# Deep Learning untuk Prediksi Gerbang XOR

```
Oleh: Tim IO-T.NET (https://io-t.net/itclab)
```

In [3]:

```
# Library yg dibutuhkan
import numpy as np
```

In [4]:

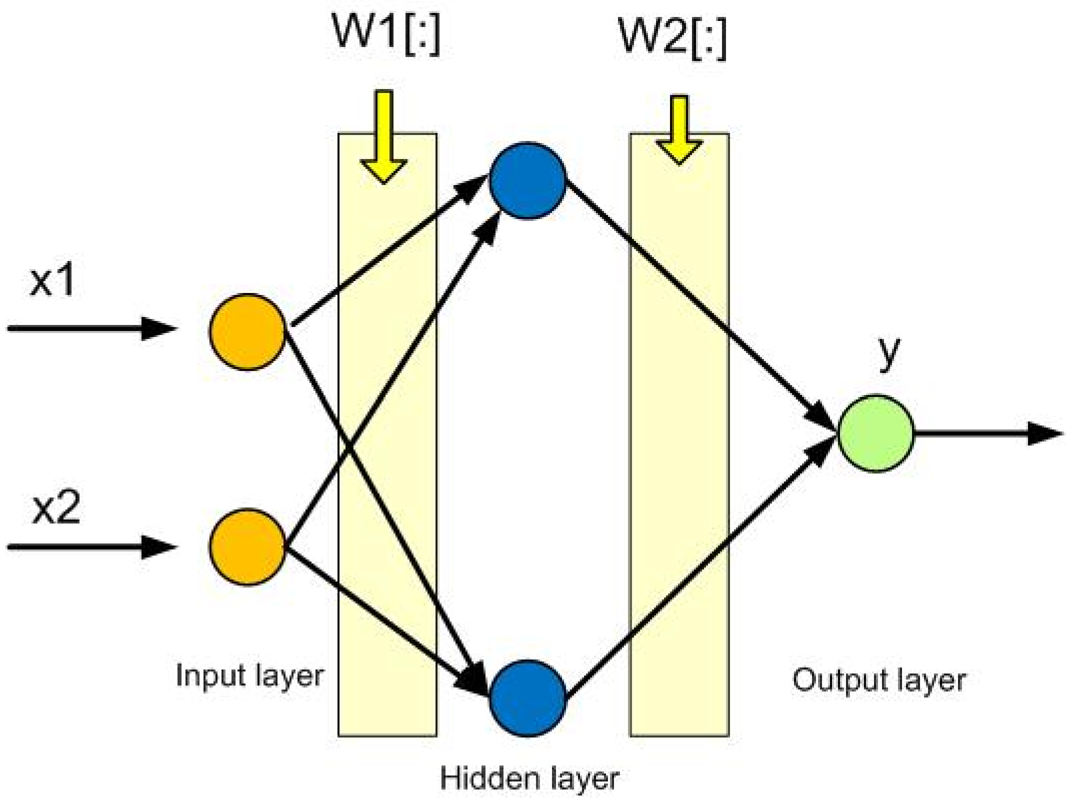
```
# Pasangan data latih

XOR_X = np.array([
    [0, 0],
    [0, 1],
    [1, 0],
    [1, 1]
])

XOR_Y = np.array([
    [0],
    [1],
    [1],
    [0]
])
```

## Arsitektur Deep Learning

Arsitektur Deep Learning dengan Dua Masukan dan Satu Keluaran


W1[:]
W2[:]
x1
x2
y
Input layer
Hidden layer
Output layer

In [6]:

```
# Impor `Sequential` dari` keras.models`
from keras.models import Sequential

# Impor `Dense` dari` keras.layers`
from keras.layers import Dense

# Inisialisasi konstruktor
model = Sequential()

# Tambahkan lapisan masukan
model.add(Dense(2, activation='sigmoid', input_shape=(2,)))

# Tambahkan satu lapisan tersembunyi
model.add(Dense(2, activation='sigmoid'))

# Tambahkan lapisan keluaran
model.add(Dense(1, activation='sigmoid'))
```

Ketikkan skrip berikut ini, untuk model Deep Learning-nya, dan dapatkan bobot-bobot dan bias awal.

