



DIGITAL TALENT SCHOLARSHIP 2019

Big Data Analytics



Pemodelan Non-Linier: ELM, SVM => SVR

Oleh: Imam Cholissodin | imamcs@ub.ac.id, Putra Pandu Adikara, Sufia Adha Putri

Asisten: Guedho, Sukma, Anshori, Aang dan Gusti

Fakultas Ilmu Komputer (Filkom) Universitas Brawijaya (UB)

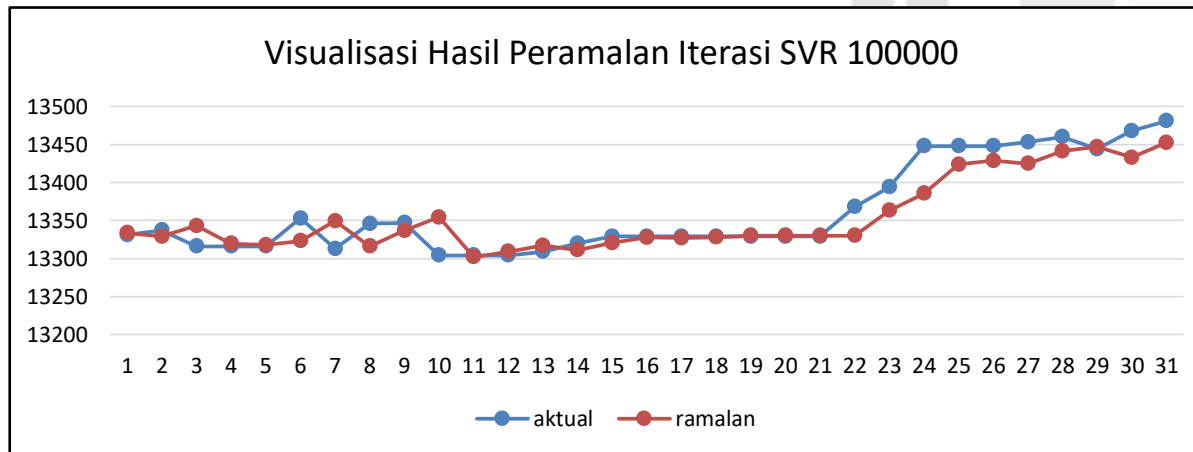
Pokok Pembahasan

- Konsep Regresi (Review)
- Analisis Teknikal dan Fundamental
- Regresi Non-Linear
 - Extreme Learning Machine (ELM) untuk Big Data
 - Support Vector Machine (SVM) untuk Regresi (Support Vector Regression (SVR)) untuk Big Data
- Tugas



Konsep Regresi (Just Review)

- **Regresi** adalah membangun model untuk memprediksi nilai (satu target atau multi-target) dari data masukan (dengan dimensi tertentu) yang telah ada.
- Goal : Membangun model regresi untuk mencari hubungan antara satu atau lebih variabel independen atau prediktor (X atau X_1, X_2, \dots, X_n) dengan variabel dependen atau respon single atau multi-target (Y atau Y_1, Y_2, \dots, Y_n).



Konsep Regresi (Just Review)

- **Prediksi** berbeda dengan **klasifikasi** (dalam *machine learning*, klasifikasi dianggap sebagai salah satu jenis dari prediksi).
- **Klasifikasi** digunakan untuk **memprediksi label kelas/kategori**.
- Klasifikasi dapat dibagi lagi menjadi dua macam, yaitu:
 - *supervised classification* (Klasifikasi) dan
 - *unsupervised classification* (**Clustering** → sudah dibahas pada pertemuan sebelumnya)
- Prediksi versus Peramalan (Jangka waktu merupakan kata kunci untuk membedakan antara prediksi dan peramalan)

Analisis Teknikal dan Fundamental

- Ada beberapa jenis pendekatan yang dilakukan untuk meramalkan, misal pada nilai tukar uang diantaranya (Madura, 2011):
 - Analisis Teknikal
 - ✓ **Melibatkan data historis** nilai tukar untuk meramalkan nilai di masa yang akan datang.
 - ✓ Prinsip yang biasanya dianut oleh para teknikalisis, bahwa **nilai kurs sudah menjadi nilai yang representatif dari seluruh informasi yang relevan** mempengaruhi kurs tersebut, kurs akan bertahan pada trend tertentu, dan nilai kurs merupakan nilai yang repetitif berulang dari pola sebelumnya (Neely et al., 2011).
 - ✓ Namun terkadang peramalan secara teknis **tidak terlalu membantu** untuk **jangka waktu** yang cukup **lama**. Banyak peneliti yang berselisih pendapat mengenai peramalan teknis, meskipun secara umum peramalan teknis memberikan konsistensi yang baik.
 - Analisis Fundamental

Analisis Teknikal dan Fundamental

- Ada beberapa jenis pendekatan yang dilakukan untuk meramalkan, misal pada nilai tukar uang diantaranya (Madura, 2011):
 - Analisis Fundamental
 - ✓ Didasari **hubungan fundamental antara variabel – variabel ekonomi terhadap nilai tukar** tersebut, seperti faktor – faktor yang mempengaruhi nilai tukar. Misalnya saja:
 - Tingkat inflasi
 - Suku bunga
 - Neraca Perdagangan (log pembayaran dari hasil jual beli barang dan jasa antara negara)
 - Hutang Publik (Public Debt),
 - Rasio Harga Ekspor dan Harga Impor, dan
 - Kestabilan Politik dan Ekonomi.
- Dari uraian sebelumnya, apakah bisa dilakukan penggabungan Analisis Teknikal dan Fundamental?

