



DEPARTEMEN TEKNIK INFORMATIKA



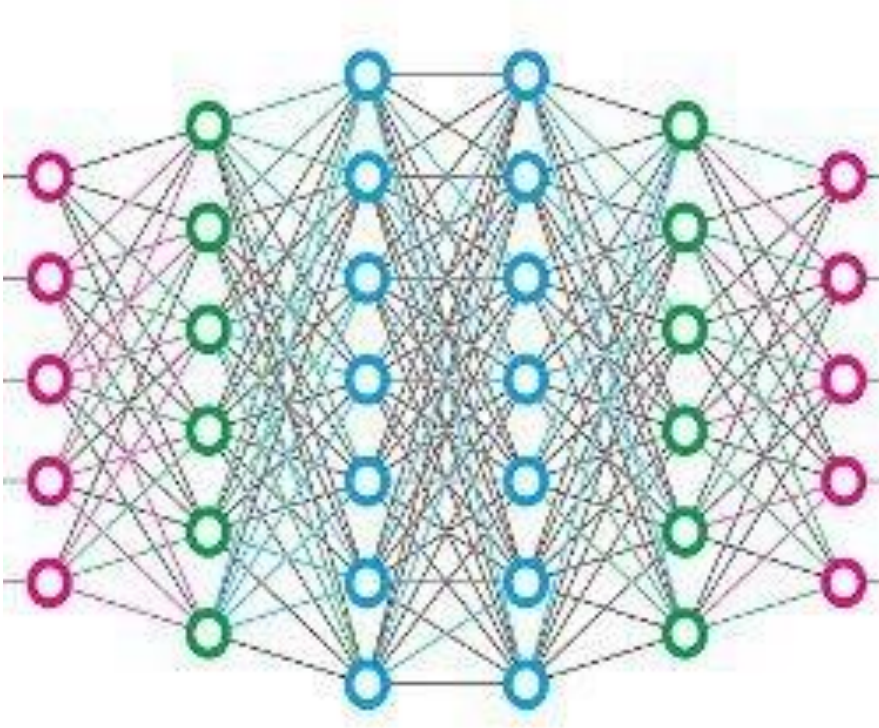
www.its.ac.id



[its_campus](#)



[institut teknologi sepuluh nopember](#)



PENGANTAR DEEP LEARNING

**Kecerdasan
Komputasional**



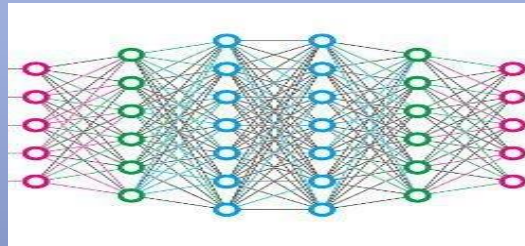
Artificial Intelligence



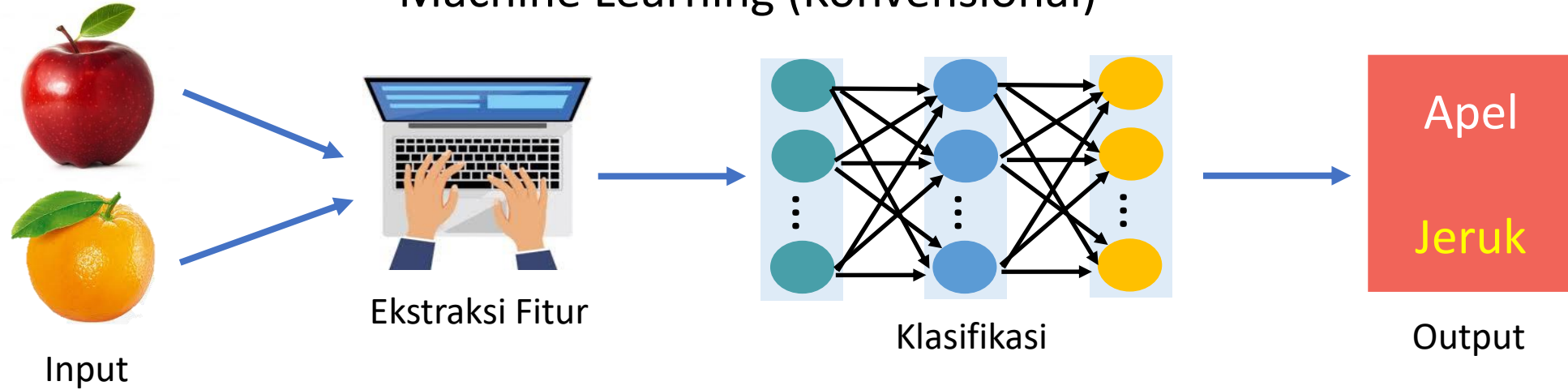
Machine Learning



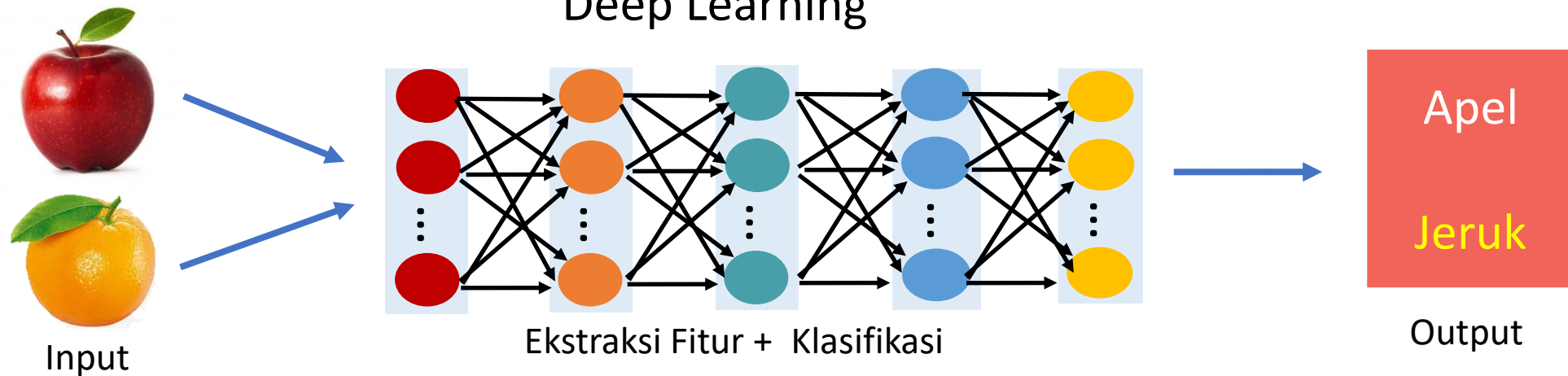
Deep Learning



Machine Learning (Konvensional)



Deep Learning



Klasifikasi Secara Konvensional

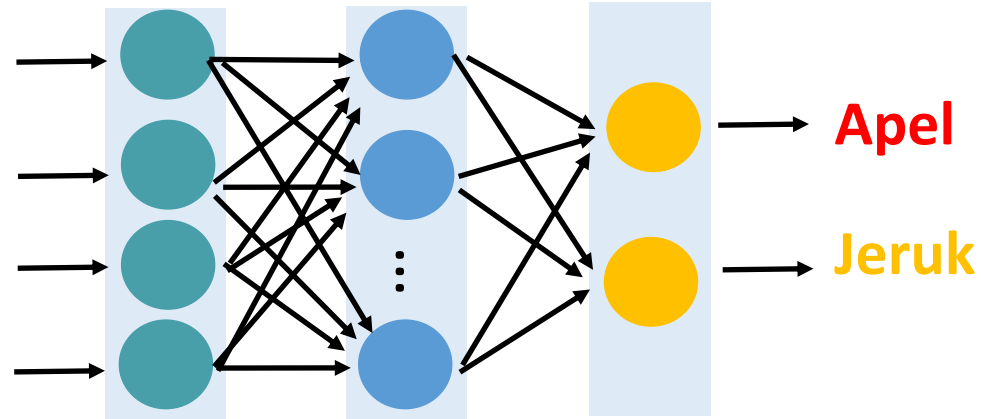
- Pendekatan klasifikasi secara konvensional umumnya melakukan ekstraksi fitur secara terpisah kemudian dilanjutkan proses pembelajaran menggunakan metode klasifikasi konvensional seperti DT, SVM, atau ANN.
- Salah satu contoh permasalahan klasifikasi adalah pengenalan obyek



Klasifikasi Secara Konvensional

Ekstraksi Fitur

Klasifikasi

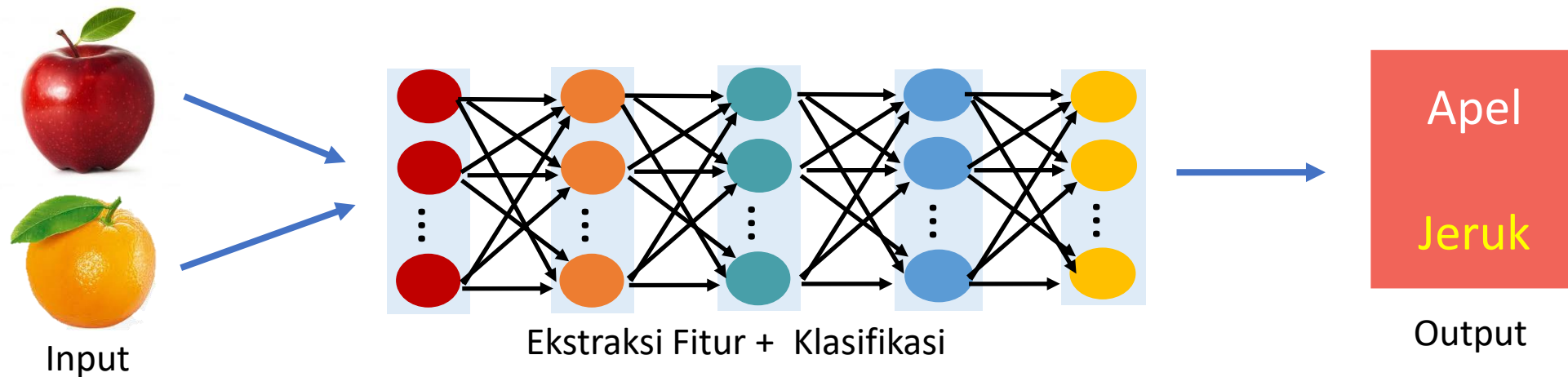


No	Warna	Bentuk	Tekstur	Ukuran	Kelas
1	Merah	Bulat	Halus	Besar	Apel
2	Hijau	Bulat	Halus	Sedang	Apel
3	Kuning	Lonjong	Kasar	Besar	Jeruk
...					
N-1	Hijau	Lonjong	Kasar	Sedang	Jeruk
N	Hijau	Bulat	Halus	Kecil	Jeruk

Kelemahan

- Memerlukan waktu dan pengetahuan lebih untuk ekstraksi fitur
- Sangat tergantung pada satu domain permasalahan saja sehingga tidak berlaku general

- Pengenalan obyek berbasis Deep learning mempelajari representasi hirarki (pola fitur) secara otomatis melalui beberapa tahapan proses *feature learning*

