

**UNIVERSITAS GUNADARMA**



**PRAKTIKUM PENGANTAR KECERDASAN BUATAN**

**MANUAL BOOK**

**“SENTIMENT ANALYSIS”**

Nama	: Fajar Satriatna	(52416566)
	Muhammad Ficki	(54416852)
	Trie Fauzan Febriansyah	(57416443)
Kelas	: 3IA01	
Fakultas	: Teknologi Industri	
Jurusan	: Teknik Informatika	
Penanggung Jawab	: Giventris Jeremia	

**Praktikum Pengantar Kecerdasan Buatan Jenjang S1**

**Universitas Gunadarma**

**2018/2019**

## **KATA PENGANTAR**

Puji syukur kehadiran Allah SWT, atas segala rahmat dan hidayah-Nya. Shalawat serta salam senantiasa tercurahkan kepada junjungan Nabi Agung Muhammad SAW yang selalu kita nantikan syafa'atnya di akhirat nanti.

Penulis mengucapkan syukur kepada Allah SWT atas limpahan nikmat sehat-Nya, baik itu berupa sehat fisik maupun akal pikiran, sehingga penulis mampu untuk menyelesaikan pembuatan makalah sebagai tugas akhir dari praktikum mata kuliah Pengantar kecerdasan buatan dengan judul "Sentiment Analysis".

Penulis tentu menyadari bahwa makalah ini masih jauh dari kata sempurna dan masih banyak terdapat kesalahan serta kekurangan di dalamnya. Untuk itu, penulis mengharapkan kritik serta saran dari pembaca untuk makalah ini, supaya makalah ini nantinya dapat menjadi makalah yang lebih baik lagi. Demikian, dan apabila terdapat banyak kesalahan pada makalah ini penulis mohon maaf yang sebesar-besarnya.

Depok, 16 Januari 2019

Penyusun

## DAFTAR ISI

<b>Kata Pengantar .....</b>	<b>2</b>
<b>Daftar Isi.....</b>	<b>3</b>
<b>Bab 1 Pendahuluan .....</b>	<b>4</b>
1.1 Latar Belakang .....	4
1.2 Rumusan Masalah .....	4
1.3 Tujuan .....	4
<b>Bab 2 Pembahasan.....</b>	<b>5</b>
2.1 Neural Network .....	5
2.2 Python 3.6 .....	8
2.3 Pandas .....	9
2.4 NumPy.....	9
2.5 CountVectorizer .....	9
2.6 Scikit-Learn .....	9
<b>Bab 3 Pembuatan .....</b>	<b>10</b>
3.1 Listing Program .....	10
3.2 Logika Program .....	12
<b>Bab 4 Kesimpulan dan Saran .....</b>	<b>14</b>
4.1 Kesimpulan .....	14
4.2 Saran .....	14
<b>Daftar Pustaka .....</b>	<b>15</b>

# **BAB 1**

## **Pendahuluan**

### **1.1 Latar Belakang**

Machine learning merupakan salah satu varian dari sistem kecerdasan buatan yang memungkinkan komputer dapat belajar tanpa diprogram secara eksplisit. Secara umum, pekerjaan machine learning yang seringkali digunakan adalah untuk mengklasifikasikan satu permasalahan menjadi beberapa kelompok.

Contoh task paling sederhananya adalah sentiment analysis. Sentiment analysis adalah satu usaha di bidang machine learning untuk menilai bagaimana penilaian publik atau kelompok orang tertentu terhadap objek yang diobservasi. Biasanya, klasifikasinya dibagi menjadi 2, yaitu positif dan negatif. Supaya mudah dipahami, saya contohkan produk iPhoneXS yang baru dipasarkan di dunia. Kita bisa melakukan sentiment analysis bagaimana masyarakat menerima produk tersebut dengan mempelajari segala komentar yang ada di social media dengan hashtag iPhoneXS (#iphoneXS). Harapannya dari sentiment analysis adalah user bisa mengetahui bagaimana penilaian publik terhadap suatu objek yang kita observasi.

### **1.2 Rumusan Masalah**

1. Menjelaskan apa itu kecerdasan buatan.
2. Menjelaskan ruang lingkup dari kecerdasan buatan.
3. Menjelaskan konsep dari word2vec.

### **1.3 Tujuan**

1. Dapat mengetahui apa itu kecerdasan buatan.
2. Dapat mengetahui ruang lingkup dari kecerdasan buatan.
3. Dapat mengetahui konsep dari word2vec.