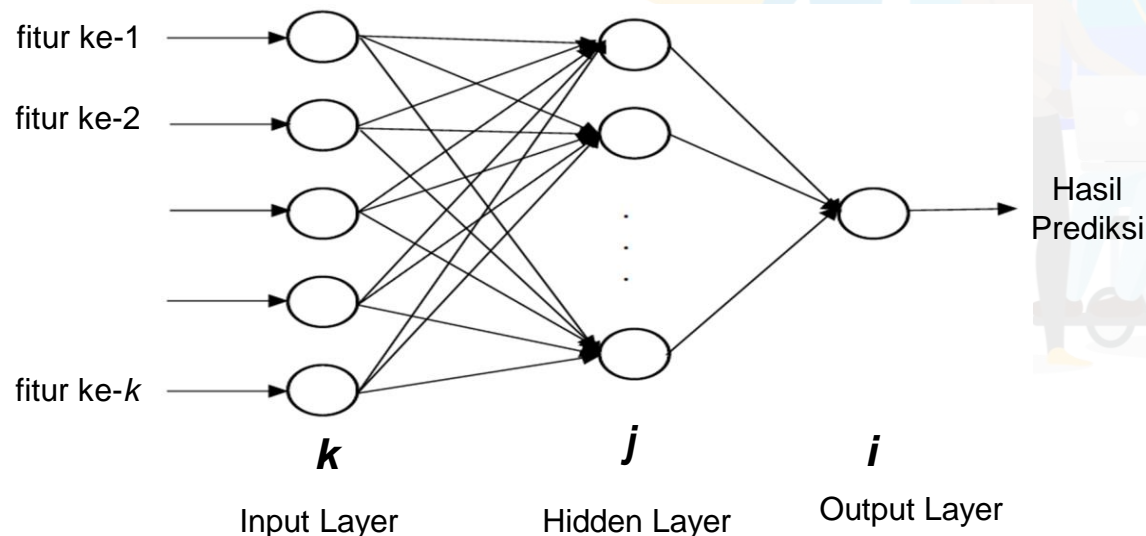
TERBUKA
UNTUK
DISABILITAS

Extreme Learning Machine (ELM) untuk Big Data

Ref: <http://bit.ly/2x8ta9S> / <http://bit.ly/2FMLnfw> &
<http://bit.ly/2Ww0wYa>

Extreme Learning Machine (ELM) untuk Big Data

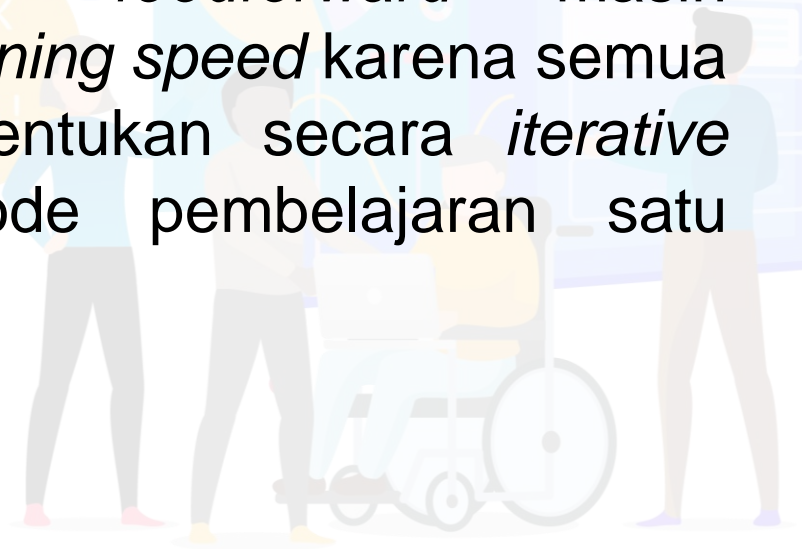
- Metode ELM ini pertama kali diperkenalkan oleh Huang (2004) (Huang, Zhu, & Siew, 2004).
- ELM merupakan jaringan syaraf tiruan (JST) *feedforward* dengan *single hidden layer* atau disebut dengan Single Hidden Layer Feedforward Neural Networks (SLFNs) (Sun, Choi, Au, & Yu, 2008).



Arsitektur ELM

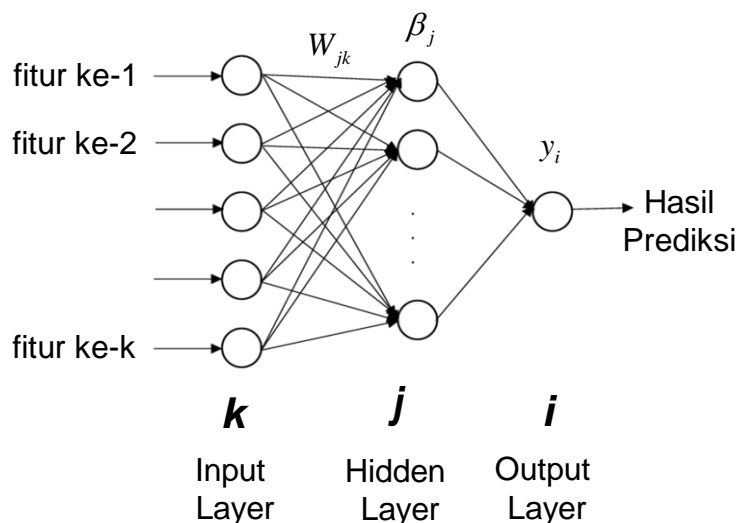
Extreme Learning Machine (ELM) untuk Big Data

- Metode pembelajaran ELM dibuat untuk mengatasi beberapa kelemahan dari jaringan syaraf tiruan feedforward, terutama dalam hal learning speed.
- Menurut Huang (2004), JST *feedforward* masih memiliki kelemahan dalam *learning speed* karena semua parameter pada jaringan ditentukan secara *iterative* dengan menggunakan metode pembelajaran satu persatu untuk setiap data.



Extreme Learning Machine (ELM) untuk Big Data

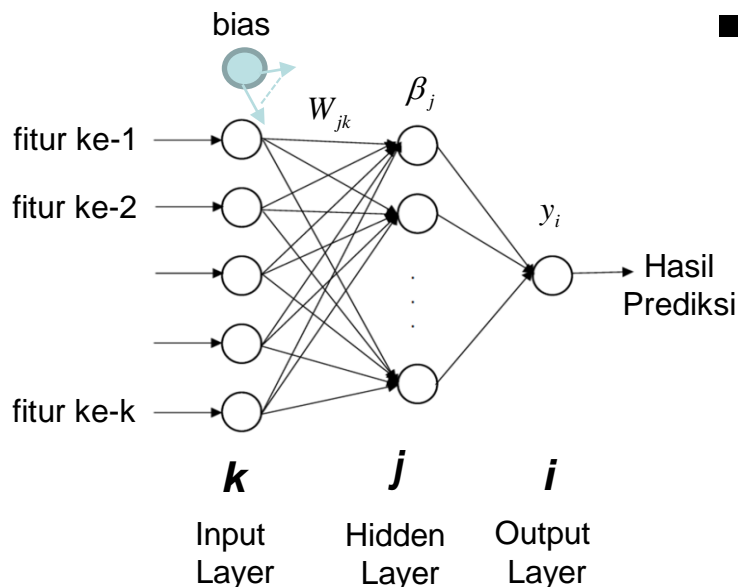
- Langkah-langkah training algoritma Extreme Learning Machine (ELM) tanpa bias:
 1. Buat random W_{jk} sebagai bobot masukan.
 2. Hitung matrik keluaran hidden layer $H = 1/(1+\exp(-X.W^T))$
 3. Hitung output weight $\hat{\beta} = H^+ . Y$ dimana $H^+ = (H^T . H)^{-1} . H^T$
 4. Hitung $\hat{Y} = H . \hat{\beta}$



- Langkah testing ELM:
 1. Diketahui W_{jk} dan $\hat{\beta}$.
 2. Hitung $H = 1/(1+\exp(-X_{test}.W^T))$
 3. Hitung $\hat{Y} = H . \hat{\beta}$
 4. Hitung nilai evaluasi, misal dengan MAPE.

Extreme Learning Machine (ELM) untuk Big Data

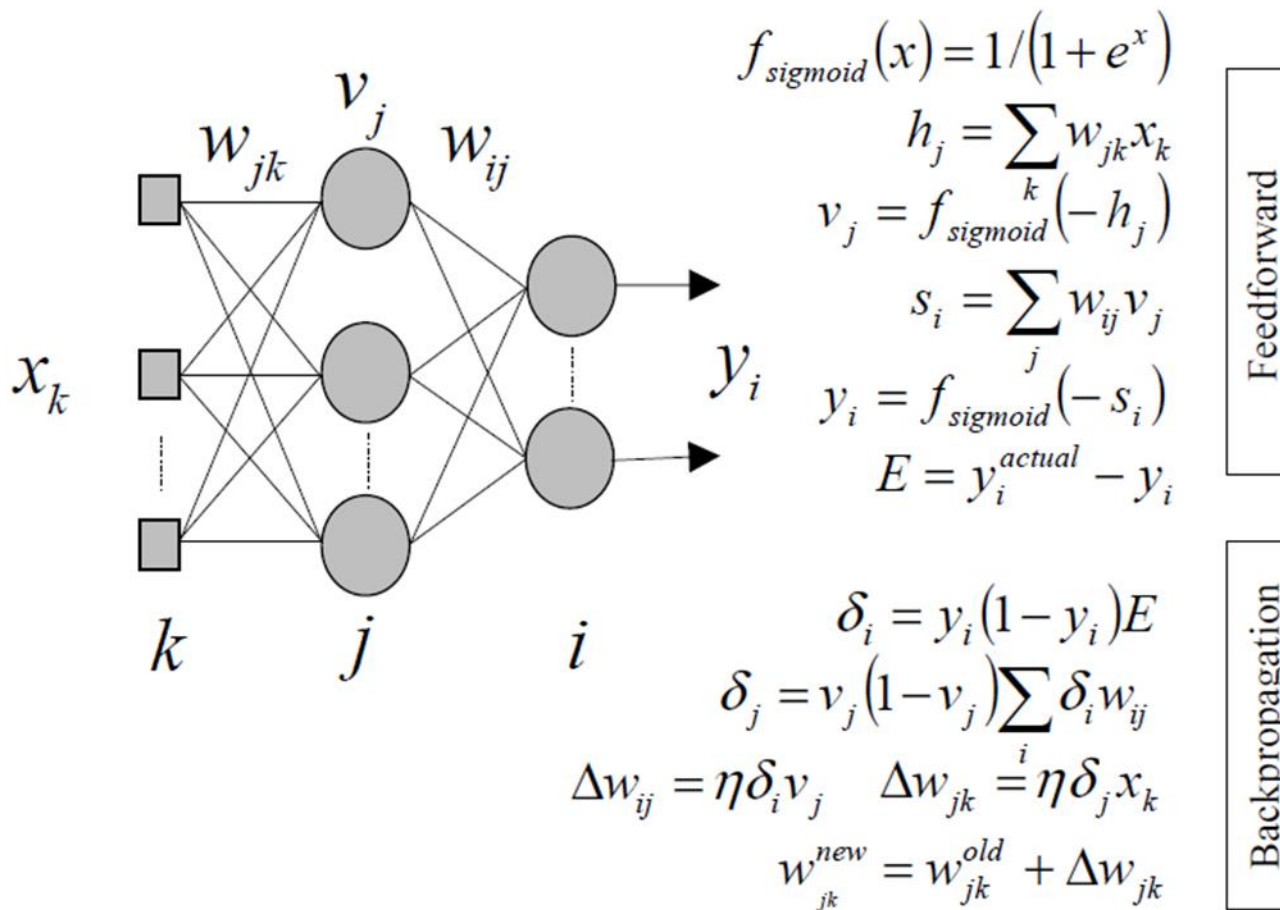
- Langkah-langkah training ELM dengan bias:
 1. Buat random W_{jk} sebagai bobot masukan dan nilai bias b jika ada. Ukuran matrik bias b adalah $[1 \times j]$
 2. Hitung matrik keluaran hidden layer $H = 1/(1+\exp(-(X.W^T + \text{ones}(N_{train}, 1)*b)))$. Dan N_{train} adalah banyaknya data training.
 3. Hitung output weight $\hat{\beta} = H^+ . Y$ dimana $H^+ = (H^T . H)^{-1} . H^T$
 4. Hitung $\hat{Y} = H . \hat{\beta}$



