

Komputasi Finansial

I Putu Indra Aristya – 1301154219



**Klasifikasi Kebangkrutan menggunakan
Multilayer Perceptron dan *k-Nearest Neighbors***

*Bankruptcy Classification using Multilayer
Perceptron and k-Nearest Neighbors*



Petunjuk Pembuatan Slide

- 1. Lama Presentasi dengan slide ditambah demo program +/- 10 menit, yang kemudian akan dilanjutkan dengan tanya jawab;**
- 2. Isi dari slide presentasi pada makalah ilmiah yang ditulis mencakup:**
 - Abstrak
 - Latar Belakang
 - Identifikasi dan Perumusan Masalah
 - Maksud dan Tujuan
 - Perumusan Matematika Yang digunakan
 - Algoritma yang digunakan
 - Tinjauan Pustaka(Tinjauan Penelitian Terdahulu (bila ada), serta teori yang digunakan untuk menjadi landasan dilakukannya penelitian)
 - Hipotesis
 - Metodologi/Rancangan Penelitian
 - Hasil Pengamatan sekaligus Kompleksitas Waktu Algoritma yang digunakan
 - Analisa
 - Kesimpulan



Latar Belakang



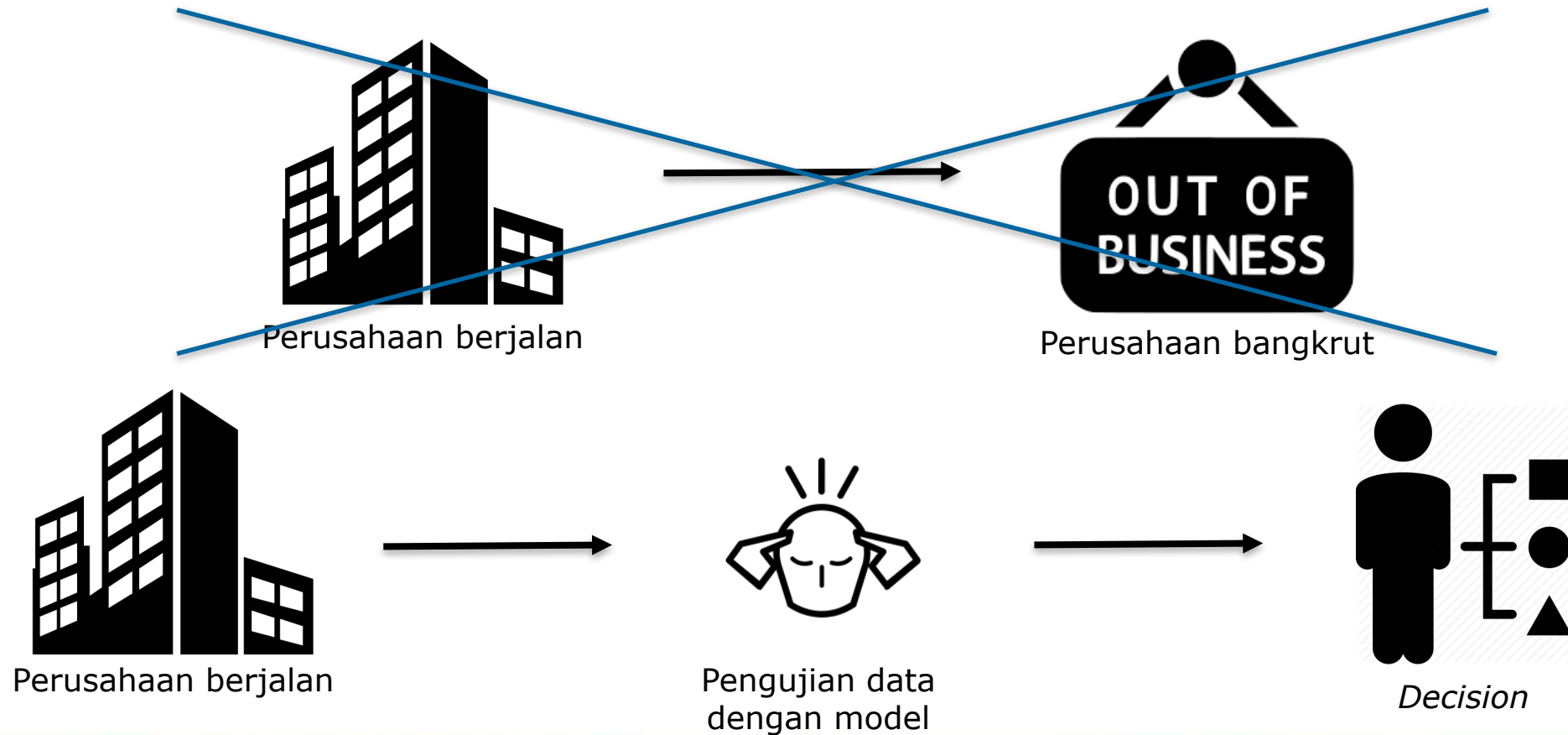
Perusahaan berjalan



Perusahaan bangkrut



Latar Belakang





Tujuan

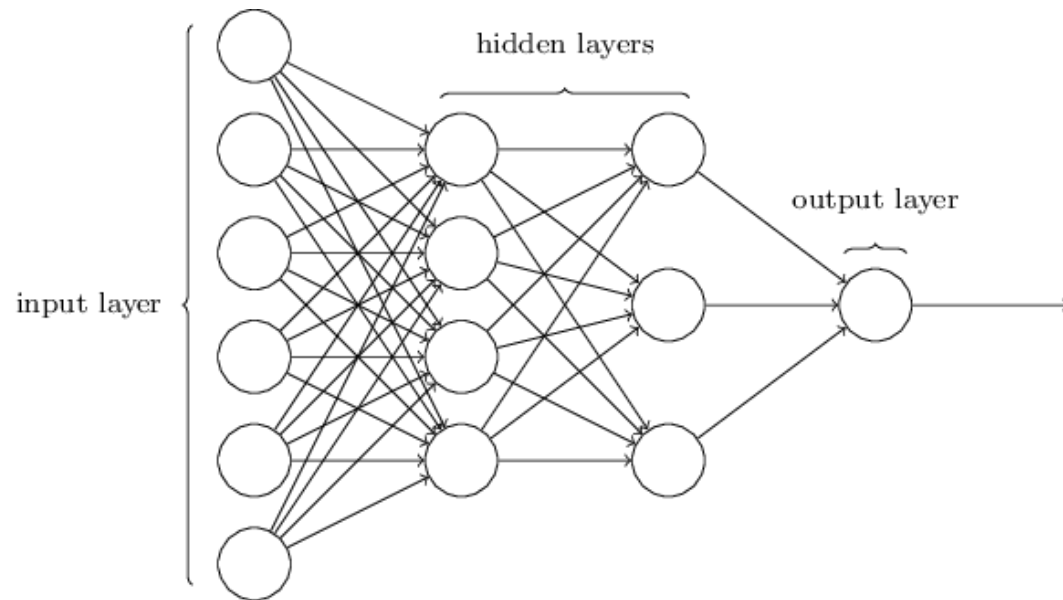
- ▶ Diharapkan dengan model yang dibuat dapat membantu perusahaan – perusahaan melakukan pengecekan berkala dari laporan keuangan dengan kebangkrutan
- ▶ Dengan ditentukannya kebangkrutan tersebut, dapat dengan cepat dan tepat menentukan keputusan yang akan dibuat selanjutnya

Permasalahan

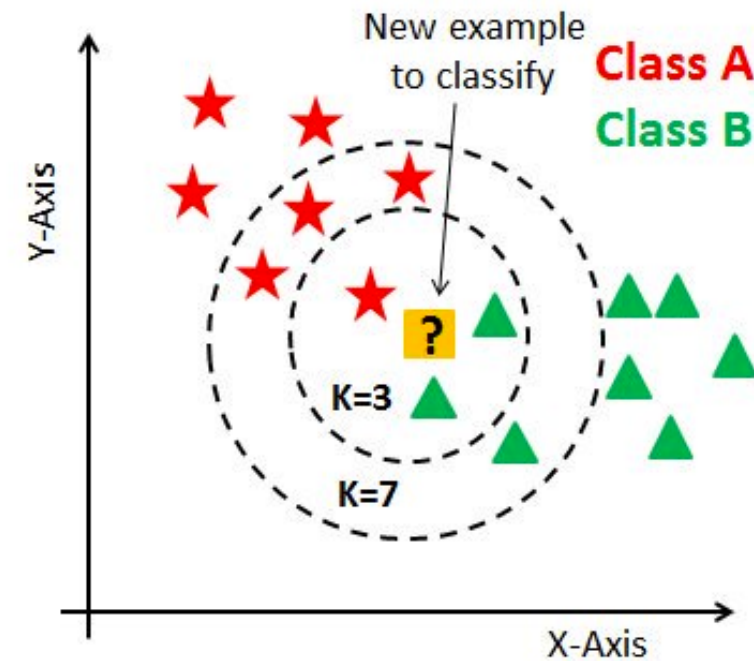
Data?

Model?

Algoritma

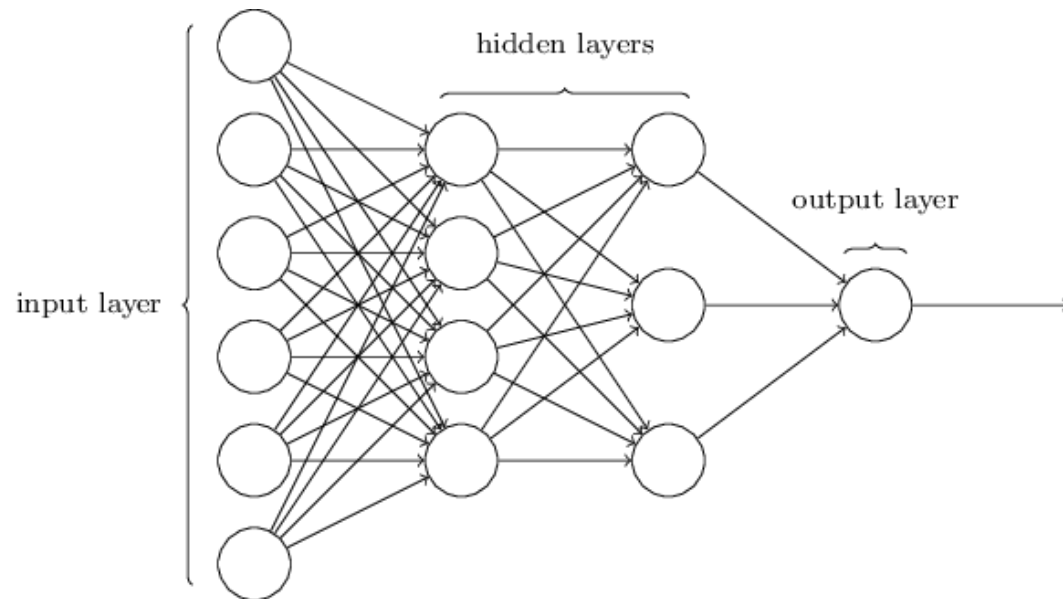


Multilayer Perceptron



K-Nearest Neighbors

Algoritma: MLP



Multilayer Perceptron

Sum Function:

- $f(x) = \text{sum}((w.x + b))$

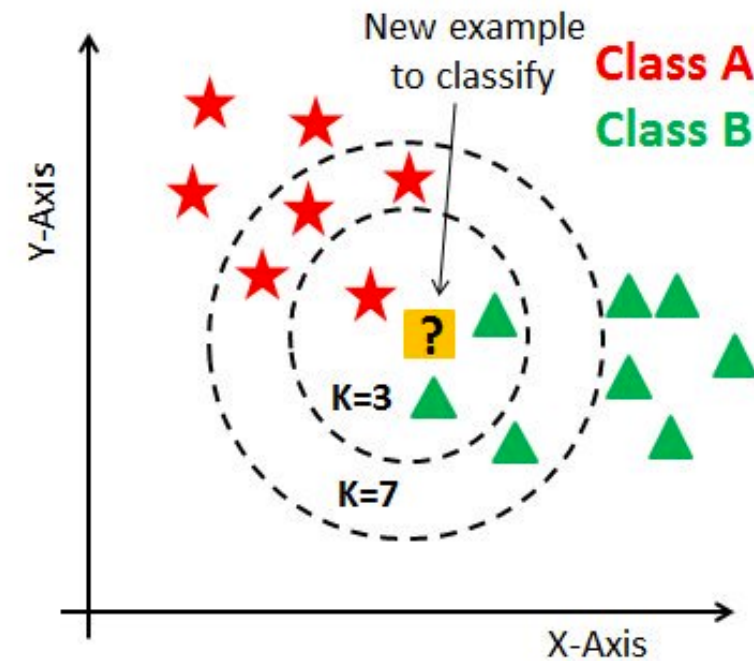
ReLU Activation:

- $f(x) = \max(0, x)$
- $f(x)' = 0; x < 0$ atau $1; x \geq 0$

Algoritma: k-Nearest Neighbors

Euclidean Distance:

- distance = $\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$

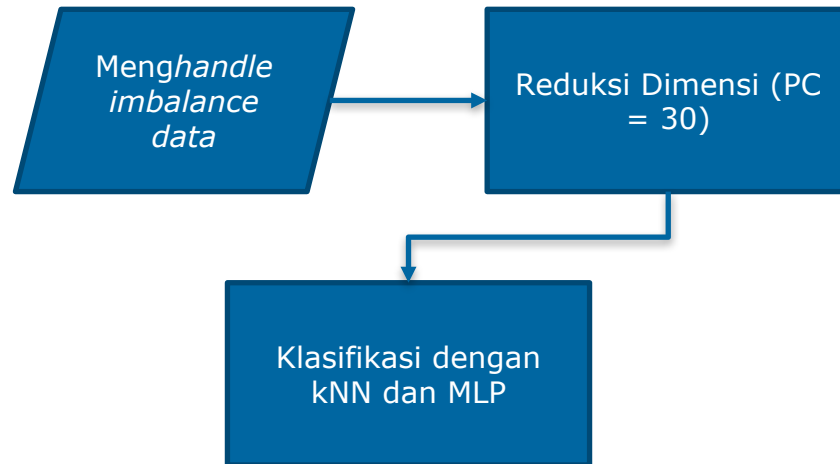


k-Nearest Neighbors

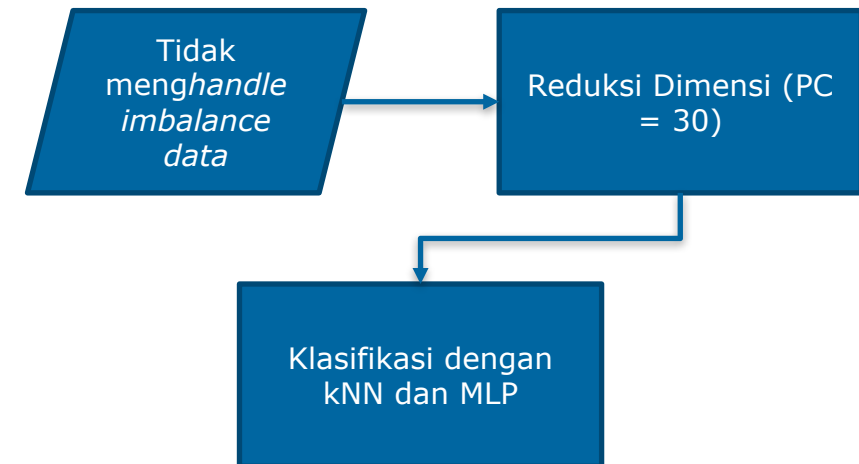
Tinjauan Pustaka

Comparative Analysis of Data Mining Methods for Bankruptcy Prediction (2011)	David L. Olson, Dursun Delen, Yanyan Meng	Decision Tree C5 93.7%
Bankruptcy Prediction using Data Mining Techniques (2017)	Manil Wagle, Zijiang Yang, Younes Benslimane	Acc NN 70.83% (semua fitur) Acc NN 85.33% (filter fitur)
Bankruptcy Prediction using Neural Network (2018)	G. Pranav Naidu, Govinda. K.	Error NN 4.4349% Error Random Forest 5.1954%.

Skenario Penelitian



Mencoba klasifikasi dengan MLP dan kNN pada data yang dilakukan *oversampling*



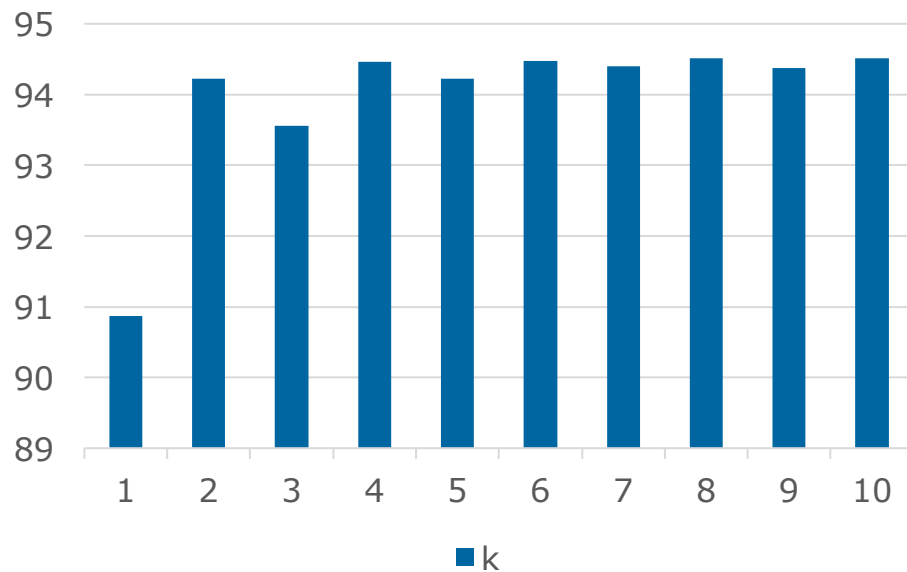
Mencoba klasifikasi dengan MLP dan kNN pada data yang **tidak** dilakukan *oversampling*