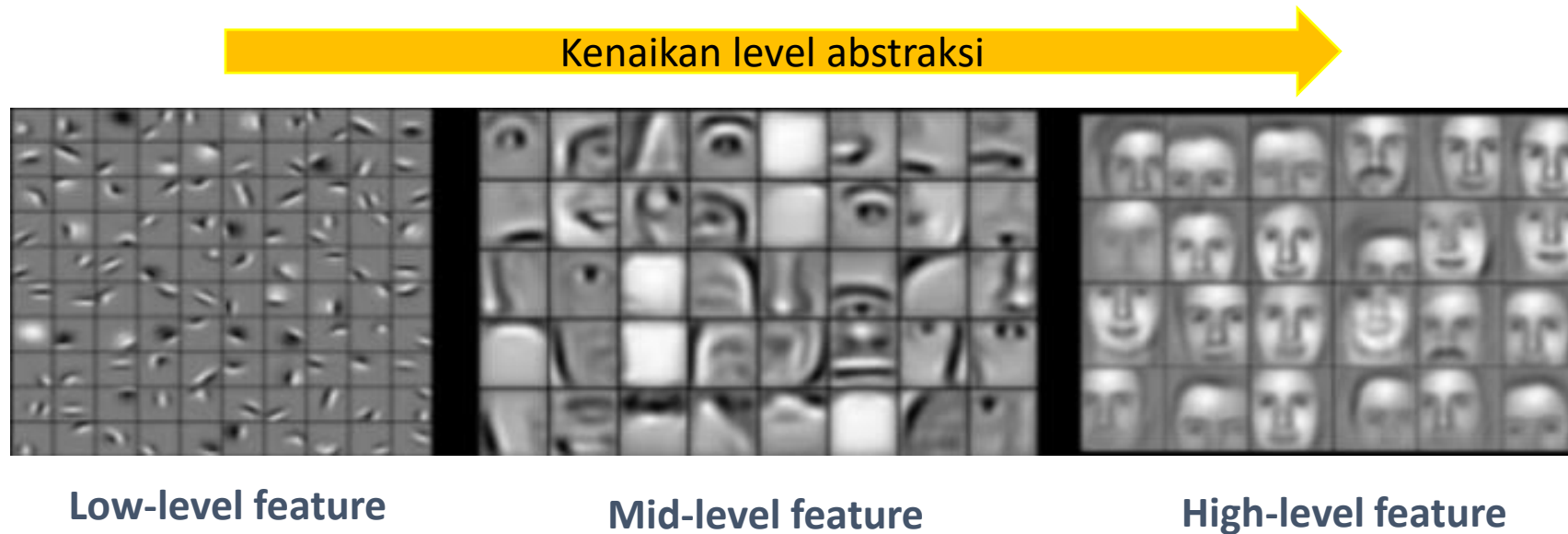


Feature learning

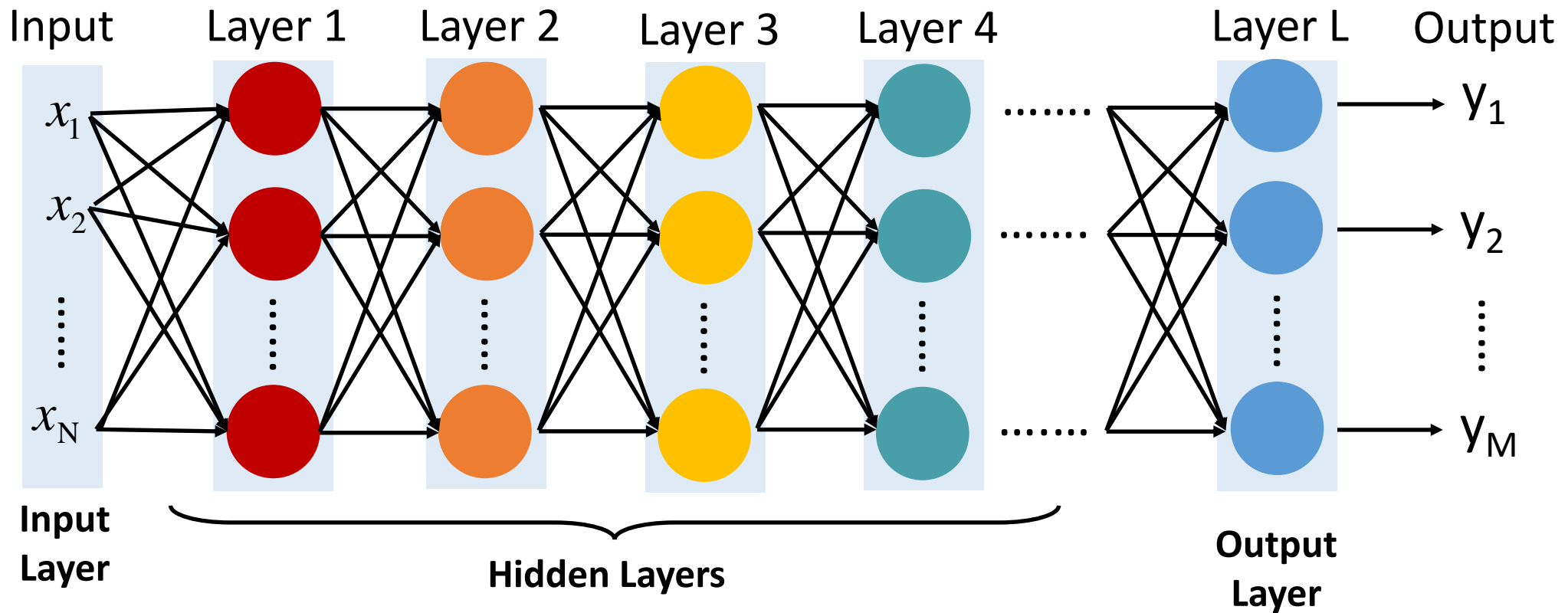


Representasi Hirarki Fitur

- Hirarki pada representasi fitur menunjukkan kenaikan level dari abstraksi.
- Setiap tahapan merupakan proses *feature learning* pada transformasi fitur



Deep Neural Network



Deep artinya banyak hidden layer

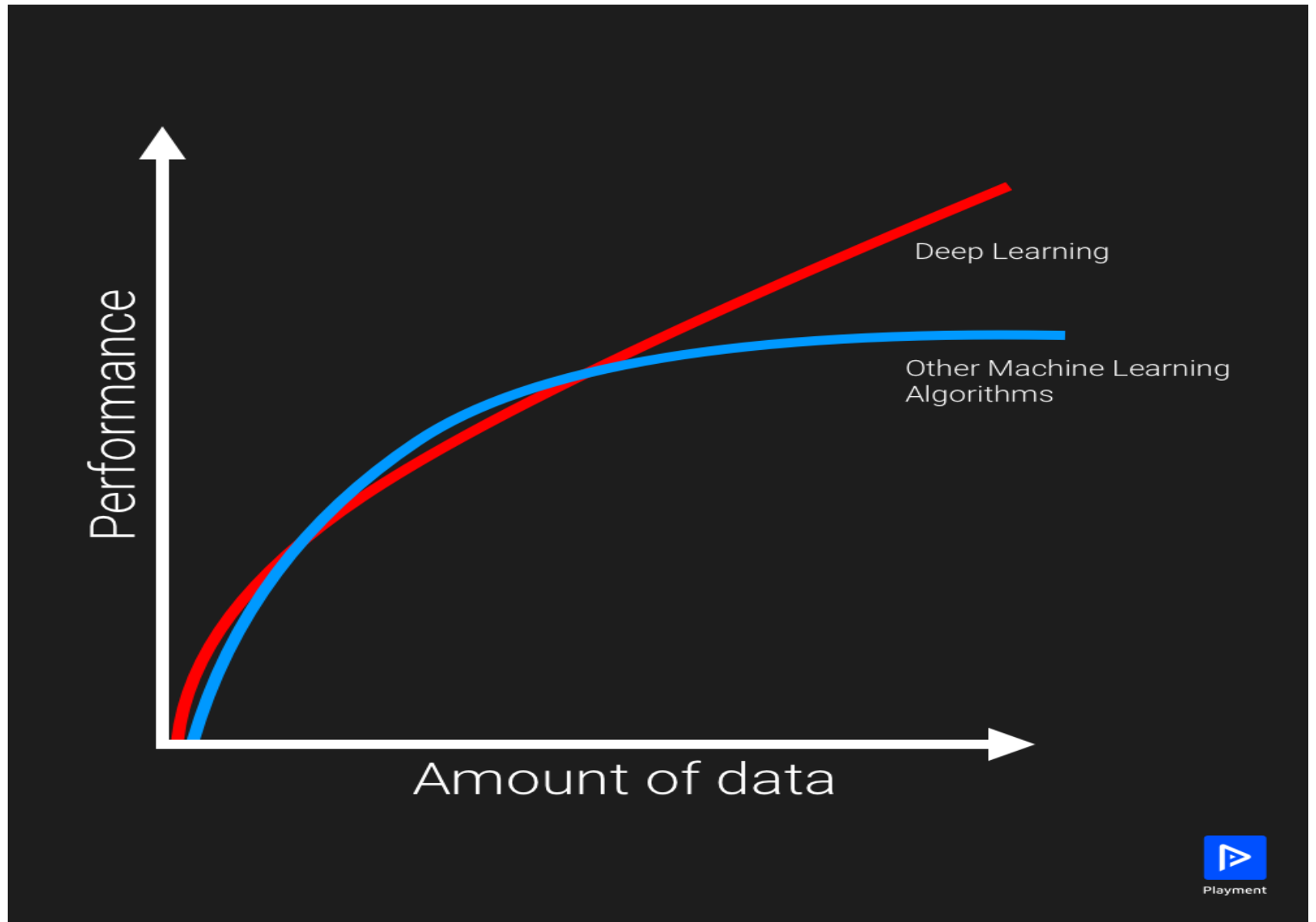
Deep vs Shallow

Layer × Size	Word Error Rate (%)	Layer × Size	Word Error Rate (%)
1 × 2k	24.2		
2 × 2k	20.4		
3 × 2k	18.4		
4 × 2k	17.8		
5 × 2k	17.2	1 × 3772	22.5
7 × 2k	17.1	1 × 4634	22.6
		1 × 16k	22.1

Manakah yang lebih baik?

Seide, Frank, Gang Li, and Dong Yu. "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks." Interspeech. 2011.

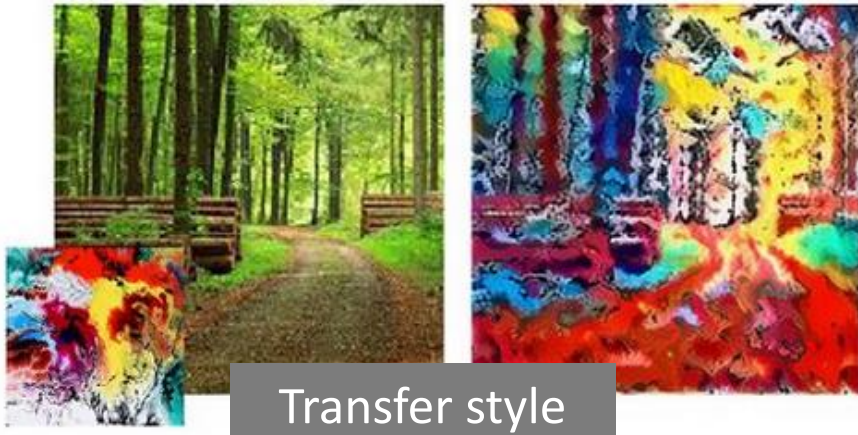
Mengapa Deep Learning?