- Langkah testing ELM:
 1. Diketahui W_{jk} , nilai bias b jika ada, dan $\hat{\beta}$.
 2. Hitung $H = 1/(1+\exp(-(X_{test} . W^T + \text{ones}(N_{test}, 1)*b)))$
 3. Hitung $\hat{Y} = H . \hat{\beta}$
 4. Hitung nilai evaluasi, misal dengan MAPE.

ELM vs ANN

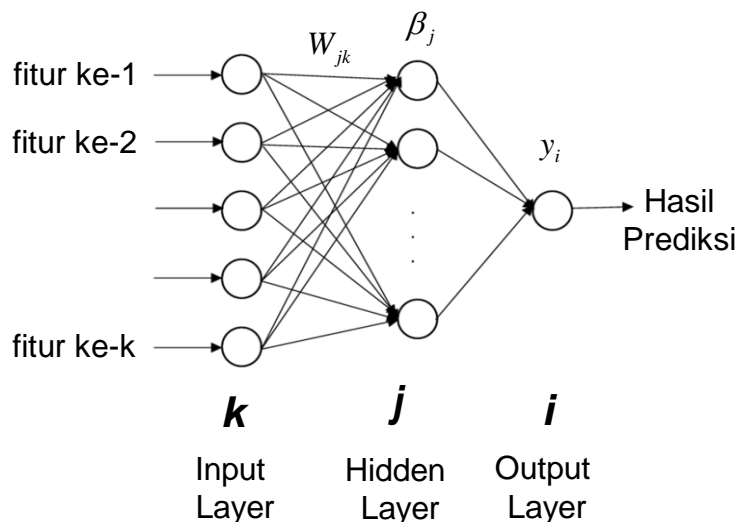
- Arsitektur Artificial Neural Network Backpropagation (ANN):



Studi Kasus Sederhana dengan ELM

- Misalkan diketahui, data *training* (terdapat 3 fitur dan *single target*) sebagai berikut:

Data	Fitur			Target
	1	2	3	
1	1	1	1	1
2	1	0	1	1
3	1	1	0	1
4	1	1	0	2
5	0	1	0	2
6	0	0	0	2
7	0	1	0	3
8	1	1	0	3
9	0	0	0	3



Diketahui,

N : Banyaknya data training = 9

k : Banyaknya dimensi input layer = 3

i : Banyaknya dimensi output layer = 1

Misal diset,

j : Banyaknya dimensi hidden layer = 3

Penyelesaian:

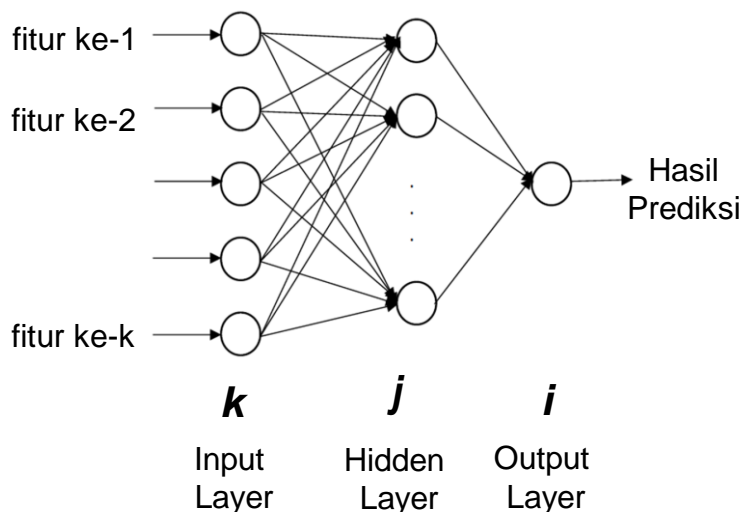
- Men-*generate* bobot input ($W_{j \times k}$) antara input layer dan hidden layer. Ukuran W adalah $[j \times k]$, yang nilainya *di-generate* antara $[-0.5, 0.5]$.

$$W = \begin{bmatrix} -0.4 & 0.2 & 0.1 \\ -0.2 & 0 & 0.4 \\ -0.3 & 0.3 & -0.1 \end{bmatrix}$$

Studi Kasus Sederhana dengan ELM

- Misalkan diketahui, data *training* (terdapat 3 fitur dan *single target*) sebagai berikut:

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$



Penyelesaian:

- Men-*generate* bobot input ($W_{j \times k}$) antara input layer dan hidden layer. Ukuran W adalah $[j \times k]$, yang nilainya *di-generate* antara $[-0.5, 0.5]$.

$$W = \begin{bmatrix} -0.4 & 0.2 & 0.1 \\ -0.2 & 0 & 0.4 \\ -0.3 & 0.3 & -0.1 \end{bmatrix}$$

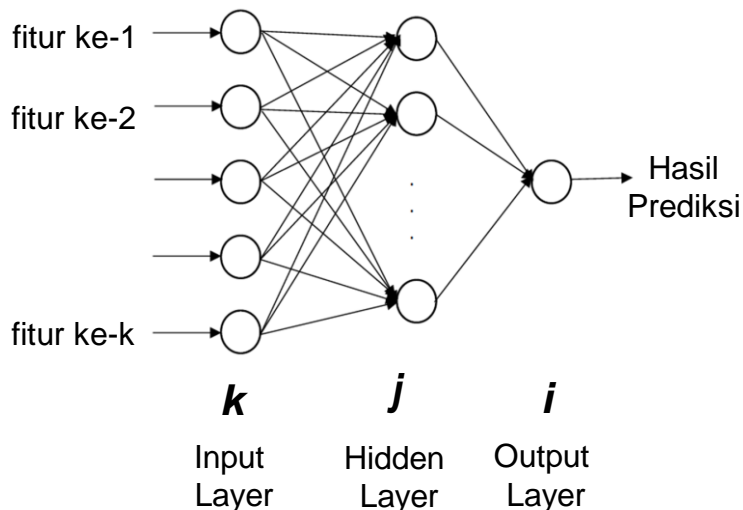
- Hitung matrik inisialisasi output hidden layer ($H_{init} = X \cdot W^T$)

$$H_{init} = \begin{pmatrix} -0.1 & 0.2 & -0.1 \\ -0.3 & 0.2 & -0.4 \\ -0.2 & -0.2 & 0 \\ -0.2 & -0.2 & 0 \\ 0.2 & 0 & 0.3 \\ 0 & 0 & 0 \\ 0.2 & 0 & 0.3 \\ -0.2 & -0.2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Studi Kasus Sederhana dengan ELM

