

# 1 Desain dan Implementasi Penambahan Data

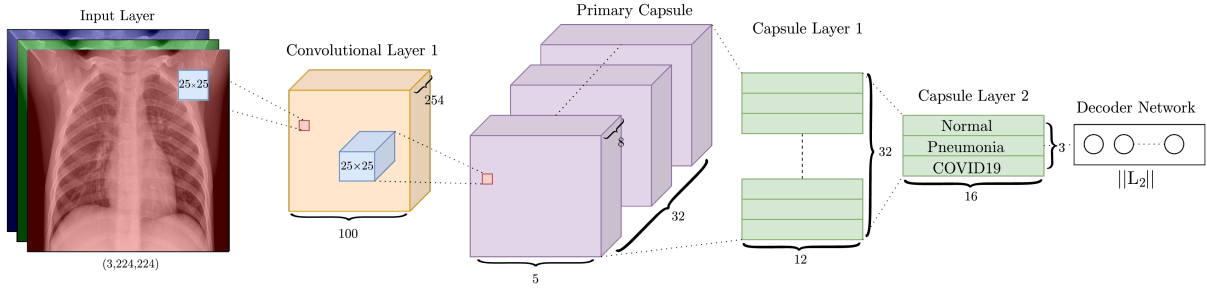


Figure 1: Arsitektur CapsNet

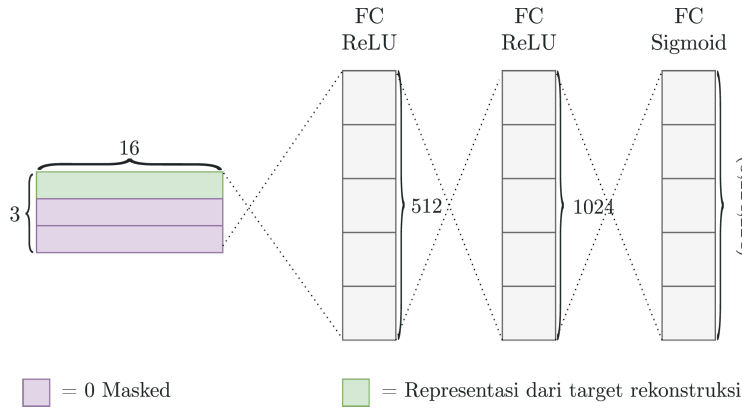


Figure 2: Arsitektur Decoder Network

## 1.1 Arsitektur CapsNet

Arsitektur CapsNet yang digunakan dalam makalah ini terdiri dari :

1. **Input Layer**  
*Input* menggunakan gambar berukuran 224x224 dengan 3 *channel* yaitu *channel red*, *green*, dan *blue* atau biasa disebut dengan RGB.
2. **Convolutional Layer 1**  
*Layer* ini akan melakukan operasi konvolusi pada gambar *input* menggunakan *out channel* atau *filter* sebanyak 256 dengan *kernel* berukuran 25x25 dan *stride* sebanyak 2. *Layer* ini akan menghasilkan *feature map*.
3. **Primary Capsule**  
*Layer* ini akan mengubah hasil konvolusi menjadi *capsule-capsule*. *Layer* ini terdiri dari beberapa layer, yaitu:
  - **Convolution Layer 2**  
*Layer* ini mendapatkan *input channel* sebanyak 256, lalu operasi konvolusi pada *layer* ini menggunakan *out channel* sebanyak 256 dengan *kernel* berukuran 25x25 dan *stride* sebanyak 16.
  - **Reshape**  
Operasi reshape digunakan untuk mendapatkan *capsule*.
  - **Squash**  
*Layer squash* adalah *layer* aktivasi dari primary capsule yang mengubah panjang vektor besar menjadi mendekati 1 dan vektor kecil menjadi 0.
4. **Capsule Layer 1**  
*Layer* ini mendapatkan *input capsule* sebanyak 800 berdimensi 8 untuk setiap *capsule*. Lalu algoritma *Routing-by-agreement* akan dijalankan pada *layer* ini dan menghasilkan *output capsule* sebanyak 32 berdimensi 12 untuk setiap *capsule*.

5. *Capsule Layer 2*

*Layer* ini mendapatkan *input capsule* sebanyak 32 berdimensi 12 untuk setiap *capsule*. Lalu algoritma *Routing-by-agreement* akan dijalankan pada layer ini dan menghasilkan *output capsule* sebanyak jumlah *class* yaitu 3 dan berdimensi 16 untuk setiap *capsule*.

6. *Decoder Network*

*Decoder Network* adalah *layer* yang digunakan CapsNet untuk menghitung *loss* dan melakukan proses rekonstruksi gambar. *Decoder Network* terdiri dari:

- *Fully Connected Layer 1*

*Layer* ini mendapatkan *input* sebanyak jumlah *output capsule* dikali dimensi *capsule* dari *layer* sebelumnya yaitu  $3 \times 16$  dan menghasilkan 512 *output neuron*. Fungsi aktivasi yang digunakan pada layer ini adalah ReLu.

- *Fully Connected Layer 2*

*Layer* ini mendapatkan *input* sebanyak jumlah *output neuron* dari *layer* sebelumnya yaitu 512 dan menghasilkan 1024 *output neuron*. Fungsi aktivasi yang digunakan pada layer ini adalah ReLu.

- *Fully Connected Layer 3*

*Layer* ini mendapatkan *input* sebanyak jumlah *output neuron* dari *layer* sebelumnya yaitu 1024 dan menghasilkan *output neuron* sebanyak ukuran gambar *input* yaitu  $224 \times 224 \times 3$ . Fungsi aktivasi yang digunakan pada layer ini adalah *sigmoid*.