Hasil Pengamatan: Hasil Akurasi Validasi kNN

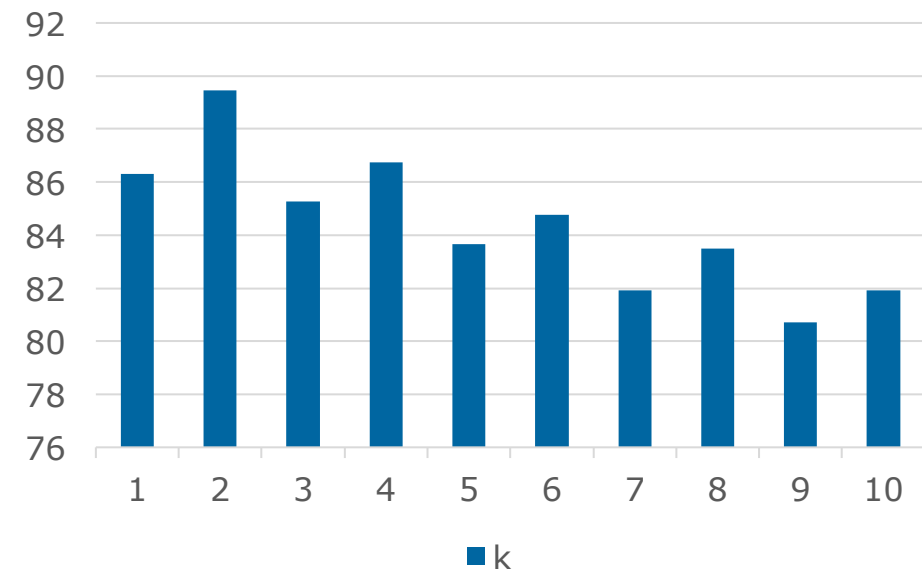
tanpa oversampling

Nilai Akurasi menggunakan kNN



dengan oversampling

Nilai Akurasi menggunakan kNN





Hasil Pengamatan: Hasil Akurasi Validasi MLP

tanpa oversampling

dengan oversampling

```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,  
              beta_2=0.999, early_stopping=False, epsilon=1e-08,  
              hidden_layer_sizes=(100,), learning_rate='constant',  
              learning_rate_init=0.001, max_iter=200, momentum=0.9,  
              n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,  
              random_state=None, shuffle=True, solver='adam', tol=0.0001,  
              validation_fraction=0.1, verbose=False, warm_start=False)
```

=

```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,  
              beta_2=0.999, early_stopping=False, epsilon=1e-08,  
              hidden_layer_sizes=(100,), learning_rate='constant',  
              learning_rate_init=0.001, max_iter=200, momentum=0.9,  
              n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,  
              random_state=None, shuffle=True, solver='adam', tol=0.0001,  
              validation_fraction=0.1, verbose=False, warm_start=False)
```

91.06%

<

69.01%



Hasil Pengamatan: Kompleksitas kNN (*training*)

Kompleksitas (jumlah operasi) kNN bergantung kepada jumlah data dan jumlah variabelnya. Pada kNN yang tidak melakukan proses pelatihan (hanya menghitung jarak antar data) memiliki kompleksitas yang cukup besar jika data yang ada semakin banyak. Dapat dihitung kompleksitasnya dengan menghitung operasi yang dilakukan menggunakan rumus sebagai berikut.

$$\text{operasi} = (\text{jumlah data latih} \times \text{jumlah data uji}) \times \text{jumlah variabel setiap data} \times \text{operasi setiap variabel}$$

tanpa oversampling

$$\begin{aligned}\text{Operasi} &= (6000 \times 3792) \times 30 \times 2 \\ &= 1.365.120.000\end{aligned}$$

dengan oversampling

$$\begin{aligned}\text{Operasi} &= (8537 \times 3792) \times 30 \times 2 \\ &= 1.942.338.240\end{aligned}$$

Hasil Pengamatan: Kompleksitas MLP

Kompleksitas MLP bergantung kepada jumlah data, *neuron* setiap *layer*-nya, jumlah *layer*-nya dan jumlah iterasi yang dilakukan. Kompleksitasnya dapat dihitung dengan menggunakan rumus sebagai berikut.

$$\text{kompleksitas} = O(\text{jumlah data} \times \text{jumlah variabel} \times \text{jumlah neuron}^{\text{jumlah layer}} \times \text{jumlah iterasi})$$

tanpa *oversampling*

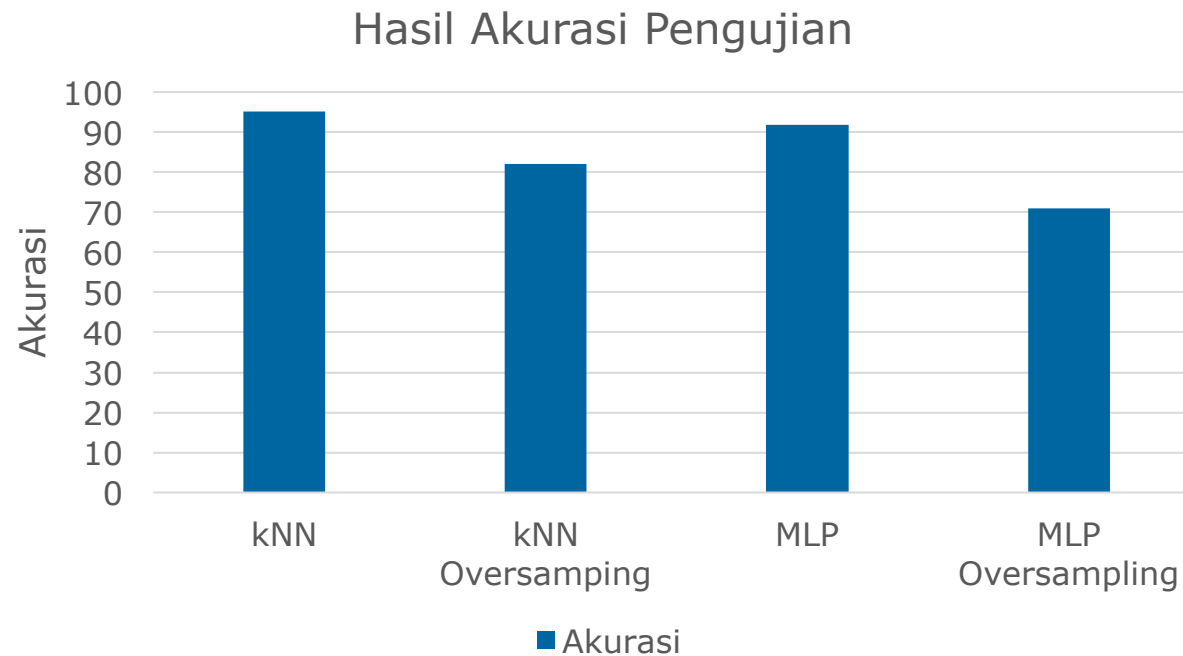
$$\begin{aligned}\text{Kompleksitas} &= O(6000 \times 30 \times 100^1 \times 200) \\ &= O(3,600,000,000)\end{aligned}$$

dengan *oversampling*

$$\begin{aligned}\text{Kompleksitas} &= O(8537 \times 30 \times 100^1 \times 200) \\ &= O(5,122,200,000)\end{aligned}$$

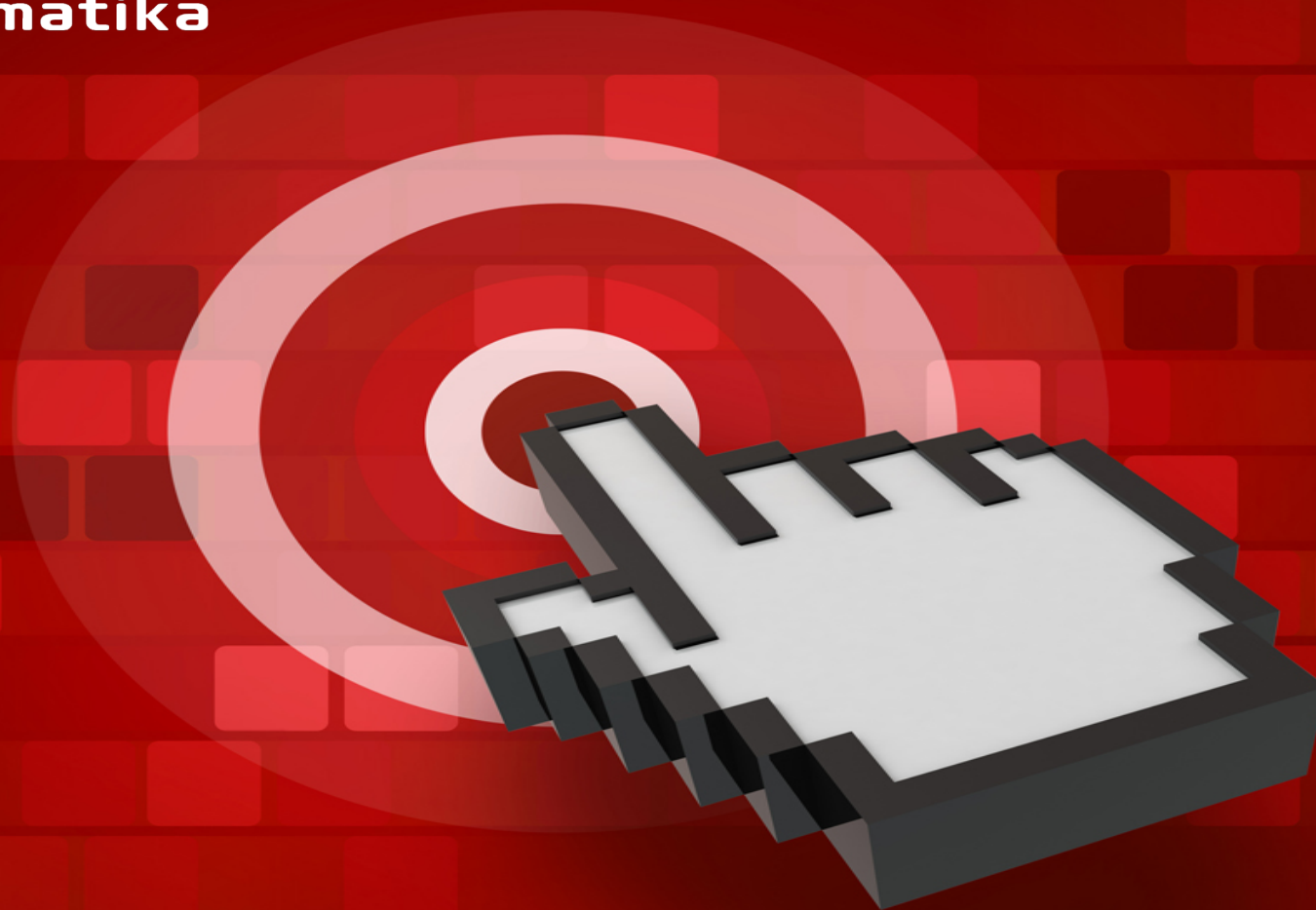


Analisa: Hasil Akurasi *Testing* kNN dan MLP



Kesimpulan

- Dengan dilakukan *oversampling* hasil yang didapatkan lebih kecil daripada tidak dilakukan *oversampling*.
- kNN lebih baik daripada MLP, karena metode MLP yang digunakan tidak dicoba beberapa arsitektur. Arsitektur yang dicoba hanya menggunakan 1 *hidden layer* dengan 100 *neuron*.
- Pada model kNN, data yang digunakan memiliki jumlah atribut hanya 30 (atribut awal 64) hasil reduksi menggunakan PCA. Atribut pada data berpengaruh jika metode klasifikasi yang digunakan kNN, karena kNN sangat rentan terhadap data dengan dimensi yang tinggi
- Selain itu, dengan PCA dapat dikurangi atribut yang tidak memberikan informasi perbedaan jarak setiap data sehingga penggunaan kNN tidak selalu lebih baik pada semua data.



THANK YOU