- Misalkan diketahui, data *training* (terdapat 3 fitur dan *single target*) sebagai berikut:

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$



Penyelesaian:

- Hitung matrik inisialisasi output hidden layer ($H_{init} = X.W^T$)

$$H_{init} =$$

$$\begin{pmatrix} -0.1 & 0.2 & -0.1 \\ -0.3 & 0.2 & -0.4 \\ -0.2 & -0.2 & 0 \\ -0.2 & -0.2 & 0 \\ 0.2 & 0 & 0.3 \\ 0 & 0 & 0 \\ 0.2 & 0 & 0.3 \\ -0.2 & -0.2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- Hitung matrik $H = 1/(1+\exp(-H_{init}))$

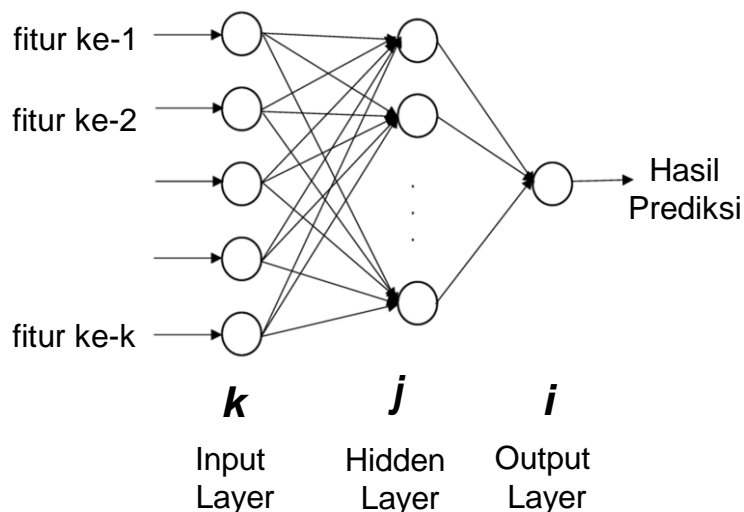
$$H =$$

$$\begin{pmatrix} 0.48 & 0.55 & 0.48 \\ 0.43 & 0.55 & 0.40 \\ 0.45 & 0.45 & 0.50 \\ 0.45 & 0.45 & 0.50 \\ 0.55 & 0.50 & 0.57 \\ 0.50 & 0.50 & 0.50 \\ 0.55 & 0.50 & 0.57 \\ 0.45 & 0.45 & 0.50 \\ 0.50 & 0.50 & 0.50 \end{pmatrix}$$

Studi Kasus Sederhana dengan ELM

- Misalkan diketahui, data *training* (terdapat 3 fitur dan *single target*) sebagai berikut:

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \end{pmatrix}$$



Penyelesaian:

3. Hitung matrik $H = 1/(1+\exp(-H_{init}))$

$H =$

$$H = \begin{pmatrix} 0.48 & 0.55 & 0.48 \\ 0.43 & 0.55 & 0.40 \\ 0.45 & 0.45 & 0.50 \\ 0.45 & 0.45 & 0.50 \\ 0.55 & 0.50 & 0.57 \\ 0.50 & 0.50 & 0.50 \\ 0.55 & 0.50 & 0.57 \\ 0.45 & 0.45 & 0.50 \\ 0.50 & 0.50 & 0.50 \end{pmatrix}$$

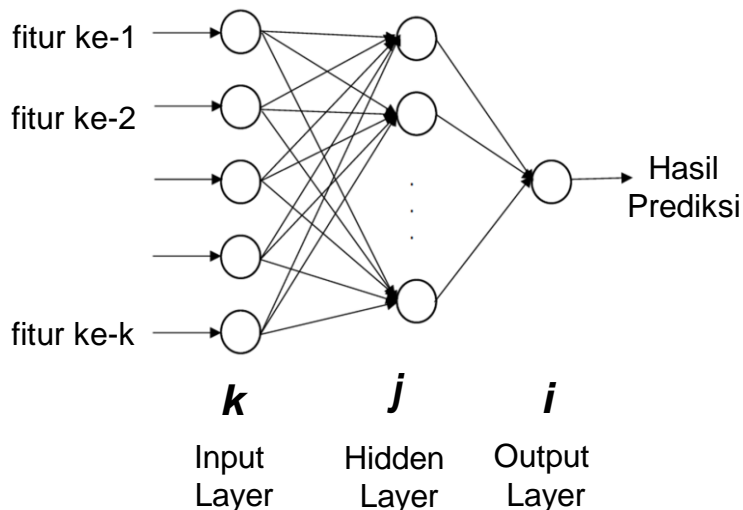
4. Hitung matrik $H^+ = (H^T \cdot H)^{-1} \cdot H^T$

$$H^+ = \begin{pmatrix} -2.18 & -1.37 & -9.11 & -9.11 & 6.69 & 7.57 & 6.69 & -9.11 & 7.57 \\ 2.58 & 3.98 & 1.68 & 1.68 & -3.13 & -1.74 & -3.13 & 1.68 & -1.74 \\ -0.21 & -2.38 & 7.33 & 7.33 & -3.13 & -5.35 & -3.13 & 7.33 & -5.35 \end{pmatrix}$$

Studi Kasus Sederhana dengan ELM

- Misalkan diketahui, data *training* (terdapat 3 fitur dan *single target*) sebagai berikut:

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \end{pmatrix}$$



Penyelesaian:

4. Hitung matrik $H^+ = (H^T \cdot H)^{-1} \cdot H^T$

$$H^+ = \begin{pmatrix} -2.18 & -1.37 & -9.11 & -9.11 & 6.69 & 7.57 & 6.69 & -9.11 & 7.57 \\ 2.58 & 3.98 & 1.68 & 1.68 & -3.13 & -1.74 & -3.13 & 1.68 & -1.74 \\ -0.21 & -2.38 & 7.33 & 7.33 & -3.13 & -5.35 & -3.13 & 7.33 & -5.35 \end{pmatrix}$$

5. Hitung $\hat{\beta}$ beta topi atau hasil nilai beta yang diestimasi

$$\hat{\beta} = H^+ \cdot Y = \begin{pmatrix} 13.06 \\ -7.74 \\ -0.98 \end{pmatrix}$$

6. Menghitung Y prediksi

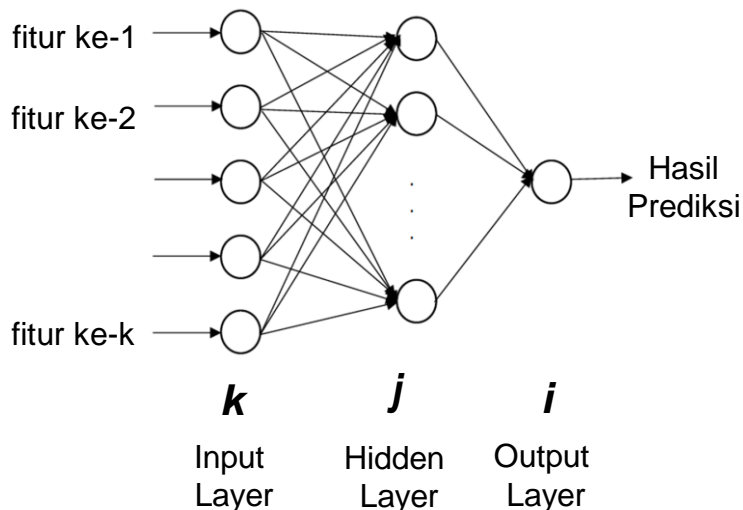
$$\hat{Y} = H \cdot \hat{\beta}$$

$$\hat{Y} = \begin{pmatrix} 1.48 \\ 0.91 \\ 1.90 \\ 1.90 \\ 2.75 \\ 2.17 \\ 2.75 \\ 1.90 \\ 2.17 \end{pmatrix}$$

Studi Kasus Sederhana dengan ELM

- Misalkan diketahui, data *training* (terdapat 3 fitur dan *single target*) sebagai berikut:

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \end{pmatrix}$$



Penyelesaian:

6. Menghitung Y prediksi

$$\hat{Y} = H \cdot \hat{\beta}$$

$$\hat{Y} =$$

1.48
0.91
1.90
1.90
2.75
2.17
2.75
1.90
2.17

7. Hitung MAPE

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{\hat{y}_i - y_i}{y_i} \times 100 \right|$$

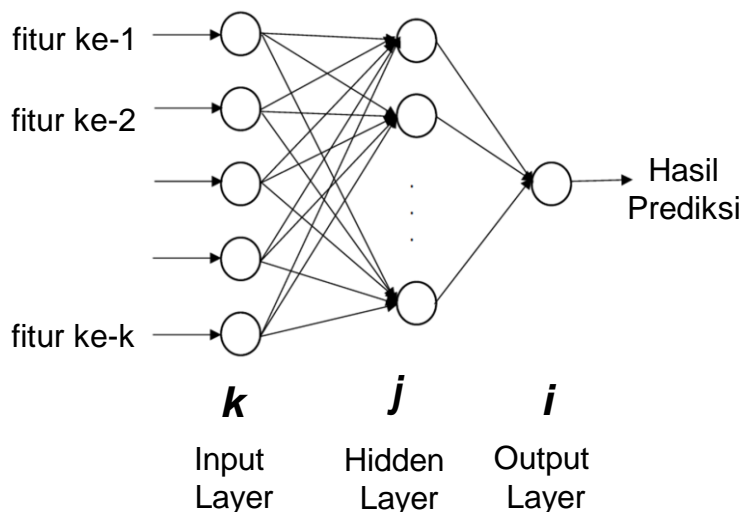
$$= 30.11$$

Pada proses training, didapatkan hasil testing menggunakan data training sendiri masih memiliki error MAPE sebesar 30.11

Studi Kasus Sederhana dengan ELM

- Misalkan diketahui, data *training* (terdapat 3 fitur dan *single target*) sebagai berikut:

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \end{pmatrix}$$



Penyelesaian:

6. Menghitung Y prediksi

$$\hat{Y} = H \cdot \hat{\beta}$$

$$\hat{Y} =$$

$$\begin{pmatrix} 1.48 \\ 0.91 \\ 1.90 \\ 1.90 \\ 2.75 \\ 2.17 \\ 2.75 \\ 1.90 \\ 2.17 \end{pmatrix}$$

7. Hitung MAPE

$$MAPE = \frac{1}{N=9} \sum_{i=1}^{N=9} \left| \frac{\hat{y}_i - y_i}{y_i} \times 100 \right|$$

$$= 30.11$$

Pada proses training, didapatkan hasil testing menggunakan data training sendiri masih memiliki error MAPE sebesar 30.11

Studi Kasus Sederhana dengan ELM

- Misalkan diketahui, data *testing* sebagai berikut:

Data *training*:

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \end{pmatrix}$$

Data *testing*:

$$X_{\text{test}} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$Y_{\text{test}} = \begin{pmatrix} 1 \\ 1 \\ 3 \end{pmatrix}$$

Diketahui,

N : Banyaknya data training = 9

k : Banyaknya input layer = 3

i : Banyaknya output layer = 1

Misal diset,

j : Banyaknya hidden layer = 3

$$W = \begin{pmatrix} -0.4 & 0.2 & 0.1 \\ -0.2 & 0 & 0.4 \\ -0.3 & 0.3 & -0.1 \end{pmatrix} \quad \hat{\beta} = \begin{pmatrix} 13.06 \\ -7.74 \\ -0.98 \end{pmatrix}$$

Penyelesaian:

- Hitung matrik inialisasi output hidden layer ($H_{\text{init}} = X_{\text{test}} \cdot W^T$)

$$H_{\text{init}} = \begin{pmatrix} -0.3 & 0.2 & -0.4 \\ -0.2 & -0.2 & 0 \\ 0.2 & 0 & 0.3 \end{pmatrix}$$

Studi Kasus Sederhana dengan ELM

- Misalkan diketahui, data *testing* sebagai berikut:

Data *training*:

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \end{pmatrix}$$

Data *testing*:

$$X_{\text{test}} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$Y_{\text{test}} = \begin{pmatrix} 1 \\ 1 \\ 3 \end{pmatrix}$$

Penyelesaian:

- Hitung matrik output hidden $H = 1/(1+\exp(-H_{\text{init}}))$

$$H = \begin{pmatrix} 0.43 & 0.55 & 0.40 \\ 0.45 & 0.45 & 0.50 \\ 0.55 & 0.50 & 0.57 \end{pmatrix}$$

- Menghitung Y prediksi

$$\hat{Y} = H \cdot \hat{\beta}, \text{ diketahui } \hat{\beta} = \begin{pmatrix} 13.06 \\ -7.74 \\ -0.98 \end{pmatrix}, \text{ maka}$$

$$\hat{Y} = \begin{pmatrix} 0.91 \\ 1.90 \\ 2.75 \end{pmatrix}$$

- Hitung MAPE

$$MAPE = \frac{1}{N=3} \sum_{i=1}^3 \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100 = 36.0048$$

Pada proses testing, didapatkan hasil prediksi dengan error MAPE sebesar 36.0048

TERBUKA
UNTUK
DISABILITAS

Support Vector Machine (SVM) untuk Regresi atau disebut sebagai **Support Vector Regression (SVR)**

Ref: <http://bit.ly/2Ww0wYa> &
<http://bit.ly/2x8ta9S> / <http://bit.ly/2FMLnfw>

Support Vector Regression (SVR)

- Konsep SVR didasarkan pada *risk minimization*, yaitu untuk mengestimasi suatu fungsi dengan cara meminimalkan batas dari *generalization error*, sehingga SVR mampu mengatasi *overfitting*.
- Fungsi regresi dari metode SVR adalah sebagai berikut (Sethu Vijayakumar & Si Wu, 1999)

ALGORITMA SEKUENSIAL TRAINING SVR :

1. Inisialisasi parameter SVR yang digunakan
2. Rumus memperoleh Matrik Hessian

$$R_{ij} = (K(x_i, x_j) + \lambda^2)$$

3. Untuk tiap training point lakukan:

$$E_i = y_i - \sum_{j=1}^l (\alpha_j^* - \alpha_j) R_{ij}$$

$$\delta\alpha_i^* = \min\{\max\{\gamma(E_i - \varepsilon), -\alpha_i^*\}, C - \alpha_i^*\}$$

$$\delta\alpha_i = \min\{\max\{\gamma(-E_i - \varepsilon), -\alpha_i\}, C - \alpha_i\}$$

$$\alpha_i^* = \alpha_i^* + \delta\alpha_i^*$$

$$\alpha_i = \alpha_i + \delta\alpha_i$$

4. Kembali ke langkah ketiga, sampai pada kondisi iterasi maksimum atau

$$\max(|\delta\alpha_i|) < \varepsilon \quad \text{dan} \quad \max(|\delta\alpha_i^*|) < \varepsilon$$

5. Dengan menggunakan fungsi peramalan berikut:

$$f(x) = \sum_{i=1}^l (\alpha_i^* - \alpha_i) (K(x_i, x) + \lambda^2)$$

Support Vector Regression (SVR)

- Contoh penyelesaian masalah *forecasting* dengan SVR:

Data Nilai Tukar
IDR-USD Juli 2015

Tanggal	Nilai Tukar
5-Jul-15	13338
6-Jul-15	13356
7-Jul-15	13332
8-Jul-15	13331
9-Jul-15	13337
10-Jul-15	13316
11-Jul-15	13316
12-Jul-15	13316
13-Jul-15	13353
14-Jul-15	13304
15-Jul-15	13304
16-Jul-15	13309



Cara membuat Dataset dengan 4 fitur:

No	Tgl/Bln/Thn	X1	X2	X3	X4	Y
1	9 Juli 2015	13338	13356	13332	13331	13337
2	10 Juli 2015	13356	13332	13331	13337	13316
3	11 Juli 2015	13332	13331	13337	13316	13316
4	12 Juli 2015	13331	13337	13316	13316	13316
5	13 Juli 2015	13337	13316	13316	13316	13353
6	14 Juli 2015	13313	13346	13347	13304	13304
7	15 Juli 2015	13346	13347	13304	13304	13304
8	16 Juli 2015	13347	13304	13304	13304	13309

Data Latih

Data Uji

Support Vector Regression (SVR)

- Dataset dengan 4 fitur yang sudah dinormalisasi:

x' = Hasil normalisasi data

x = Nilai data yang akan dinormalisasi

x_{max} = Nilai maksimum dari keseluruhan data, misal mulai dari tahun 2006 – 2015

x_{min} = Nilai minimum dari keseluruhan data, misal mulai dari tahun 2006 – 2015

Misal didapatkan, $x_{min} = 9634$ dan $x_{max} = 14728$

$$x' = \frac{(x - x_{min})}{x_{max} - x_{min}}$$

$$x' = \frac{(13338 - 9634)}{14728 - 9634} = 0.727129$$

No	Tgl/Bln/Thn	X1	X2	X3	X4	Y
1	9 Juli 2015	0.727129957	0.730663526	0.725952101	0.725755791	0.726933647
2	10 Juli 2015	0.730663526	0.725952101	0.725755791	0.726933647	0.72281115
3	11 Juli 2015	0.725952101	0.725755791	0.726933647	0.72281115	0.72281115
4	12 Juli 2015	0.725755791	0.726933647	0.72281115	0.72281115	0.72281115
5	13 Juli 2015	0.726933647	0.72281115	0.72281115	0.72281115	0.730074598
6	14 Juli 2015	0.72281115	0.72281115	0.72281115	0.730074598	0.722222222
7	15 Juli 2015	0.72281115	0.72281115	0.730074598	0.722222222	0.728700432
8	16 Juli 2015	0.72281115	0.730074598	0.722222222	0.728700432	0.728896741

Data Latih

Data Uji

Support Vector Regression (SVR)

■ Proses Perhitungan Jarak Data:

No	Tgl/Bln/Thn	X1	X2	X3	X4
1	9 Juli 2015	0.727129957	0.730663526	0.725952101	0.725755791
2	10 Juli 2015	0.730663526	0.725952101	0.725755791	0.726933647

Keterangan:
X1, X2, X3, X4 → menyatakan fitur data atau pola-pola data, sedangkan x_1 dan x_2 menyatakan data ke-1 dan data ke-2.

$$\begin{aligned} \|x_1 - x_2\|^2 &= (0.7271299 - 0.7306635)^2 + (0.7306635 - 0.7259521)^2 \\ &+ (0.7259521 - 0.7257557)^2 + (0.7257557 - 0.7269336)^2 \\ &= 3.61095 \times 10^{-5} \end{aligned}$$

Data ke-i	1	2	3	4	5
1	0	3.61095×10^{-5}	3.51075×10^{-5}	3.43368×10^{-5}	8.02348×10^{-5}
2	3.61095×10^{-5}	0	4.06184×10^{-5}	5.07152×10^{-5}	4.94435×10^{-5}
3	3.51075×10^{-5}	4.06184×10^{-5}	0	1.84209×10^{-5}	2.66293×10^{-5}
4	3.43368×10^{-5}	5.07152×10^{-5}	1.84209×10^{-5}	0	1.83823×10^{-5}
5	8.02348×10^{-5}	4.94435×10^{-5}	2.66293×10^{-5}	1.83823×10^{-5}	0

Support Vector Regression (SVR)

- Perhitungan Matriks Hessian:

$$[R]_{1,2} = K(x_1, x_2) + \lambda^2 = \exp\left(-\frac{3.61095 \times 10^{-5}}{2 \times (0.7)^2}\right) + 4.32^2 = 19.6623631542$$

R _{ij}	1	2	3	4	5
1	19.6624000000	19.6623631542	9.6623641766	19.6623649631	19.6623181311
2	19.6623631542	19.6624000000	19.6623585535	19.6623482512	19.6623495488
3	19.6623641766	19.6623585535	19.6624000000	19.6623812034	19.6623728276
4	19.6623649631	19.6623482512	19.6623812034	19.6624000000	19.6623812427
5	19.6623181311	19.6623495488	19.6623728276	19.6623812427	19.6624000000

Support Vector Regression (SVR)

- Hitung Nilai Error, $\delta\alpha_i^*$ dan $\delta\alpha_i$, serta α_i dan α_i^* :

$$E_i = y_i - \sum_{j=1}^l (\alpha_j^* - \alpha_j) R_{ij}$$

$$\begin{aligned} E_1 &= 0.72693 - ((0.0073646 - 0) * 19.6624 - (0.0072136 - 0) * 19.6623 \\ &\quad - (0.0072136 - 0) * 19.6623 - (0.0072136 - 0) * 19.6623 \\ &\quad - (0.0075296 - 0) * 19.6623 \\ &= 0.009548583120820 \end{aligned}$$

$$\delta\alpha_i^* = \min\{\max\{\gamma(E_i - \varepsilon), -\alpha_i^*\}, C - \alpha_i^*\}$$

$$\begin{aligned} \delta\alpha_1^* &= (\min(\max(0.00406 * (0.0095485 - 0.0004) \\ &\quad , 0.0073646), 100 - 0.0073646)) \\ &= 0.000037222650829 \end{aligned}$$

$$\delta\alpha_i = \min\{\max\{\gamma(-E_i - \varepsilon), -\alpha_i\}, C - \alpha_i\}$$

$$\begin{aligned} \delta\alpha_1 &= (\min(\max(0.00406 * (0.0095485 - 0.0004) \\ &\quad , -0), 100 - 0)) = 0 \end{aligned}$$

$$\alpha_i^* = \alpha_i^* + \delta\alpha_i^*$$

$$\alpha_1^* = 0.0073646102 + 0.0000372226 = 0.0074018329$$

$$\alpha_i = \alpha_i + \delta\alpha_i \quad \alpha_1 = 0 + 0 = 0$$

E_i
0.009548583120820
0.005426086065460
0.005426086065460
0.005426086065460
0.012689533258236

$\delta\alpha_i^*$	$\delta\alpha_i$
0.000037222650829	0
0.000020449532368	0
0.000020449532368	0
0.000020449532368	0
0.000050002169657	0

α_i^*	α_i
0.007401832903405	0
0.007234101718795	0
0.007234101718795	0
0.007234101718795	0
0.007529628091680	0

Support Vector Regression (SVR)

- Testing Data Latih Dan Data Uji:

$$f(x) = \sum_{i=1}^l (\alpha_i^* - \alpha_i)(K(x_i, x) + \lambda^2)$$

$$f(x) = ((0.00740183290 - (0)) \times 19.6624) + ((0.00723410171 - (0)) \times 19.662363) + ((0.00723410171 - (0)) \times 19.662364) + ((0.00723410171 - (0)) \times 19.662364) + ((0.00752962809 - (0)) \times 19.662318) = 0.720306367977400$$

Data Latih ke-i	Nilai Aktual Kurs	F(x)
1	0.726933647428347	0.720306367977400
2	0.722811150372988	0.720306436786951
3	0.722811150372988	0.720306858015035
4	0.722811150372988	0.720306852670471
5	0.730074597565764	0.720306460367159

Data Uji ke-i	Nilai Aktual Kurs	F(x)
1	0.722222222	0.720304504002685
2	0.728700432	0.720304342412368
3	0.728896741	0.720304580743711

$$R_{ij} = (K(x_i, x_j) + \lambda^2)$$

Lihat tabel Perhitungan Matriks Hessian, baris ke-1 kolom ke-1

Support Vector Regression (SVR)

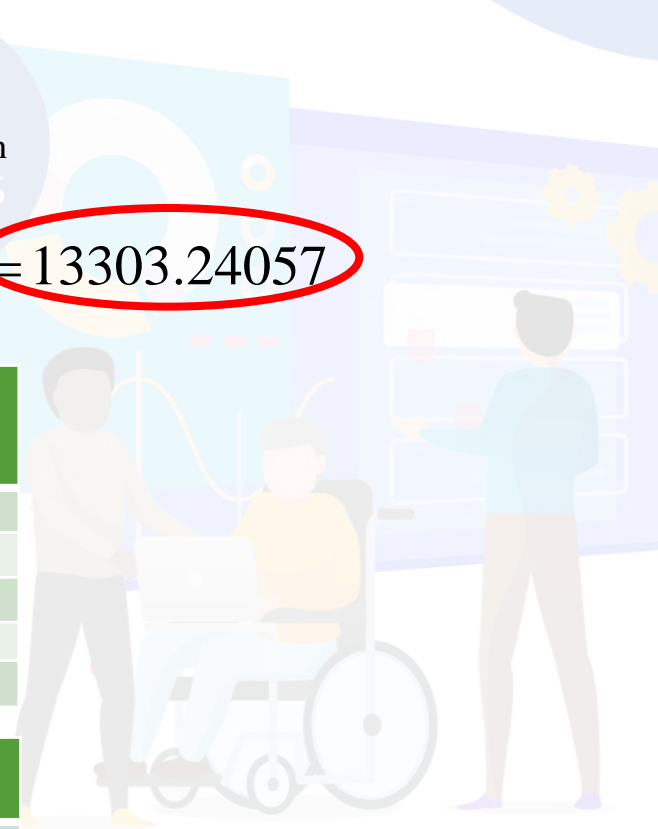
■ Denormalisasi Peramalan:

$$x' = \frac{(x - x_{\min})}{x_{\max} - x_{\min}} \Rightarrow x = x'(x_{\max} - x_{\min}) + x_{\min}$$

$$x = (0.720306367977 \times (14728 - 9634)) + 9634 = 13303.24057$$

Data Latih ke-i	F(x)	F(x) Denormalisasi	Nilai Aktual
1	0.720306367977400	13303.24057	13337
2	0.720306436786951	13303.24309	13316
3	0.720306858015035	13303.24396	13316
4	0.720306852670471	13303.24388	13316
5	0.720306460367159	13303.2422	13353

Data Uji ke-i	F(x)	F(x) Denormalisasi	Nilai Aktual
1	0.720304504002685	13303.23114	13304
2	0.720304342412368	13303.23032	13304
3	0.720304580743711	13303.23153	13309



Support Vector Regression (SVR)

- MAPE:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{\hat{y}_i - y_i}{y_i} \times 100 \right|$$

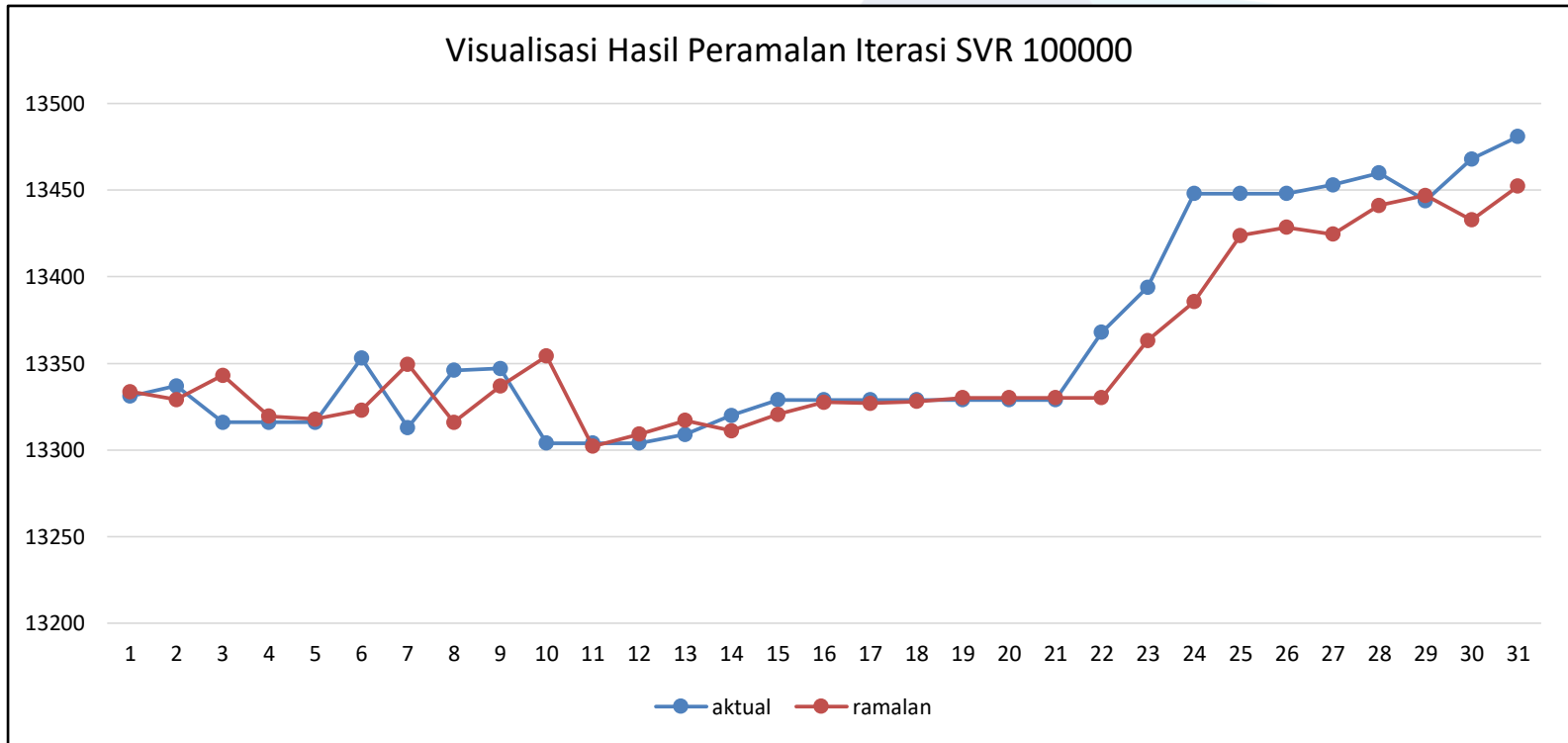
$$MAPE_{data\ latihan} = \frac{1}{5} (abs(\frac{13303.24057 - 13337}{13337} \times 100) + abs(\frac{13303.24309 - 13316}{13316} \times 100) + abs(\frac{13303.24396 - 13316}{13316} \times 100) + abs(\frac{1303.24388 - 13316}{13316} \times 100) + abs(\frac{1333.2422 - 13353}{13353} \times 100))$$

$$= 0.182630337$$

	Data Latih	Data Uji
NILAI	0.253126116	0.073378327
	0.095801356	0.320468153
	0.095794862	0.327927367
	0.095795468	
	0.372633884	
MAPE	0.182630337	0.240591282

Support Vector Regression (SVR)

- Contoh Grafik Hasil Peramalan Nilai Kurs:



MAPE = 0.127551



DIGITAL
TALENT
SCHOLARSHIP

Latihan langsung di Kelas Ke-1 & Pembahasan Link kode “<http://bit.ly/2KmzL5g>”

Silahkan dicoba dijalankan dengan Jupyter notebook yang Anda buat sebelumnya di Ubuntu 16.04 atau dengan SageMaker notebook (JupyterLab) yang baru Anda buat hari ini.