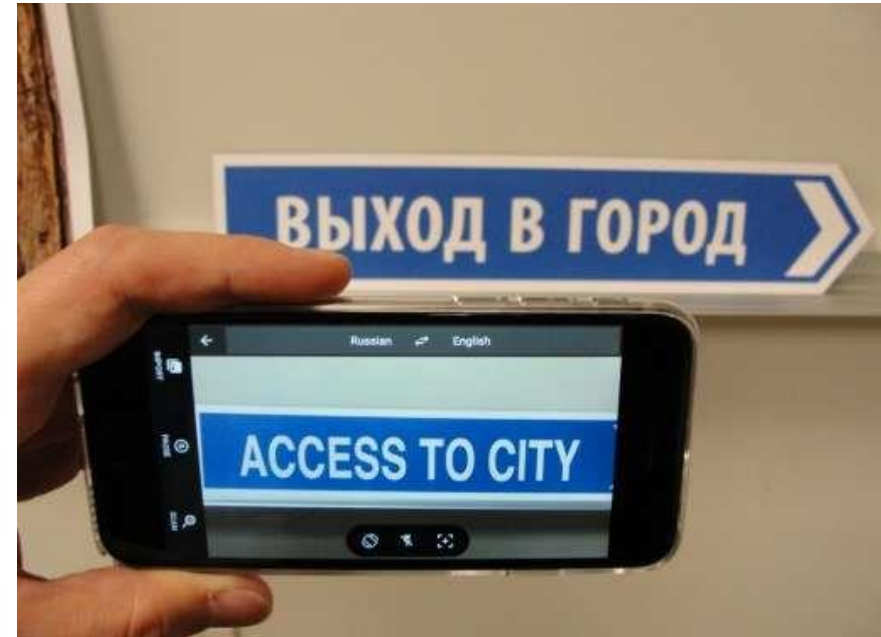
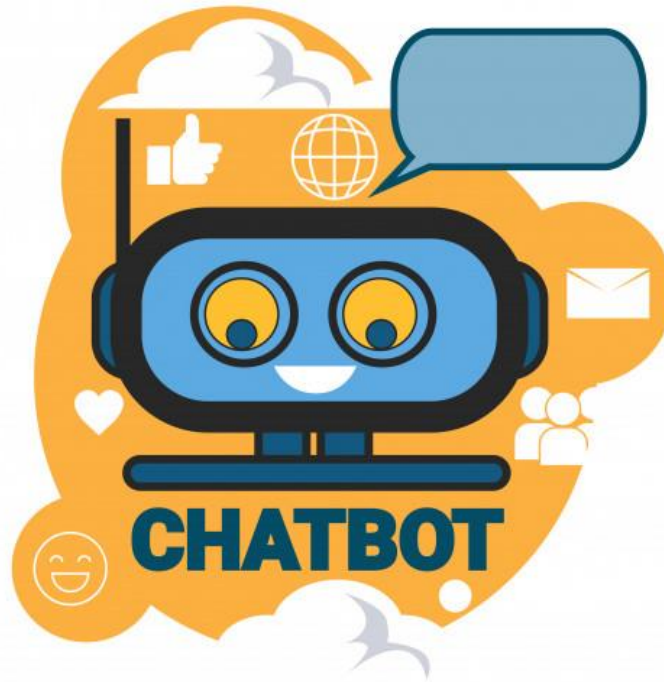
Arsitektur Deep Learning

- *Convolutional Neural Network (CNN)*
- *Sequence Modelling*
 - *Recurrent Neural Networks (RNN)*
- *Generative Modelling*
 - *Generative Adversarial Network (GAN)*
- *Deep Reinforcement Learning*

Aplikasi *Convolutional Neural Network*

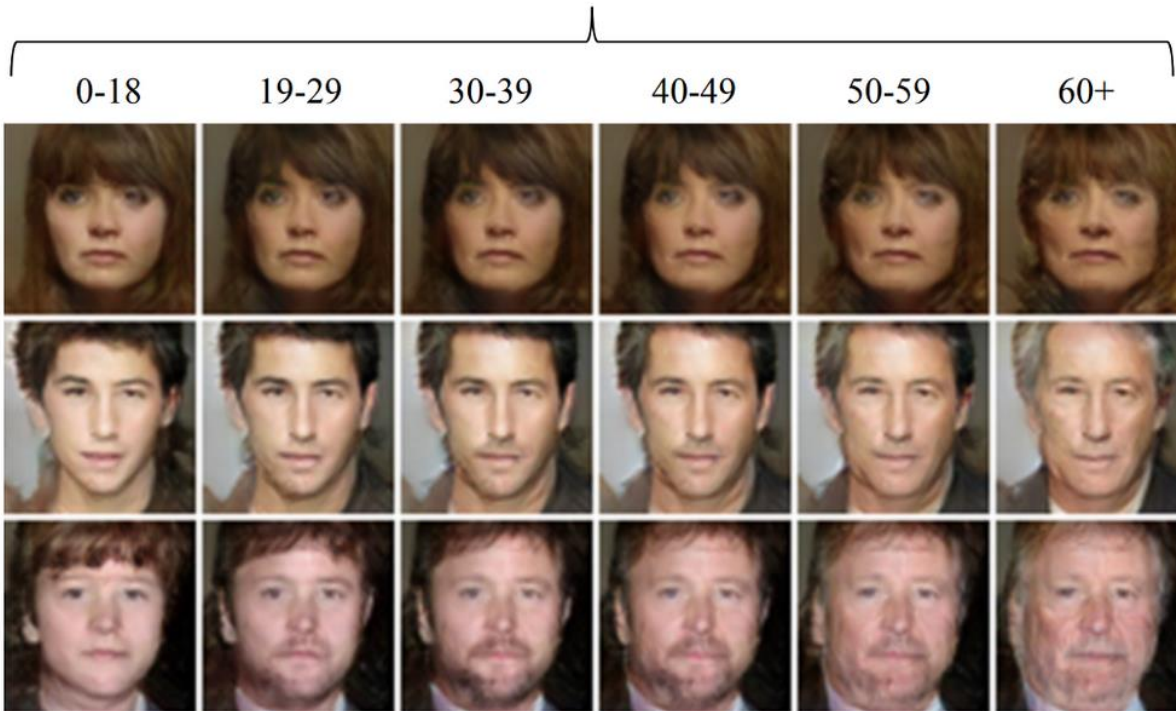


Aplikasi *Sequence Modelling*



Aplikasi *Generative Modelling*

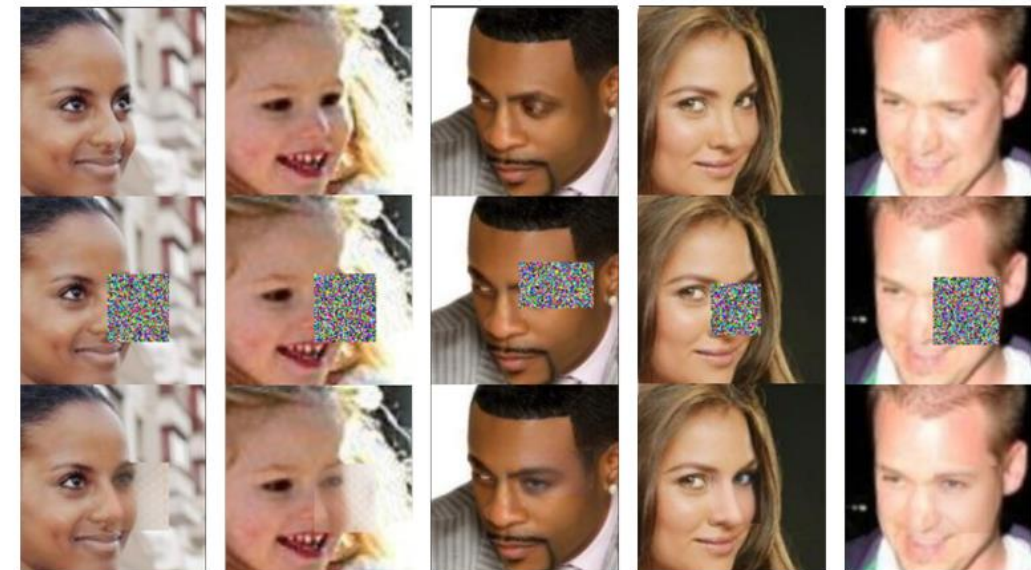
Face Aging



Original

Masking

Inpainting
Result



Zebras \leftrightarrow Horses

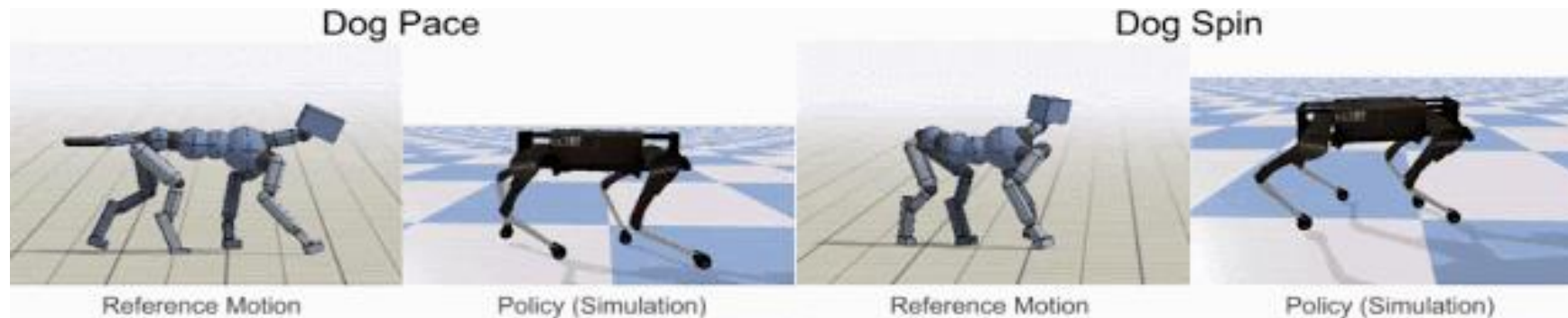
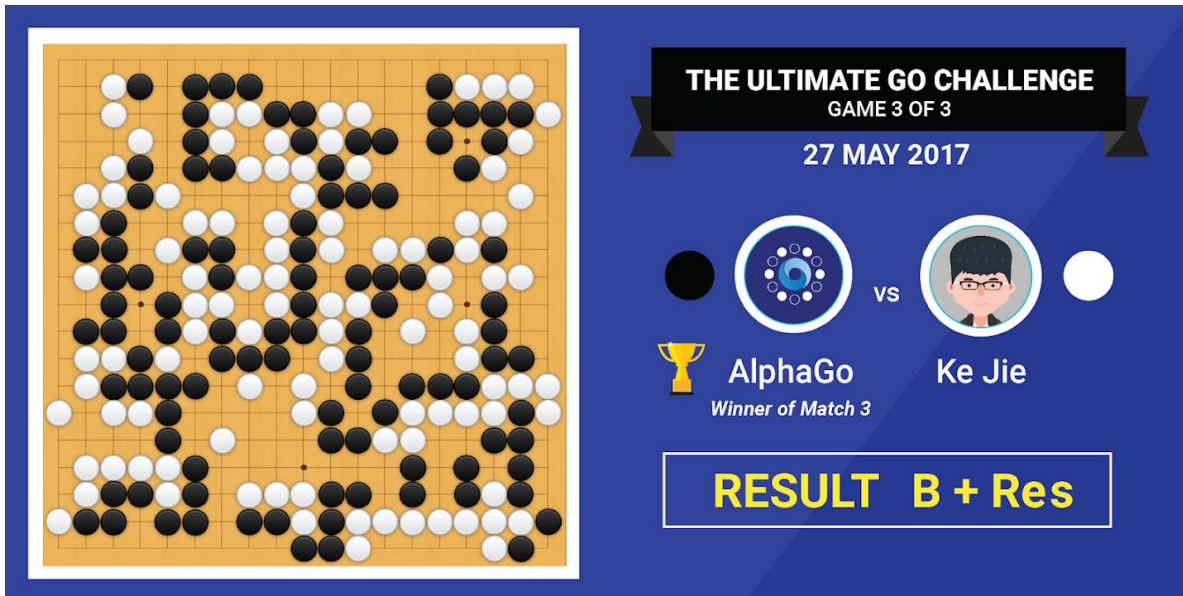


zebra \rightarrow horse

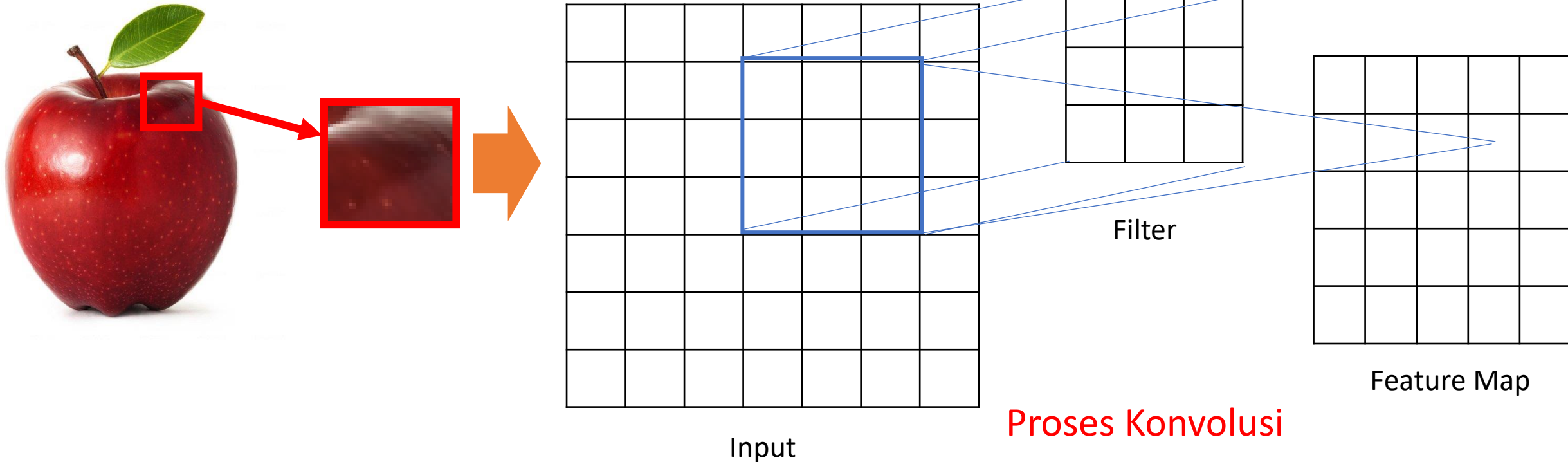


horse \rightarrow zebra

Aplikasi *Deep Reinforcement Learning*



- CNN merupakan salah satu varian *Neural Network* dengan beberapa layer konvolusi dan layer lainnya.
- Layer konvolusi mempunyai sejumlah filter yang mengaplikasikan operasi konvolusi.