## BAB 2

### Pembahasan

#### 2.1 Neural Networks

Neural Networks (Jaringan Saraf Tiruan) menurut Haykin didefinisikan sebagai berikut :

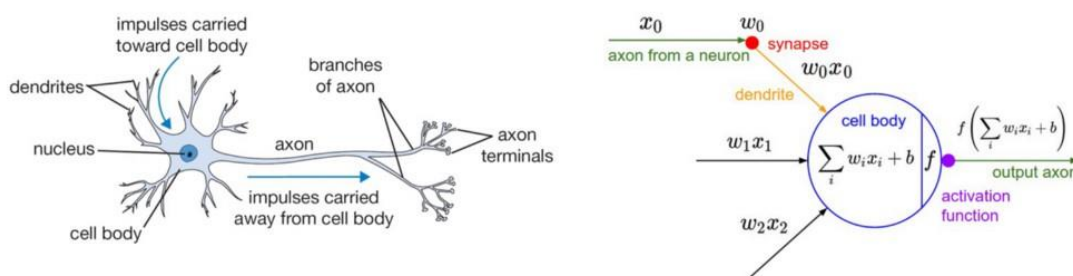
“Sebuah neural network (JST: Jaringan Saraf Tiruan) adalah prosesor yang terdistribusi paralel, terbuat dari unit-unit yang sederhana, dan memiliki kemampuan untuk menyimpan pengetahuan yang diperoleh secara eksperimental dan siap pakai untuk berbagai tujuan. Neural network ini meniru otak manusia dari sudut :

1) Pengetahuan diperoleh oleh network dari lingkungan, melalui suatu proses pembelajaran.

2) Kekuatan koneksi antar unit yang disebut synaptic weights, berfungsi untuk menyimpan pengetahuan yang telah diperoleh oleh jaringan tersebut.”

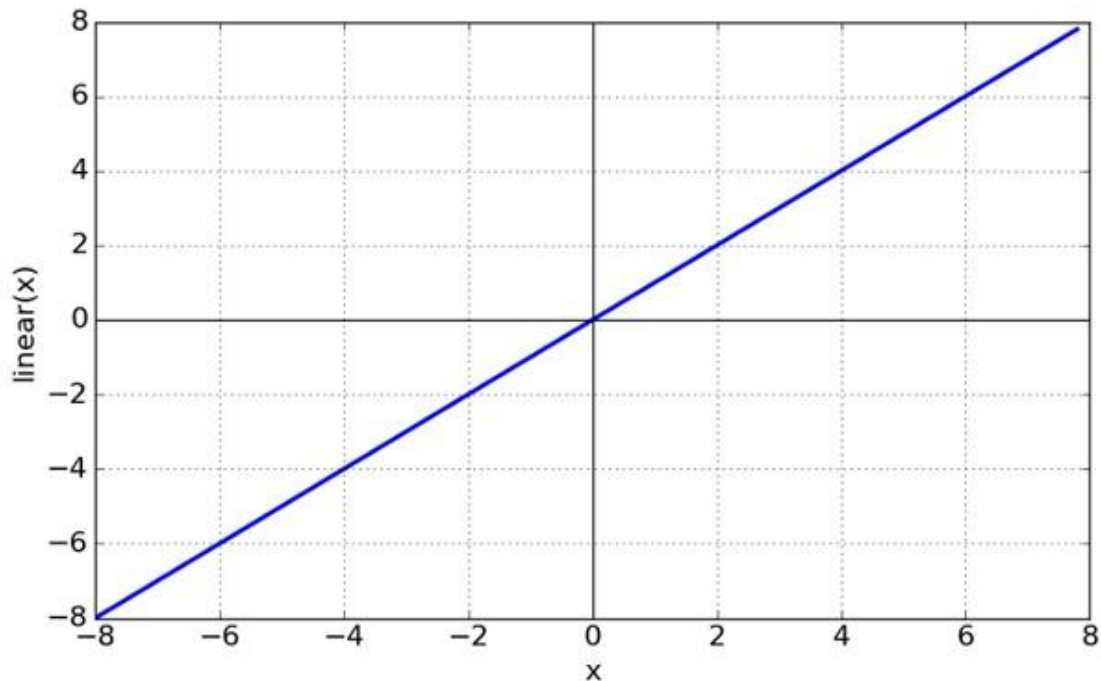
Secara sederhana, JST adalah sebuah alat pemodelan data statistik non-linier. JST dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data. Menurut suatu teorema yang disebut "teorema penaksiran universal", JST dengan minimal sebuah lapis tersembunyi dengan fungsi aktivasi non-linier dapat memodelkan seluruh fungsi terukur Borel apapun dari suatu dimensi ke dimensi lainnya. Neural network adalah model yang terinspirasi oleh bagaimana neuron dalam otak manusia bekerja. Tiap neuron pada otak manusia saling berhubungan dan informasi mengalir dari setiap neuron tersebut.