Lab-Sesi27-1

```
# Init Python alg. Extreme Learning Machine (ELM)
#by:
#1. Imam Cholissodin (imamcs@ub.ac.id)
#2. Galih Ariwanda (galihariwanda01@gmail.com)
#3. Django Developer
#4. Apache Hadoop, Spark, etc
#5. All of My Teams

# a. Perbaiki koding berikut sehingga jalan di jupyter notebook anda .:D

import pandas as pd
import numpy as np
dataset = request.FILES["inputDataset"]#dataset_dump.csv
persentase_data_training = 90
banyak_fitur = int(request.POST["banyakFitur"])
banyak_hidden_neuron = int(request.POST["banyakHiddenNeuron"])

dataset = pd.read_csv(dataset, delimiter=',', names = ['Tanggal', 'Harga'], usecols=['Harga'])
minimum = int(dataset.min()-10000)
maksimum = int(dataset.max()+10000)
new_banyak_fitur = banyak_fitur + 1
hasil_fitur = []
for i in range((len(dataset)-new_banyak_fitur)+1):
    kolom = []
    j = i
    while j < (i+new_banyak_fitur):
        kolom.append(dataset.values[j][0])
        j += 1
    hasil_fitur.append(kolom)
hasil_fitur = np.array(hasil_fitur)
data_normalisasi = (hasil_fitur - minimum)/(maksimum - minimum)

data_training = data_normalisasi[int(persentase_data_training*len(data_normalisasi)/100)]
data_testing = data_normalisasi[int(persentase_data_training*len(data_normalisasi)/100):]

#Training ELM
bobot = np.random.rand(banyak_hidden_neuron, banyak_fitur)
bias = np.random.rand(banyak_hidden_neuron)
h = 1/(1 + np.exp(-(np.dot(data_training[:, :banyak_fitur], np.transpose(bobot)) + bias)))
h_plus = np.dot(np.linalg.inv(np.dot(np.transpose(h), h)), np.transpose(h))
output_weight = np.dot(h_plus, data_training[:, banyak_fitur])

#Testing ELM
h = 1/(1 + np.exp(-(np.dot(data_testing[:, :banyak_fitur], np.transpose(bobot)) + bias)))
predict = np.dot(h, output_weight)
predict = predict * (maksimum - minimum) + minimum

#MAPE
aktual = np.array(hasil_fitur[int(persentase_data_training*len(data_normalisasi)/100):, banyak_fitur])
mape = np.sum(np.abs((aktual - predict)/aktual)*100)/len(predict)

#return render(request, 'week1.html', {
#    'y_aktual': list(aktual),
#    'y_prediksi': list(predict),
#    'mape': mape

# b. Lalu plotting grafik y_aktual dan y_prediksi dalam satu grafik
```

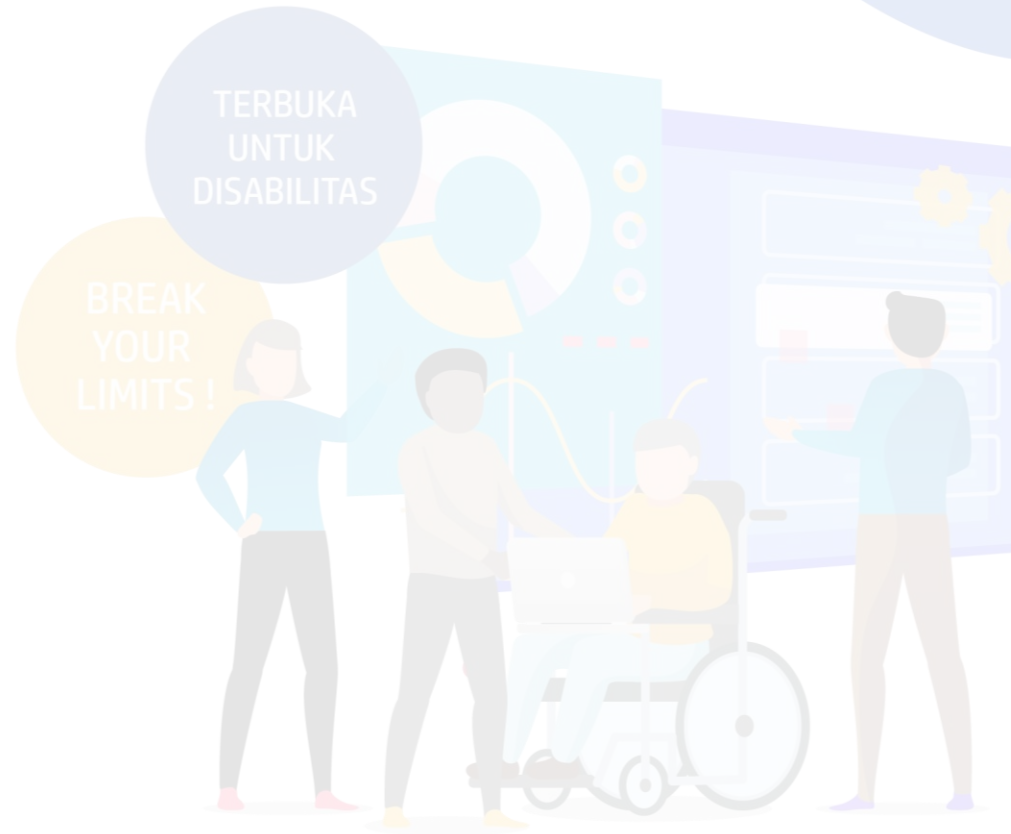
BREAK
YOUR
LIMITS !

TERBUKA
UNTUK
DISABILITAS



Latihan langsung di Kelas Ke-2 & Pembahasan

- Tidak ada

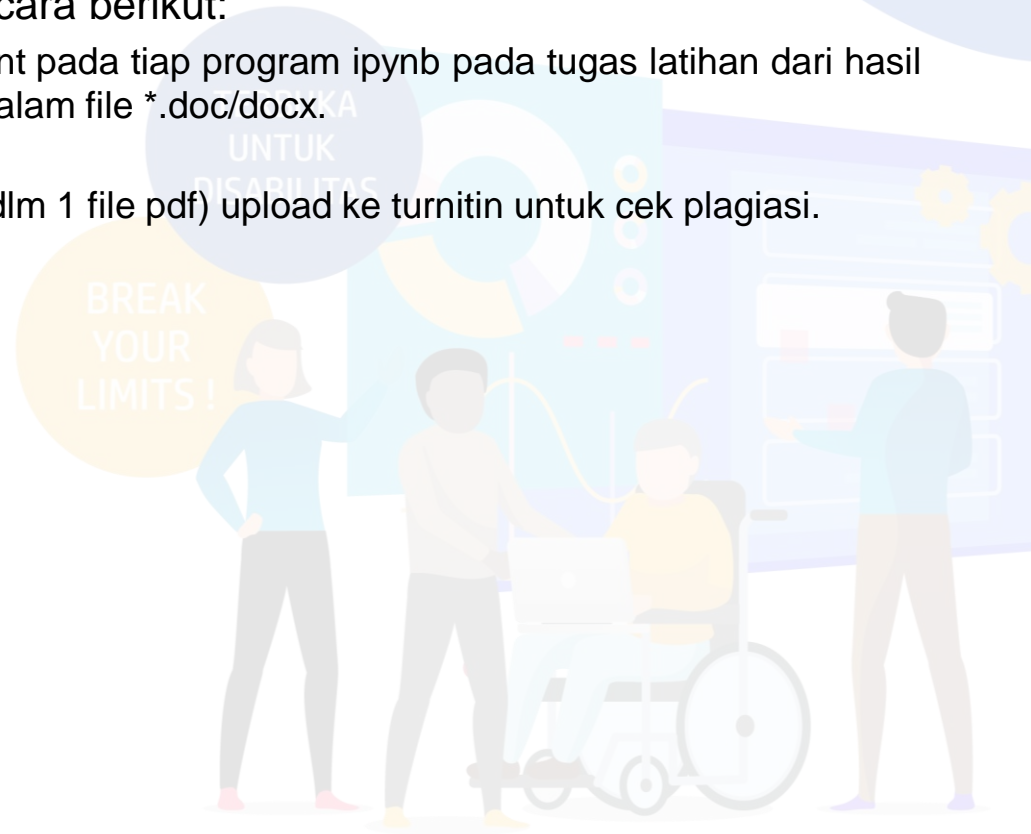


Tugas Individu

1. Buatlah rangkuman materi dengan cara berikut:

- Menambahkan penjelasan/comment pada tiap program ipynb pada tugas latihan dari hasil “Latihan langsung di Kelas Ke-1” dalam file *.doc/docx.

*semua bentuk tugas tersebut (merger dlm 1 file pdf) upload ke turnitin untuk cek plagiasi.





DIGITAL TALENT SCHOLARSHIP 2019

Big Data Analytics



Terimakasih

Oleh: Imam Cholissodin | imamcs@ub.ac.id, Putra Pandu Adikara, Sufia Adha Putri

Asisten: Guedho, Sukma, Anshori, Aang dan Gusti

Fakultas Ilmu Komputer (Filkom) Universitas Brawijaya (UB)