Arsitektur CNN

Convolution

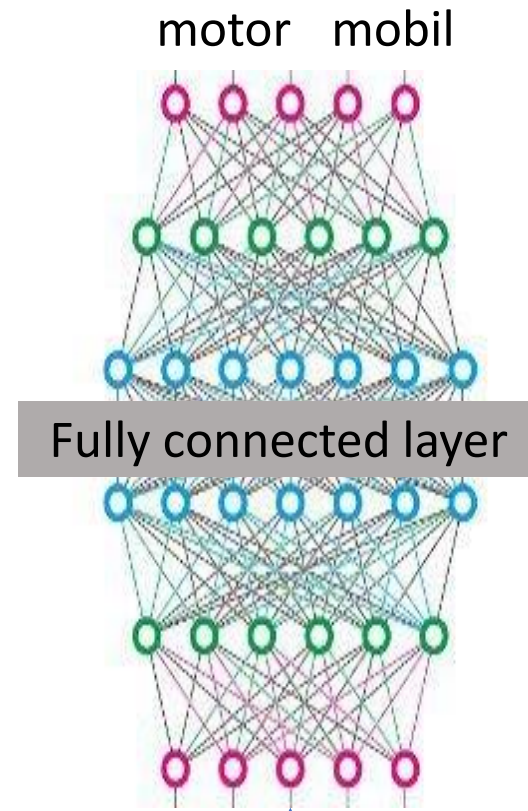
Max Pooling

Convolution

Max Pooling

Bisa dilakukan berulang kali

Flatten



Proses Konvolusi

$$(I * K)(i, j) = \sum_m \sum_n I(m, n) K(i + m, j + n)$$

Layer Konvolusi

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

/

*

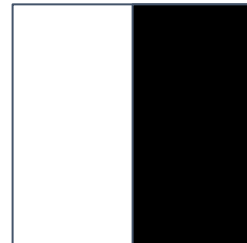
1	-1	-1
-1	1	-1
-1	-1	1

K

Proses Konvolusi

Citra

30	30	30	0	0	0
30	30	30	0	0	0
30	30	30	0	0	0
30	30	30	0	0	0
30	30	30	0	0	0
30	30	30	0	0	0



Filter

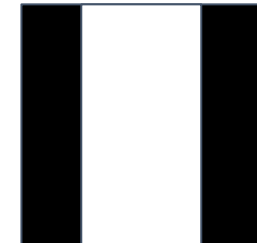
1	0	-1
1	0	-1
1	0	-1

*

=

Convolved
Feature

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



Ukuran Convoluted Feature (Feature Map)

Ukuran output gambar hasil proses konvolusi dapat dihitung dengan rumus sebagai berikut:

Contoh:

$$((N - F)/s) + 1$$

N : input size

F : filter size

s : stride

$$\text{stride} = 1 \rightarrow ((5-3)/1) + 1 = 3$$

$$\text{stride} = 2 \rightarrow ((5-3)/2) + 1 = 2$$

Stride adalah besarnya step yang digunakan ketika melakukan sliding filter

Proses Konvolusi

Layer Konvolusi

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Citra 6 x 6

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

⋮ ⋮

Tiap filter menangkap sebuah pola (3 x 3)

Proses Konvolusi

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Citra 6 x 6

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Layer Konvolusi

Proses Konvolusi

stride=1

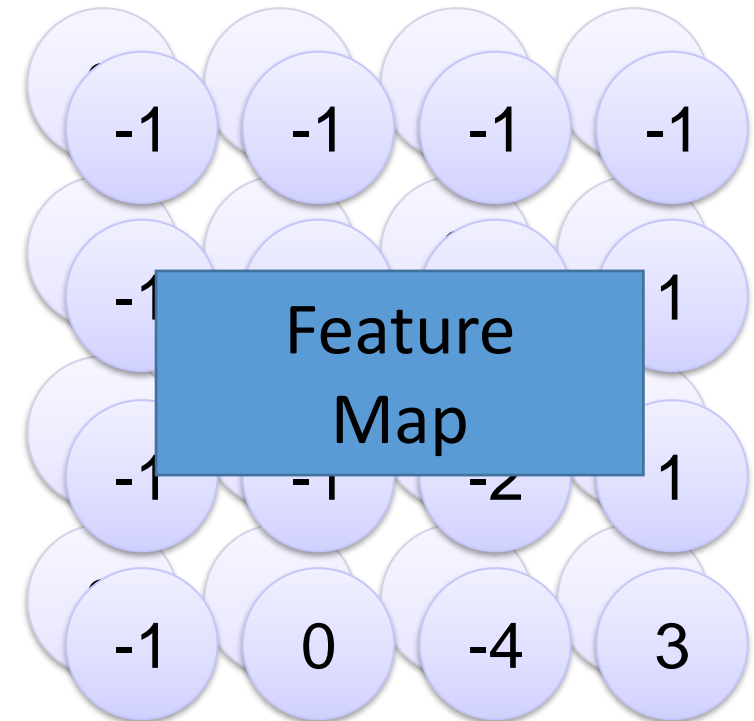
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Citra 6 x 6

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

Layer Konvolusi

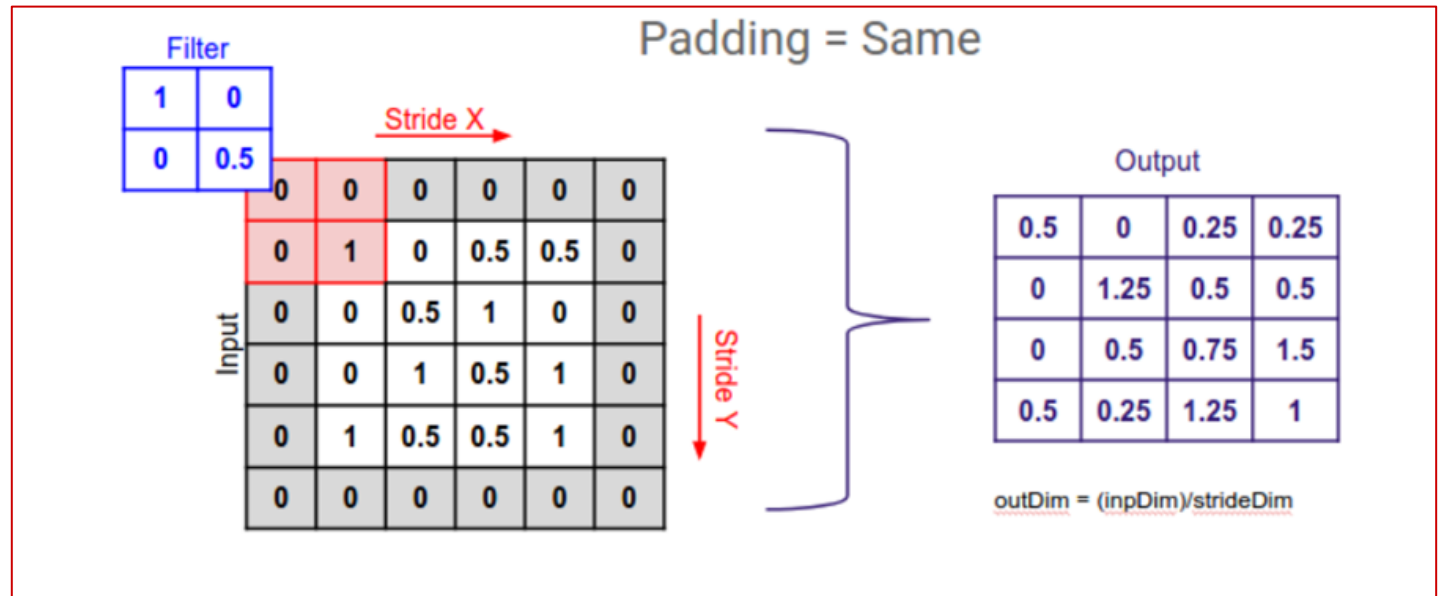


Dua citra 4 x 4
Membentuk 2 *feature map*

Padding

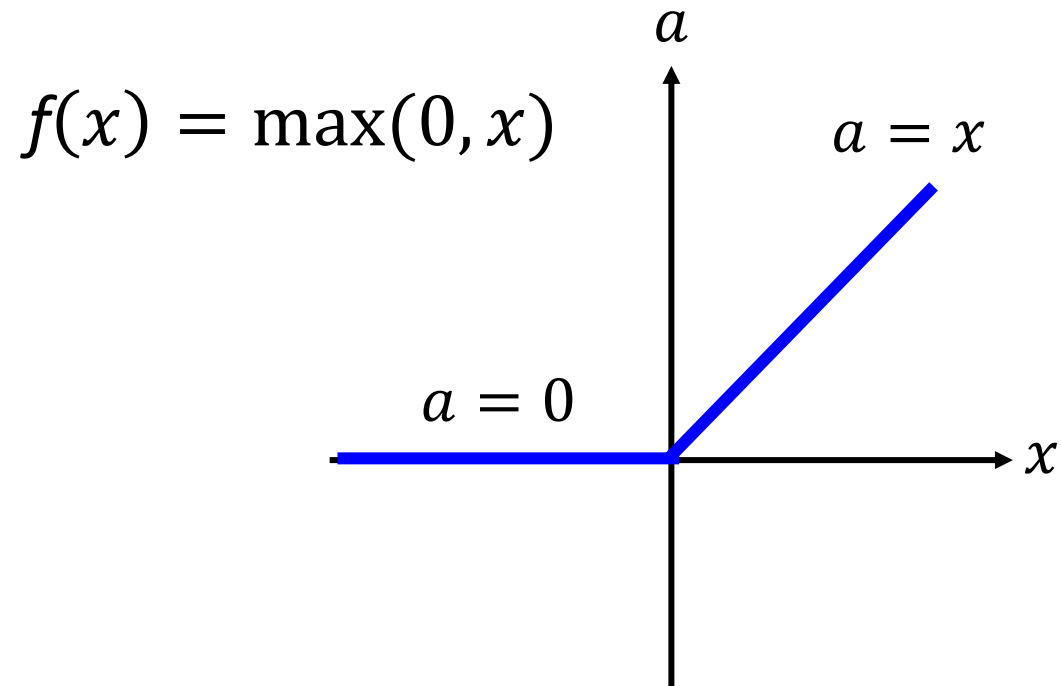
- Layer tambahan untuk menangani **tepi dari citra** untuk menghindari hilangnya informasi pada corner sebuah citra
- Zero Padding

0	0	0	0	0	0			
0								
0								
0								
0								



Aktivasi ReLU

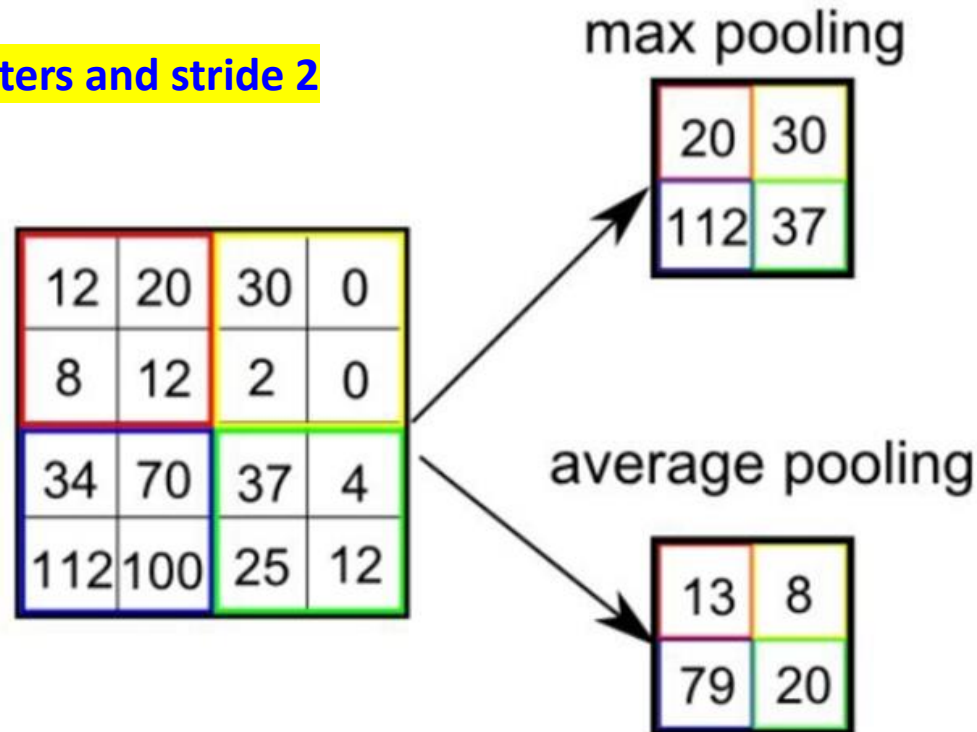
- Rectified Linear Unit (ReLU)
 - Komputasi cepat
 - *Infinite sigmoid*
 - Menangani *vanishing gradient*



POOLING

- Pooling layer digunakan untuk mengurangi ukuran gambar menjadi lebih kecil (downsample) dan mengekstrak *salient features*.
- Ada dua tipe pooling yaitu **Maximum pooling** dan **Average pooling**

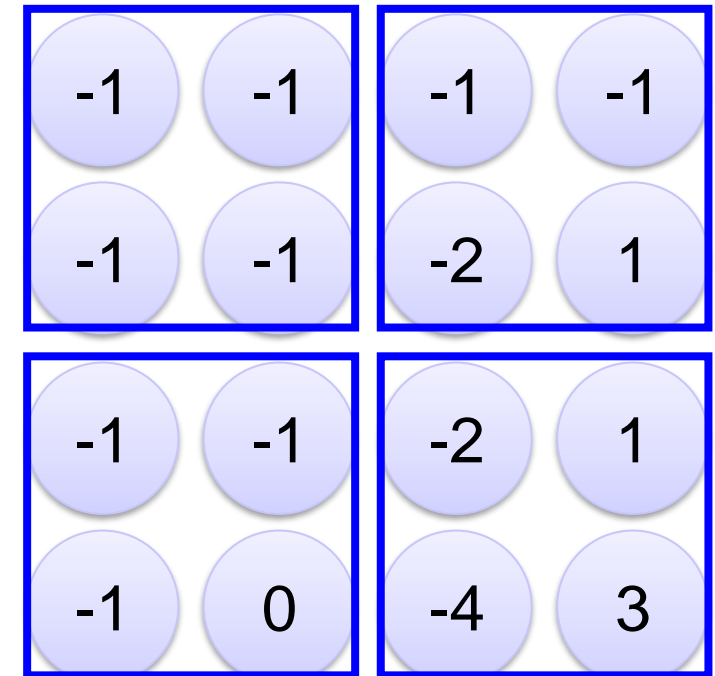
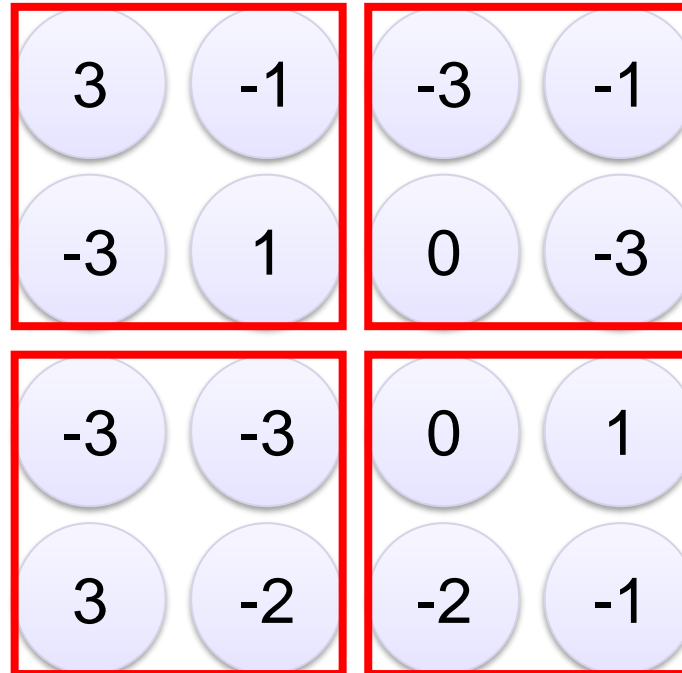
Pooling with 2x2 filters and stride 2



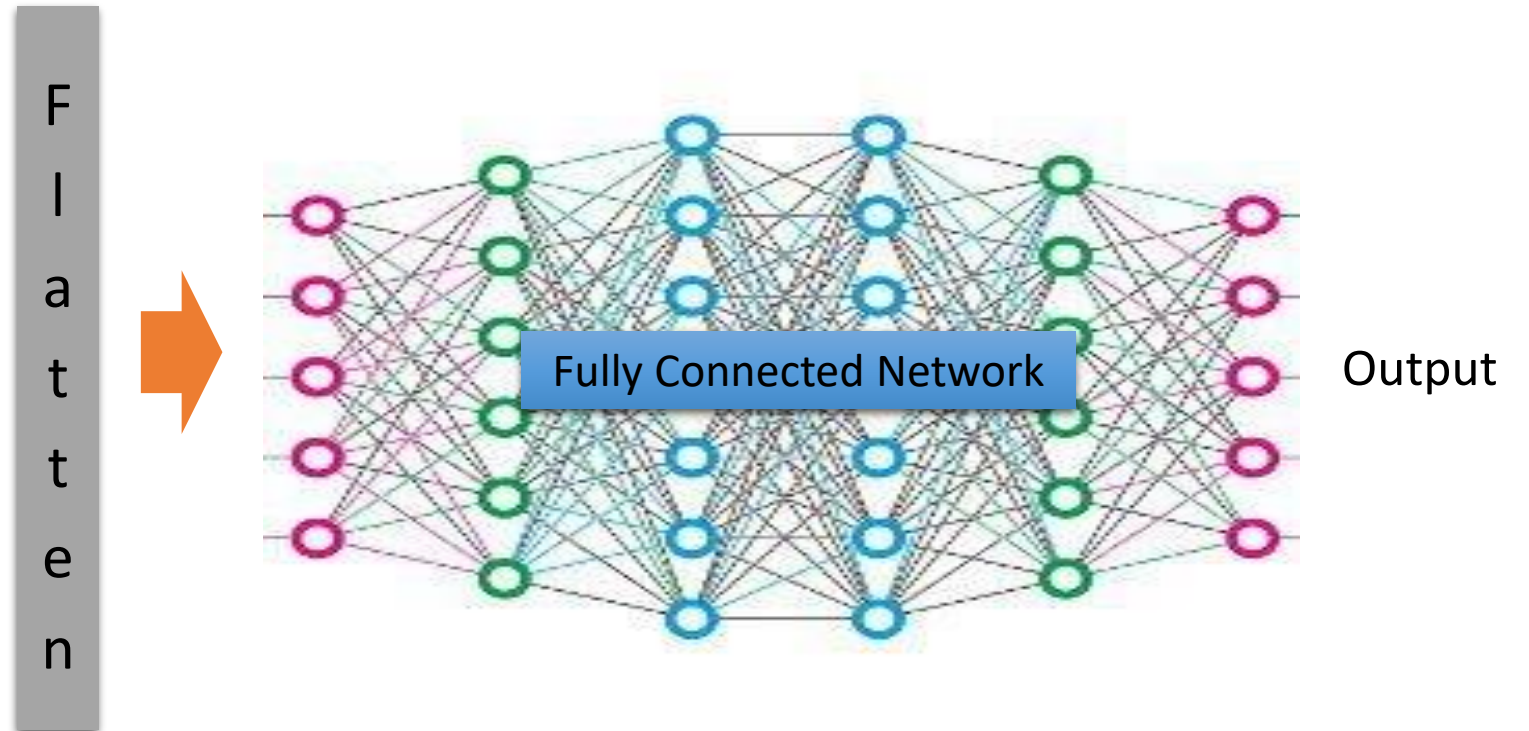
Max Pooling

Feature Map

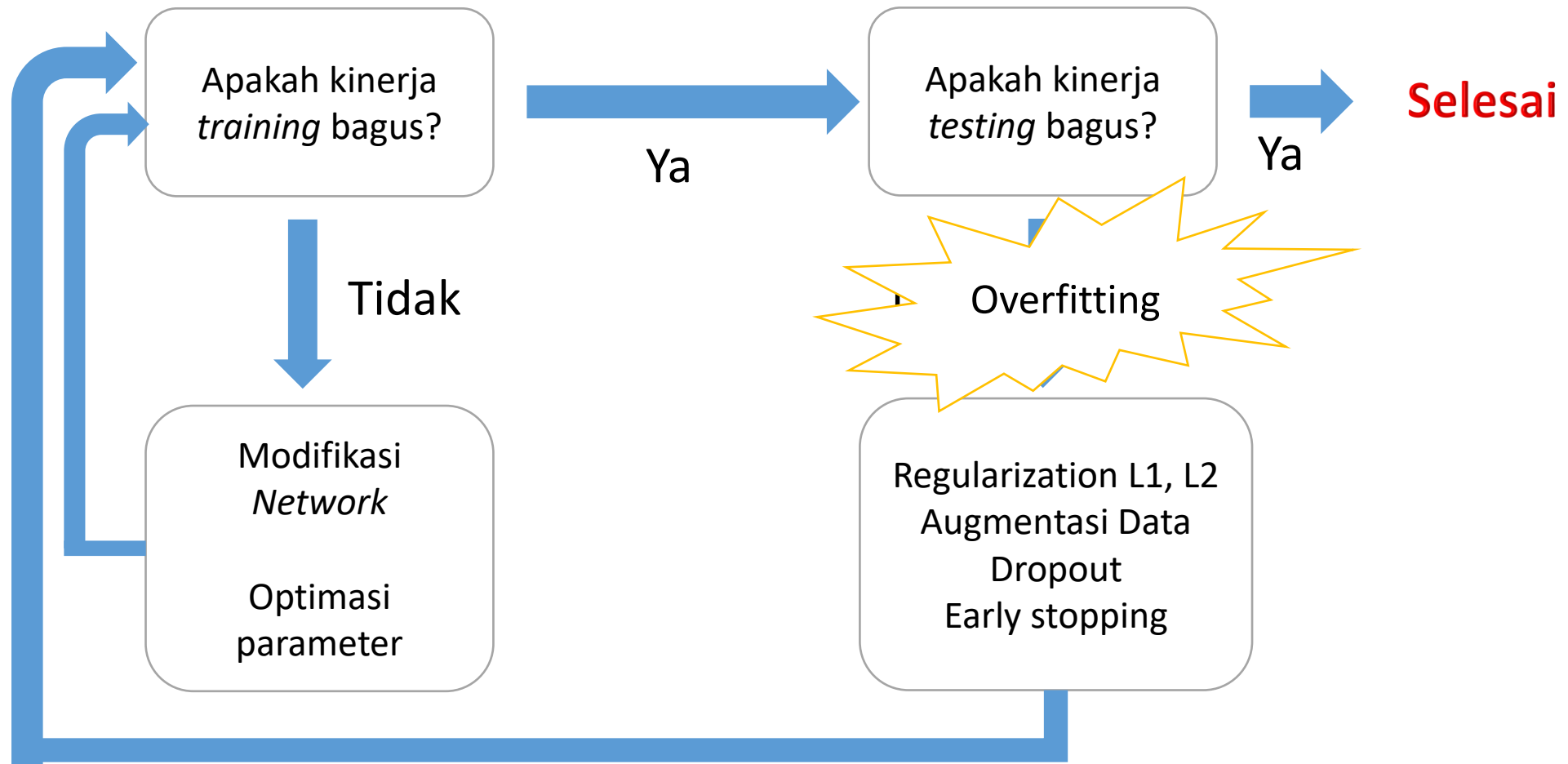
Layer
Max
Pooling



Full Connected Network Layer



Strategi Proses Pembelajaran



<http://www.gizmodo.com.au/2015/04/the-basic-recipe-for-machine-learning-explained-in-a-single-powerpoint-slide/>



Modifikasi *Network*

- Merubah arsitektur
- Merubah fungsi aktivasi

Memilih Metode Optimasi

- *Adaptive learning rate*

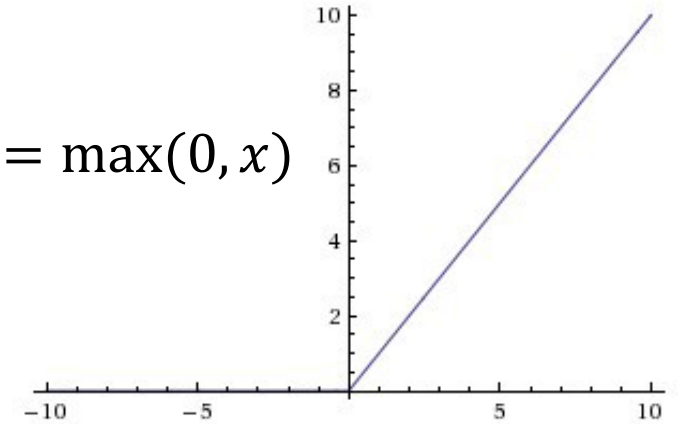
Mencegah *Overfitting*

- *Dropout*
- *Augmentasi data*

Modifikasi Network

- Merubah arsitektur, misalnya menambah jumlah hidden layer, jumlah neuron, atau jenis arsitektur lain
- Merubah fungsi aktivasi, misalnya menggunakan ReLU

$$f(x) = \max(0, x)$$



Rectified Linear Unit function (ReLU)

Optimasi parameter

Nilai *learning rate* berpengaruh pada perhitungan bobot baru, umumnya penggunaan *learning rate* yang menyesuaikan nilai gradien (*adaptive learning rate*) menunjukkan kinerja model yang lebih baik. Contoh *algoritma adaptive learning rate*:

- Adagrad [John Duchi, JMLR 2011]
- Adadelata [Matthew D. Zeiler, arXiv 2012]
- Adam [Diederik P. Kingma, ICLR 2015]
- AdaSecant [Caglar Gulcehre, arXiv 2014]
- RMSprop <https://www.youtube.com/watch?v=O3sxAc4hxZU>

Mencegah *Overfitting*

Regularization

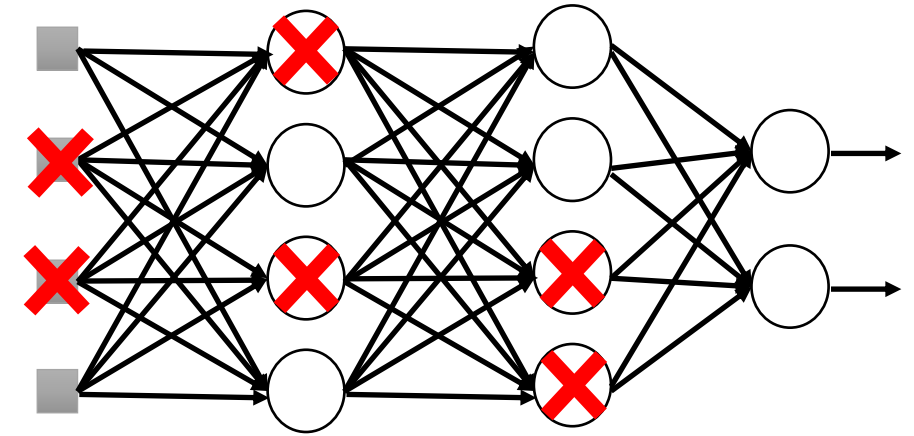
Regularisasi dilakukan untuk mengurangi *generalization error* dengan mencegah model lebih kompleks. Penerapan regularisasi dengan cara menambahkan *regularization term* pada semua parameter (bobot) ke fungsi obyektif.

- **Regularization L1 norm**
 - Menambahkan *sum of the absolute weights* sebagai *penalty term* ke fungsi obyektif
- **Regularization L2 norm (*weight decay*)**
 - Menambahkan *sum of the squared weights* sebagai *penalty term* ke fungsi obyektif

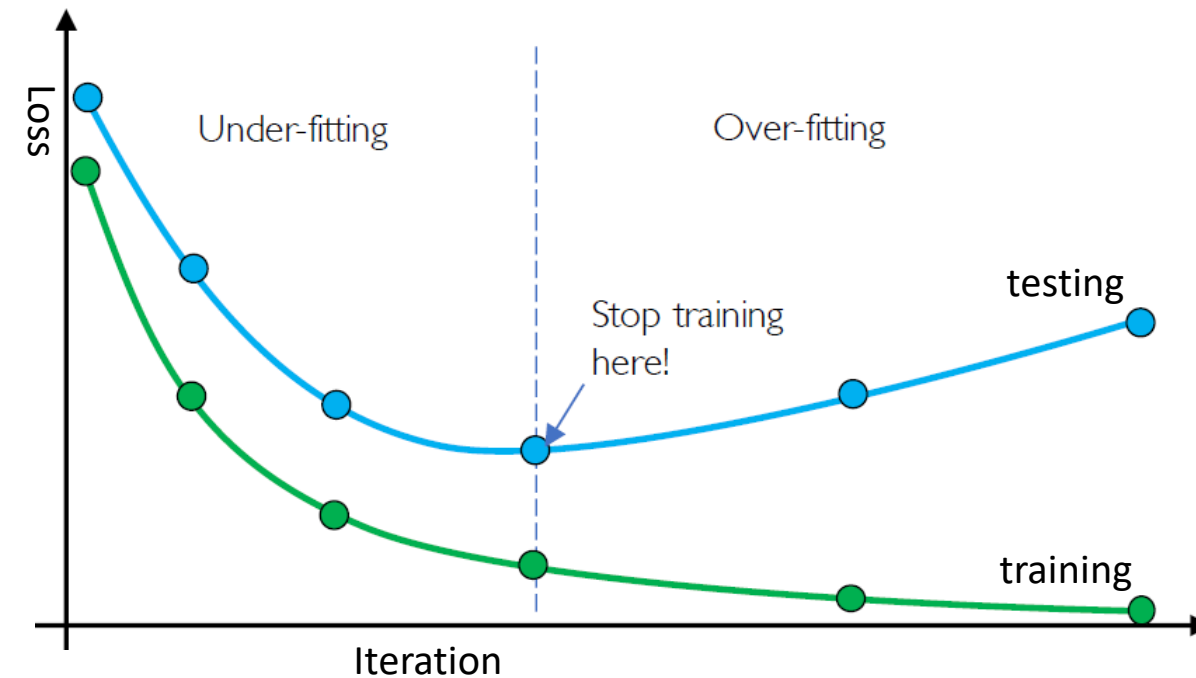
Mencegah Overfitting

Cara meregulasi parameter untuk menghindari *overfitting* sehingga model lebih general

- **Dropout**
 - Umumnya dilakukan pada fully connected layer
 - **Selama proses training, setiap iterasi sebelum menghitung gradient**
 - Setiap neuron diset tidak aktif dengan prosentase dropout $p\%$
- **Early stopping**
 - Iterasi pada saat training dihentikan jika *generalization error* mulai naik



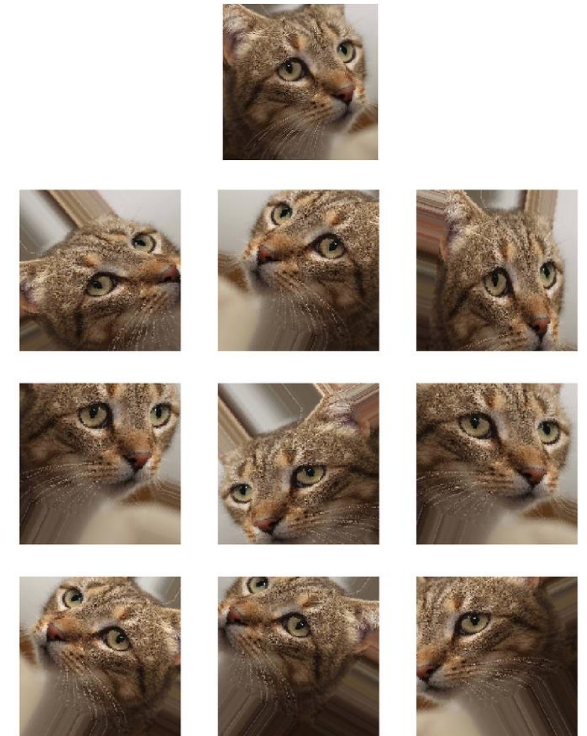
Struktur network berubah lebih ringan



Mencegah *Overfitting*

Menambah Data Latih (Augmentasi Data)

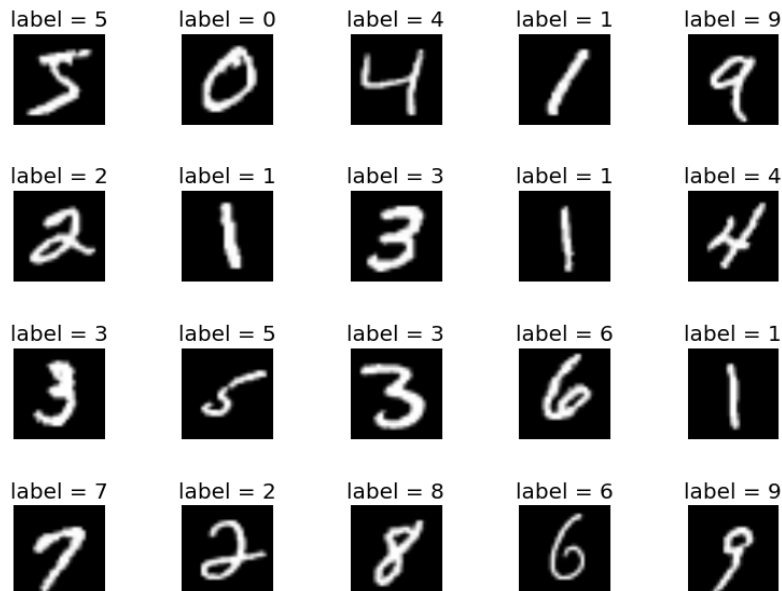
- Proses memperbanyak variasi data latih, sehingga model yang dihasilkan lebih baik dalam memprediksi data uji
- Metode augmentasi data yang digunakan tergantung dari jenis data input
- Metode *oversampling* data numerik: smote, adasyn, dan sebagainya
- Contoh augmentasi data citra: rotasi, translasi, flip, dan zoom



MNIST Handwritten Digits

Aplikasi CNN pada Pengenalan Angka

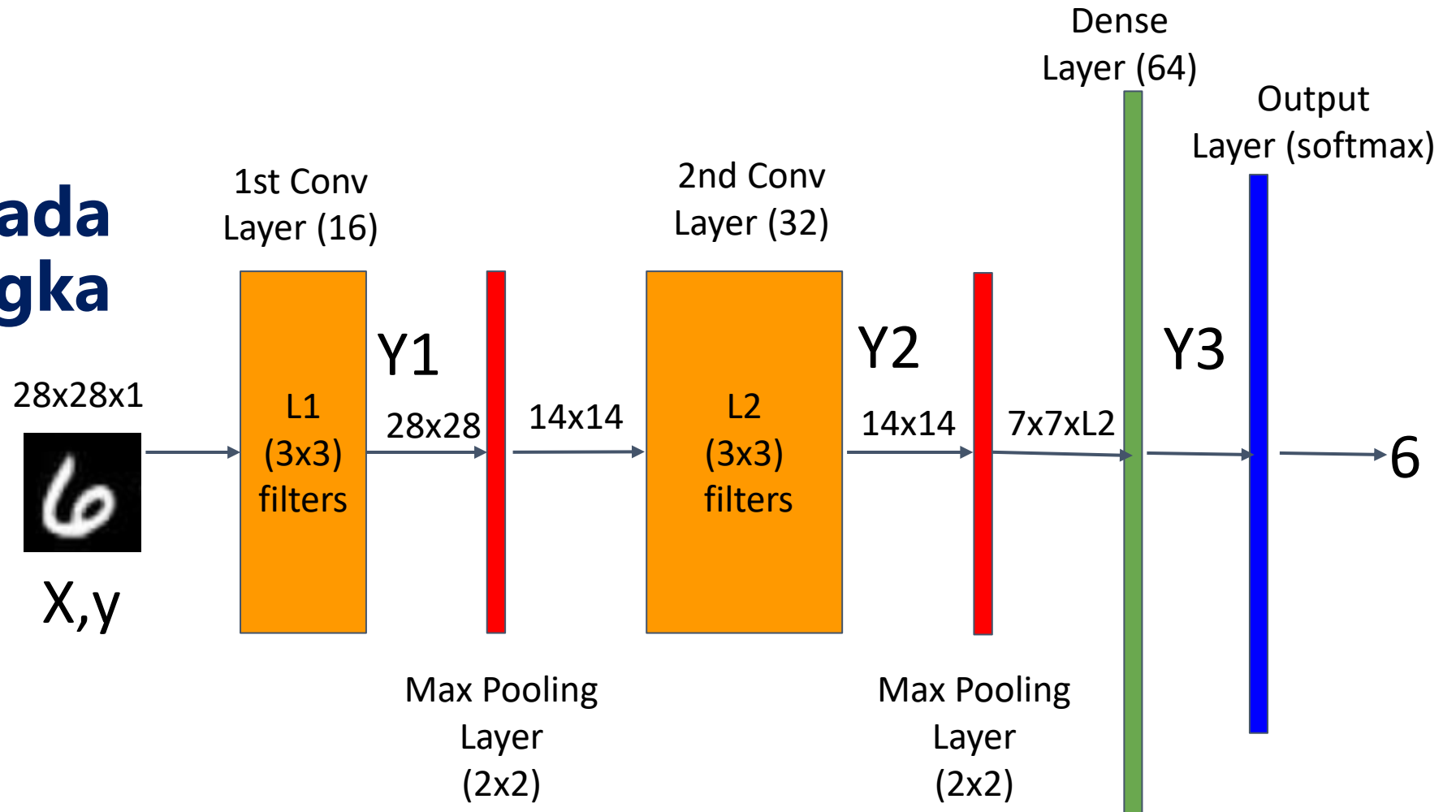
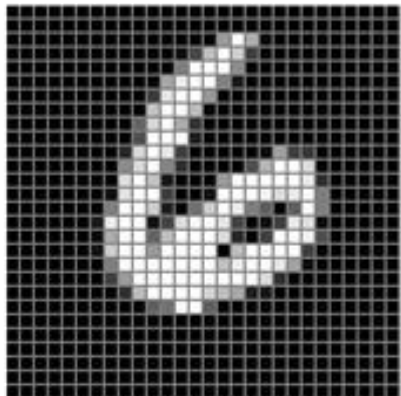
- MNIST dataset dibagi menjadi 3 bagian:
 - 55,000 training data
 - 10,000 test data
 - 5,000 validation data
- Setiap gambar berukuran 28×28 pixels
- Label kelas berupa *one hot encoded*



0	[1 0 0 0 0 0 0 0 0 0]
1	[0 1 0 0 0 0 0 0 0 0]
2	[0 0 1 0 0 0 0 0 0 0]
3	[0 0 0 1 0 0 0 0 0 0]
4	[0 0 0 0 1 0 0 0 0 0]
5	[0 0 0 0 0 1 0 0 0 0]
6	[0 0 0 0 0 0 1 0 0 0]
7	[0 0 0 0 0 0 0 1 0 0]
8	[0 0 0 0 0 0 0 0 1 0]
9	[0 0 0 0 0 0 0 0 0 1]

MNIST Handwritten Digits

Aplikasi CNN pada Pengenalan Angka



Aplikasi CNN pada Pengengenalan Angka

CNN dengan Keras

```
model = Sequential()
model.add(Conv2D(16,(3,3),activation='relu',input_shape=(28,28,1),padding='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(32,(3,3),activation='relu',padding='same'))
model.add(MaxPooling2D(2,2))
model.add(Flatten())
model.add(Dense(64,activation='relu'))
model.add(Dense(10,activation='softmax'))
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 28, 16)	160
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 16)	0
conv2d_2 (Conv2D)	(None, 14, 14, 32)	4640
max_pooling2d_2 (MaxPooling2D)	(None, 7, 7, 32)	0
flatten_1 (Flatten)	(None, 1568)	0
dense_1 (Dense)	(None, 64)	100416
dense_2 (Dense)	(None, 10)	650
Total params: 105,866		
Trainable params: 105,866		
Non-trainable params: 0		

CNN dengan Keras

Aplikasi CNN pada Pengengenan Angka

```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['acc'])
history = model.fit(X_train,y_train,epochs=10,batch_size=100,validation_data=(X_test,y_test))
model.evaluate(X_test,y_test)
```

313/313 [=====] - 1s 4ms/step - loss: 0.0277 - acc: 0.9906

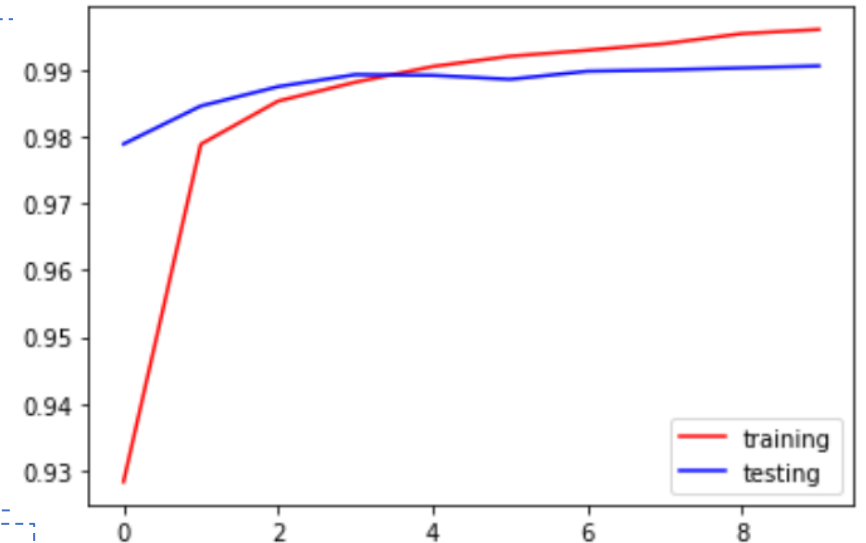
Out[12]: [0.02769605629146099, 0.9905999898910522]

```
acc = history.history['acc']
val_acc = history.history['val_acc']

epochs = range(10)

import matplotlib.pyplot as plt
plt.plot(epochs,acc,'r',label='training')
plt.plot(epochs,val_acc,'b',label='testing')
plt.legend()
```

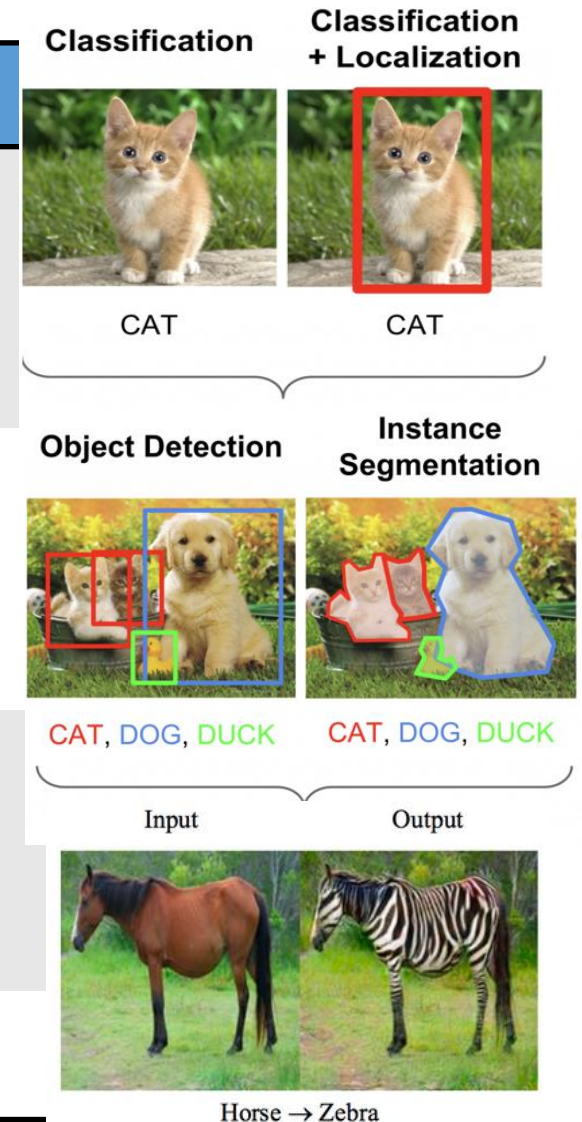
```
pred = model.predict(X_test)
print('label actual:',np.argmax(y_test[0]))
print('label prediction:',np.argmax(pred[0]))
```



label actual: 7
label prediction: 7

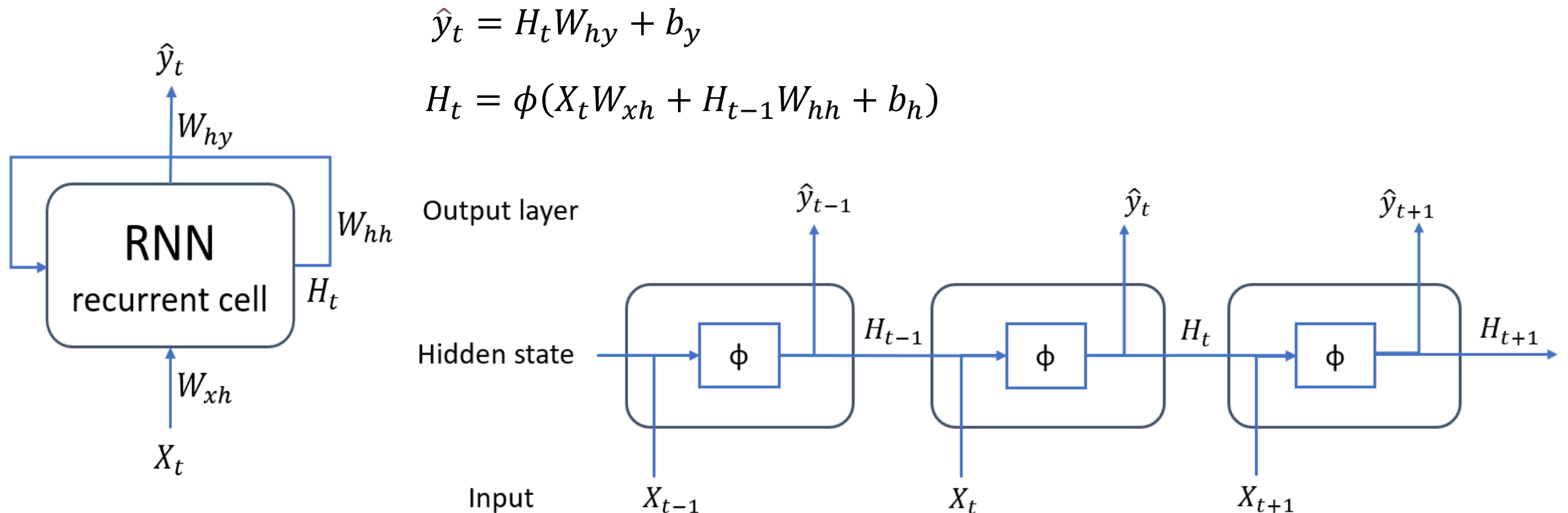
Varian dari Arsitektur CNN dan Tipe Aplikasinya

Aplikasi	Arsitektur CNN
Image Classification	<ul style="list-style-type: none"> LeNet-5 (1998) AlexNet (2012) GoogLeNet/Inception (2014) VGGNet (2014) ResNet (2015)
Object Detection	<ul style="list-style-type: none"> R-CNN (2013) Fast R-CNN (2014) Faster R-CNN (2015) Single Shot Detector (SSD) (2016) YOLO (2016), YOLOv3 (2018), YOLOv4 (2020), YOLOv5 (2020)
Semantic (Instance) Segmentation	<ul style="list-style-type: none"> Fully Convolutional Network (FCN) (2015) U-Net (2015) Feature Pyramid Network (FPN) (2016) Mask R-CNN (2017) DeepLab (2016), DeepLabv3 (2017), DeepLabv3+ (2018)
Generative model	<ul style="list-style-type: none"> Autoencoders, Variational Autoencoders (VAE) Generative Adversarial Network (GAN)

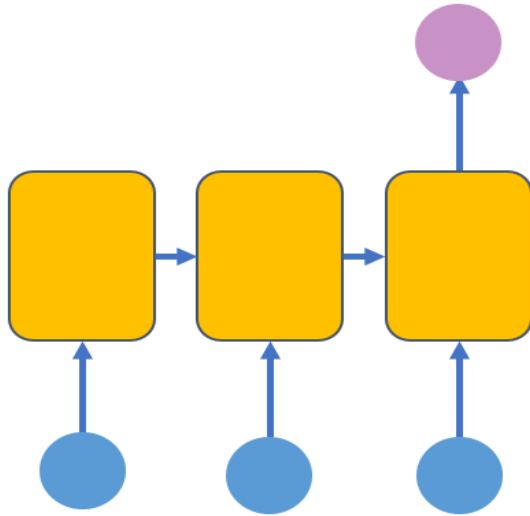


Recurrent Neural Network

Recurrent Neural Network (RNN) adalah salah satu arsitektur ANN yang mampu merepresentasikan data *sequential* misalnya teks, DNA, suara, *time series*, dan sebagainya



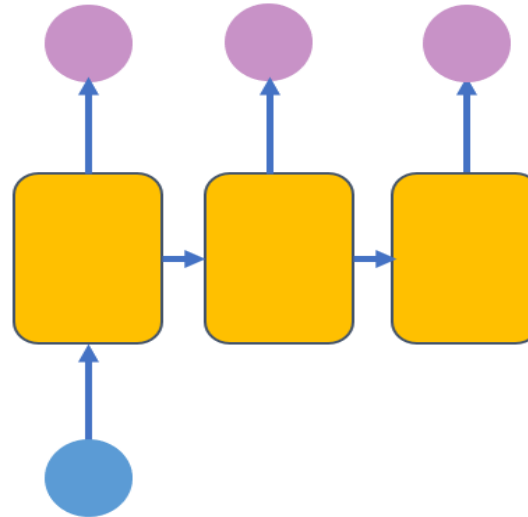
Tipe Arsitektur RNN dan Aplikasinya



Many to One

Applications:

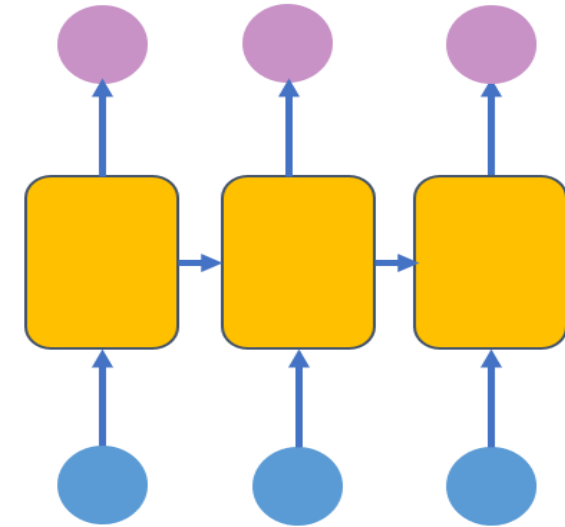
- Sentiment classification
- Opinion mining
- Speech recognition
- Automated answer scoring



One to Many

Applications:

- Image captioning
- Text generation

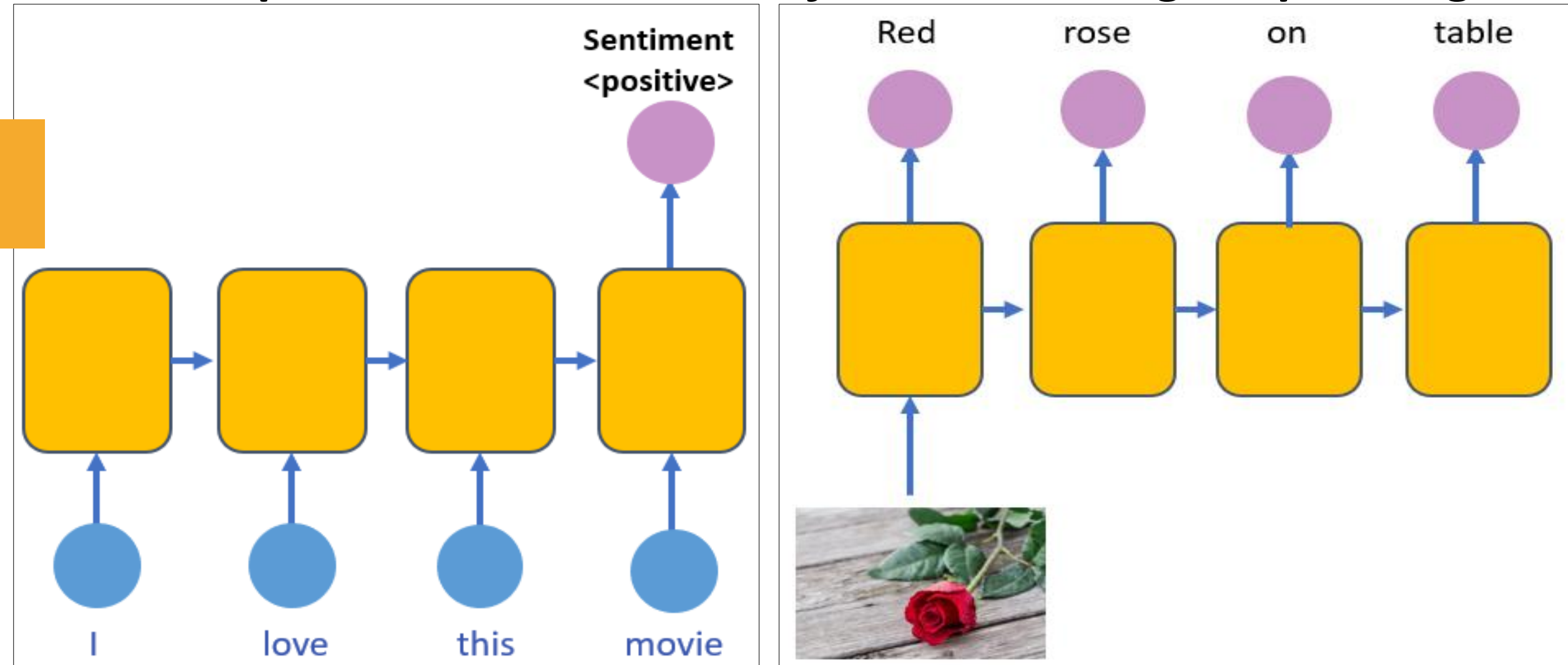


Many to Many

Applications:

- Translation
- Forecasting
- Chatbot
- Music generation

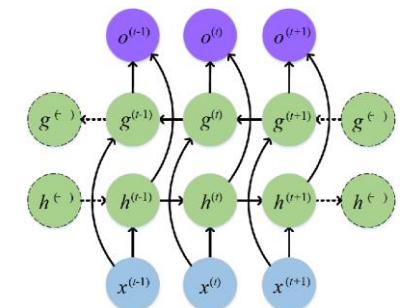
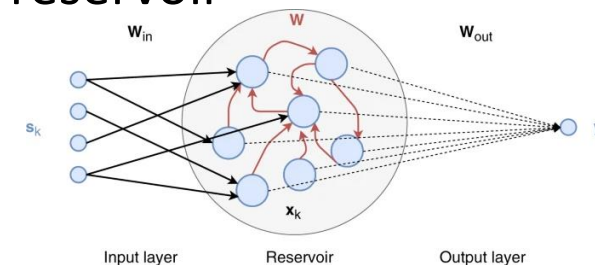
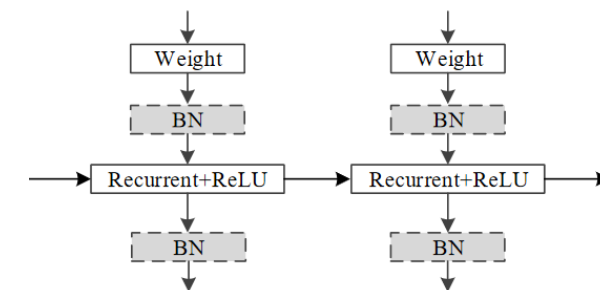
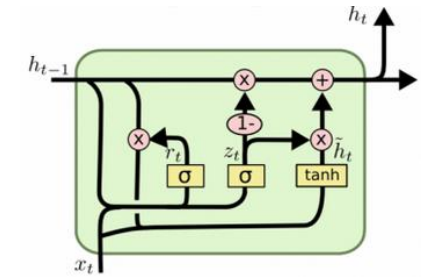
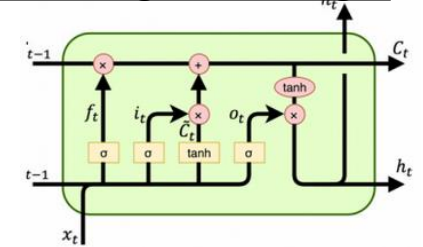
Contoh Aplikasi: *Sentiment classification & Image captioning*



Varian Arsitektur RNN

- **Long Short-Term Memory (LSTM):** merupakan salah satu jenis arsitektur RNN yang terdiri dari beberapa unit yaitu input gate, output gate, dan forget gate
- **Gate Recurrent Unit (GRU):** merupakan simplifikasi dari arsitektur LSTM dengan menggabungkan input gate dan forget gate sehingga jumlah parameter lebih sedikit
- **Independently RNN (IndRNN):** arsitektur RNN dimana setiap neuron dalam satu layer independen dari yang lain
- **Bi-directional RNN:** merupakan arsitektur RNN menghubungkan dua hidden layer dari arah yang berlawanan ke output yang sama.
- **Echo State Network (ESN):** ide dasar ESN adalah untuk membuat jaringan berulang yang terhubung secara acak, yang disebut reservoir

<http://dprogrammer.org/rnn-lstm-gru>



Summary

FCN

Data numerik

Jumlah hidden layer
sesuai kompleksitas
permasalahan

Klasifikasi dan regresi

CNN

Data citra, video

Convolution & Pooling
layer

Klasifikasi, deteksi obyek,
instance segmentation,
generate citra sintetis

RNN

Data text, signal, suara,
time-series

Konsep recurrent dan
memperhatikan urutan
input

Klasifikasi, regresi,
generate text, translation



- TERIMA KASIH -