#### Activation Function



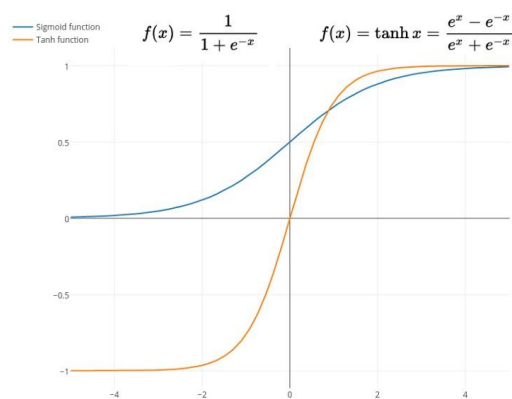
Activation function berfungsi untuk menentukan apakah neuron tersebut harus “aktif” atau tidak berdasarkan dari weighted sum dari input. Secara umum terdapat 2 jenis activation function, Linear dan Non-Linear Activation function.

## Linear Function



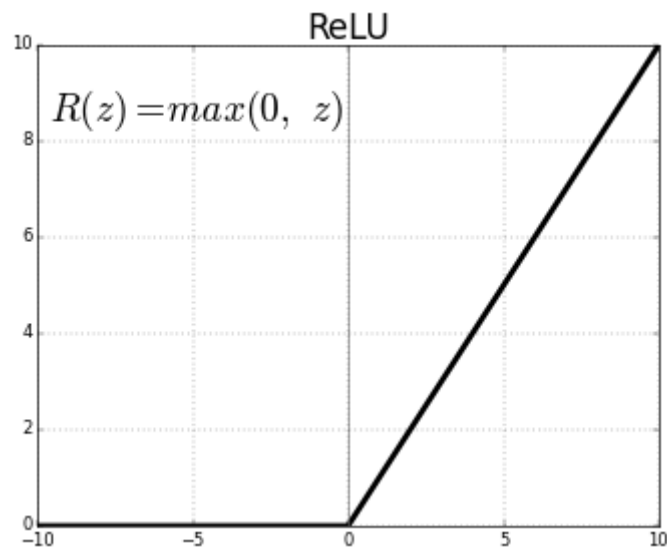
Bisa dikatakan secara “default” activation function dari sebuah neuron adalah Linear. Jika sebuah neuron menggunakan linear function, maka keluaran dari neuron tersebut adalah weighted sum dari input + bias.

## Sigmoid and Tanh Function (Non-Linear)



Sigmoid function mempunyai rentang antara 0 hingga 1 sedangkan rentang dari Tanh adalah -1 hingga 1. Kedua fungsi ini biasanya digunakan untuk klasifikasi 2 class atau kelompok data. Namun terdapat kelemahan dari kedua fungsi ini, nanti akan coba saya jelaskan di part berikutnya.

## ReLU (Non-Linear)



Pada dasarnya ReLU melakukan “treshold” dari 0 hingga infinity.

## Training a Neural Network

Pada Supervised Learning menggunakan Neural Network, pada umumnya Learning terdiri dari 2 tahap, yaitu training dan evaluation. Namun kadang terdapat tahap tambahan yaitu testing, namun sifatnya tidak wajib.

Pada tahap training setiap weight dan bias pada tiap neuron akan diupdate terus menerus hingga output yang dihasilkan sesuai dengan harapan. Pada tiap iterasi akan dilakukan proses evaluation yang biasanya digunakan untuk menentukan kapan harus menghentikan proses training (stopping point)

Pada part selanjutnya, akan saya bahas bagaimana proses training pada neural network lebih mendalam. Namun pada part ini akan dijelaskan garis besarnya saja. Proses training terdiri dari 2 tahap :

- Forward Pass
- Backward Pass

## Forward Pass

$$dot_j = \sum_i^3 w_{ji}x_i + b_j$$

$$h_j = \sigma(dot_j) = \max(0, dot_j)$$

Forward pass atau biasa juga disebut forward propagation adalah proses dimana kita membawa data pada input melewati tiap neuron pada hidden layer sampai kepada output layer yang nanti akan dihitung errornya

Persamaan diatas adalah contoh forward pass pada arsitektur pertama (lihat gambar arsitektur diatas) yang menggunakan ReLU sebagai activation function. Dimana  $i$  adalah node pada input layer (3 node input),  $j$  adalah node pada hidden layer sedangkan  $h$  adalah output dari node pada hidden layer.

## Backward Pass

Error yang kita dapat pada forward pass akan digunakan untuk mengupdate setiap weight dan bias dengan learning rate tertentu.

Kedua proses diatas akan dilakukan berulang-ulang sampai didapatkan nilai weight dan bias yang dapat memberikan nilai error sekecil mungkin pada output layer (pada saat forward pass)

## 2.2 Python 3.6

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa 5 yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

Python mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. Python dapat digunakan untuk berbagai



keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi.

### **2.3 Pandas**

Pandas adalah librari analisis data yang memiliki struktur data yang kita perlukan untuk membersihkan data mentah ke dalam sebuah bentuk yang cocok untuk analisis (yaitu tabel). Penting untuk dicatat di sini bahwa karena pandas melakukan tugas penting seperti menyelaraskan data untuk perbandingan dan penggabungan set data, penanganan data yang hilang, dll, 4 itu telah menjadi sebuah library.

### **2.4 numPy**

NumPy adalah modul yang menyediakan objek-objek matematika yang memudahkan dalam melakukan perhitungan. Untuk menggunakan SciPy, biasanya kita membutuhkan NumPy. Objek utama yang disediakan NumPy adalah array yang dapat berperan sebagai matrix. Array ini tidak sama dengan array biasa pada bahasa-bahasa pemrograman secara umum.

### **2.5 CountVectorizer**

CountVectorizer adalah perpustakaan yang dapat merubah fitur teks menjadi sebuah representasi vector.

### **2.6 Scikit-Learn**

Scikit-learn adalah perpustakaan pembelajaran mesin perangkat lunak bahasa pemrograman Python untuk berbagai klasifikasi, regresi dan algoritma clustering termasuk vector machines, random forests, gradient boosting, kmeans and DBSCAN, dan dirancang untuk beroperasi dengan perpustakaan numerik dan ilmiah Python NumPy dan SciPy.

## BAB 3

### Pembuatan

#### 3.1 Listing Program

```
import pandas as pd

import numpy as np


from google.colab import files

uploaded = files.upload()


with open("ProjekData.txt", "r") as text_file:

    lines = text_file.read().split("\n")


with open("coba1.txt", "r") as text_file:

    lines = text_file.read().split("\n")


newLines = [line.split("\t") for line in lines if len(line.split("\t")) == 2 and
line.split("\t")[1] != ""]


train_documents = [line[0] for line in newLines]


train_labels = [int(line[1]) for line in newLines]
```

```
from sklearn.feature_extraction.text import CountVectorizer

count_vectorizer = CountVectorizer(binary="true")
train_documents = count_vectorizer.fit_transform(train_documents)

train_documents

from sklearn.naive_bayes import BernoulliNB

classifier = BernoulliNB().fit(train_documents, train_labels)

classifier.predict(count_vectorizer.transform(["Ini sangat keren!"]))

classifier.predict(count_vectorizer.transform(["Kubunuh kau"]))

def predictionOutput(sentence):
    prediction = classifier.predict(count_vectorizer.transform([sentence]))
    if(prediction[0] == 1):
        print("This is a Positive Sentiment Sentence")
    elif (prediction[0] == 0):
        print("This is a Negative Sentiment Sentence")

predictionOutput("Bacot")
```

```
predictionOutput("Alhamdulillah baik")

predictionOutput( "Amsyong cuy")

predictionOutput( "Sama nih aku juga")
```

### 3.2 Logika Program

```
import pandas as pd
import numpy as np
```

Untuk mengimport library yaitu pandas dan numpy.

```
from google.colab import files
uploaded = files.upload()
```

Untuk mengupload data berupa ProjekData.txt, coba1.txt

```
[3] #membuka data yang telah di upload
    with open("ProjekData.txt", "r") as text_file:
        lines = text_file.read().split("\n")

[4] with open("coba1.txt", "r") as text_file:
        lines = text_file.read().split("\n")
```

Untuk membuka data yang telah diupload.

```
newLines = [line.split("\t") for line in lines if len(line.split("\t")) == 2 and line.split("\t")[1] != '']
```