In [7]:

```
# Bentuk keluaran model
model.output_shape

# Ringkasan model
model.summary()

# Konfigurasi model
model.get_config()

# Buat daftar semua tensor bobot
model.get_weights()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_3 (Dense)              (None, 2)                 6         
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 6         
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 3         
=================================================================
Total params: 15
Trainable params: 15
Non-trainable params: 0
_________________________________________________________________
```

Out[7]:

```
[array([[ 0.47920144, -0.10224676],
        [-0.6128937 , -1.1116478 ]], dtype=float32),
 array([0., 0.], dtype=float32),
 array([[-0.10563552, -0.4309035 ],
        [ 0.9424602 , -0.3766948 ]], dtype=float32),
 array([0., 0.], dtype=float32),
 array([[ 0.8593122],
        [-1.3878404]], dtype=float32),
 array([0.], dtype=float32)]
```

Untuk pelatihan Deep Learning silahkan ketikkan skrip berikut.

In [13]:

```
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

model.fit(XOR_X, XOR_Y,epochs=1000, batch_size=1, verbose=1)
```

```
Epoch 1/1000
4/4 [==============================] - 1s 2ms/step - loss: 1.4030e-08 - accuracy: 1.0000
Epoch 2/1000
4/4 [==============================] - 0s 3ms/step - loss: 1.4030e-08 - accuracy: 1.0000
Epoch 3/1000
4/4 [==============================] - 0s 3ms/step - loss: 1.4030e-08 - accuracy: 1.0000
Epoch 4/1000
4/4 [==============================] - 0s 3ms/step - loss: 1.4030e-08 - accuracy: 1.0000
Epoch 5/1000
4/4 [==============================] - 0s 3ms/step - loss: 1.4030e-08 - accuracy: 1.0000
Epoch 6/1000
4/4 [==============================] - 0s 3ms/step - loss: 1.4030e-08 - accuracy: 1.0000
Epoch 7/1000
4/4 [==============================] - 0s 3ms/step - loss: 1.4030e-08 - accuracy: 1.0000
Epoch 8/1000
4/4 [==============================] - 0s 3ms/step - loss: 1.4030e-08 - accuracy: 1.0000
Epoch 9/1000
4/4 [==============================] - 0s 3ms/step - loss: 1.4030e-08 - accuracy: 1.0000
Epoch 10/1000
4/4 [==============================] - 0s 3ms/step - loss: 1.4030e-08 - accuracy: 1.0000
```

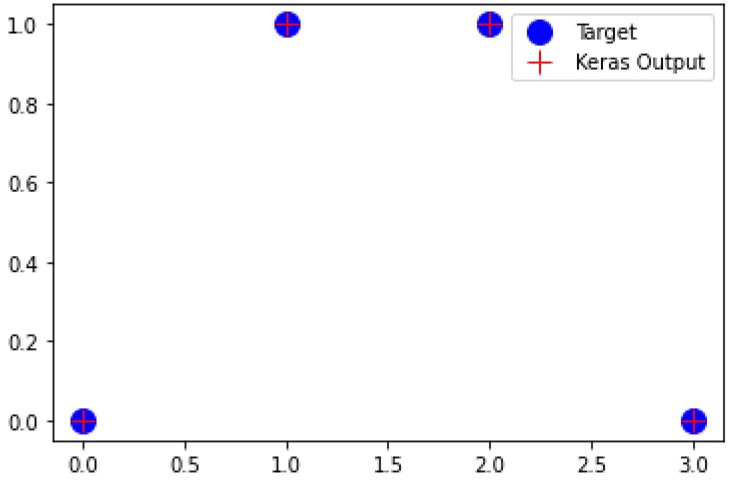
In [14]:

```
Hasil_Prediksi_Keras = model.predict(XOR_X)
print(Hasil_Prediksi_Keras)
```

```
[[1.4407077e-09]
 [1.0000000e+00]
 [1.0000000e+00]
 [1.3886289e-09]]
```

In [15]:

```
import matplotlib.pyplot as plt
plt.plot(XOR_Y, 'bo', label='Target', linewidth=2, markersize=12)
plt.plot(Hasil_Prediksi_Keras, 'r+', label='Keras Output', linewidth=2, markersize=12)
plt.legend(loc='upper right')
plt.show()
```



In [16]:

```
from sklearn.metrics import mean_squared_error
from math import sqrt
mse2  = mean_squared_error(XOR_Y, Hasil_Prediksi_Keras)
rmse2 = sqrt(mean_squared_error(XOR_Y, Hasil_Prediksi_Keras))
print('MSE =',mse2)
print('RMSE =',rmse2)
```

```
MSE = 1.0009821840653178e-18
RMSE = 1.000490971506149e-09
```

In [ ]: