | kernel | kernel3 | kernel4 |
| Terendah | 26.83% | 33.90% | 41.79% | 3.62% | 4.48% | 4.18% |
|  | kernel2 | kernel2 | kernel2 | kernel2 | kernel2 | kernel2 |

* akurasi tertinggi didapat ketika menggunakan **kernel**(kernel\_size=5) dan **kernel3**(kernel\_size=3)
* akurasi terendah didapat ketika menggunakan **kernel2**. hal ini disebabkan karena kernel2 merupakan model yang kernel\_size nya 1 semua

hasil percobaan mengubah layer

|  | test yang displit | | | test foto sendiri | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | dataset 500 | dataset 1000 | dataset 2000 | dataset 500 | dataset 1000 | dataset 2000 |
| layer | 93.66% | 97.76% | 98.94% | 18.45% | 13.62% | 17.24% |
| layer1 | 49.83% | 76.69% | 98.89% | 11.55% | 8.28% | 14.48% |
| layer2 | 73.07% | 93.35% | 98.74% | 15.00% | 14.83% | 13.97% |
| Tertinggi | 93.66% | 97.76% | 98.94% | 18.45% | 14.83% | 17.24% |
|  | layer | layer | layer | layer | layer2 | layer |
| Terendah | 49.83% | 76.69% | 98.74% | 11.55% | 8.28% | 13.97% |
|  | layer1 | layer1 | layer2 | layer1 | layer1 | layer2 |

* Akurasi tertinggi didapat ketika menggunakan **layer** yang mana model tersebut terdiri atas 5 layer.
* Akurasi terendah didapat ketika menggunakan **layer1** yang mana model tersebut terdiri atas 7 layer.

hasil percobaan mengubah jumlah epoch

|  | test yang displit | | | test foto sendiri | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | dataset 500 | dataset 1000 | dataset 2000 | dataset 500 | dataset 1000 | dataset 2000 |
| epoch5 | 93.66% | 97.76% | 98.94% | 18.45% | 13.62% | 17.24% |
| epoch3 | 98.41% | 83.60% | 99.29% | 18.10% | 9.66% | 15.00% |
| epoch7 | 95.93% | 99.50% | 99.78% | 14.83% | 18.28% | 14.83% |
| Tertinggi | 98.41% | 99.50% | 99.78% | 18.45% | 18.28% | 17.24% |
|  | epoch3 | epoch7 | epoch7 | epoch5 | epoch7 | epoch5 |
| Terendah | 93.66% | 83.60% | 98.94% | 14.83% | 9.66% | 14.83% |
|  | epoch5 | epoch3 | epoch5 | epoch7 | epoch3 | epoch7 |

# Akurasi tertinggi didapat ketika menggunakan **epoch7** (7 epoch)

* Akurasi terendah didapatkan balance (sama) untuk epoch3, epoch5, epoch7

# **Kesimpulan :**

Dari berbagai percobaan yang kami lakukan, kami dapat menyimpulkan bahwa model mempengaruhi akurasi dari hasil prediksi. Ada 3 hal dari model yang memiliki pengaruh, yaitu : jumlah filter, ukuran kernel, jumlah layer.

* Penggunaan **jumlah filter** yang baik adalah dengan melakukan penambahan secara bertahap dan jumlah filter harus 2^n.
* Penggunaan **ukuran kernel** yang baik adalah dengan menggunakan bilangan ganjil bulat. Pada umumnya ukuran kernel yang digunakan adalah 3x3 dan 5x5
* Penggunaan **jumlah layer** yang baik adalah secukupnya saja dengan menambahkan jumlah layer bukan berarti dapat meningkatkan akurasi prediksi