Untuk mengkombinasi semua file yang sudah dibuka.

```
train_documents = [line[0] for line in newLines]
```

```
train_labels = [int(line[1]) for line in newLines]
```

Untuk memisahkan data ke Train Features & Train Labels.

```
from sklearn.feature_extraction.text import CountVectorizer
```

Untuk mengimport library sklearn dari CountVectorizer.

```
count_vectorizer = CountVectorizer(binary="true")
train_documents = count_vectorizer.fit_transform(train_documents)
```

Untuk menghitung jumlah vector pada train\_document.

```
train_documents
```

Untuk melihat isi dokumen.

```
from sklearn.naive_bayes import BernoulliNB
```

Untuk mengimport BernoulliNB dari sklearn.

```
[12] classifier = BernoulliNB().fit(train_documents, train_labels) #fungsi klasifikasi menggunakan bernauliNB  
[13] classifier.predict(count_vectorizer.transform(["Ini sangat keren!"])) #melatih data untuk kalimat positif  
array([1])  
[14] classifier.predict(count_vectorizer.transform(['Kubunuh Kau'])) #melatih data untuk kalimat negatif  
array([0])
```

Untuk mengklasifikasikan kalimat yang diinputkan.

```
def predictionOutput(sentence):  
    prediction = classifier.predict(count_vectorizer.transform([sentence]))  
    if(prediction[0] == 1):  
        print("This is a Positive Sentiment Sentence")  
    elif (prediction[0] == 0):  
        print("This is a Negative Sentiment Sentence")
```

Kondisi untuk menampilkan output, dimana jika nilai arraynya 1 maka dia akan positif statemen dan jika arraynya 0 maka akan menampilkan negative statemen.

```
[16] #output  
predictionOutput("Bacot")  
This is a Negative Sentiment Sentence
```

```
[17] predictionOutput("Alhamdulillah, baik")  
This is a Positive Sentiment Sentence
```

```
[18] predictionOutput("Amsyong cuy")  
This is a Negative Sentiment Sentence
```

```
predictionOutput("Sama nih aku juga")  
This is a Positive Sentiment Sentence
```

## BAB 4

### Kesimpulan dan Saran

#### 4.1 Kesimpulan

Sentiment Analysis adalah penambangan kontekstual teks yang mengidentifikasi dan mengekstrak informasi subjektif dalam sumber, dan membantu para pembisnis untuk memahami sentimen sosial dari merek, produk atau layanan mereka saat memantau percakapan online. Secara umum, sentimen analisis terbagi menjadi 2 kategori besar yaitu *Coarse-grained sentiment analisis* dan *Fined-grained sentiment analisis*. Machine Learning memerlukan dataset untuk digunakan sebagai data training. Oleh karena itu, dibutuhkan effort untuk mengumpulkan dan melakukan class tag pada sampel dataset tersebut, selain itu proses training juga membutuhkan waktu. Akurasi dari pendekatan klasifikasi machine learning sangat baik, akan tetapi performa klasifikasinya domain dependent terhadap dataset yang digunakan pada saat training.

Sentiment analysis terdiri dari 3 subproses besar. Masing-masing subproses ini bisa dijadikan bahan / topik riset secara terpisah karena masing-masing subproses ini membutuhkan teknik yang tidak mudah yakni *Subjectivity Classification*, *Orientation Detection*, dan *Opinion Holder and Target Detection*.

#### 4.2 Saran

Menyadari bahwa penulis masih jauh dari kata sempurna, kedepannya penulis akan lebih fokus dan details dalam menjelaskan tentang makalah di atas dengan sumber – sumber yang lebih banyak yang tentunya dapat di pertanggungjawabkan.

## Daftar Pustaka

<https://cran.r-project.org/web/packages/text2vec/vignettes/glove.html>  
<https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/>  
<http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/>  
<https://moondoggiesmusic.com/contoh-kata-pengantar/>  
<https://frendypratamablog.wordpress.com/2018/01/03/makalah-neural-network/>  
<https://blog.kmkonline.co.id/were-doing-machine-learning-9d4075d46cc3>  
<https://en.wikipedia.org/wiki/Scikit-learn>  
<https://code.tutsplus.com/id/tutorials/introducing-pandas--cms-26514>  
<https://copycat91.wordpress.com/2010/06/19/python-scipy-numpy-matplotlib-nice/>  
<https://medium.com/i-a/simple-nlp-sentiment-analysis-with-google-colaboratory-761a5391b57c>  
<https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac>