

# SEGMENTASI PELANGGAN MENGGUNAKAN PYTHON



NISA HANUM HARANI  
FIKRI ALDI NUGRAHA

# **SEGMENTASI PELANGGAN MENGGUNAKAN PYTHON**



---

# **SEGMENTASI PELANGGAN MENGGUNAKAN PYTHON**

---

**Nisa Hanum Harani**

**Fikri Aldi Nugraha**

Informatics Research Center



**Kreatif Industri Nusantara**

**Penulis:**

Nisa Hanum Harani

Fikri Aldi Nugraha

ISBN : 978-602-53897-6-4

**Editor:**

Fikri Aldi Nugraha

**Penyunting:**

Cahyo Prianto

**Desain sampul dan Tata letak:**

Dinda Majesty

**Penerbit:**

Kreatif Industri Nusantara

**Redaksi:**

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

**Distributor:**

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2020

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat  
menahan lelahnya  
belajar, Maka kamu harus  
sanggup menahan  
perihnya Kebodohan.’  
Imam Syafi’i*

# CONTRIBUTORS

---

Nisa Hanum Harani, Fikri Aldi Nugraha,

Informatics Research Center, Politeknik Pos Indonesia, Bandung, Indonesia

# **CONTENTS IN BRIEF**

---

<b>1 Pengenalan Data Mining</b>	<b>1</b>
<b>2 Segmentasi Dan Profiling Pelanggan</b>	<b>35</b>
<b>3 Pengenalan Algoritma</b>	<b>49</b>
<b>4 Pengenalan Python</b>	<b>97</b>
<b>5 Penerapan Segmentasi Pelanggan Menggunakan Python</b>	<b>145</b>



# DAFTAR ISI

---

Daftar Gambar	xiv
Daftar Tabel	xxi
Kata Pengantar	xxiii
Acknowledgments	xxv
Acronyms	xxvii
Glossary	xxix
Introduction	xxxi

*Nisa Hanum Harani, Fikri Aldi Nugraha.*

<b>1 Pengenalan Data Mining</b>	<b>1</b>
1.1 Apa Itu Data Mining?	1
1.2 Sejarah Data Mining	6
1.3 Tujuan Data Mining	7

## **x DAFTAR ISI**

1.4	Fungsi Data Mining	8
1.5	Tipe Data Pada Data Mining	11
1.6	Perkembangan Data Mining	11
1.7	Implementasi Data Mining	14
1.8	Pengelompokan Data Mining	19
1.9	Cara Kerja Data Mining	22
1.10	Algoritma Data Mining	23
1.11	CRISP-DM	25
1.12	Tools Data Mining	28
<b>2</b>	<b>Segmentasi Dan <i>Profiling</i> Pelanggan</b>	<b>35</b>
2.1	Segmentasi Pelanggan	35
2.2	<i>Profiling</i> Pelanggan	39
2.3	Perlunya Mempelajari Perilaku Pelanggan	41
2.4	Jenis-Jenis Pelanggan	42
2.5	Macam-Macam Model Perilaku Pelanggan	43
2.6	Faktor Yang Mempengaruhi Perilaku Pelanggan	44
<b>3</b>	<b>Pengenalan Algoritma</b>	<b>49</b>
3.1	Apa itu Algoritma?	49
3.2	Kriteria Algoritma	50
3.3	Jenis-Jenis Proses Algoritma	51
3.4	Prinsip Kerja Algoritma	52
3.5	<i>Clustering</i>	52
3.6	Manfaat <i>Clustering</i>	54
3.7	Konsep Dasar <i>Clustering</i>	55
3.8	Syarat <i>Clustering</i>	55
3.9	Metode <i>Clustering</i>	57

3.9.1	<i>Hierarchical Clustering</i>	57
3.9.2	<i>Partitional Clustering</i>	58
3.10	Algoritma K-Means	60
3.11	Implementasi K-Means	64
3.12	Kelebihan dan Kekurangan Algoritma K-Means	64
3.13	Permasalahan Terkait Algoritma K-Means	66
3.14	Contoh Perhitungan <i>Clustering</i>	69
3.15	Metode Penentuan K	90
3.15.1	Metode <i>Elbow</i>	90
3.16	Metode Evaluasi <i>Cluster</i>	92
3.16.1	Silhouette index	93
3.16.2	Davies-Bouldin Index	94
3.16.3	Calinski Harabasz Index	95
<b>4</b>	<b>Pengenalan Python</b>	<b>97</b>
4.1	Apa itu Python?	97
4.2	Sejarah Python	99
4.3	Mengapa Harus Menggunakan Python?	101
4.4	Lisensi Python	103
4.5	Kelebihan dan Kekurangan Python	104
4.6	Cakupan Aplikasi Python	106
4.7	Instalasi Python	107
4.8	Jupyter Notebook	128
4.9	<i>Library</i> Python	136
4.9.1	Numpy	140
4.9.2	Pandas	141
4.9.3	Seaborn	142
4.9.4	Scikit-learn	143

**xii DAFTAR ISI**

4.9.5 Matplotlib	144
<b>5 Penerapan Segmentasi Pelanggan Menggunakan Python</b>	<b>145</b>
5.1 Mengapa Perlu Dilakukan Segmentasi Pelanggan?	145
5.2 Penerapan CRISP-DM Pada Segmentasi Pelanggan	149
5.2.1 Analisis Permasalahan	149
5.2.2 Pengumpulan Data	150
5.2.3 Pengolahan Data	150
5.2.4 Pemodelan	151
5.2.4.1 Import Module	152
5.2.4.2 Baca Data	153
5.2.4.3 Normalisasi Data	154
5.2.4.4 Split Data	155
5.2.4.5 Penentuan Jumlah Cluster Terbaik	164
5.2.4.6 K-Means Clustering	165
5.2.4.6.1 Simulasi Pertama	165
5.2.4.6.2 Simulasi Kedua	181
5.2.4.6.3 Simulasi Ketiga	196
5.2.4.7 Pengkajian Pemodelan dan Evaluasi	210
5.2.4.7.1 Pengkajian Pemodelan	210
5.2.4.7.2 Evaluasi	213
Daftar Pustaka	217



# DAFTAR GAMBAR

---

Gambar 1.1 Ilustrasi Data Mining	3
Gambar 1.2 Proses Knowledge Discovery Database	4
Gambar 1.3 Bidang Ilmu Data Mining	14
Gambar 1.4 Proses Data Mining menurut CRISP-DM	26
Gambar 1.5 Logo Rapidminer	28
Gambar 1.6 Logo Weka	31
Gambar 1.7 Logo R	32
Gambar 3.1 Prinsip Kerja Algoritma	52
Gambar 3.2 Hierarchical Clustering	57
Gambar 3.3 Cluster Awal	59
Gambar 3.4 Proses Iterasi	59
Gambar 3.5 Cluster Akhir	60
Gambar 3.6 Contoh Dataset yang Memiliki Bentuk Khusus	68
Gambar 3.7 Diagram Alir Rancangan Penentuan Cluster Atau Segmentasi Data Pelanggan	70
Gambar 3.8 Struktur Tabel Data Pelanggan	72

Gambar 3.9 Grafik Metode Elbow	92
Gambar 4.1 Logo Bahasa Pemrograman Python	98
Gambar 4.2 Download Python	108
Gambar 4.3 Eksekusi <i>Installer</i> Python	109
Gambar 4.4 Tampilan Awal Installer Python	110
Gambar 4.5 Proses Instalasi Python	111
Gambar 4.6 Proses Instalasi Python Berhasil	112
Gambar 4.7 Pengecekan Python Melalui CMD	112
Gambar 4.8 Pengecekan Python Melalui Python Shell	113
Gambar 4.9 Pengujian Python Melalui Python Shell	114
Gambar 4.10 Download Anaconda	116
Gambar 4.11 Eksekusi Installer Anaconda	117
Gambar 4.12 Tampilan Awal Instalasi Anaconda	118
Gambar 4.13 License Agreement Anaconda	119
Gambar 4.14 Pemilihan Tipe Instalasi Anaconda	120
Gambar 4.15 Pemilihan Lokasi Instalasi Anaconda	121
Gambar 4.16 Pemilihan Opsi Lanjutan Instalasi Anaconda	122
Gambar 4.17 Proses Instalasi Anaconda	123
Gambar 4.18 Notifikasi Proses Instalasi Anaconda Selesai	124
Gambar 4.19 Notifikasi Rekomendasi Editor Untuk Menulis Kode	125
Gambar 4.20 Ucapan Terimakasih Telah Menginstal Anaconda	126
Gambar 4.21 Pengecekan Anaconda Melalui CMD	127
Gambar 4.22 Cara Menampilkan Anaconda Navigator	127
Gambar 4.23 Tampilan Awal Anaconda Navigator	128
Gambar 4.24 Menjalankan Jupyter Notebook Melalui Anaconda Navigator	132
Gambar 4.25 Tampilan Awal Jupyter Notebook	132
Gambar 4.26 Membuat File Baru Pada Jupyter Notebook	133
Gambar 4.27 Mengeksekusi Baris Perintah Pada Jupyter Notebook	134

**xvi DAFTAR GAMBAR**

Gambar 4.28 Menyimpan File Pada Jupyter Notebook	135
Gambar 4.29 Mengganti Nama File Pada Jupyter Notebook	135
Gambar 4.30 Menjalankan Jupyter Notebook Melalui Shortcut	136
Gambar 4.31 Logo Library Numpy	140
Gambar 4.32 Logo Library Pandas	141
Gambar 4.33 Logo Library Seaborn	142
Gambar 4.34 Logo Library Scikit-learn	143
Gambar 4.35 Logo Library Matplotlib	144
Gambar 5.1 Import Module atau Library	153
Gambar 5.2 Baca Data atau Load Data	153
Gambar 5.3 Normalisasi Data	154
Gambar 5.4 <i>Split Data</i> Simulasi Pertama	157
Gambar 5.5 <i>Split Data</i> Simulasi Kedua	160
Gambar 5.6 <i>Split Data</i> Simulasi Ketiga	163
Gambar 5.7 Penentuan Jumlah <i>Cluster</i> Terbaik	165
Gambar 5.8 Inisialisasi <i>Train data</i> Simulasi Pertama	166
Gambar 5.9 Model <i>Clustering Train Data</i> Simulasi Pertama	166
Gambar 5.10 Centroid <i>Train Data</i> Simulasi Pertama	167
Gambar 5.11 Proses <i>Clustering Train Data</i> Simulasi Pertama	167
Gambar 5.12 Hasil <i>Clustering Train Data</i> Simulasi Pertama	168
Gambar 5.13 Visualisasi Atribut lama_langganan dan total_tagihan Pada <i>Train Data</i> Simulasi Pertama	169
Gambar 5.14 Visualisasi Atribut lama_langganan dan jumlah_layanan Pada <i>Train Data</i> Simulasi Pertama	170
Gambar 5.15 Visualisasi Atribut jumlah_layanan dan total_tagihan Pada <i>Train Data</i> Simulasi Pertama	171
Gambar 5.16 Silhouette Score Pada <i>Train Data</i> Simulasi Pertama	172
Gambar 5.17 Davies Bouldin Index Pada <i>Train Data</i> Simulasi Pertama	173

Gambar 5.18 Calinski Harabasz Index Pada <i>Train Data</i>	
Simulasi Pertama	173
Gambar 5.19 Inisialisasi <i>Test Data</i> Simulasi Pertama	174
Gambar 5.20 Model <i>Clustering Test Data</i> Simulasi Pertama	174
Gambar 5.21 Centroid <i>Test Data</i> Simulasi Pertama	175
Gambar 5.22 Proses <i>Clustering Test Data</i> Simulasi Pertama	175
Gambar 5.23 Hasil <i>Clustering Test Data</i> Simulasi Pertama	176
Gambar 5.24 Visualisasi Atribut lama_langganan dan total_tagihan Pada <i>Test Data</i> Simulasi Pertama	177
Gambar 5.25 Visualisasi Atribut lama_langganan dan jumlah_layanan Pada <i>Test Data</i> Simulasi Pertama	178
Gambar 5.26 Visualisasi Atribut jumlah_layanan dan total_tagihan Pada <i>Test Data</i> Simulasi Pertama	179
Gambar 5.27 Silhouette score pada <i>Test Data</i> Simulasi Pertama	180
Gambar 5.28 Davies Bouldin Index pada <i>Test Data</i> Simulasi Pertama	180
Gambar 5.29 Calinski Harabasz Index pada <i>Test Data</i> Simulasi Pertama	181
Gambar 5.30 Inisialisasi <i>Train Data</i> Simulasi Kedua	181
Gambar 5.31 Model <i>Clustering</i> Simulasi Kedua	182
Gambar 5.32 Centroid <i>Train Data</i> Simulasi Kedua	182
Gambar 5.33 Proses <i>Clustering Train Data</i> Simulasi Kedua	183
Gambar 5.34 Hasil <i>Clustering Train Data</i> Simulasi Kedua	184
Gambar 5.35 Visualisasi Atribut lama_langganan dan total_tagihan Pada <i>Train Data</i> Simulasi Kedua	185
Gambar 5.36 Visualisasi Atribut lama_langganan dan jumlah_layanan Pada <i>Train Data</i> Simulasi Kedua	186
Gambar 5.37 Visualisasi Atribut jumlah_layanan dan total_tagihan Pada <i>Train Data</i> Simulasi Kedua	187
Gambar 5.38 Silhouette Score Pada <i>Train Data</i> Simulasi Kedua	188
Gambar 5.39 Davies Bouldin Index pada <i>Train Data</i> Simulasi Kedua	188

## **xviii DAFTAR GAMBAR**

Gambar 5.40 Calinski Harabasz Index Pada <i>Train Data</i> Simulasi Kedua	189
Gambar 5.41 Inisialisasi <i>Test Data</i> Simulasi Kedua	189
Gambar 5.42 Model <i>Clustering Test Data</i> Simulasi Kedua	190
Gambar 5.43 Centroid <i>Test Data</i> Simulasi Kedua	190
Gambar 5.44 Proses <i>Clustering Test Data</i> Simulasi Kedua	191
Gambar 5.45 Hasil <i>Clustering Test Data</i> Simulasi Kedua	191
Gambar 5.46 Visualisasi Atribut lama_langganan dan total_tagihan Pada <i>Test Data</i> Simulasi Kedua	192
Gambar 5.47 Visualisasi Atribut lama_langganan dan jumlah_layanan Pada <i>Test Data</i> Simulasi Kedua	193
Gambar 5.48 Visualisasi Atribut jumlah_layanan dan total_tagihan Pada <i>Test Data</i> Simulasi Kedua	194
Gambar 5.49 Silhouette Score Pada <i>Test Data</i> Simulasi Kedua	195
Gambar 5.50 Davies Bouldin Index Pada <i>Test Data</i> Simulasi Kedua	195
Gambar 5.51 Calinski Harabasz Index Pada <i>Test Data</i> Simulasi Kedua	196
Gambar 5.52 Inisialisasi <i>Train Data</i> Simulasi Ketiga	196
Gambar 5.53 Model <i>Clustering</i> Simulasi Ketiga	197
Gambar 5.54 Centroid <i>Train Data</i> Simulasi Ketiga	197
Gambar 5.55 Proses <i>Clustering Train Data</i> Simulasi Ketiga	198
Gambar 5.56 Hasil <i>Clustering Train Data</i> Simulasi Ketiga	198
Gambar 5.57 Visualisasi Atribut lama_langganan dan total_tagihan Pada <i>Train Data</i> Simulasi Ketiga	199
Gambar 5.58 Visualisasi Atribut lama_langganan dan jumlah_layanan Pada <i>Train Data</i> Simulasi Ketiga	200
Gambar 5.59 Visualisasi Atribut jumlah_layanan dan total_tagihan Pada <i>Train Data</i> Simulasi Ketiga	201
Gambar 5.60 Silhouette Score Pada <i>Train data</i> Simulasi Ketiga	202
Gambar 5.61 Davies Bouldin Index Pada <i>Train data</i> Simulasi Ketiga	202
Gambar 5.62 Calinski Harabasz Index Pada <i>Train data</i> Simulasi Ketiga	203

Gambar 5.63 Inisialisasi <i>Test Data</i> Simulasi Ketiga	203
Gambar 5.64 Model <i>Clustering Test Data</i> Simulasi Ketiga	204
Gambar 5.65 Centroid <i>Test Data</i> Simulasi Ketiga	204
Gambar 5.66 Proses <i>Clustering Test Data</i> Simulasi Ketiga	205
Gambar 5.67 Hasil <i>Clustering Test Data</i> Simulasi Ketiga	205
Gambar 5.68 Visualisasi Atribut lama_langganan dan total_tagihan Pada <i>Test Data</i> Simulasi Ketiga	206
Gambar 5.69 Visualisasi Atribut lama_langganan dan jumlah_layanan Pada <i>Test Data</i> Simulasi Ketiga	207
Gambar 5.70 Visualisasi Atribut jumlah_layanan dan total_tagihan Pada <i>Test Data</i> Simulasi Ketiga	208
Gambar 5.71 Silhouette Score pada <i>Test Data</i> Simulasi Ketiga	209
Gambar 5.72 Davies Bouldin Index pada <i>Test Data</i> Simulasi Ketiga	209
Gambar 5.73 Calinski Harabasz Index pada <i>Test Data</i> Simulasi Ketiga	210



# DAFTAR TABEL

---

Tabel 1.1 Tipe Data Dalam Data Mining	11
Tabel 2.1 Karakteristik Pelanggan	38
Tabel 3.1 Atribut Yang Digunakan Untuk <i>Clustering</i>	76
Tabel 3.2 Hasil Normalisasi	78
Tabel 3.3 <i>Centroid</i> awal	79
Tabel 3.4 Hasil Perhitungan Jarak Data Dengan <i>Centroid</i>	80
Tabel 3.5 Hasil Pemetaan <i>Centroid</i> Awal	81
Tabel 3.6 Hasil Perhitungan Jarak Data Dengan Centroid	84
Tabel 3.7 Hasil Pemetaan Centroid Iterasi Pertama	85
Tabel 3.8 Hasil Perhitungan Jarak Data Dengan Centroid	88
Tabel 3.9 Hasil Pemetaan <i>Centroid</i> Iterasi Kedua	89
Tabel 4.1 Release Version Python	100
Tabel 5.1 Dataset Pelanggan	151
Tabel 5.2 Hasil Evaluasi Performance K-Means Simulasi 1	211
Tabel 5.3 Hasil Evaluasi Performance K-Means Simulasi 2	212
Tabel 5.4 Hasil Evaluasi Performance K-Means Simulasi 3	212



# KATA PENGANTAR

---

Penulisan buku ini dimaksudkan untuk memberikan arahan atau sarana pembelajaran kepada pembaca mengenai data mining serta penerapan data mining dalam kehidupan sehari-hari yang kemudian proses data mining tersebut dibuat dengan menggunakan tools bahasa pemrograman python.

Nisa Hanum Harani, Fikri Aldi Nugraha

Bandung, Jawa Barat

Januari, 2020



# **ACKNOWLEDGMENTS**

---

Puji syukur penulis haturkan kepada Allah SWT atas berkah, rahmat dan hidayahnya sehingga penulis dapat menyelesaikan buku yang berjudul "Segmentasi Pelanggan Menggunakan Python". Dalam menulis buku ini banyak orang yang telah memotivasi, dan menyampaikan saran bagi penulis. Dalam kesempatan yang berharga ini, penulis bermaksud menyampaikan rasa terima kasih dan penghargaannya kepada mereka semua. Pertama, penghargaan terdalam dari penulis diberikan kepada ibunda tercinta untuk do'a dan dukungannya. Penulis juga ucapan terima kasih kepada pembimbing yang mana tanpa bantuan, dukungan dan saran pembimbing penulis, buku ini mungkin tidak akan terselesaikan. Buku ini jauh dari sempurna, tetapi diharapkan dapat bermanfaat tidak hanya bagi penulis, tetapi juga bagi pembaca. Untuk alasan ini, penulis menyambut baik saran dan kritik bijaksana yang konstruktif.

F. A. N.



# **ACRONYMS**

---

CRISP-DM	Cross-Industry Standard Process for Data Mining
WEKA	Waikato Environment for Knowledge Analysis
GUI	Graphical User Interface
GPL	General Public License



# **GLOSSARY**

---

Algoritma	Langkah-langkah atau urutan untuk memecahkan suatu masalah berdasarkan tahapan logis secara sistematis dalam periode tertentu.
Clustering	Salah satu teknik yang digunakan untuk mengelompokan data menjadi beberapa kelompok berdasarkan kesamaan karakteristik masing-masing anggota.
Data	Kumpulan fakta yang terekam atau sebuah entitas yang tidak memiliki arti dan selama ini terabaikan.
K-Means	Salah satu algoritma clustering non hirarki yang menggunakan metode partisi untuk pembagian data kedalam masing-masing kelompok.
Model	tiruan dari suatu sistem ataupun kejadian yang sebenarnya yang didalamnya memiliki sebuah informasi penting yang dapat dianalisis untuk penelitian lebih lanjut.
Pelanggan	Orang yang membeli produk atau jasa yang telah dipasarkan oleh perusahaan, dimana pemberlilan tersebut tidak hanya terjadi sekali melainkan dapat terjadi berulang-ulang.



# **INTRODUCTION**

---

NISA HANUM HARANI, FIKRI ALDI NUGRAHA.

Informatics Research Center

Bandung, Jawa Barat, Indonesia

Data mining merupakan suatu proses penggalian data untuk mencari suatu informasi, salah satunya adalah pada penentuan segmentasi pelanggan. Kemudian proses penggalian data tersebut dilakukan dengan menggunakan tools bahasa pemrograman python.



# BAB 1

---

## PENGENALAN DATA MINING

---

### 1.1 Apa Itu Data Mining?

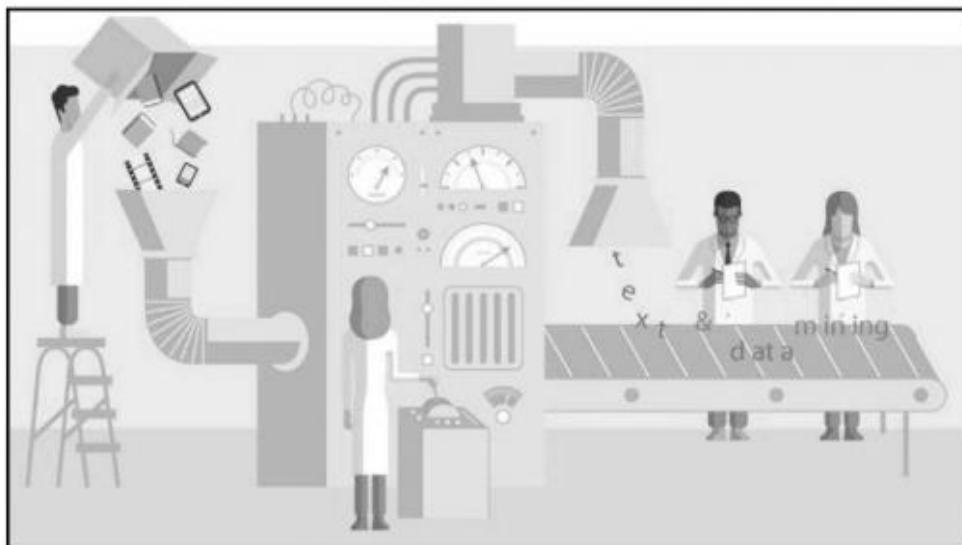
Dalam definisi sederhana, data mining merupakan ekstraksi informasi ataupun pola yang penting dan menarik dari data yang terdapat dalam sebuah *database* dalam jumlah besar. Data mining juga dapat diartikan sebagai suatu istilah yang digunakan untuk menguraikan pengetahuan di dalam *database*. Data mining merupakan gabungan dari 2 (dua) kata yang masing-masing kata tersebut memiliki definisi yang berbeda yaitu sebagai berikut.

1. Data yaitu kumpulan fakta yang terekam atau sebuah entitas yang tidak memiliki arti dan selama ini terabaikan.
2. Mining yaitu proses penambangan.

## 2 PENGENALAN DATA MINING

Sehingga data mining dapat diartikan sebagai proses penambangan data yang menghasilkan sebuah *output* berupa pengetahuan. Berikut merupakan definisi data mining yang dikutip dari beberapa sumber, yaitu:

1. Data mining merupakan teknik untuk melakukan analisis otomatis dari data dalam jumlah besar dengan kompleksitas tinggi dengan tujuan untuk menemukan pola yang biasanya tidak disadari keberadaannya (Pramudiono, 2006).
2. Data mining dapat didefinisikan sebagai sebuah analisis dari kumpulan data untuk menemukan keterkaitan yang tidak diduga dan meringkas data dengan cara yang berbeda sehingga data tersebut dapat dipahami dan bermanfaat bagi pemilik data (Larose, 2005).
3. Data mining merupakan hasil kolaborasi dari beberapa bidang keilmuan yang menggabungkan teknik dari *machine learning*, pengenalan pola, statistik, *database*, serta visualisasi untuk dapat menangani permasalahan pengambilan informasi dari database dalam jumlah besar. (Larose, 2005)
4. Data mining merupakan sebuah proses ekstraksi dari suatu data menjadi sebuah informasi, pengetahuan maupun pola dari data yang berjumlah besar (Written, Ian H. Frank, 2011).
5. Data mining menurut Gartner Group merupakan suatu proses untuk menemukan hubungan yang berarti pola dan kecenderungan dengan memeriksa dalam sekumpulan data yang tersimpan dalam penyimpanan (*database*) dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika (Larose, 2005).



Gambar 1.1 Ilustrasi Data Mining

Dari definisi yang telah disampaikan dapat disimpulkan bahwa data mining merupakan proses penambangan data dalam jumlah besar untuk memperoleh pengetahuan dengan menyatukan teknik dari berbagai bidang yaitu pembelajaran mesin, pengenalan pola, statistic, database, dan visualisasi informasi.

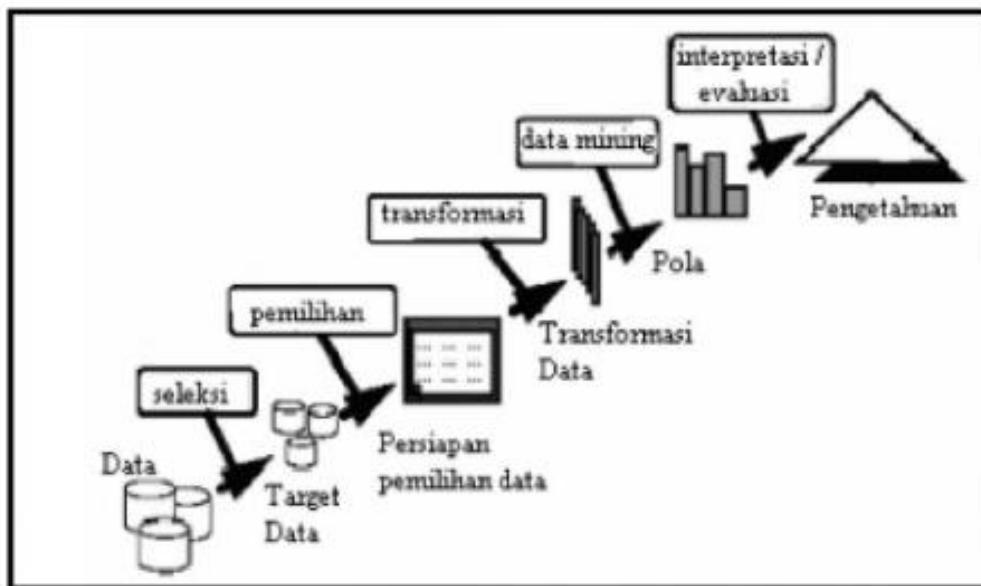
Data mining tidak hanya digunakan untuk melakukan penambangan data pada data transaksi saja. Penelitian di bidang data mining saat ini sudah merambah ke sistem database lanjut seperti *object oriented database*, *image/spatial database*, *time-series data/temporal database*, *text* (dikenal dengan nama *text mining*) dan *multimedia database*. Kemudian data mining juga memiliki karakteristik sebagai berikut:

- a. Data mining sangat berkaitan dengan penemuan sesuatu yang tersembunyi serta pola data tertentu yang belum diketahui sebelumnya.
- b. Data yang diproses menggunakan teknik data mining biasanya dalam jumlah besar. Biasanya data tersebut digunakan untuk membuat hasil lebih dapat dipercaya.

#### 4 PENGENALAN DATA MINING

- c. Data mining berguna untuk membuat keputusan kritis.

Dalam jurnal ilmiah data mining dikenal dengan istilah *Knowledge Discovery in Database* (KDD). Proses KDD secara garis besar dapat dijelaskan sebagai berikut (Fayyad, 1996).



Gambar 1.2 Proses Knowledge Discovery Database

##### 1. *Data Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi saat KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses data mining disimpan dalam suatu berkas yang terpisah dari database operasional.

##### 2. *Pre-processing/Cleaning*

Proses *cleaning* merupakan tahapan awal dari pemrosesan data pada data mining serta menjadi fokus dalam KDD. Terdapat beberapa tugas dalam proses *cleaning* diantaranya membuang duplikasi data, memeriksa data yang tidak konsisten, serta memperbaiki kesalahan yang terdapat pada data. Kemudian dilakukan juga dilakukan proses *enrichment* atau proses untuk

memperkaya data yang sudah ada dengan menggunakan data atau informasi lain yang relevan serta diperlukan untuk kebutuhan KDD.

### 3. *Transformation*

*Coding* merupakan sebuah proses untuk mentransformasikan data yang telah dipilih sehingga data tersebut dapat memenuhi kriteria untuk dilakukan proses data mining. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

### 4. *Data Mining*

*Data mining* merupakan proses untuk mencari pola ataupun informasi menarik dalam kumpulan data dengan menggunakan teknik atau metode tertentu. Metode atau algoritma yang digunakan sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

### 5. *Interpretation/Evaluation*

Pola atau informasi hasil dari proses data mining perlu divisualisasikan dalam bentuk yang mudah dipahami oleh berbagai pihak yang berkepentingan. Pada tahap ini dilakukan pula pemeriksaan apakah pola atau informasi yang dihasilkan bertentangan dengan fakta atau hipotesis yang telah ada sebelumnya.

Perlukah kita mempelajari data mining? Perlu, karena manusia banyak sekali menghasilkan data dalam jumlah besar dari berbagai bidang baik dalam bidang Bisnis, Kedokteran, Cuaca, Olahraga, Politik dan sebagainya. Dari bidang olahraga kita mengetahui berapa perolehan gol dari setiap tim dalam satu musim.

Pada bidang bisnis khususnya saham, kita memperoleh data dari Bursa Efek Jakarta, kapan harga saham naik ataupun turun. Kemudian pada bidang cuaca kita memiliki data curah hujan, tingkat kelembaban, dan lain sebagainya yang diperoleh dari BMKG.

## 1.2 Sejarah Data Mining

Tahun 90-an merupakan awal kemunculan data mining. Data mining memang termasuk ke dalam salah satu cabang ilmu computer yang relatif baru. Dan hingga saat ini orang-orang masih memperdebatkan data mining untuk ditempatkan di dalam bidang ilmu mana, karena data mining menyangkut berbagai aspek seperti *database*, kecerdasan buatan (*artificial intelligence*), statistik, dan sebagainya. Namun demikian, terdapat beberapa pihak yang berpendapat bahwa data mining tidak lebih dari sebuah *machine learning* atau analisa statistik yang berjalan di atas sebuah *database*. Namun beberapa pihak berpendapat bahwa *database* memiliki peranan yang penting dalam data mining, dikarenakan data mining mengakses data yang ukurannya besar, dan oleh karena itu database memiliki peranan penting terutama dalam proses optimisasi querinya. Data mining hadir dengan dilatarbelakangi terjadinya ledakan jumlah data yang dialami saat ini dimana banyak organisasi baik itu milik pemerintah maupun non-pemerintah yang telah mengumpulkan banyak data sekian tahun lamanya.

Data tersebut hampir semuanya dimasukkan ke dalam komputer yang digunakan untuk menangani transaksi sehari-hari. Bisa dibayangkan berapa transaksi perbankan yang terjadi dari sebuah bank dalam seharinya dan betapa besarnya data yang dimiliki oleh bank tersebut jika aktivitas tersebut telah berjalan beberapa tahun. Dari hal tersebut timbul sebuah pertanyaan, apakah data tersebut akan tersimpan begitu saja lalu pada akhirnya dibuang, ataukah kita dapat menggali informasi yang berguna untuk organisasi dari sekian banyak data tersebut.

Data mining melakukan eksplorasi terhadap basis data untuk menemukan pola yang tersembunyi serta mencari informasi yang mungkin saja tidak disadari oleh pelaku bisnis karena hal tersebut diluar pemikiran mereka.

Sementara itu pelaku bisnis tersebut memiliki kebutuhan untuk memanfaatkan kumpulan data yang sudah dimiliki, melihat peluang tersebut para peneliti menciptakan sebuah teknologi baru yang dapat menjawab kebutuhan tersebut, yaitu data mining. Teknologi tersebut sekarang sudah banyak digunakan oleh berbagai perusahaan untuk memecahkan berbagai persoalan bisnis. Meningkatnya kebutuhan dunia bisnis yang selalu ingin memperoleh nilai tambah dari data yang telah dikumpulkan telah mendorong penerapan teknik analisis data yang berasal dari berbagai bidang seperti statistika, kecerdasan buatan, dan lain sebagainya untuk mengolah data berskala besar tersebut.

Pada awalnya data mining hanya diimplementasikan dalam dunia bisnis, namun saat ini data mining juga diimplementasikan dalam bidang lain yang memiliki kebutuhan untuk analisis data dalam skala besar seperti bidang bioinformasi dan pertahanan negara.

### **1.3 Tujuan Data Mining**

Penggunaan data mining tidak hanya sekedar hanya menggabungkan beberapa bidang untuk memperoleh suatu informasi saja, lebih dari itu memiliki tujuan utama yaitu sebagaimana dijelaskan berikut ini (Hoffer, Presscott, dan McFadden, 2007).

#### *a. Explanatory*

Untuk menjelaskan beberapa kondisi yang terdapat di dalam penelitian, seperti mengapa penjualan handphone meningkat di suatu negara.

#### *b. Confirmatory*

Untuk mempertegas hipotesis, seperti halnya suatu promosi yang dilakukan pada social media lebih menarik perhatian banyak orang daripada promosi yang dilakukan pada media cetak.

c. *Exploratory*

Untuk menganalisis data yang memiliki hubungan yang baru. Misalnya, pola apa yang cocok diterapkan untuk strategi promosi penjualan.

## 1.4 Fungsi Data Mining

Teknik data mining tidak hanya digunakan untuk menemukan pola namun juga dapat digunakan untuk meperediksi tren masa kini. Selain itu data mining juga memiliki keuntungan yang kompetitif termasuk di dalamnya peningkatan pendapatan, berkurangnya pengeluaran, dan meningkatnya kemampuan pasar. Menurut Han dan Kamber (2006), data mining dibagi menjadi 2 kategori yakni.

### 1. Prediktif

Data mining dikategorikan sebagai prediktif dapat diefinisikan bahwa data mining dapat digunakan untuk melakukan prrediksi terhadap nilai dari atribut tertentu berdasarkan pada nilai atribut lain. Atribut yang diprediksi secara umum dikenal sebagai target atau biasa disebut variabel tak bebas, sedangkan atribut-atribut yang digunakan untuk membuat prediksi dikenal dengan istilah explanatory atau variable bebas.

### 2. Deskriptif

Tujuan dari tugas deskriptif adalah untuk menurunkan pola-pola (korelasi, trend, cluster, teritori, dan anomali) yang meringkas hubungan pokok dalam data. Tugas tersebut sering berupa penyelidikan serta seringkali memerlukan teknik *post-processing* untuk melakukan validasi dan untuk memberikan hasil yang lebih jelas.

Kemudian Han dan Kamber (2006) berpendapat bahwa itu data mining juga memiliki beberapa fungsionalitas diantaranya yaitu *Concept/Class Description: Characterization and Discrimination, Mining Frequent Patterns, Associations, and Correlations, Clasifiaction and Prediction, Cluster Analysis, Outlier analysis and Evolution analysis*.

Berikut ini merupakan penjelasan dari masing-masing fungsi yang telah disebutkan.

### 1. *Concept/Class Description: Characterization and Discrimination*

Data characterization adalah ringkasan dari semua karakteristik atau fitur dari data yang telah diperoleh dari target kelas atau fitur dari data yang telah diperoleh dari target kelas. Data yang sesuai dengan kelas yang telah ditentukan oleh pengguna biasanya dikumpulkan di dalam database. Sedangkan data discrimination adalah perbandingan antara fitur umum objek data target kelas dengan fitur umum objek dari satu atau satu set kelas lainnya. Target diambil melalui query database.

### 2. *Mining Frequent Patterns, Associations, and Correlations*

*Frequent patterns* merupakan istilah pola yang sering muncul di dalam data. *Frequent pattern* memiliki jenis yang beragam termasuk di dalamnya pola, sekelompok item set, sub-sequence, dan sub struktur. Sebuah *frequent patterns* biasnaya mengacu pada satu set item yang sering muncul bersama dalam sebuah kumpulan data transaksional.

Association analysis adalah pencarian aturan-aturan asosiasi yang menunjukkan kondisi-kondisi nilai atribut yang sering terjadi bersama dalam sekumpulan data. Association analysis sering digunakan untuk menganalisa *Market Basket Analysis* dan data transaksi.

### 3. *Classification and Prediction*

Klasifikasi merupakan proses untuk membentuk sebuah model ataupun fungsi yang dapat menggambarkan serta membedakan kelas data atau konsep dengan tujuan untuk memprediksi kelas dari data yang belum diketahui kelasnya. Model turunan yang dibuat harus berdasarkan hasil dari analisis dari training data. Model yang diturunkan tersebut dapat direpresentasikan dalam berbagai bentuk seperti if-then klasifikasi, decision tree, dan sebagainya.

Teknik *classification* bekerja dengan mengelompokkan data berdasarkan *training data* dan nilai atribut klasifikasi. Aturan pengelompokan

## 10 PENGENALAN DATA MINING

tersebut akan digunakan untuk klasifikasi data bar uke dalam kelompok yang ada. Classification dapat direpresntasikan dalam bentuk pohon keputusan. Setiap node dalam pohon keputusan menyatakan suatu tes terhadap atribut dataset, sedangkan setiap cabang menyatakan hasil dari test tersebut. Pohon keputusan tersebut dapat didefinisikan sebagai kumpulan aturan dalam bentuk IF condition THEN outcome. (Mewati Ayub, 2007: 7)

### 4. *Cluster Analysis*

*Cluster* merupakan kumpulan dari objek data yang memiliki kemiripan satu dengan yang lain dalam suatu kelompok yang sama dan berbeda dengan objek data yang terdapat pada kelompok lain. Sedangkan, clustering atau analisis *cluster* adalah proses pengelompokan satu set benda-benda fisik atau abstrak ke dalam kelas objek yang sama. Tujuannya adalah untuk menghasilkan pengelompokan objek yang mirip satu sama lain dalam suatu kelompok. Semakin besar kemiripan objek dalam suatu cluster serta semakin besar pula perbedaan dalam suatu cluster makan kualitas analisis cluster tersebut semakin baik.

### 5. *Outlier Analysis*

*Outlier* merupakan objek data yang tidak dapat mengikuti perilaku umum dari data. Outlier dianggap sebagai noise atau pengecualian. Analisis data outliers dapat dianggap sebagai noise atau pengecualian. Analisis data outlier dinamakan outlier mining. Teknik ini biasanya digunakan dalam fraud detection dan rare events analysis.

### 6. *Evolution Analysis*

Analisis evolusi data dapat digunakan untuk menjelaskan serta memodelkan trend dari objek yang memiliki perilaku yang dapat berubah setiap waktu. Teknik ini dapat meliputi krakterisasi, diskriminasi, asosiasi, klarifikasi, atau *clusteing* dari data yang berkaitan dengan waktu.

## 1.5 Tipe Data Pada Data Mining

Secara garis besar terdapat 2 (dua) tipe data yang harus dipahami dalam data mining yaitu:

1. Numeric merupakan tipe data yang bisa di kalkulasi.
2. Nominal merupakan tipe data yang tidak bisa di kalkulasi baik tambah, kurang, kali, maupun bagi.

Contoh pemanfaatan tipe data dapat terlihat pada Tabel 1.1 berikut.

Tabel 1.1 Tipe Data Dalam Data Mining

No	NAMA	V1	V2	V3	Ket
1	Dini	0.25	73.6	79.3	Gagal
2	Dino	3.75	98.9	87	Lulus
3	Dina	3.85	99	85	Lulus
4	Dani	0.56	60.3	65	Gagal
5	Dana	3.15	95.7	84.3	Lulus
6	Danu	0.35	52.6	56	Gagal
7	Doni	1.72	68.3	73	Gagal
8	Dono	0.75	79.4	80	Gagal

Numeric

Nominal

## 1.6 Perkembangan Data Mining

Perkembangan awal dari data mining dimulai pada tahun 1763 ketika Thomas Bayes mempublikasikan Teorema Bayes. Teori ini sangat penting dalam data mining, karena memungkinkan estimasi suatu kejadian berdasarkan kejadian yang telah berlangsung. Pada tahun 1805 mulai berkembang teori regresi yang mempelajari hubungan antar variabel. Regresi menjadi salah satu alat penting dalam data mining.

Penggunaan komputer untuk mengolah data dalam jumlah besar dimulai ketika Alan Turing memperkenalkan ide mesin pengolah data yang bersifat

## 12 PENGENALAN DATA MINING

universal pada tahun 1936. Tahun 1943 Warren McCulloch dan Walters Pitts menciptakan konsep dasar jaringan syaraf tiruan. Konsep jaringan syaraf tiruan berusaha meniru cara kerja otak manusia dalam mengingat pola. Pengembangan sistem basis data yang pesat mulai tahun 1970 memungkinkan manusia untuk menyimpan dan mengelola data berukuran besar. Perkembangan itu diikuti pula oleh perkembangan berbagai algoritma untuk pengolahan data, misalnya algoritma genetika pada tahun 1975 dan *Support Vector Machines (SVM)* pada tahun 1992.

Perkembangan pesat data mining, baik dari segi perangkat keras maupun algoritma, memungkinkan implementasi data mining dalam berbagai bidang.

Kemajuan luar biasa yang terus berlanjut pada data mining yang didorong oleh beberapa faktor, antara lain (Larose,2005) :

1. Mempunyai pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data, sehingga seluruh perusahaan memiliki akses ke dalam *database* yang andal.
3. Peningkatan akses data melalui navigasi web dan intranet.
4. Adanya tekanan kompetisi bisnis untuk meingkatkan penguasaan pasar dalam globalisasi ekonomi.
5. Perkembangan teknologi dari perangkat lunak untuk mendukung proses data mining.
6. Perkembangan yang hebat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan.

Terdapat beberapa hal penting yang perlu diperhatikan terkait dengan data mining diantaranya adalah sebagai berikut.

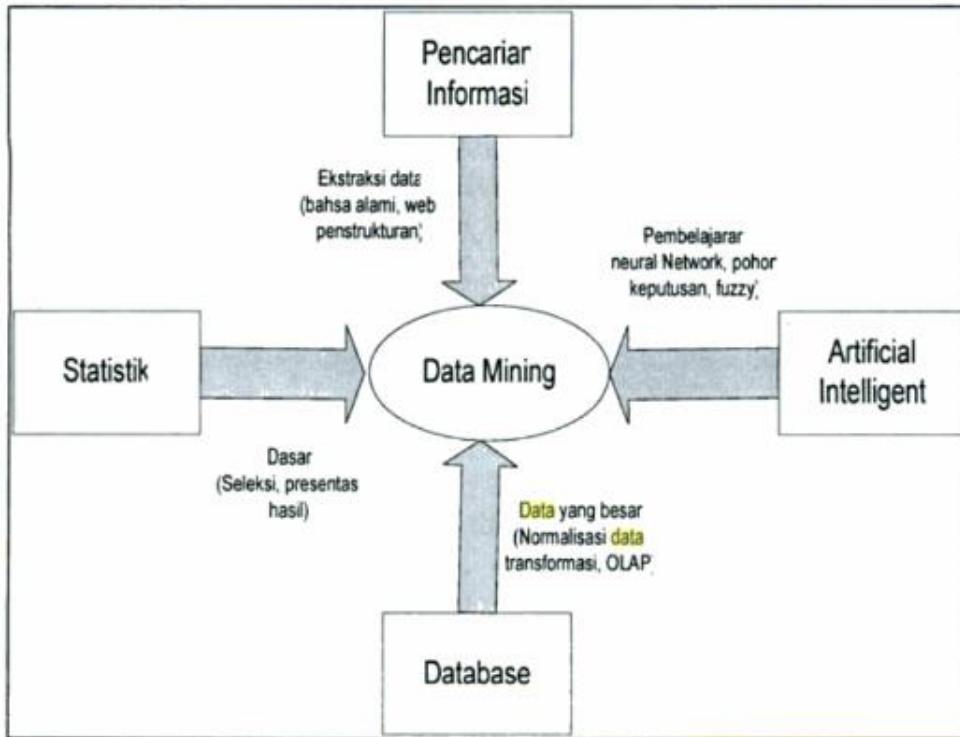
1. Data mining adalah suatu proses otomatis terhadap data yang sudah ada.
2. Data yang digunakan pada proses data mining merupakan data dalam jumlah besar.

3. Tujuan data mining adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

Keterkaitan antara satu pola dengan yang lain dapat berupa keterkaitan antara dua dimensi atau lebih. Misalnya dalam dimensi produk dapat melihat keterkaitan pembelian suatu produk dengan produk yang lain. Selain itu, hubungan juga dapat dilihat antara dua atau lebih atribut dan dua atau lebih objek (Ponniah, 2001). Berry dan Linoff berpendapat dalam bukunya yang berjudul "*Data Mining Technique for Marketing, Sales, and Customers Support*" bahwa data mining dapat dijadikan sebagai suatu proses untuk melakukan eksplorasi dan analisis baik secara otomatis maupun semiotomatis terhadap data dalam jumlah yang besar dengan tujuan untuk menemukan pola atau aturan yang berarti. (Larose, 2005)

Buku *Mastering Data Mining* yang terbit pada tiga tahun kemudian juga mendefinisikan ulang dari pengertian data mining serta memberikan pernyataan bahwa terdapat penyesalan dalam penggunaan frasa "secara otomatis maupun semiotomatis" karena kami merasa hal tersebut memberikan fokus lebih pada teknik otomatis dan kurang pada eksplorasi dan analisis. Hal tersebut memberikan pemahaman yang salah bahwa data mining merupakan produk yang dapat dibeli dibandingkan keilmuan yang harus dikuasai (Larose,2005).

Dari pernyataan tersebut menegaskan bahwa dalam data mining otomatisasi tidak mengantikan campur tangan manusia. Manusia harus ikut aktif dalam setiap fase proses data mining. Kehebatan data mining yang terdapat saat ini memungkinkan terjadinya kesalahan penggunaan yang berakibat fatal. Pengguna mungkin menrapkan analisis yang tidak tepat terhadap kumpulan data dengan menggunakan pendekatan yang berbeda. Oleh karena itu, dibutuhkan pemahaman tentang statsitik dan struktur model matematika yang mendasari kerja perangkat lunak (Larose,2005).



Gambar 1.3 Bidang Ilmu Data Mining

Terdapat kesulitan dalam mendefinisikan data mining, salah satu diantaranya adalah kenyataan bahwa data mining mewarisi banyak aspek dan teknik dari bidang ilmu yang sudah mapan terlebih dahulu. Gambar 1.3 menunjukkan bahwa data mining memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent*), machine learning, statistik, data base, dan juga information retrieval (Pramudiono, 2006).

## 1.7 Implementasi Data Mining

Data mining memiliki penerapan yang sangat luas di berbagai bidang, hal tersebut dikarenakan aktivitas yang dilakukan banyak menghasilkan data. Beikut ini merupakan contoh penerapan data mining dalam berbagai bidang.

## **1. Kesehatan**

Pada bidang kesehatan data mining memiliki potensi yang besar untuk memperbaiki kesehatan. Analisis terhadap data praktik terbaik untuk meningkatkan perawatan serta menimalkan biaya. Selain itu juga data mining diimplementasikan untuk diagnosis sebuah penyakit dan digunakan untuk memprediksi volume pasien dari setiap kategori. Selain itu data mining juga digunakan oleh perusahaan asuransi untuk menghindari kecurangan.

## **2. Analisis pasar**

Analisis pasar digunakan untuk melakukan pemodelan dengan menarik kesimpulan bahwa jika seseorang membeli item tertentu, maka cenderung akan membeli item lainnya. Teknik ini memungkinkan seorang penjual memahami perilaku dari pelanggannya sehingga penataan item pada toko akan disesuaikan dengan hasil analisis tersebut. Selain itu analisis tersebut juga untuk perbandingan antara pelanggan dalam suatu kelompok demografis yang berbeda.

## **3. Pendidikan**

Terdapat bidang baru yang muncul dalam Pendidikan, yaitu Educational Data Mining (EDM). Bidang tersebut berkaitan dengan metode pengembangan dengan menemukan pengetahuan baru yang berasal dari lingkungan Pendidikan. Tujuan dari EDM dapat diidentifikasi sebagai perilaku belajar siswa di masa yang akan datang, mempelajari dampak dari dukungan pendidikan serta untuk memajukan pengetahuan ilmiah tentang pembelajaran.

## **4. Rekayasa Manufaktur**

Dalam bidang manufaktur data mining dapat digunakan untuk menemukan proses manufaktur yang kompleks. Data mining dapat juga digunakan untuk membuat suatu rancangan sistem yang digunakan untuk

## 16 PENGENALAN DATA MINING

mengekstrak hubungan antara arsitektur produk, portofolio produk, data kebutuhan pelanggan, serta perkembangan produk.

### 5. CRM

*Customer Relationship Management* merupakan bagian yang memiliki tugas untuk mengakuisisi dan mempertahankan pelanggan, serta meningkatkan loyalitas pelanggan dan menerapkan strategi yang berfokus pada pelanggan. Untuk menjaga hubungan yang benar dengan pelanggan maka pelaku bisnis harus mengumpulkan data dan menganalisis informasi. Dengan menggunakan teknologi data mining, data yang telah dikumpulkan dapat digunakan untuk analisis guna menghasilkan informasi yang bermanfaat.

### 6. *Fraud Detection* (Deteksi Penipuan/Kecurangan)

Metode tradisional yang digunakan untuk melakukan deteksi terhadap kecurangan memakan waktu dan kompleks dalam pelaksanaannya.

Data mining berperan membantu dalam menemukan pola yang berarti serta mengubah data menjadi sebuah informasi, dimana setiap informasi yang valid dan berguna merupakan pengetahuan. Sebuah sistem diakatakan sempurna adalah sistem yang dapat melindungi semua informasi pengguna. Metode yang mendapat pengawasan yaitu pengumpulan catatan sampel, karena catatan ini tergolong curang atau tidak palsu. Kemudian dibangun sebuah model dengan menggunakan data ini dan algoritma dibuat untuk mengidentifikasi apakah rekamana itu salah atau tidak.

### 7. *Intrusion Detection*

Gangguan dapat didefinisikan sebagai setiap tindakan yang membahayakan integritas dan kerahasiaan sumber daya. Langkah yang dilakukan untuk menghindari gangguan tersebut meliputi otentifikasi pengguna, meminimalkan kesalahan pemrograman, dan pelindungan terhadap

informasi. Data mining dapat membantu memperbaiki deteksi intrusi dengan menambah tingkat fokus terhadap deteksi anomali. Hal tersebut membantu analis untuk membedakan aktivitas yang terjadi sehari-hari pada jaringan. Data mining juga dapat berguna untuk membantu mengekstrak data yang lebih relevan dengan masalah yang ada.

## **8. Deteksi Kebohongan**

Penegakan hukum bisa dilakukan dengan menggunakan teknik data mining untuk menyelidiki kejahatan, memantau komunikasi tersangka yang dianggap sebagai teroris. Hal ini termasuk ke dalam teks mining. Proses ini berjalan untuk menemukan pola yang berarti dalam data yang biasanya berupa teks yang tidak terstruktur. Hasil pengumpulan sampel data dari penelitian sebelumnya akan dibandingkan dengan model untuk melakukan deteksi terhadapa kebohongan yang dilakukan dimana model yang dibuat diciptakan sesuai dengan kebutuhan.

## **9. Segmentasi Pelanggan**

Penelitian terhadap suatu pasar dapat memabntu untuk melakukan segmentasi pelanggan, namun dengan penggunaan data mining hal tersebut dapat dilakukan lebih mendalam serta dapat meningkatkan efektivitas pasar. Selain itu data mining juga dapat digunakan untuk menyelaraskan pelanggan menjadi segmen yang berbeda dan dapat melakukan penentuan kebutuhan berdasarkan pelanggan. Dalam buku ini akan dilakukan simulasi dalam implementasi data mining yaitu segmentasi data pelanggan pada sebuah perusahaan yang akan dijelaskan pada bab selanjutnya.

## **10. Perbankan/Keuangan**

Komputerisasi data dalam jumlah besar pada perbankan pelanggan mempengaruhi bertambahnya data yang diperoleh dari setiap transaksi terutama transaksi baru. Hal tersebut menjadikan data mining memiliki

## 18 PENGENALAN DATA MINING

kontribusi yang besar untuk memecahkan masalah bisnis di bidang perbankan dan keuangan dengan menemukan pola, sebab-akibat, serta korelasi dalam informasi bisnis dan harga pasar yang tidak jelas terlihat oleh manajer dikarenakan volume data yang terlalu besar atau dihasilkan terlalu cepat. Para ahli melakukan penyaringan dan pengolahan terhadap data sehingga para manajer dapat memperoleh informasi untuk segmentasi, penargetan, perolahan, penahanan, serta pemeliharaan pelanggan yang lebih baik

### **11. Pengawasan Perusahaan**

Pengawasan perusahaan merupakan pemantauan terhadap perilaku seseorang atau kelompok oleh perusahaan. Data yang diperoleh biasanya sering digunakan untuk tujuan pemasaran atau dijual ke perusahaan lain, namun dibagi secara regular dengan instansi pemerintah terkait. Hal tersebut dapat digunakan oleh para pelaku bisnis untuk menyesuaikan produk mereka yang diharapkan oleh pelanggan. Data tersebut juga dapat digunakan untuk tujuan pemasaran langsung, seperti iklan bertarget pada mesin pencari Google, dimana iklan ditargetkan pada pengguna dengan menganalisis riwayat pencarian mereka.

### **12. Analisis Riset**

Data mining memiliki peran yang sangat membantu dalam pembersihan data, pra-pengolahan data serta integrasi database. Penemuan yang didapat oleh peneliti dari data yang serupa dalam database dapat membawa perubahan dalam penelitian. Identifikasi dan korelasi antar aktivitas apapun dapat diketahui. Kemudian visualisasi data yang dilakukan oleh data mining dapat memberi gambaran yang jelas tentang data.

### **13. Investigasi Kriminal**

Proses yang bertujuan untuk mengidentifikasi karakteristik kejahatan disebut dengan kriminologi.

Analisis kejahatan meliputi penjajakan dan deteksi kejahatan serta hubungannya dengan penjahat. Volume dataset kejahatan yang tinggi juga kompleksitas hubungan antara data semacam ini menjadikan kriminologi sebagai bidang yang tepat untuk menerapkan teknik data mining. Hasil laporan kejahatan berbasis teks dapat diubah menjadi dile pengolah kata dimana informasi yang diperoleh bisa digunakan untuk melakukan proses pencocokan kejahatan.

## 14. Bioinformatika

Pendekatan data mining dalam bioinformatika membantu untuk mengekstrak pengetahuan yang berguna dari kumpulan data yang dikumpulkan dalam biologi serta bidang ilmu kehidupan lainnya yang terkait seperti kedokteran dan ilmu saraf.

## 1.8 Pengelompokan Data Mining

Berdasarkan tugas yang dapat dilakukan, data mining dibagi ke dalam beberapa kelompok, yaitu (Larose, 2005):

### 1. Deskripsi

Peneliti dan analis terkadang secara sederhana ingin mencoba menemukan cara untuk menggambarkan pola serta kecenderungan yang terdapat dalam data. Misal, petugas pengumpulan suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup professional akan sedikit didukung dalam pemilihan kepala daerah. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan terhadap suatu pola atau kecenderungan.

### 2. Estimasi

Estimasi memiliki kesamaan dengan klasifikasi, kecuali pada variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Sebuah model

dibangun dengan menggunakan record lengkap yang menyediakan nilai dari variable target sebagai nilai prediksi. Selanjutnya pada tahap peninjauan berikutnya estimasi nilai dari variable target dibuat berdasarkan nilai dari variable prediksi. Misal, akan dilakukan estimasi tekanan darah sistolik pada seorang pasien di sebuah rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variable prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya. Contoh lain yaitu estimasi nilai indeks prestasi kumulatif mahasiswa program pascasarjana berdasarkan nilai indeks prestasi mahasiswa tersebut pada saat mengikuti program sarjana.

### 3. Prediksi

Prediksi memiliki karakteristik yang hamper sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang.

Contoh prediksi dalam dunia bisnis dan penelitian adalah sebagai berikut.

- Prediksi harga beras dalam tiga bulan yang akan datang.
- Prediksi persentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikan.

Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi bisa juga digunakan (dalam keadaan yang tepat) untuk prediksi.

### 4. Klasifikasi

Pada klasifikasi, terdapat target variable kategori. Misal, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang dan pendapatan rendah.

Contoh lain klasifikasi dalam dunia bisnis maupun penelitian adalah sebagai berikut.

- Menentukan apakah suatu transaksi pada sebuah kartu kredit merupakan transaksi yang curang atau bukan.
- Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk
- Mendiagnosis penyakit seorang pasien untuk mendapatkan informasi penyakit tersebut termasuk dalam kategori apa.

## 5. Pengklusteran

Pengklusteran bisa juga disebut dengan pengelompokan *record*, pengamatan, atau memperhatikan dan memebentuk kelas objek-objek yang memiliki kemiripan. Kluster merupakan kumpulan dari berbagai *record* yang memiliki kemiripan satu dengan yang lain serta memiliki ketidakmiripan dengan *record* dalam kluster lain.

Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. Dalam pengklusteran tidak dilakukan proses klasifikasi, estimasi, atau memprediksi nilai dari variable target. Tetapi dalam algoritma pengklusteran terdapat pembagian data ke dalam masing-masing kelompok yang berdasarkan kemiripan (homogen).

Contoh pengklusteran dalam dunia bisnis maupun penelitian adalah:

- Mendapatkan kelompok-kelompok pelanggan untuk target pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
- Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku finansial dalam baik dan mencurigakan.
- Melakukan pengklusteran terhadap ekspresi dari sebuah gen yang berfungsi untuk mendapatkan kemiripan perilaku dari gen dalam jumlah besar.

## 6. Asosiasi

Dalam data mining, asosiasi memiliki tugas menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut dengan *Market Basket Analysis* atau analisis keranjang belanja.

Contoh asosiasi dalam dunia bisnis maupun penelitian adalah:

- Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respons positif terhadap penawaran peningkatan layanan yang diberikan.
- Mencari barang yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan pada supermarket.

## 1.9 Cara Kerja Data Mining

Cara kerja data mining yaitu menggali hal-hal penting yang belum diketahui sebelumnya atau memprediksi apa yang akan terjadi. Teknik yang digunakan untuk melaksanakan tugas ini disebut pemodelan. Pemodelan adalah sebuah kegiatan untuk membangun sebuah model pada situasi yang telah diketahui “jawabannya” dan kemudian menerapkannya pada situasi lain yang akan dicari jawabannya.

Data Mining untuk menentukan pola-pola dalam data. Pola tersebut harus memiliki arti dan serta dapat memberikan keuntungan pada berbagai pihak yang berkepentingan. Karakteristik data mining sebagai berikut:

- a. Data mining berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu yang tidak diketahui sebelumnya.
- b. Penggunaan data dalam jumlah besar untuk data mining bertujuan untuk membuat hasil lebih mudah dipercaya.
- c. Data mining berguna untuk membuat keputusan yang kritis, terutama strategi (Davies,2004), juga dapat digunakan untuk pengambilan keputusan dimasa depan berdasarkan informasi yang diperoleh dari data

masa lalu. Tergantung pada aplikasinya, data bisa berupa data mahasiswa, pasien, dll. Banyak Kasus dalam sehari-hari yang dapat diselesaikan dengan data mining diantaranya adalah sebagai berikut.

- 1) Memprediksi berapa jumlah mahasiswa baru di perguruan tinggi berdasarkan data pendaftar pada tahun-tahun sebelumnya.
- 2) Memprediksi nilai IPK berdasarkan nilai IP setiap semester sebelumnya.
- 3) Produk apa yang akan dibeli pelanggan secara bersamaan jika membeli produk di swalayan.

Tentu masih banyak lagi contoh-contoh dalam bidang lain atau kasus lain yang kaitannya dengan panggalian data sehingga bisa menghasilkan pengetahuan baru dan informasi baru yang dapat menjadi strategi dalam mengembangkan suatu bidang usaha.

## 1.10 Algoritma Data Mining

Proses pemecahan masalah yang dilakukan saat pengolahan data dalam data mining yang bertujuan untuk menemukan sebuah pola yang tersembunyi dalam data tidak lepas dari sebuah algoritma. Penggunaan algoritma itu sendiri disesuaikan dengan melihat informasi apa yang ingin didapat serta data yang akan diolah menggunakan data mining. Berikut ini merupakan algoritma yang popular dan sering digunakan dalam data mining.

### 1. Klasifikasi

Pada data mining proses klasifikasi bertujuan untuk mengelompokkan data menjadi beberapa kelompok.

Proses pengelompokan data mengacu pada data yang telah diketahui terlebih dahulu kelompok atau kelasnya. Data yang belum memiliki kelompok ditentukan kelompoknya melalui proses pembandingan dengan data yang sudah diketahui kelompoknya. Berikut merupakan algoritma klasifikasi populer dan sering digunakan.

## 24 PENGENALAN DATA MINING

- a. *Decision tree*
- b. *Naïve bayes*
- c. *K-nearest neighbor (KNN)*
- d. Jaringan syaraf tiruan
- e. Algoritma genetika
- f. *Support vector machine (SVM)*

### 2. *Clustering.*

*Clustering* memiliki tujuan yang sama dengan klasifikasi yaitu mengelompokan data. Namun, proses pengelompokan data pada clustering tidak menggunakan data lain yang sudah diketahui kelompoknya sebagai pembanding. Pengelompokan clustering berlangsung otonom dengan cara membandingkan semua data yang belum memiliki kelas dan membaginya kedalam beberapa kelas berdasarkan kemiripan antara data. Beberapa algoritma clustering yang banyak digunakan adalah sebagai berikut.

- a. *K-means.*
- b. *Density-based spatial clustering of applications with noise (DBSCAN).*
- c. *Expectation-Maximization (EM).*
- d. *Fuzzy C-Means.*
- e. *Hierarchical clustering.*
- f. *Gaussian mixtures.*

### 3. Regresi

Regresi berbeda dengan klasifikasi dan clustering yang bertujuan dalam mengelompokan data. Regresi bertujuan untuk melakukan prediksi atau peramalan. Konsep dasar regresi pada data mining diturunkan dari teori statistika.

Pada dasarnya, regresi berusaha mengidentifikasi relasi antar beberapa variable terikat dengan variable bebas. Selanjutnya model matematika yang telah dihasilkan dapat digunakan untuk memperkirakan nilai dari suatu variable terikat berdasarkan nilai variable bebasnya. Berikut ini beberapa algoritma regresi yang banyak digunakan adalah sebagai berikut.

- a. Regresi linear sederhana.
- b. Regresi linear berganda.

#### **4. Association Rule.**

*Association rule* merupakan metode pencarian pola relasi antar data dalam sebuah kumpulan data. Berdasarkan pola tersebut, kemunculan suatu data dapat diprediksi berdasarkan kemunculan data lainnya. Algoritma *association rule* yang popular diantaranya adalah sebagai berikut.

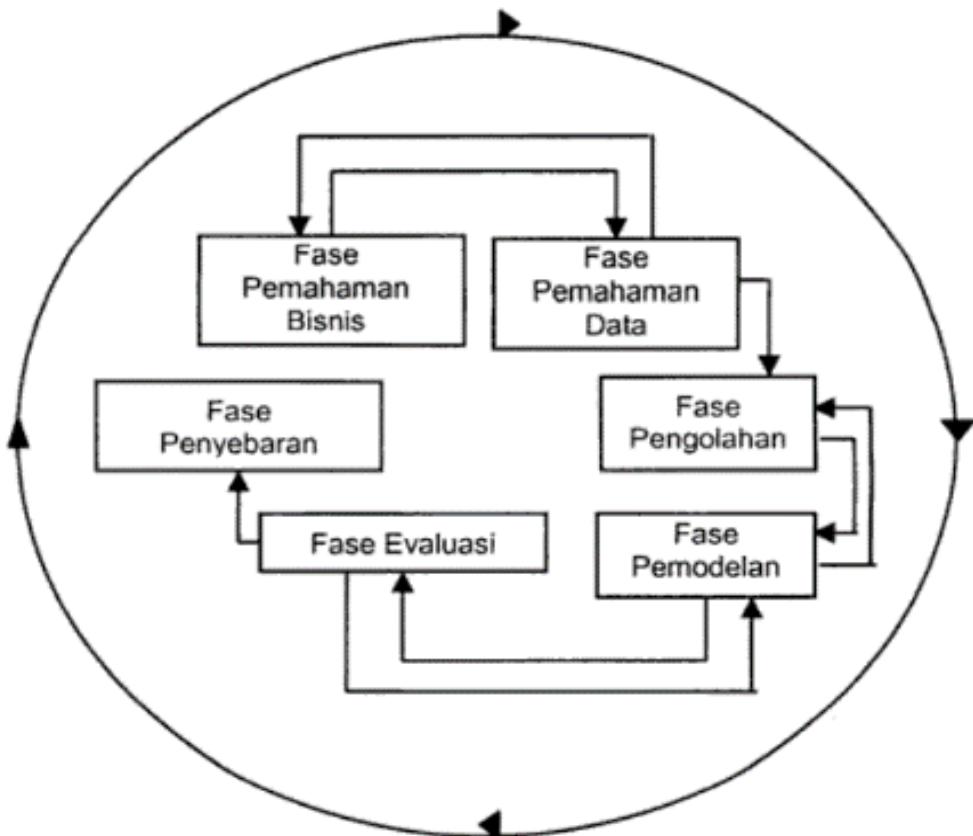
- a. Apriori.
- b. Eclat.
- c. *Frequent-pattern growth (FP-growth)*.

### **1.11 CRISP-DM**

Pada tahun 1996 analis dari beberapa industri seperti DaimlerChrysler, SPSS dan NCR mengembangkan *Cross-Industry Standard Process for Data Mining* (CRISP-DM). CRISP DM merupakan standar proses untuk data mining yang berfungsi sebagai strategi pemecahan masalah secara umum dari baik dalam bisnis ataupun dalam penelitian. Sebuah proyek data mining yang menggunakan standar CRISP-DM memiliki siklus hidup yang terbagi dalam enam fase. Keseluruhan fase berurutan yang ada tersebut bersifat adaptif. Fase berikutnya dalam urutan bergantung kepada keluaran dari fase sebelumnya. Hubungan penting antarfase yang digambarkan oleh panah. Misalkan, jika proses berada fase modeling. Dengan melihat perilaku serta karakteristik dari model, memungkinkan proses harus kembali kepada fase data preparation

## 26 PENGENALAN DATA MINING

untuk perbaikan lebih lanjut terhadap data atau berpindah pada fase evaluation.



Gambar 1.4 Proses Data Mining menurut CRISP-DM

Terdapat enam fase CRISP-DM (Larose, 2005) :

1. Fase Pemahaman Bisnis (*Business Understanding Phase*)
  - a. Penentuan tujuan proyek dan kebutuhan secara setail dalam lingkup bisnis atau unit penelitian secara keseluruhan.
  - b. Menerjemahkan tujuan dan batasan bisnis menjadi sebuah formula dari permasalahan ke dalam proses yang dapat dilakukan oleh data mining.
  - c. Menyiapkan strategi awal untuk mencapai tujuan

2. Fase Pemahaman Data (*Data Understanding Phase*)
  - a. Mengumpulkan data
  - b. Menggunakan analisis penyelidikan data untuk menegentralisir lebih lanjut data dan pencarian pengetahuan awal
  - c. Mengevaluasi kualitas data
  - d. Jika diharapkan, pilihlah sebagian kecil kelompok data yang memungkinkan terdapat pola dari permasalahan.
3. Fase Pengolahan Data (*Data Preparation Phase*)
  - a. Siapkan diri data awal, kumpulan data yang akan digunakan untuk keseluruhan fase berikutnya. Fase ini merupakan pekerjaan perlu dilaksanakan secara intensif
  - b. Pilih kasus dan variabel yang akan dianalisis dan sesuai analisis yang akan dilakukan.
  - c. Lakukan perubahan pada beberapa variabel jika dibutuhkan.
  - d. Siapkan data awal sehingga siap untuk ke perangkat pemodelan.
4. Fase Pemodelan (*Modeling Phase*)
  - a. Pilih dan aplikasikan teknik pemodelan yang sesuai.
  - b. Kalibrasi aturan model untuk mengoptimalkan hasil.
  - c. Perhatikan bahwa beberapa teknik mungkin dapat digunakan digunakan untuk menyelesaikan permasalahan data mining yang sama.
  - d. Proses dapat kembali ke fase pengolahan data untuk memperbaiki data ke dalam bentuk yang sesuai dengan spesifikasi kebutuhan teknik data mining tertentu jika diperlukan.
5. Fase Evaluasi (*Evaluation Phase*)
  - a. Mengevaluasi satu atau lebih model yang digunakan dalam fase pemodelan untuk mendapatkan kualitas dan efektivitas sebelum disebarluaskan untuk digunakan.
  - b. Menetapkan apakah terdapat model yang memenuhi tujuan pada fase awal

- c. Menentukan apakah terdapat permasalahan penting dari bisnis atau penelitian yang tidak tertangani dengan baik.
  - d. Pengambilan keputusan berdasarkan hasil yang dieporeleh dari proses pengolahan data menggunakan data mining.
6. Fase Penyebaran (*Deployment Phase*)
- a. Menggunakan model yang dihasilkan. Terbentuknya model tidak menandakan telah terselesaikannya proyek.
  - b. Pembuatan laporan serta implementasi dari proses data mining secara paralel pada departemen lain.

## 1.12 Tools Data Mining

Tools data mining digunakan untuk menunjang proses kerja dari data mining. Tools tersebut dibuat untuk mendefinisikan dan mencapai berbagai tujuan serta untuk membantu mendapatkan informasi yang lebih terperinci. Berikut ini beberapa tools atau software yang digunakan dalam data mining.

### 1. Rapidminer



Gambar 1.5 Logo Rapidminer

Rapidminer merupakan software yang bersifat *open source*. Rapidminer merupakan salahsatu solusi untuk melakukan analisis terhadap data mining, text mining dan analisis prediksi. Rapidminer menggunakan berbagai teknik deskriptif serta prediksi dalam memberikan pengetahuan kepada pengguna sehingga pengguna dapat mengambil keputusan yang paling baik.

Rapidminer merupakan software yang berdiri sendiri serta memiliki fungsi untuk analisis data dan dapat dijadikan sebagai mesin data mining yang dapat diintegrasikan pada produknya sendiri. Rapidminer ditulis dengan menggunakan bahasa pemrograman java sehingga dapat bekerja pada semua sistem operasi.

Rapidminer memiliki beberapa sifat sebagai berikut.

1. Ditulis dengan bahasa pemrograman java sehingga dapat dijalankan di berbagai sistem operasi.
2. Proses penemuan pengetahuan dimodelkan sebagai operator trees.
3. Representasi XML internal untuk memastikan format standar pertukaran data.
4. Bahasa scripting memungkinkan untuk eksperimen skala besar dan otomatisasi eksperimen.
5. Konsep multi-layer untuk menjamin tampilan data yang efisien dan menjamin penangan data.
6. Memiliki GUI, command line mode, dan java API yang dapat dipanggil dari program lain.

Selain itu Rapidminer juga memiliki beberapa fitur diantaranya adalah sebagai berikut.

1. Banyaknya algoritma data mining, seperti decision tree dan self-organization map.
2. Bentuk grafis seperti diagram histogram, tree chart dan 3D Scatter plots.

## 30 PENGENALAN DATA MINING

3. Banyaknya variasi plugin, seperti text plugin untuk melakukan analisis teks.
4. Menyediakan berbagai prosedur yang digunakan dalam data mining dan *machine learning* termasuk didalamnya ETL (*extraction, transformation, loading*), *data preprocessing*, visualisasi, *modelling* serta evaluasi.
5. Proses data mining tersusun atas operator-operator yang nestable, dideskripsikan dengan XML, dan dibuat dengan GUI.
6. Mengintegrasikan proyek data mining Weka dan statistika R.

## 2. Weka



Gambar 1.6 Logo Weka

Weka merupakan software terintegrasi yang berisi implementasi dari metode-metode data mining. Weka dikembangkan oleh Universitas Wakaito, Selandia Baru menggunakan bahasa pemrograman java. Oleh karena itu, Weka merupakan singkatan dari *Waikato Environment for Knowledge Analysis*. Dengan mengadopsi konsep *open source software*, menjadikan Weka dapat digunakan dan dimodifikasi siapapun secara gratis.

Weka memiliki keunggulan jika dibandingkan dengan perangkat lunak data mining lainnya. Penggunaan Weka murah karena aplikasi tersebut berlisensi *GNU General Public License (GPL)*, yang artinya dapat digunakan secara gratis. Penggunaan bahasa pemrograman java dalam pengembangan Weka menyebabkan Weka dapat diinstal pada hampir semua sistem operasi,

sepanjang sistem operasi tersebut mendukung *Java Virtual Machine*. Berbagai macam algoritma data mining, mulai dari pemrosesan awal sampai dengan pemodelan data, telah disertakan dalam Weka sehingga memudahkan pengguna dalam menganalisis data. Apabila algoritma yang akan digunakan tidak tersedia pada Weka, pengguna dapat menambahkan algoritma tersebut melalui bahasa pemrograman java. Penggunaan Weka pun tergolong mudah karena telah dibekali dengan antarmuka grafis atau *Graphical User Interface* (GUI) sehingga pengguna dapat menggunakannya tanpa perlu meulis datu baris kode pun.

### 3. R



Gambar 1.7 Logo R

R adalah nama bahasa pemrograman komputer yang ditujukan secara khusus untuk menangani komputasi statistic dan memudahkan penyajian grafik. Bahasa ini diciptakan oleh Ross Ihaka dan Robert Gentleman di

Universitas Auckland, Selandia Baru. Versi pertama dirilis pada tahun 1995. Adapun nama R disematkan berdasarkan nama depan kedua penciptanya.

Bahasa R tergolong sebagai bahasa skrip, yakni bahasa yang memungkinkan perintah-perintah yang digunakan ditulis dalam skrip, berekstensi .r, yang disimpan dalam bentuk berkas teks. Selain itu dengan menggunakan konsol, dimungkinkan untuk memberikan perintah secara interaktif. Dalam hal ini, begitu perintah diberikan dan tombol enter ditekan, perintah tersebut akan dieksekusi dan hasilnya akan segera terlihat. R menggunakan lisensi *GNU General Public License* (GPL) sehingga dapat digunakan atau bebas dipakai oleh siapa saja selain itu bahasa R juga dapat berjalan hampir di semua sistem operasi baik itu Windows, Linux, maupun MacOS X.

Bahasa R sangat cocok digunakan untuk menangani operasi yang melibatkan vektor dan matriks. Dengan bahasa R, operasi vektor dan matriks dapat dikerjakan dengan perintah yang sangat singkat. Dengan demikian, R dapat digunakan sebagai alternative terhadap bahasa MATLAB maupun Octave. Aplikasi utama R adalah untuk menangani komputasi statistic dan memudahkan dalam penyajian grafik. Namun, dengan paket-paket tambahan yang juga bersifat gratis, R dapat digunakan untuk menangani pengolahan citra (*image processing*), pembelajaran mesin (*machine learning*), amupun data besar (*big data*).

Bahasa Pemrograman R memiliki kelebihan diantaranya adalah sebagai berikut.

1. Dapat diinstal pada semua sistem operasi baik itu Windows, Linux ataupun MacOS (multiplatform).
2. Memiliki dukungan yang banyak, baik itu dari komunitas yang secara aktif melakukan interaksi satu sama lain melalui internet serta dengan disertakan juga buku petunjuk manual yang sengaja disematkan oleh pengembang yaitu R help yang menyatu dalam *software*.

### 34 PENGENALAN DATA MINING

3. Memiliki tingkat efektif yang cukup tinggi dalam hal pengelolaan data serta fasilitas penyimpanan. File hasil output dari R cenderung memiliki ukuran yang lebih kecil dibanding dengan bahasa pemrograman yang lain.
4. Memiliki fitur yang lengkap untuk melakukan perhitungan dalam bentuk array.
5. Meyediakan berbagai tools yang digunakan untuk melakukan operasi yang berkaitan dengan statistic terintegrasi untuk analisis data.
6. Memiliki visualisasi grafik yang menarik dan fleksibel.
7. Bersifat *open source* sehingga siapun bebas mengembangkan R sesuai dengan kebutuhan.

## **BAB 2**

---

# **SEGMENTASI DAN *PROFILING* PELANGGAN**

---

### **2.1 Segmentasi Pelanggan**

Segmentasi terus menjadi konsep pemasaran yang penting juga dalam konteks *relationship marketing*. Meningkatkan hubungan dengan pelanggan menjadi lebih menarik dan akan menghasilkan pemahaman yang lebih baik tentang kebutuhan pelanggan. Segmentasi adalah proses membagi pelanggan menjadi beberapa *cluster* dengan kategori loyalitas pelanggan untuk membangun strategi pemasaran. Segmentasi pelanggan adalah salah satu langkah awal dalam membuat model bisnis.

## 36 SEGMENTASI DAN PROFILING PELANGGAN

Menurut Don Pepper dan Martha Roger dalam bukunya yang berjudul “*Managing Customer Relationship: A Strategic Framework*”. Melakukan kategori pelanggan sebagai berikut:

1. *Most Valuable Customer* (MVC), yaitu pelanggan dengan nilai paling tinggi bagi perusahaan. Merupakan pelanggan yang memberikan keuntungan terbesar bagi perusahaan.
2. *Most Groable Customer*, yaitu pelanggan yang tanpa disadari memiliki potensi besar.
3. *Below Zeros*, yaitu pelanggan yang memberikan keuntungan lebih sedikit daripada biaya untuk memberikan pelayanan.
4. *Migrators*, merupakan pelanggan yang ada pada posisi diantara *below zeros* dan *most growable customer* serta perlu dilakukan analisis agar dapat diketahui kategori asalnya.

Segmentasi pelanggan dapat didefinisikan sebagai pembagian basis pelanggan menjadi kelompok yang berbeda dan konsisten secara internal dengan karakteristik serupa dimana memungkinkan perusahaan untuk mengembangkan strategi pemasaran yang berbeda sesuai dengan karakteristik pelanggan. Pemahaman seperti itu akan membantu perusahaan dalam mempertahankan pelanggan dan menciptakan nilai tambah bagi pelanggan itu melalui pengembangan hubungan pelanggan. Akibat dari manajemen hubungan pelanggan (*Customer Relationship Management - CRM*).

Segmentasi pelanggan juga mempresentasikan elemen kunci dalam identifikasi pelanggan dalam *customer relationship management* (Ngai dkk, 2009). *Customer relationship management* berguna untuk meningkatkan hubungan dengan pelanggan, memfokuskan dalam hal mengintegrasikan nilai, harapan, dan perilaku pelanggan dengan melakukan analisa data dari transaksi pelanggan (Peppard, 2000).

Untuk mencapai tujuan *customer relationship management*, maka biasanya perusahaan memanfaatkan teknologi informasi untuk membantu perusahaan dalam mengatur hubungan pelanggan dengan suatu cara yang sistematis untuk meningkatkan loyalitas pelanggan dan meningkatkan keuntungan bisnis secara menyeluruh (Kalakota & Robinson, 1999).

Karakteristik pelanggan dapat direpresentasikan oleh beberapa kategori variabel yang terkait dengan pengelompokan, seperti berikut ini:

- *Demographics*: Umur, jenis kelamin, besarnya keluarga, besarnya kediaman, siklus kehidupan keluarga, pemasukan, pekerjaan atau profesi, pendidikan, kepemilikan rumah, status sosial ekonomi, agama, kewarganegaraan.
- *Psychographics*: kepribadian, gaya hidup, nilai-nilai, sikap.
- *Behaviour*: manfaat yang dicari, status pembelian, tingkat penggunaan produk, frekuensi pembelian.
- *Geographic*: negara, provinsi, kota, kode pos, iklim.

Skema segmentasi yang berbeda dapat dikembangkan menurut tujuan bisnis yang spesifik dari organisasi. Segmentasi umumnya digunakan melalui riset pasar untuk mendapatkan wawasan tentang sikap pelanggan, keinginan, pandangan, preferensi, dan opini tentang perusahaan dan kompetisi. Segmentasi pelanggan berdasarkan pada riset pasar dan demografi seringkali membutuhkan pemahaman karakteristik semua pelanggan agar lebih efektif mengetahui segmen apa yang menjadi menarik pelanggan. Penggalian data dapat mengembangkan segmentasi pelanggan yang juga mengidentifikasi segmentasi pada perilaku pelanggan. Selain data penelitian eksternal atau pasar, data transaksi dan pembayaran pelanggan juga dapat digunakan untuk mendapatkan wawasan tentang perilaku pelanggan. Segmentasi dengan cara tersebut, dapat mengalokasikan pelanggan untuk membentuk kelompok berdasarkan jumlah pengeluaran mereka. Hal ini dapat

38 SEGMENTASI DAN *PROFILING* PELANGGAN

digunakan untuk mengidentifikasi pelanggan yang bernilai tinggi dan memprioritaskan pelayanan. Karakter dari pelanggan dijelaskan pada Tabel 2.1.

Tabel 2.1 Karakteristik Pelanggan

Kelas Pelanggan	Karakteristik
<i>Superstar</i>	<ul style="list-style-type: none"> <li>a. Pelanggan dengan loyalitas tinggi.</li> <li>b. Mempunyai nilai monetary yang tinggi.</li> <li>c. Mempunyai frekuensi yang tinggi.</li> <li>d. Mempunyai transaksi paling tinggi.</li> </ul>
<i>Golden Customer</i>	<ul style="list-style-type: none"> <li>a. Mempunyai nilai monetary teringgi yang kedua.</li> <li>b. Frequency yang tinggi.</li> <li>c. Mempunyai rata-rata transaksi.</li> </ul>
<i>Typical Customer</i>	Mempunyai rata-rata nilai monetary dan rata-rata transaksi.
<i>Occasional customer</i>	<ul style="list-style-type: none"> <li>a. Nilai monetary terendah kedua setelah dormant customer</li> <li>b. Nilai recency paling rendah</li> <li>c. Transaksi paling tinggi</li> </ul>

Lanjutan Tabel 2.1 Karakteristik Pelanggan

<i>Everyday shopper</i>	a. Memiliki peningkatan transaksi b. Transaksi yang rendah c. Mempunyai nilai monetary sedang sampai dengan rendah.
<i>Dormant customer</i>	a. Mempunyai frequency dan monetary yang paling rendah b. Nilai recency yang paling rendah

## 2.2 Profiling Pelanggan

*Customer Profiling* merupakan langkah yang dilakukan untuk memetakan dan mendalami profil pelanggan atau pelanggan dengan lebih baik. Pemetaan profil pelanggan dapat dilakukan dengan kombinasi data eksplisit (informasi mengenai pelanggan yang didapatkan dari proses pendaftaran dan survei) dan data implisit (informasi perilaku pelanggan yang didapatkan dengan pengamatan langsung).

Istilah pelanggan diartikan sebagai dua jenis pelanggan, yaitu: pelanggan individu dan pelanggan organisasi. Pelanggan individu membeli barang dan jasa untuk digunakan sendiri, maupun oleh anggota keluarga yang lain. Pelanggan individu sering disebut pelanggan akhir karena langsung digunakan oleh individunya. Sedangkan pelanggan organisasi meliputi organisasi bisnis, yayasan, kantor, dan lembaga lainnya. Jenis pelanggan organisasi membeli produk dan jasa untuk menjalankan kegiatan organisasinya (tidak dikonsumsi sendiri). Perilaku pelanggan merupakan sebuah tindakan memiliki keterlibatan secara langsung dalam usaha untuk mendapatkan, mengonsumsi, dan menghabiskan produk atau jasa, termasuk dalam proses keputusan yang mendahului dan menyertai tindakan tersebut.

Kebutuhan dan keinginan pelanggan selalu menjadi perhatian utama bagi pemilik usaha, yaitu dengan selalu memperhatikan perilaku pelanggannya. Oleh sebab itu, suatu perusahaan dituntut untuk selalu memperhatikan perilaku pelanggan dan menyesuaikan pengenalan produknya kepada pelanggan dengan mengadakan penyempurnaan dan perbaikan terhadap produknya serta menyesuaikan kembali kebutuhan mereka untuk saat ini maupun kebutuhan masa depan. Berikut ini merupakan definisi perilaku pelanggan menurut beberapa ahli.

1. Perilaku Pelanggan atau *consumer behavior* merupakan sebuah studi tentang unit pembelian atau *Buying units* serta proses pertukaran baik yang melibatkan upaya memperoleh, konsumsi, dan pembuangan barang maupun jasa, pengalaman, serta ide-ide. Proses pertukaran merupakan unsur mendasar dari perilaku pelanggan. Pertukaran terjadi antara pelanggan dengan perusahaan, disamping itu juga terjadi di antara perusahaan pada situasi pembelian industrial. Akhirnya, pertukaran juga terjadi diantara pelanggan sendiri, seperti pada saat tetangga meminjam secangkir gula atau mesin pemotong rumput (John C. Mowen and Minor, 2002).
2. Perilaku pelanggan dapat didefinisikan sebagai tindakan yang dilakukan secara langsung dalam upaya mendapatkan, mengkonsumsi, dan menghabiskan produk dan jasa, termasuk dalam proses pengambilan keputusan yang mendahului dan menyertai tindakan ini (Engel, et al 2008).
3. Perilaku pelanggan menurut Shiffman adalah perilaku yang ditunjukkan dalam mencari, membeli, menggunakan, menilai dan menentukan produk jasa dan gagasan. Sedangkan menurut Philip perilaku pelanggan merupakan bidang ilmu perilaku pelanggan mempelajari bagaimana individu, kelompok maupun organisasi dapat memilih, memakai serta memanfaatkan barang, jasa, gagasan atau pengalaman dalam rangka memuaskan kebutuhan dan hasrat mereka.

4. Menurut Carl McDaniel perilaku pelanggan menggambarkan bagaimana pelanggan membuat keputusan pembelian dan bagaimana mereka menggunakan serta mengatur pembelian barang atau jasa. Dari beberapa pengertian diatas disimpulkan bahwa setiap pelanggan dalam membeli produk mempunyai perilaku yang berbeda antara satu dengan yang lain.
5. Perilaku pelanggan adalah perilaku dari pelanggan akhir, individu dan rumah tangga, yang membeli barang dan jasa untuk konsumsi pribadi. Faktor-faktor yang mempengaruhi perilaku pelanggan adalah kebudayaan, sosial, pribadi, psikologis (Kotler dan Keller, 2007).

Berdasarkan pendapat para ahli tersebut, maka dapat disimpulkan bahwa perilaku pelanggan adalah tindakan yang dilakukan oleh individu, kelompok, atau organisasi yang secara langsung terlibat atau berhubungan dengan proses pengambilan keputusan yang meliputi tindakan mengevaluasi, mendapatkan, dan mengkonsumsi produk, baik barang maupun jasa yang dapat dipengaruhi lingkungan, termasuk sebelum dan sesudah proses pengambilan keputusan pembelian. Perilaku pelanggan adalah dinamis, berarti bahwa perilaku seorang pelanggan, kelompok pelanggan, ataupun masyarakat luas selalu berubah dan bergerak sepanjang waktu. Perilaku pelanggan melibatkan pertukaran. Itu merupakan hal terakhir yang ditekankan dalam definisi perilaku pelanggan, yaitu pertukaran di antara individu. Hal tersebut membuat definisi dari perilaku pelanggan tetap konsisten dengan definisi pemasaran yang sejauh ini juga menekankan pertukaran. Namun pada kenyataannya, peran pemasaran adalah untuk menciptakan pertukaran dengan pelanggan melalui formulasi dan penerapan strategi pemasaran.

### **2.3 Perlunya Mempelajari Perilaku Pelanggan**

Kajian atau studi tentang perilaku pelanggan yang dilakukan oleh para ahli menyimpulkan bahwa mempelajari perilaku dari pelanggan itu harus

## 42 SEGMENTASI DAN *PROFILING* PELANGGAN

dilakukan dikarenakan akan memiliki dampak yang dapat membantu para pelaku bisnis untuk melakukan hal berikut ini.

1. Merancang bauran pemasaran.
2. Menetapkan segmentasi.
3. Merumuskan positioning dan pembedaan produk.
4. Memformulasikan analisis lingkungan bisnisnya.
5. Mengembangkan riset pemasaran.

Selain itu, analisis perilaku pelanggan juga memiliki peranan penting dalam merancang kebijakan publik. Bagi orang yang memiliki peranan penting pada suatu negara, kajian ini diperlukan untuk merumuskan kebijakannya dalam kerangka perlindungan pelanggan.

Dengan mengetahui perilaku pelanggan mungkin dapat dimanfaatkan untuk kepentingan pengembangan kemampuan seorang pelaku bisnis dalam menjalankan tugasnya.

### 2.4 Jenis-Jenis Pelanggan

Keputusan pelanggan untuk pembelian dan mengonsumsi suatu produk sangat dipengaruhi oleh berbagai faktor. Sebagai seorang individu, konsumsi suatu produk akan dipengaruhi oleh persepsi, proses pembelajaran dan memori, motivasi dan nilai, konsep diri, sikap, kepribadian dan gaya hidup. Sebagai pengambil keputusan, hal ini akan tergantung dari tipe keputusan (rutin atau jarang), situasi pembelian yang dihadapi, kelompok atau orang yang mempengaruhi dan menjadi acuan. Selanjutnya, kebudayaan dan subbudaya juga memiliki pengaruh kepada perilaku pelanggan.

Kata pelanggan (*consumer*) lebih umum menjelaskan setiap orang yang terlibat dengan suatu kegiatan, seperti yang tercantum pada definisi perilaku pelanggan di atas, yaitu mengevaluasi, memperoleh, menggunakan, dan membuang barang atau jasa.

Dengan demikian, pelanggan terkait dengan hubungannya dengan perusahaan tertentu, sedangkan pelanggan tidak. Pelanggan umum adalah seseorang yang mempunyai kebutuhan ataupun dorongan untuk melakukan pembelian produk tertentu, selanjutnya membuang produk tersebut dalam 3 tahap proses konsumsi (Solomon, 2002). Pelanggan memiliki beberapa peran dalam ketiga proses tersebut, yaitu berikut ini.

1. Pencetus ide (*Initiator*).
2. Pembeli (*Purchaser/Buyer*).
3. Membayar (*Payer*).
4. Pengguna/pemakai (*User*).
5. Pemberi pengaruh (*Influencer*).
6. Pengambil keputusan (*Decision Maker*).
7. Pelanggan organisasi atau kelompok, di mana satu orang atau sekelompok orang akan membuat keputusan untuk organisasi.

## 2.5 Macam-Macam Model Perilaku Pelanggan

Model perilaku pelanggan yang dikemukakan oleh Henry Assael (1998) terdapat beberapa faktor yang mempengaruhi perilaku pelanggan. Dengan model perilaku pelanggan yang sederhana Henry Assael menunjukkan bahwa interaksi antara pemasar dengan pelanggan perlu dilakukan karena dapat menimbulkan adanya proses untuk merasakan dan mengevaluasi informasi merek produk, mempertimbangkan berbagai alternatif merek dapat memenuhi kebutuhan pelanggan dan pada akhirnya memutuskan merek apa yang akan dibeli pelanggan. Model perilaku pelanggan merupakan sebuah gambaran ataupun kerangka yang mencerminkan atau menjelaskan setiap tahapan yang akan dilakukan oleh pelanggan dalam memutuskan untuk melakukan keputusan pembelian.

## 2.6 Faktor Yang Mempengaruhi Perilaku Pelanggan

Terdapat banyak faktor yang mampu mempengaruhi perilaku pelanggan dalam melakukan sebuah pembelian. Faktor-faktor tersebut berawal dari dalam diri pelanggan serta dari luar pelanggan. Keputusan pembelian yang dilakukan oleh seorang pembeli sangat dipengaruhi oleh berbagai faktor baik itu kebudayaan, sosial, pribadi dan psikologi dari pembeli. Sebagian besar adalah faktor-faktor yang tidak dapat dikendalikan oleh pemasar, tetapi harus benar-benar diperhitungkan. Faktor-Faktor yang mempengaruhi Perilaku Pelanggan

### 1. Faktor Budaya

Menurut Sumarwan (2004) budaya adalah segala nilai, pemikiran, simbol yang mempengaruhi perilaku, sikap, kepercayaan dan kebiasaan seseorang dan masyarakat. Adapun unsur-unsur budaya antara lain budaya, subbudaya dan kelas sosial.

### 2. Subbudaya

Setiap kebudayaan dapat terbagi menjadi sebuah subbudaya yang lebih kecil yang bermanfaat untuk memberikan identifikasi dan sosialisasi yang lebih spesifik untuk para anggotanya. Subbudaya tersebut dapat dibedakan menjadi empat bagian yakni kelompok nasionalisme, kelompok keagamaan, kelompok ras, serta area geografis.

### 3. Kelas sosial

Kelas sosial merupakan kelompok yang bersifat homogen dan dapat bertahan lama dalam suatu kelompok masyarakat serta tersusun dalam sebuah hirarki yang mana dalam keanggotaannya memiliki sebuah nilai, minat, maupun perilaku yang sama.

#### 4. Faktor Sosial

- Kelompok Referensi

Kelompok referensi seseorang terdiri dari seluruh kelompok yang mempunyai pengaruh langsung maupun tidak langsung terhadap sikap atau perilaku seseorang. Beberapa diantaranya adalah kelompok primer yaitu kelompok yang memiliki interaksi yang cukup berkesinambungan seperti, keluarga, teman, tetangga, dan teman sejawat. Kelompok sekunder merupakan kelompok yang cenderung lebih resmi dan yang mana interaksi yang terjadi kurangv berkesinambungan. Kelompok yang seseorang ingin menjadi anggotanya disebut kelompok aspirasi. Kemudian kelompok diasosiatif merupakan sebuah kelompok yang nilai atau perlakunya tidak disukai oleh individu.

- Keluarga

Pengaruh Keluarga yaitu keluarga memberikan pengaruh yang besar dalam perilaku pembelian. Para pelaku pasar telah memeriksa peran dan pengaruh suami, istri, dan anak dalam pembelian produk yang berbeda. Sebagai contoh, anak-anak dapat memberikan pengaruh yang besar terhadap keputusan yang melibatkan restoran fast food. Faktor sosial terdiri dari kelompok acuan, keluarga, peran dan status (Setiadi, 2003). Dalam kehidupan pembeli keluarga dibedakan menjadi dua yaitu.

- 1) Keluarga orientasi

Merupakan orangtua seseorang. Dari orangtualah seseorang mendapatkan pandangan tentang agama, politik, ekonomi, dan merasakan ambisi pribadi nilai atau harga diri dan cinta.

- 2) Keluarga prokreasi

Merupakan pasangan hidup serta anak-anak dari seseorang, keluarga ini merupakan organisasi pembeli dari seorang pelanggan yang paling penting dalam suatu masyarakat dan telah diteliti secara intensif.

- Peran dan Status

Setiap orang umumnya berpartisipasi dalam kelompok selama hidupnya, baik itu keluarga, organisasi ataupun klub. Posisi seseorang yang berada dalam sebuah kelompok dapat diidentifikasi melalui peran dan status yang dimiliki.

## 5. Faktor Pribadi

Keputusan pembelian juga dipengaruhi oleh karakteristik pribadi, antara lain sebagai berikut.

### 1) Umur dan tahapan dalam siklus hidup

Konsumsi yang dilakukan oleh seseorang juga dipengaruhi oleh tahapan siklus hidup keluarga. Beberapa penelitian juga pernah melakukan identifikasi terhadap tahapan-tahapan dalam siklus hidup psikologis. Orang-orang dewasa biasanya mengalami perubahan atau transformasi tertentu pada saat mereka menjalani hidupnya.

### 2) Pekerjaan

Para pelaku bisnis mengidentifikasi kelompok-kelompok pekerja yang memiliki minat di atas rata-rata terhadap produk dan jasa tertentu.

### 3) Kepribadian dan konsep diri

Dalam hal ini yang dimaksud kepribadian adalah karakteristik psikologis yang berbeda dan setiap orang yang memandang responsnya terhadap lingkungan yang relative konsisten. Kepribadian merupakan suatu unsur yang sangat berguna dalam hal analisis perilaku pelanggan. Bila keragaman dari kepribadian dapat diklasifikasikan dan memiliki korelasi yang kuat antara jenis-jenis kepribadian tersebut dan berbagai pilihan produk atau merek.

4) Situasi ekonomi

Keadaan atau situasi ekonomi yang dimaksud dari sesorang terdiri dari pendapatan yang dapat dibelanjakan, tabungan dan hartanya, kemampuan untuk meminjam dan sikap terhadap mengeluarkan lawan menabung.

5) Gaya hidup.

Merupakan pola hidup seseorang di dunia yang diekspresikan dengan kegiatan, minat dan pendapat seseorang. Gaya hidup menggambarkan seseseorang secara utuh yang berinteraksi dengan lingkungannya. Gaya hidup juga dapat mencerminkan sesuatu di balik kelas sosial seseorang.

## 6. Faktor Psikologis

Keputusan pembelian yang dilakukan seseorang dipengaruhi oleh empat faktor utama psikologis yaitu.

1) Motivasi

Manusia memiliki beberapa kebutuhan diantaranya bersifat biogenic, kebutuhan biogenic yaitu kebutuhan yang timbul dari suatu keadaan bilogis tertentu, seperti rasa lapar, haus, resah tidak nyaman. Adapun kebutuhan lain bersifat psikogenik, yaitu kebutuhan yang timbul dari keadaan psikologis tertentu, seperti kebutuhan untuk diakui oleh orang lain ataupun sebagainya. Untuk mencukupi kebutuhan tersebut diperlukanlah sebuah motivasi dimana hal tersebut dapat mendorong seseorang untuk melakukan sesuatu.

2) Persepsi

Persepsi merupakan sebuah proses memilih, mengorganisasikan, serta mengartikan masukan informasi yang

dilakukan oleh seseorang untuk menciptakan suatu gambaran yang lebih berarti.

3) Pembelajaran

Pembelajaran biasa juga disebut dengan proses belajar dapat menjelaskan perubahan dalam perilaku seseorang yang timbul diakibatkan dari pengalaman.

4) Keyakinan dan sikap.

Keyakinan dan sikap merupakan salahsatu faktor yang penting, karena keyakinan merupakan sebuah gagasan deskriptif yang dimiliki seseorang terhadap sesuatu.

Keputusan pembelian terhadap sebuah produk yang dilakukan oleh seseorang memiliki keterkaitan yang cukup rumit dan dipengaruhi oleh berbagai faktor yang telah disebutkan sebelumnya. Namun faktor-faktor tersebut sangat berguna dalam hal mengidentifikasi pembeli atau pelanggan yang mungkin memiliki minat terhadap suatu produk.

## BAB 3

---

### PENGENALAN ALGORITMA

---

#### 3.1 Apa itu Algoritma?

Dalam pekerjaan sehari-hari, sangat diperlukan untuk memiliki pengetahuan serta pemahaman terlebih dahulu terkait dengan pekerjaan yang akan dilakukan agar tujuan, proses serta hasil dari pekerjaan tersebut dapat terlaksana. Demikian pula dalam pemrograman komputer, seorang programmer harus memiliki pemahaman yang cukup tentang logika serta konsep dari pemrograman, agar dalam pelaksanaannya program yang dibuat dapat memenuhi kebutuhan yang ada. Sebuah program tidak akan terlepas dari kinerja sebuah komputer, dimana dalam hal tersebut terdapat logika dasar dari setiap siklus yang ada dalam komputer seperti *input*, *process*, hingga *output*.

Algoritma merupakan sebuah pemahaman dasar yang harus dimiliki, karena dengan algoritma ini dapat membantu menata tahapan-tahapan dalam penyelesaian masalah menggunakan komputer, dalam hal ini membuat program.

Algoritma berasal dari nama seorang penulis dan juga seorang ahli matematika yang bernama Abu Ja'far Muhammad Ibnu Musa al-Khawarizmi yang mana oleh orang barat kata al-Khawarizmi dibaca *algorism* yang kemudian lambat laun berubah menjadi *algorithm* atau diterjemahkan dalam bahasa Indonesia menjadi algoritma. Salahsatu karya terkenal dari al-Khawarizmi yaitu *al-Kitab al-mukhtasar fi hisab al-jabr wa'l-muqabala* (*rules of restoration and reduction* atau buku rangkuman untuk kalkulasi dengan melengkapkan dan menyeimbangkan).

Algoritma dapat didefinisikan sebagai langkah-langkah atau urutan untuk memecahkan suatu masalah berdasarkan tahapan logis secara sistematis dalam periode tertentu. Dalam literatur yang lain, definisi dari algoritma adalah langkah-langkah perhitungan dasar untuk mengubah suatu masukanan menjadi keluaran.

## 3.2 Kriteria Algoritma

Menurut Donald E.Kurth sebuah algoritma yang baik harus memiliki kriteria sebagai berikut.

### 1. *Input*

Sebuah algoritma harus memiliki sebuah baik itu masukan dari pengguna ataupun data yang diinisialisasikan atau dibangkitkan dalam suatu algoritma.

### 2. *Output*

Dalam suatu algoritma harus memiliki satu atau lebih *output*. Algoritma yang tidak memiliki output merupakan sebuah algortima yang sia-sia untuk dilakukan.

Oleh karena itu, tujuan dibuatnya algoritma adalah untuk mendapatkan sebuah *output*.

### 3. *Finiteness*

Algoritma yang dijalankan harus memiliki sebuah jaminan untuk berhenti setelah melakukan suatu proses atau setelah output yang diinginkan tercapai.

### 4. *Definiteness*

Setiap pernyataan yang terdapat dalam suatu algoritma harus pasti dan tidak memiliki makna ganda sehingga tidak membingungkan pembaca dari algoritma tersebut, sehingga diharapkan mampu untuk memberikan hasil atau output sesuai dengan yang diharapkan.

### 5. *Effectiveness*

Sebuah algoritma sebisa mungkin harus dilaksanakan dalam jangka waktu yang wajar serta efektif, sehingga segala aktivitas yang dilakukan merupakan aktivitas yang memiliki dampak dan tidak ada aktivitas yang sia-sia dalam proses pengerjaannya.

## 3.3 Jenis-Jenis Proses Algoritma

Secara umum, algoritma dibedakan menjadi beberapa jenis yaitu sebagai berikut.

### 1. Algoritma Sekuensial (*Sequential*)

Algoritma sekuensial adalah langkah-langkah pemecahan suatu masalah yang dilakukan sesuai dengan penulisannya. Jika salahsatu langkah atau urutannya dirubah maka kemungkinan akan mempengaruhi output yang dihasilkan.

### 2. Algoritma Percabangan (*Branching*)

Dalam suatu algoritma tertentu, sebuah aksi terkadang akan dilakukan jika terdapat kondisi yang terpenuhi ataupun tidak dilakukan tergantung dari

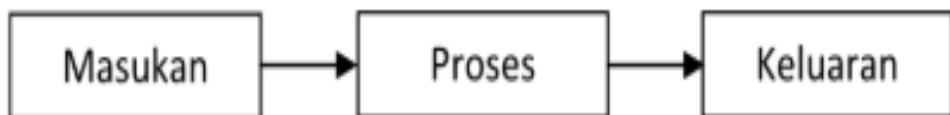
situasi yang terjadi. Pada algoritma percabangan ini, hanya ada satu aksi yang dijalankan dari sejumlah pilihan aksi yang diberikan.

### 3. Algoritma Perulangan (*Looping*)

Dalam algoritma dikenal istilah perulangan. Perulangan yang dimaksud adalah satu atau beberapa aksi yang dijalankan secara berulang sesuai dengan kondisi dan kebutuhan.

## 3.4 Prinsip Kerja Algoritma

Berdasarkan definisi, algoritma merupakan sebuah deskripsi dari pelaksanaan dalam suatu proses, dimana proses yang dikerjakan akan sesuai dengan algoritma yang ditulis atau dibuat. Prinsip kerja dari sebuah algoritma yaitu adanya suatu masukan (*input*) yang kemudian diproses hingga menghasilkan sebuah keluaran (*output*) seperti yang digambarkan pada Gambar 3.1.



Gambar 3.1 Prinsip Kerja Algoritma

## 3.5 *Clustering*

*Clustering* merupakan salahsatu teknik yang digunakan untuk mengelompokan data. *Clustering* adalah sebuah proses yang digunakan untuk mengelompokan data menjadi beberapa kelompok atau *cluster* sehingga memiliki tingkat kemiripan yang tinggi terhadap anggota yang lain dalam satu kelompok serta memiliki kemiripan yang minimum dengan cluster yang lain (Tan, 2006). Menurut Rui Xu dan Donald (2009), *clustering* merupakan proses membagi kumpulan data menjadi beberapa kelompok dimana anggota dari masing-masing kelompok memiliki kesamaan.

Gagasan terkait dengan penglompokan data atau *clustering*, memiliki sifat yang sederhana dan cukup dekat dengan cara berpikir manusia. Selain itu, sebagian besar dari data yang diperoleh dalam jumlah besar terkadang memiliki banyak masalah yang dapat dilihat dari beberapa sifat yang melekat serta mengalami proses pengelompokan-pengelompokan secara natural (Hammuda dan Karay, 2003).

*Clustering* dapat digunakan sebagai langkah awal atau pendahuluan dalam sebuah proses pengumpulan data. Dengan adanya *cluster-cluster* yang dihasilkan tersebut dapat digunakan untuk masukan lebih lanjut dalam suatu teknik yang berbeda, seperti natural yang disebutkan sebelumnya dapat diperoleh sebagai jarak dari pembaharuan formula Lance-Williams (Lance dan Williams, 1967).

Analisis kluster atau *clustering* merupakan proses membagi data pada suatu himpunan ke dalam beberapa kelompok atau *cluster* yang memiliki kesamaan data dalam suatu kelompok yang memiliki kesamaan yang lebih besar daripada kesamaan data tersebut dengan data yang berada pada kelompok yang lain (Jang, Sun, dan Mizutani, 2004). Analisis kluster juga dapat dikatakan sebagai teknik multivariat dengan tujuan utama yaitu untuk mengelompokan objek berdasarkan karakteristik yang dimiliki dari masing-masing objek. Analisis kluster mengklasifikasi objek sehingga setiap objek yang memiliki kesamaan paling dekat dengan objek lainnya yang berada pada cluster yang sama. Solusi yang diperoleh dari hasil analisis kluster bersifat tidak unik, anggota dari masing-masing *cluster* dari masing-masing penyelesaian bergantung pada beberapa elemen prosedur serta beberapa solusi yang berbeda dapat diperoleh dengan mengubah salahsatu elemen atau lebih. Secara keseluruhan, solusi untuk analisis kluster bergantung pada variable-variabel yang dijadikan sebagai dasar untuk menilai kesamaan tersebut. Pengurangan atau penambahan dari masing-masing variable yang relevan dapat mempengaruhi hasil dari *clustering*.

*Clustering* merupakan suatu proses membagi kumpulan objek data ke dalam suatu himpunan yang biasa disebut dengan cluster. Objek yang berada di dalam cluster tersebut memiliki kesamaan antara satu dengan yang lainnya dan memiliki perbedaan dengan anggota kelompok dari *cluster* lain. Proses pembagian data dilakukan dengan menggunakan sebuah algoritma *clustering*. Maka dari itu, hal tersebut sangat berguna serta dapat menemukan kelompok atau *cluster* yang belum dikenal dalam data. *Clustering* banyak diimplementasikan pada berbagai aplikasi seperti *business intelligence*, pengenalan pola citra, *web search*, bidang ilmu biologi, serta untuk keamanan (*security*).

Dalam *business intelligence*, *clustering* dapat digunakan untuk membagi *customer* atau pelanggan ke dalam banyak kelompok sesuai dengan karakteristik yang diharapkan dari masing-masing kelompok. Selain itu, *clustering* dikenal sebagai data segmentasi dikarenakan *clustering* dapat melakukan pembagian data terhadap banyak data set ke dalam banyak kelompok atau *cluster* berdasarkan kemiripan dari masing-masing data. Kemudian *clustering* juga dikenal sebagai *outlier detection*.

### 3.6 Manfaat *Clustering*

Teknik *clustering* memiliki beberapa manfaat dalam penggunaannya. Manfaat tersebut diantaranya adalah sebagai berikut.

1. *Clustering* adalah metode segmentasi data yang sangat berguna dalam melakukan prediksi dan analisis terhadap masalah bisnis tertentu (Berson dan Smith, 2001). Contohnya segmentasi pasar dan pelanggan, *marketing* dan pemetaan pada zonasi wilayah.
2. *Clustering* juga berguna untuk melakukan identifikasi obyek dalam berbagai bidang seperti *computer vision* dan *image processing*.

### 3.7 Konsep Dasar *Clustering*

Hasil yang baik dari sebuah *clustering* akan memiliki tingkat kesamaan yang tinggi dalam suatu kelompok atau *cluster* dan memiliki tingkat kesamaan yang rendah dengan anggota *cluster* lain. Tingkat kesamaan yang dimaksud tersebut merupakan hasil pengukuran secara numerik terhadap dua objek. Hasil kesamaan antar dua objek akan memiliki nilai yang semakin tinggi jika kedua objek tersebut dibandingkan. Begitupun sebaliknya. Kualitas yang dimiliki dari hasil *clustering* bergantung pada metode yang digunakan. Terdapat 4 tipe data yang dikenal dalam *clustering*. Adapun keempat tipe data tersebut adalah sebagai berikut.

1. Variable berskala interval.
2. Variable biner.
3. Variable nominal, ordinal, dan rasio.
4. Variable dengan tipe lainnya.

### 3.8 Syarat *Clustering*

Algoritma *clustering* memiliki syarat sekaligus tantangan yang harus dipenuhi. Syarat tersebut adalah sebagai berikut (Han dan Kamber, 2012).

#### 1. Skalabilitas

Sebuah algoritma clustering harus memiliki kemampuan untuk mengolah data dalam jumlah yang besar. Penggunaan data dalam jumlah besar pada saat ini sudah sangat biasa digunakan di berbagai bidang contohnya saja database. Database ini tidak hanya berisi ratusan objek, melainkan berisi lebih dari jutaan objek.

#### 2. Kemampuan analisa beragam bentuk data

Selain memiliki kemampuan skalabilitas yang tinggi, algoritma *clustering* juga harus mampu untuk dapat diaplikasikan dalam berbagai

bentuk data, seperti pada tipe data yang telah disebutkan sebelumnya yaitu nominal, ordinal maupun gabungannya.

### 3. Mampu menampilkan hasil *cluster* dengan bentuk yang berbeda

Pada umumnya algoritma *clustering* yang menggunakan metode *Euclidian* atau *Manhattan* memiliki hasil berbentuk bulat. Namun sebenarnya, hasil clustering dapat memiliki bentuk yang aneh dan tidak sama antara satu dengan yang lain. Oleh karena itu, dibutuhkan kemampuan untuk menganalisa *cluster* dalam berbagai bentuk pada suatu algortima *clustering*.

### 4. Kemampuan untuk dapat menangani noise

Data yang diperoleh dari suatu pengumpulan data tidak selalu dalam keadaan yang baik. Terkadang terdapat beberapa data dalam kondisi rusak, sulit dimengerti ataupun hilang. Dengan adanya sistem inilah, suatu algoritma *clustering* dituntut untuk dapat menangani data yang rusak tersebut.

### 5. Sensitifitas terhadap perubahan input

Perubahan serta penambahan data yang terjadi pada sebuah masukan dapat mengakibatkan terjadinya perubahan pada cluster yang telah ada bahkan dapat menyebabkan perubahan yang sangat mencolok jika menggunakan algoritma *clustering* yang memiliki tingkat sensitifitas yang rendah.

### 6. Mampu melakukan *clustering* untuk data dimensi tinggi

Dimensi atau atribut yang dimiliki dari sekumpulan data dapat berbeda-beda. Oleh karena itu sangat diperlukan sebuah algoritma *clustering* yang dapat menangani data dengan perbedaan jumlah dimensi yang tidak sedikit.

### 7. Interpretasi dan kegunaan

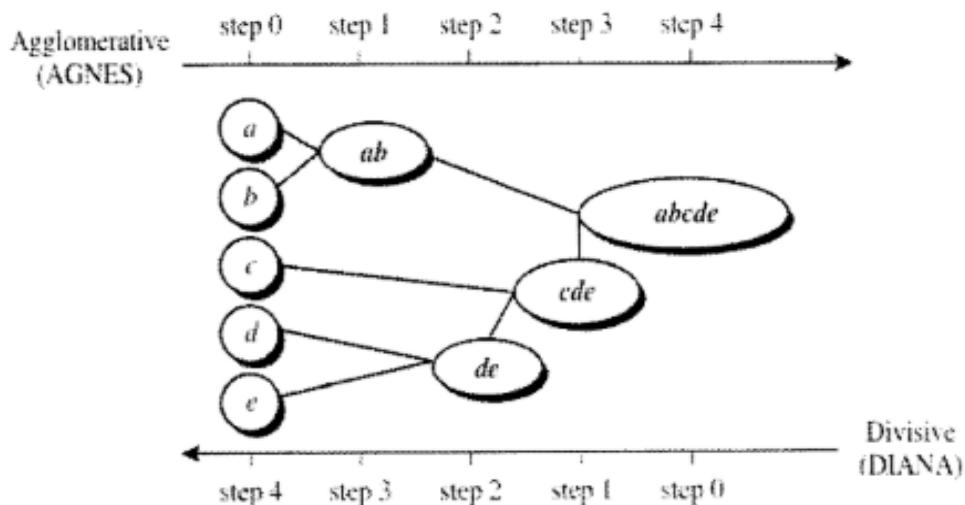
Hasil yang diperoleh dari proses pengolahan data dengan menggunakan algoritma *clustering* harus dapat divisualisasikan serta memiliki informasi yang berguna untuk penelitian.

### 3.9 Metode *Clustering*

Secara umum metode *clustering* dapat dibagi menjadi dua yaitu *hierarchical clustering* dan *partitional clustering* (Tan, 2011). Kemudian sebagai tambahan terdapat juga metode Density-Based dan Grid-Based yang sering juga diterapkan dalam pengimplementasian *clustering*.

#### 3.9.1 *Hierarchical Clustering*

Dalam metode *hierarchical clustering* pengelompokan data dilakukan menggunakan suatu bagan yang berbentuk hirarki. Pada bagan tersebut terdapat penggabungan dua kelompok yang terdekat pada setiap iterasinya ataupun pembagian dari seluruh data set ke dalam suatu *cluster* seperti yang terlihat pada Gambar 3.2.



Gambar 3.2 *Hierarchical Clustering*

Berikut ini tahapan-tahapan dalam melakukan *Hierarchical Clustering*.

1. Identifikasi *item* yang memiliki jarak terdekat.
2. Gabungkan *item* tersebut kedalam suatu *cluster*.
3. Hitung jarak antar *cluster*.
4. Ulangi tahapan tersebut dari awal hingga semua terhubung.

Contoh metode *hierarchical clustering* diantaranya yaitu sebagai berikut.

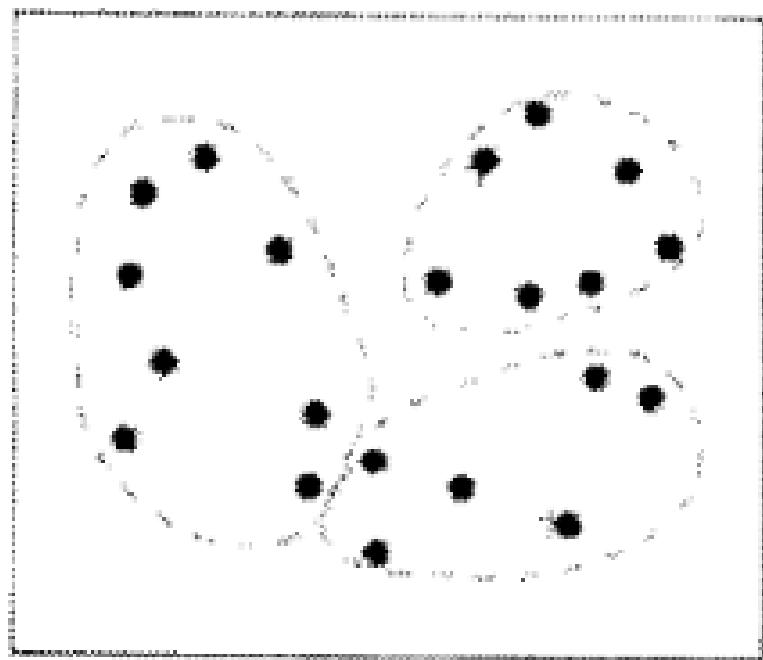
1. *Single Linkage*.
2. *Complete Linkage*.
3. *Average Linkage*.
4. *Average Group Linkage*.

### **3.9.2 Partitional Clustering**

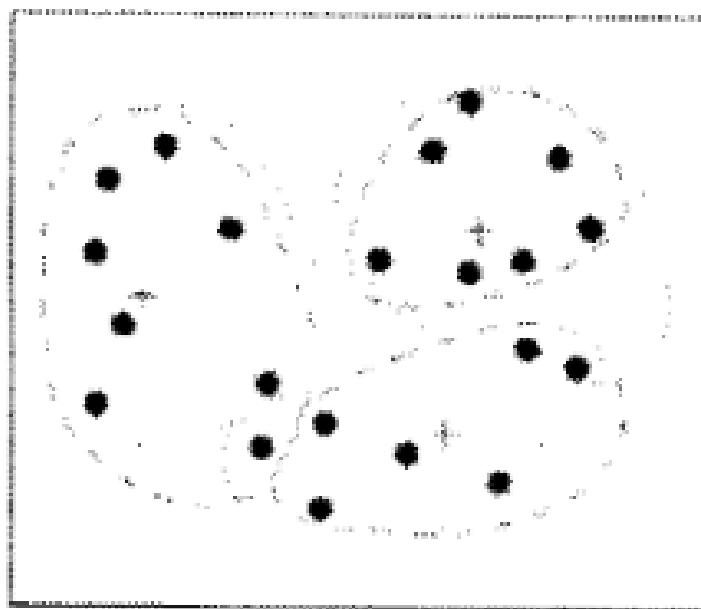
*Partitional clustering* merupakan pengelompokan data kedalam sejumlah *cluster* tanpa adanya struktur hirarki antara satu dengan yang lainnya. Dalam metode ini setiap *cluster* memiliki titik pusat cluster atau biasa disebut dengan *centroid*. Kemudian secara umum metode ini memiliki sebuah fungsi dan tujuan yaitu meminimumkan jarak atau *dissimilarity* dari keseluruhan data terhadap pusat *cluster* masing-masing.

Contoh metode *partitional clustering* diantaranya yaitu sebagai berikut.

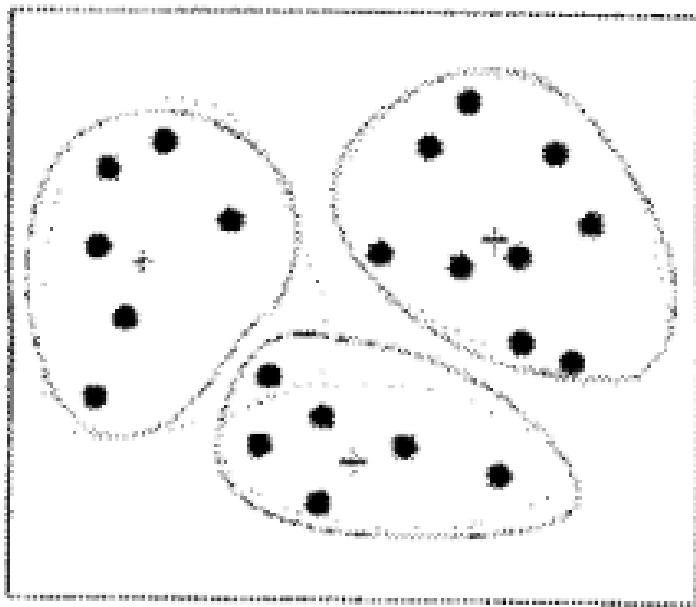
1. *K-Means*.
2. *Fuzzy K-Means*.
3. *Mixture Modelling*.



Gambar 3.3 *Cluster Awal*



Gambar 3.4 Proses Iterasi

Gambar 3.5 *Cluster Akhir*

### 3.10 Algoritma K-Means

Algoritma K-Means merupakan algoritma *clustering* yang paling sederhana dan umum digunakan. Hal tersebut dikarenakan kemampuan yang dimiliki K-Means yaitu dapat mengelompokan data dalam jumlah yang besar dengan waktu komputasi yang dibutuhkan relatif cepat dan efisien. K-means merupakan salahsatu algoritma *clustering* dengan metode partisi atau *partitioning method* yang memisahkan data ke dalam kelompok yang berbeda. Penggunaan *partitioning method* yang dilakukan secara iteratif, menjadikan K-Means memiliki kemampuan untuk meminimalkan rata-rata jarak dari setiap data ke *cluster* nya. Algoritma K-Means pertama kali diusulkan oleh MacQueen pada tahun 1967 kemudian dikembangkan oleh Hartigan dan Wong pada tahun 1975 dengan tujuan algoritma K-Means dapat digunakan dapat membagi M *data point* dalam N dimensi kedalam sejumlah k *cluster*,

dimana proses *clustering* tersebut dilakukan dengan meminimalkan jarak *sum squares* antara data dengan masing-masing pusat *cluster* atau *centroid-based*.

Dalam penerapannya algoritma K-Means memerlukan tiga parameter yang keseluruhannya ditentukan pengguna yaitu jumlah *cluster* k, inisialisasi *cluster* dan jarak sistem. K-Means biasanya dijalankan secara independen dengan inisialisasi yang berbeda sehingga menghasilkan *cluster* akhir yang berbeda karena algoritma ini pada prinsipnya hanya mengelompokan data menuju *local minimal*. Salah satu cara yang dapat dilakukan untuk mengatasi *local minimal* adalah dengan menerapkan algoritma K-Means, untuk jumlah K yang diberikan dengan memberikan beberapa nilai *initial partition* yang berbeda kemudian selanjutnya dipilih partisi dengan nilai kesalahan kuadarat terkecil (Jain, 2009).

Algoritma K-Means merupakan model *centroid* yang artinya bahwa algoritma K-Means menggunakan centroid untuk membuat sebuah cluster. *Centroid* adalah titik tengah suatu *cluster*. *Centroid* dapat berupa nilai. Nilai dari *centroid* tersebut dijadikan sebagai nilai yang digunakan untuk menghitung jarak suatu objek data terhadap *centroid* titik pusat *cluster*. Suatu objek data termasuk dalam *cluster* jika memiliki jarak terpendek terhadap centroid *cluster* tersebut. Selain itu algoritma K-means memiliki aturan dalam proses *clustering* yaitu sebagai berikut.

1. Berapa jumlah *cluster* yang perlu dimasukkan.
2. Hanya memiliki atribut bertipe numerik.

K-Means memiliki kemampuan untuk mengelompokkan data dalam jumlah yang cukup besar dengan waktu komputasi yang dibutuhkan untuk relatif cepat dan efisien. Namun, K-Means mempunyai kelemahan yang diakibatkan oleh penentuan pusat awal *cluster*. Dalam algoritma K-Means, inisiasi nilai pusat awal *cluster* yang diberikan akan mempengaruhi bentuk dari *cluster*.

Teknik yang digunakan dalam algoritma K-Means menjadikan ketidakmiripan sebagai tolak ukur untuk pengelompokan objek. Ketidakmiripan dapat diterjemahkan dalam konsep jarak. Dua objek dikatakan mirip jika jarak dua objek tersebut dekat. Sehingga bila nilai jarak semakin tinggi nilai jarak, maka nilai ketidakmiripannya semakin tinggi. Tahapan awal yang dilakukan pada proses pengelompokan data dengan menggunakan algoritma K-Means adalah pembentukan titik awal centroid  $c_j$ .

Pada umumnya pembentukan titik awal centroid dibangkitkan secara acak. Nilai dari centroid  $c_j$  merupakan nilai dari centroid yang dibangkitkan pertama kali sesuai dengan jumlah *cluster*. Setelah k centroid terbentuk kemudian dilakukan perhitungan terhadap jarak dari setiap data  $x_i$  dengan menggunakan centroid ke- $j$  sampai k, kemudian dinotasikan dengan menjadi  $d(x_i, c_j)$ . Terdapat beberapa ukuran jarak yang digunakan sebagai ukuran kemiripan suatu instance data, salah satunya adalah jarak Euclidean. Perhitungan jarak Euclidean seperti pada Persamaan 3.1.

Duran dan Odell (1974) menyatakan jika semakin kecil, kesamaan antara dua  $d(X_i, C_j)$  unit pengamatan semakin dekat. Jarak Euclidian tersebut dapat digunakan apabila semua fitur yang terdapat dalam dataset tidak saling berkorelasi. Jika terdapat fitur yang berkorelasi maka menggunakan konsep jarak Mahalanobis. Agusta (2007) menyatakan kelanjutan dari jarak tersebut dicari yang terdekat sehingga data akan mengelompok berdasarkan centroid yang paling dekat. Proses selanjutnya adalah lakukan pembaruan pada titik centroid dengan cara menghitung rata-rata jarak keseluruhan data terhadap centroid. Selanjutnya akan kembali lagi ke proses awal. Proses iterasi atau pengulangan akan terus dilakukan sampai didapatkan centroid yang tetap

artinya titik centroid tersebut sudah tidak berubah lagi. Atau iterasi dihentikan berdasarkan jumlah iterasi maksimal yang ditentukan.

Algoritma K-Means secara iteratif dapat meningkatkan variasi dari nilai dari masing-masing cluster dimana obyek selanjutnya ditempatkan dalam *cluster* terdekat, dihitung dari titik tengah *cluster*. Titik tengah baru tersebut dapat ditentukan apabila semua data telah ditempatkan dalam *cluster* terdekat. Proses penentuan *centroid* dan penempatan data dalam *cluster* diulangi sampai nilai tengah dari semua *cluster* yang terbentuk tidak berubah lagi (Han dkk, 2012).

Sebelum melakukan proses *clustering* dilakukan tahap persiapan data, salah satu langkah yang dilakukan adalah menganalisis atribut serta nilai dari atribut yang akan digunakan untuk melakukan *clustering*. Apabila terdapat nilai yang berbeda atau memiliki rentang yang berbeda maka diperlukan adanya proses normalisasi terlebih dahulu. Normalisasi adalah proses transformasi untuk merubah nilai data. Normalisasi data yang dilakukan dengan menggunakan metode *Min-Max* merupakan metode normalisasi yang dapat menghasilkan transformasi linier dari data asal dimana normalisasi menggunakan metode *Min-Max* ini dapat memetakan sebuah nilai yang disimbolkan dengan  $v$  dari  $A$  menjadi  $v'$  dalam jangkauan dari nilai minimal dan maksimal yang baru. Normalisasi juga dapat digunakan untuk menyamakan skala atribut data kedalam *range* yang lebih spesifik yang lebih kecil seperti -1 sampai 1 atau 0 – 1. Untuk melakukan proses normalisasi dapat dilakukan dengan menggunakan Persamaan 3.2.

$$\text{Nilai normalisasi} = \frac{(nilai awal - nilai minimal)}{(nilai maksimal - nilai minimal)} \dots\dots\dots(3.2)$$

### 3.11 Implementasi K-Means

Algoritma K-Means telah mendapatkan perluasan atau *extension* terhadap kemampuannya hingga saat ini. Kumar dan Wasan, 2010 mencatat ada tiga varian dari algoritma K-Means hasil modifikasi yaitu algoritma global K-Means (Likas dkk, 20013), algoritma *efficient* K-Means (Zhang dkk, 2003) dan algoritma X-Means (Pelleg dan Moore, 2000).

Peningkatan akan kemampuan tersebut antara lain dengan diusulkannya *fast adaptive* K-Means *clustering* algoritma (Darken dan Moody, 1990), *intelligent* K-Means (Mirkin, 2005), algoritma *improved genetic* K-Means (Guo dkk, 2006), *constrained intelligent* K-Means (Amorim, 2008) serta usulan terkait dengan *shift-based initialization* pada K-Means (Cabria dan Gondra, 2012).

Penggunaan algoritma K-Means dengan menggunakan data spasial hingga saat ini telah diimplementasikan pada berbagai aplikasi diantaranya adalah *clustering* daerah resiko kebakaran di wilayah perkotaan (Lizhi dan Aizhu, 2008), identifikasi *cluster* pepohonan dengan menggunakan data dari citra satelit (Fan dkk, 2010) serta perencanaan sistem transportasi yang memiliki keterkaitan dengan penentuan jumlah lokasi yang sesuai untuk digunakan sebagai pusat layanan *cassava* (Tangkitjaroenongkol, 2011).

### 3.12 Kelebihan dan Kekurangan Algoritma K-Means

Setiap algoritma dapat memiliki kelebihan dan kekurangan dalam pengoperasian serta pengimplementasiannya, hal tersebut tidak menutup kemungkinan bahwa pada algoritma K-Means pun demikian. Berikut ini merupakan kelebihan serta kekurangan dari algoritma K-Means.

## Kelebihan

1. Algoritma K-Means memiliki kemudahan dalam pengimplementasian serta pengoperasiannya.
2. Waktu yang dibutuhkan untuk proses yang dilakukan oleh algoritma K-Means untuk melakukan proses pembelajaran relatif cepat.
3. Memiliki tingkat fleksibel yang tinggi serta adaptasi dalam penggunaan algoritma ini dapat dilakukan dengan mudah.
4. Penggunaan algoritma K-Means sangat umum, sehingga ketika terdapat *error* saat pengimplementasian akan banyak sekali dokumentasi yang diperlukan untuk melakukan penyelesaiannya.
5. Prinsip yang digunakan oleh algoritma K-Means yang sederhana sehingga dapat dijelaskan secara umum dan non statistik.

## Kekurangan

1. Saat algoritma K-Means pertama kali dijalankan, nilai K dinisialisasikan secara acak sehingga dalam hal penglompokan data hasil yang berbeda-beda pada setiap percobaan yang dilakukan. Namun, jika nilai yang didapat secara acak tersebut digunakan untuk inisialisasi hasilnya kurang baik, maka hasil yang didapat dari pengelompokan menjadi tidak optimal.
2. Apabila terjebak dalam sebuah permasalahan yang biasanya dinamakan *Curse of Dimensionality*. Hal tersebut akan terjadi apabila salahsatu data yang digunakan sebagai data latih memiliki dimensi yang sangat banyak. Misalnya, bila terdapat data dengan terdiri dari 2 atribut saja maka dimensinya hanya 2 juga. Namun, situasinya akan berbeda jika terdapat 20 atribut maka akan ada 20 dimensi yang dimiliki pula. Salahsatu cara kerja algoritma *clustering* ini adalah untuk memperoleh jarak terdekat diantara dari masing-masing k titik dengan titik lainnya. Pencarian jarak terdekat tersebut mudah dilakukan apabila atribut yang dimiliki hanya 2 atau dua dimensi saja, namun bila lebih dari itu hal tersebut akan menjadi lebih sulit untuk dilakukan proses perhitungan dari jarak terdekat.

### 3.13 Permasalahan Terkait Algoritma K-Means

Dalam penggunaannya beberapa permasalahan sering ditemukan dalam algoritma K-Means. Beberapa permasalahan yang sering muncul tersebut diantaranya adalah sebagai berikut.

1. Sering ditemukannya beberapa model *clustering* yang berbeda.

Penyebab dari permasalahan ini biasanya disebabkan oleh adanya perbedaan pada proses inisialisasi dari masing-masing anggota *cluster*. Dalam proses inisialisasi biasanya digunakan proses inisialisasi secara acak. Dalam suatu kajian tentang perbandingan, proses ininisialisasi yang dilakukan secara acak memiliki kecenderungan untuk memberikan hasil yang lebih baik baik serta independen, namun memiliki kekurangan yaitu lambatnya kecepatan untuk konvergen.

2. Penentuan nilai K, untuk jumlah *cluster* yang paling tepat.

Permasalahan ini merupakan permasalahan utama dalam algoritma K-Means. Oleh karena itu diperlukan pendekatan untuk penentuan nilai K yang optimal agar memperoleh hasil *clustering* optimal.

3. Terjadi kegagalan dalam penentuan kriteria untuk memperoleh hasil yang konvergensi.

Kegagalan untuk konvergen, dapat dimungkinkan terjadi dalam metode Hard K-Means ataupun Fuzzy K-Means. Kemungkinan dari hal tersebut akan semakin besar terjadi dalam metode Hard K-Means, dikarenakan setiap data yang berada dalam dataset dialokasikan secara tegas (*hard*) untuk menjadi bagian dari suatu cluster tertentu.

Perpindahan data yang terjadi dari suatu dataset ke dalam suatu *cluster*, dapat mempengaruhi berubahnya karakteristik dari suatu model clustering sehingga menyebabkan data yang telah berpindah tersebut lebih cocok untuk berada pada cluster semula sebelum data tersebut dipindahkan, begitu pula sebaliknya.

Pada Fuzzy K-Means walaupun terjadi permasalahan ini, kemungkinan terjadinya adalah sangat kecil, dikarenakan setiap data dilengkapi dengan *membership function* untuk menjadi anggota dari sebuah *cluster*.

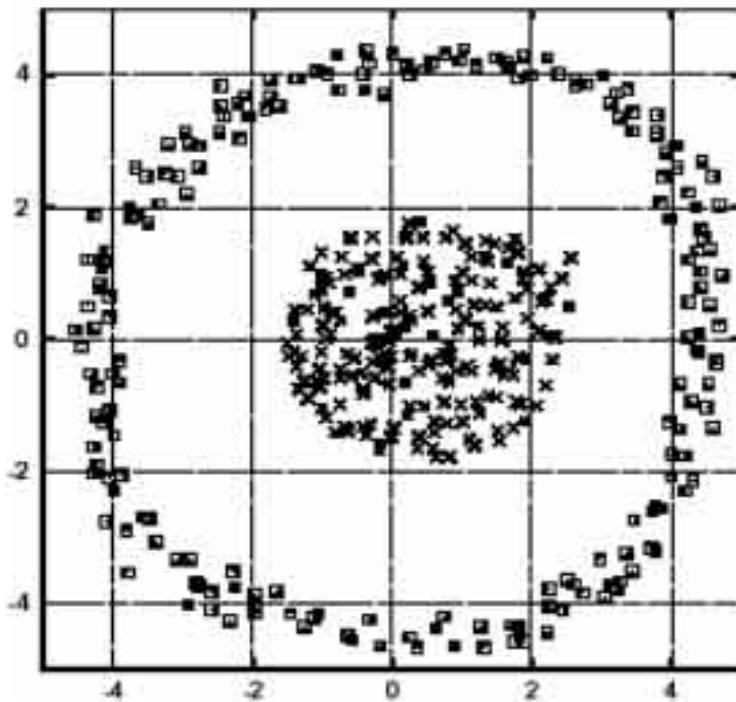
#### 4. *Outliers*.

Permasalahan ini hampir terjadi pada setiap metode yang digunakan untuk memodelkan data. Dalam metode K-Means hal ini dapat menjadi sebuah permasalahan yang cukup serius dan menentukan terhadap hasil dari *clustering*. Hal yang perlu diperhatikan dalam mendekripsi *outliers* di dalam proses penegloppokan data termasuk bagaimana menentukan apakah suatu data merupakan *outliers* dari suatu *cluster* tertentu serta apakah data dalam jumlah kecil yang membentuk suatu *cluster* dapat dianggap sebagai sebuah *outliers*. Proses tersebut memerlukan sebuah pendekatan yang berbeda dengan proses pendekripsi *outliers* tersebut dalam suatu dataset yang hanya memiliki satu populasi yang sama.

#### 5. Bentuk dari *cluster*.

Bentuk dari suatu *cluster* yang diperoleh di dalam algoritma K-Means merupakan hal yang perlu dicermati. Berbeda dengan metode *clustering* lainnya. K-Means secara umum tidak memperhatikan bentuk dari tiap-tiap cluster yang menjadi dasar terbentuknya suatu model, walapun biasanya secara natural sebuah *cluster* berbentuk bulat.

Diperlukan beberapa pendekatan untuk dataset yang memiliki bentuk yang tidak biasa.



Gambar 3.6 Contoh Dataset yang Memiliki Bentuk Khusus

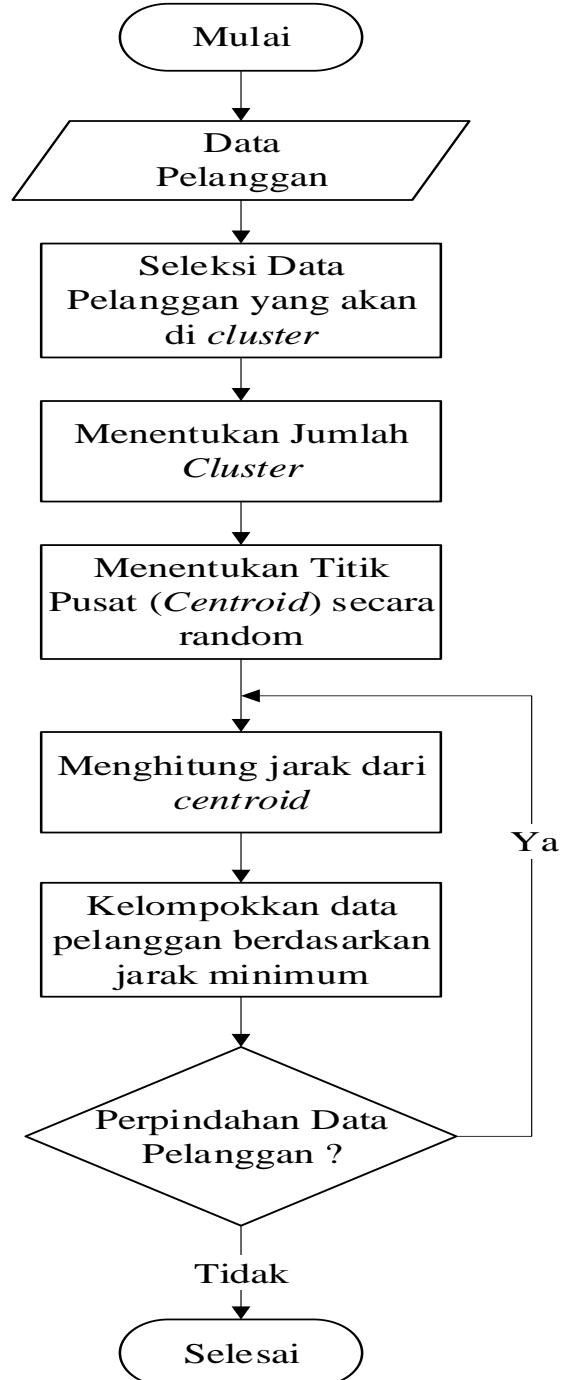
#### 6. *Overlapping*.

Permasalahan yang terkait dengan *overlapping* ini dapat dikatakan sebagai permasalahan yang sering sekali diabaikan dikarenakan biasanya permasalahan ini sulit untuk dideteksi. Hal tersebut bias terjadi pada metode Hard K-Means maupun metode Fuzzy K-Means, dikarenakan pada dasarnya, metode tersebut tidak dilengkapi dengan fungsi untuk pendekripsi apakah terdapat *cluster* tersembunyi didalam suatu *cluster*.

Permasalahan tersebut harus sangat diperhatikan dalam proses *clustering* atau pengelompokan data menggunakan algoritma K-Means sehingga dapat dilakukan langkah antisipasi agar beberapa permasalahan tersebut tidak terjadi.

### 3.14 Contoh Perhitungan *Clustering*

Pada pembahasan ini akan dijelaskan bagaimana contoh implementasi pengolahan data menggunakan algoritma K-Means. Dalam hal ini algoritma K-Means akan diaplikasikan untuk *clustering* data pelanggan. Sebelum melakukan pengolahan data, terlebih dahulu harus dipahami langkah-langkah yang dilakukan dalam proses *clustering* agar prosesnya berjalan dengan baik. Berikut ini merupakan diagram alir rancangan *clustering* yang diimplementasikan dalam segmentasi pelanggan.



Gambar 3.7 Diagram Alir Rancangan Penentuan *Cluster* Atau Segmentasi Data Pelanggan

Pada gambar 3.7 merupakan langkah-langkah yang dilakukan untuk melakukan *clustering* menggunakan algoritma K-Means.

1. Data pelanggan: pengumpulan dan penyiapan data pelanggan yang akan di *cluster*.
  2. Seleksi data pelanggan: dilakukan untuk memilih atribut yang akan digunakan untuk proses *clustering*.
  3. Penentuan jumlah *cluster*: menentukan jumlah *cluster* atau nilai k yang diinginkan.
  4. Penetuan titik pusat : dilakukan untuk menentukan nilai *centroid* pada setiap *cluster*.
  5. Menghitung jarak dari centroid: proses untuk menghitung jarak minimum antara data dengan titik pusat atau *centroid*, dapat dilakukan dengan menggunakan Persamaan 3.1.

## Keterangan :

$d_{ij}$  : Euclidian distance, Jarak antara data pelanggan i dan j

$p$  : Dimensi data yang digunakan

$X_{ik}$  : Data pelanggan ke-*i*

$C_{jk}$  : Centroid ke-j

6. Pengelompokkan data: pada tahap pengelompokkan ini dilakukan berdasarkan hasil perhitungan *euclidian distance* setiap data pelanggan. Suatu data pelanggan akan menjadi anggota dari *cluster* ke-k apabila jarak data tersebut ke pusat *cluster* bernilai paling kecil jika dibandingkan dengan jarak ke pusat *cluster* lainnya.

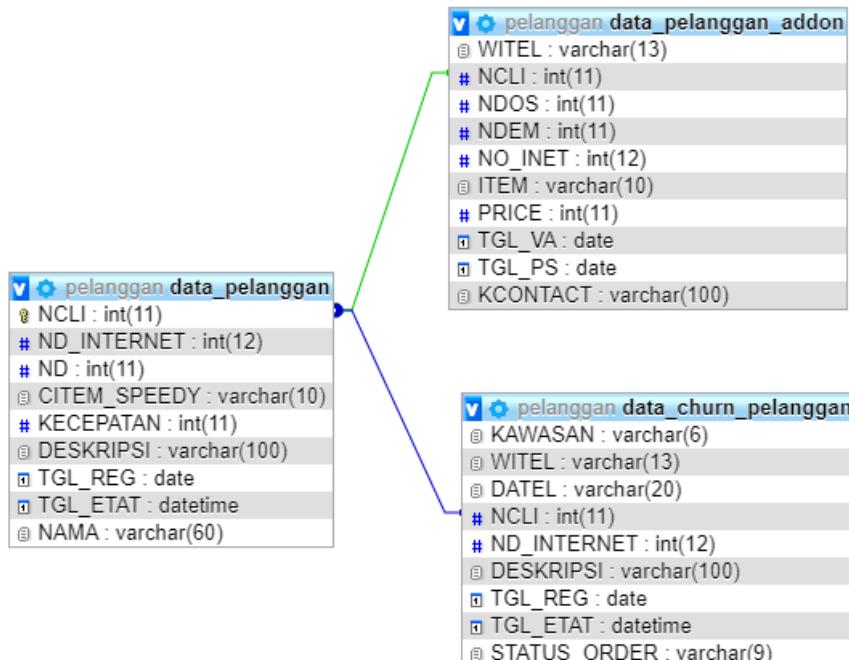
## 72 PENGENALAN ALGORITMA

7. Hitung nilai *centroid* baru berdasarkan data yang mengikuti *cluster* masing-masing. Untuk memperoleh titik *centroid* baru dilakukan dengan cara menghitung nilai rata-rata dari data yang ada pada *cluster* yang sama.
8. Apabila tidak ada data yang berpindah ke *cluster* lain maka proses *clustering* selesai. Ulangi dari langkah ke tiga hingga langkah ke lima apabila terdapat perpindahan data ke *cluster* lain.

Berikut ini merupakan penerapan proses segmentasi data pelanggan sesuai dengan diagram alir pada Gambar 3.7.

### • Penjelasan Data

Pada pembahasan ini akan dijelaskan terkait dengan data yang akan digunakan dalam proses clustering. Data yang akan digunakan merupakan data pelanggan dimana hasil akhir dari *clustering* ini adalah pengelompokan data ke dalam masing-masing *cluster* sesuai dengan kemiripan dari masing-masing data.



Gambar 3.8 Struktur Tabel Data Pelanggan

Data yang digunakan merupakan contoh data pelanggan yang diperoleh dari arsip perusahaan telekomunikasi untuk digunakan dalam proses segmentasi pelangga. Adapun data tersebut terdiri dari beberapa tabel dan dari masing-masing tabel tersebut memiliki atribut dan tipe data yang berbeda disesuaikan dengan nisinya yang nantinya akan diinputkan ke dalam tabel. Untuk lebih detail terkait dengan atribut dan tipe data bisa dilihat pada Gambar 3.8. Berikut ini merupakan penjelasan dari masing-masing tabel beserta dengan atributnya.

### 1. Data Pelanggan Indihome

Pada tabel data pelanggan indihome berisi data pelanggan yang berlangganan layanan jaringan internet. Adapun tabel tersebut terdiri dari beberapa atribut diantaranya adalah NCLI yang merupakan nomor pelanggan yang mana NCLI ini merupakan primary key dalam struktur database pelanggan. Primary key itu sendiri merupakan sebuah data yang dijadikan sebagai patokan atau identitas utama dari tiap data pelanggan yang membedakan dengan data pelanggan yang lain. Kemudian ND\_INTERNET ini merupakan nomor internet yang dimiliki oleh pelanggan; ND merupakan nomor telefon dari pelanggan; CITEM\_SPEEDY merupakan kode untuk kecepatan internet yang dipilih oleh pelanggan; KECEPATAN merupakan jumlah kecepatan yang diperoleh oleh pelanggan sesuai dengan paket langganan yang digunakan; DESKRIPSI merupakan penjelasan dari paket digunakan oleh masing-masing pelanggan; TGL\_REG merupakan tanggal registrasi dari pelanggan untuk berlangganan paket internet; TGL\_ETAT merupakan tanggal pelanggan dapat menikmati atau menggunakan layanan internet, tanggal ini bisa berbeda tiap pelanggan dikarenakan bila terjadi antrian dalam proses pemasangan setelah melakukan proses registrasi maka secara otomatis layanan internet yang dinikmati oleh pelanggan waktunya dapat berbeda. Kemudian yang terakhir adalah NAMA yaitu nama dari

## 74 PENGENALAN ALGORITMA

pelanggan atau orang yang bertanggung jawab untuk membayar tagihan layanan.

### 2. Data Pelanggan *Add On*

Tabel data pelanggan *Add On* merupakan data pelanggan yang berlangganan layanan tambahan produk digital yang disediakan oleh perusahaan untuk melengkapi layanan internet yang telah digunakan. Karena bersifat sebagai tambahan, maka layanan *add on* ini bersifat opsional dan pelanggan tidak diwajibkan untuk berlangganan. Pada tabel ini terdapat beberapa atribut yang menjadi identitas bahwa pelanggan telah berlangganan layanan *add on*. Atribut tersebut diantarnya adalah WITEL merupakan atribut yang menjelaskan wilayah dari pelanggan tersebut mendaftarkan wilayah, dikarenakan perusahaan telekomunikasi memiliki jangkauan yang luas oleh karena itu perusahaan memiliki sebuah kantor perwakilan di masing-masing wilayah yang setingkat kabupaten untuk memenuhi kebutuhan layanan pelanggan dan dikenal dengan istilah witel, namun dalam satu witel tersebut terdiri dari beberapa kabupaten/kota; NCLI merupakan no pelanggan; NDOS; NDEM merupakan no permintaan, jadi saat pelanggan melakukan *request* layanan maka akan dibuatkan nomor permintaan; NO\_INET merupakan nomor internet pelanggan; ITEM merupakan atribut yang berisi kode untuk layanan *add on* yang akan dipesan oleh pelanggan ; TGL\_VA; TGL\_PS merupakan tanggal layanan *add on* dipasang atau dikonfigurasi untuk dapat digunakan oleh pelanggan; KCONTACT merupakan identitas dari pelanggan yang berisi nama dan nomor *handphone* pelanggan yang dapat dihubungi .

### 3. Data *Churn* pelanggan

Tabel data *churn* pelanggan berisi data pelanggan yang berhenti berlangganan layanan. Adapun atribut yang terdapat dalam tabel tersebut diantaranya adalah KAWASAN merupakan atribut yang mendeskripsikan pelanggan berada di wilayah mana sesuai dengan pembagian wilayah yang dilakukan oleh perusahaan seluruh Indonesia; WITEL merupakan atribut yang menjelaskan wilayah dari pelanggan tersebut mendaftarkan wilayah, dikarenakan perusahaan telekomunikasi memiliki jangkauan yang luas oleh karena itu perusahaan memiliki sebuah kantor perwakilan di masing-masing wilayah yang setingkat kabupaten untuk memenuhi kebutuhan layanan pelanggan dan dikenal dengan istilah witel, namun dalam satu witel tersebut dapat terdiri dari beberapa kabupaten/kota; DATEL merupakan informasi wilayah cakupan perusahaan yang lebih kecil terhadap tempat tinggal pelanggan (setingkat kecamatan) dimana datel ini merupakan wilayah cakupan pelanggan yang berada dalam sebuah witel; NCLI merupakan nomor pelanggan; ND\_INTERNET merupakan nomor internet pelanggan; DESKRIPSI merupakan informasi dari paket yang diambil oleh pelanggan; TGL\_REG merupakan tanggal yang dicatat oleh sistem saat pelanggan melakukan registrasi; TGL\_ETAT merupakan tanggal yang dicatat oleh sistem dimana pelanggan dapat menggunakan layanan; STATUS\_ORDER merupakan status dari order pelanggan. Pada tabel *churn* ini status order yang digunakan yaitu CHURN\_OUT dikarenakan pada tabel ini dikhkususkan untuk mendata pelanggan yang berhenti menggunakan layanan .

- **Seleksi Data Pelanggan**

Proses ini dilakukan untuk memilih atritbut yang akan digunakan untuk proses *clustering*, pemilihan tersebut didasarkan pada item yang memang bisa dijadikan

## 76 PENGENALAN ALGORITMA

sebagai identitas pelanggan serta relevan untuk dijadikan sebagai data yang digunakan untuk melakukan segmentasi pelanggan.

Dari sekian banyak atribut yang ada, dipilihlah beberapa atribut yang dapat digunakan untuk mewakili setiap pelanggan diantaranya adalah nomor pelanggan atau NCLI, lama berlangganan, jumlah layanan yang digunakan serta total tagihan namun dalam proses *clustering*, atribut yang digunakan hanya 3 atribut kecuali nomor pelanggan atau NCLI.

Tabel 3.1 Atribut Yang Digunakan Untuk *Clustering*

NCLI	LAMA_LANGGANAN	JUMLAH_LAYANAN	JUMLAH_TAGIHAN_HARUS_DIBAYAR
39684298	11	1	418000
39713960	11	2	511500
39716635	11	2	698500
39818227	11	2	291500
68228	11	2	517000
39937697	11	1	352000
40011078	11	2	847000
39817407	11	1	621500
39864058	11	2	1100000
39952623	11	1	902000
39891161	11	1	352000
40033755	11	2	291500
39786171	11	2	2090000
39995005	11	2	621500
32745610	11	2	814000
380513	1	2	1358500
39765624	11	1	286000
39600770	11	1	275000
39999904	11	1	275000
39892083	11	1	495000

Pada pembahasan seleksi data pelanggan dilakukan juga proses normalisasi. Hal ini dilakukan untuk menyamakan *range* nilai dari masing-masing atribut yang dipilih. Proses normalisasi dilakukan dengan menggunakan Persamaan 3.2. Berikut ini merupakan contoh perhitungan normalisasi.

### 1. Lama Langganan

<b>Lama Langganan</b>	
Max	11
Min	1

$$\begin{aligned}
 X_{11} &= (X_{\text{Lama langganan NCLI 1}} - X_{\min}) / (X_{\max} - X_{\min}) \\
 &= (11 - 1) / (11 - 1) \\
 &= 1
 \end{aligned}$$

### 2. Jumlah Layanan

<b>Jumlah Layanan</b>	
Max	2
Min	1

$$\begin{aligned}
 X_{21} &= (X_{\text{Jumlah Layanan NCLI 1}} - X_{\min}) / (X_{\max} - X_{\min}) \\
 &= (1 - 1) / (2 - 1) \\
 &= 0
 \end{aligned}$$

### 3. Total Tagihan

<b>Tagihan</b>	
Max	2090000
Min	275000

$$\begin{aligned}
 X_{31} &= (X_{\text{Total Tagihan NCLI 1}} - X_{\min}) / (X_{\max} - X_{\min}) \\
 &= (418000 - 275000) / (2090000 - 275000) \\
 &= 0.08
 \end{aligned}$$

Perhitungan dilakukan sampai semua nilai atribut dinormalisasi. Hasil dari proses normalisasi dapat dilihat pada Tabel 3.2.

Tabel 3.2 Hasil Normalisasi

NCLI	LAMA_LANGGANAN	JUMLAH_LAYANAN	TAGIHAN
39684298	1.00	0.00	0.08
39713960	1.00	0.25	0.13
39716635	1.00	0.25	0.23
39818227	1.00	0.25	0.01
39813816	1.00	0.00	0.32
30389436	1.00	0.25	0.32

Lanjutan Tabel 3.2 Hasil Normalisasi

NCLI	LAMA_LANGGANAN	JUMLAH_LAYANAN	TAGIHAN
39950721	1.00	0.00	0.48
39872630	1.00	0.25	0.48
39880310	1.00	0.00	0.03
616655	1.00	0.50	0.20
39792742	0.90	0.25	0.19
39802417	1.00	0.00	0.08
40012860	1.00	0.25	0.24
110281	1.00	0.50	0.24
39739034	1.00	0.00	0.23
39742383	1.00	0.25	0.23
40007439	1.00	0.00	0.35
39884232	1.00	0.25	0.35
39884288	1.00	0.25	0.35
639184	1.00	0.75	0.87

- **Penentuan Jumlah Cluster**

Pada tahap ini dilakukan penentuan jumlah *cluster* atau nilai K. Adapun nilai K yang digunakan pada penelitian ini berjumlah 3.

- **Penentuan Titik Pusat**

Pada tahap ini dilakukan penentuan nilai centroid pada setiap *cluster*, nilai dari *centroid* itu sendiri ditentukan secara acak.

Tabel 3.3 *Centroid* awal

K	NCLI	LAMA_LANGGANAN	JUMLAH_LAYANAN	TAGIHAN
0	30389436	1.00	0.25	0.32
1	110281	1.00	0.50	0.24
2	639184	1.00	0.75	0.87

- Perhitungan Jarak Data Centroid

#### Inisialisasi

Pada tahap inisialisasi merupakan perhitungan jarak antara data dengan nilai *centroid* awal (*euclidian distance*) dengan menggunakan persamaan 3.1.

1. Perhitungan jarak minimum data ke-1 (1.00, 0, 0.08)

❖ DC0

$$\begin{aligned} & \sqrt{(1 - 1)^2 + (0 - 0.25)^2 + (0.08 - 0.32)^2} \\ & = 0.346554469 \end{aligned}$$

❖ DC1

$$\begin{aligned} & \sqrt{(1 - 1)^2 + (0 - 0.5)^2 + (0.08 - 0.24)^2} \\ & = 0.52497619 \end{aligned}$$

❖ DC2

$$\begin{aligned} & \sqrt{(1 - 1)^2 + (0 - 0.75)^2 + (0.08 - 0.87)^2} \\ & = 1.089311709 \end{aligned}$$

2. Perhitungan jarak minimum data ke-2 (1.00, 0.25, 0.13)

❖ DC0

$$\begin{aligned} & \sqrt{(1 - 1)^2 + (0.25 - 0.25)^2 + (0.13 - 0.32)^2} \\ & = 0.19 \end{aligned}$$

## ❖ DC1

$$\begin{aligned} & \sqrt{(1 - 1)^2 + (0.25 - 0.5)^2 + (0.13 - 0.24)^2} \\ & = 0.273130006 \end{aligned}$$

## ❖ DC2

$$\begin{aligned} & \sqrt{(1 - 1)^2 + (0.25 - 0.75)^2 + (0.13 - 0.87)^2} \\ & = 0.893084542 \end{aligned}$$

Lakukan perhitungan yang sama pada data yang lain sehingga menghasilkan nilai jarak data terhadap *centroid* sebagai berikut.

Tabel 3.4 Hasil Perhitungan Jarak Data Dengan *Centroid*

Pada Inisialisasi

NO	NCLI	DC0	DC1	DC2
1	39684298	0.346554	0.524976	1.089312
2	39713960	0.19	0.27313	0.893085
3	39716635	0.09	0.2502	0.812158
4	39818227	0.31	0.339706	0.994786
5	39813816	0.25	0.50636	0.930054
6	30389436	0	0.262488	0.743303
7	39950721	0.296816	0.554617	0.84534
8	39872630	0.16	0.346554	0.634114
9	39880310	0.382884	0.54231	1.126099
10	616655	0.277308	0.04	0.715122
11	39792742	0.164012	0.273861	0.849941
12	39802417	0.346554	0.524976	1.089312
13	40012860	0.08	0.25	0.804301
14	110281	0.262488	0	0.677791
15	39739034	0.265707	0.5001	0.985951
16	39742383	0.09	0.2502	0.812158
17	40007439	0.251794	0.511957	0.912634
18	39884232	0.03	0.27313	0.721388
19	39884288	0.03	0.27313	0.721388
20	639184	0.743303	0.677791	0

Penentuan data berada pada *cluster* tertentu didasarkan pada jarak minimum perhitungan *euclidian distance* sehingga didapat hasil sebagai berikut.

Tabel 3.5 Hasil Pemetaan *Centroid* Awal

NO	NCLI	DC0	DC1	DC2	HASIL
1	39684298	*			0
2	39713960	*			0
3	39716635	*			0
4	39818227	*			0
5	39813816	*			0
6	30389436	*			0
7	39950721	*			0
8	39872630	*			0
9	39880310	*			0
10	616655		*		1
11	39792742	*			0
12	39802417	*			0
13	40012860	*			0
14	110281		*		1
15	39739034	*			0
16	39742383	*			0
17	40007439	*			0
18	39884232	*			0
19	39884288	*			0
20	639184			*	2

82 PENGENALAN ALGORITMA

#### • Iterasi Pertama

Setelah data terbagi ke dalam *cluster* pada tahap inisialisasi, untuk melanjutkan tahap iterasi pertama diperlukan adanya penentuan centroid baru untuk menghitung jarak menggunakan euclidian *distance*.

Untuk menentukan centroid baru dapat dilakukan dengan menggunakan Persamaan 3.3.

Lokasi centroid setiap kelompok diambil dari rata-rata (mean) semua nilai data pada setiap fiturnya. Dimana rata-rata semua nilai dihitung dengan membagi jumlah data dengan jumlah atribut untuk pembagi pada perhitungan atribut baru.

• C0

$$\left( \frac{(1+1+1+1+1+1+1+1+1+1+0.9+1+1+1+1+1+1+1+1+1+1)}{17} \right)$$

$\equiv 0.994117647$

$\equiv 0.147058824$

$$\left( \frac{(0.08 + 0.13 + 0.23 + 0.01 + 0.32 + 0.32 + 0.48 + 0.48 + 0.03 + 0.19 + 0.08 + 0.24 + 0.23 + 0.23 + 0.35 + 0.35 + 0.35)}{17} \right)$$

= 0.241176471

Sehingga *centroid* C0 yang baru bernilai (0.994117647, 0.147058824, 0.241176471).

- C1

$$\left( \frac{(1+1)}{2} \right)$$

$$= 1$$

$$\left( \frac{(0.5+0.5)}{2} \right)$$

$$= 0.5$$

$$\left( \frac{(0.2+0.24)}{2} \right)$$

$$= 0.22$$

Sehingga *centroid* C1 yang baru bernilai (1, 0.5, 0.22).

- C2

$$\left( \frac{(1)}{1} \right)$$

$$= 1$$

$$\left( \frac{(0.75)}{1} \right)$$

$$= 0.75$$

$$\left( \frac{(0.87)}{17} \right)$$

$$= 0.87$$

Sehingga *centroid* C2 yang baru bernilai (1, 0.75, 0.87).

Hitung kembali jarak euclidian *distance* antara data dengan centroid baru seperti pada tahap sebelumnya, sehingga didapat hasil sebagai berikut.

Tabel 3.6 Hasil Perhitungan Jarak Data Dengan *Centroid*

Pada Iterasi Pertama

NO	NCLI	DC0	DC1	DC2
1	39684298	0.218263	0.51923	1.089312
2	39713960	0.15163	0.265707	0.893085
3	39716635	0.103713	0.2502	0.812158
4	39818227	0.253129	0.326497	0.994786
5	39813816	0.166955	0.509902	0.930054
6	30389436	0.129787	0.269258	0.743303
7	39950721	0.280531	0.56356	0.84534
8	39872630	0.260131	0.360694	0.634114
9	39880310	0.257403	0.534883	1.126099
10	616655	0.355384	0.02	0.715122
11	39792742	0.148573	0.270924	0.849941
12	39802417	0.218263	0.51923	1.089312
13	40012860	0.103116	0.250799	0.804301
14	110281	0.352992	0.02	0.677791
15	39739034	0.1476	0.5001	0.985951
16	39742383	0.103713	0.2502	0.812158
17	40007439	0.18304	0.516624	0.912634
18	39884232	0.149913	0.28178	0.721388
19	39884288	0.149913	0.28178	0.721388
20	639184	0.871201	0.696419	0

Tentukan kembali data berada pada *cluster* yang sesuai dengan jarak minimum perhitungan *euclidian distance* sehingga didapat hasil sebagai berikut.

Tabel 3.7 Hasil Pemetaan *Centroid* Iterasi Pertama

NO	NCLI	DC0	DC1	DC2	HASIL
1	39684298	*			0
2	39713960	*			0
3	39716635	*			0
4	39818227	*			0
5	39813816	*			0
6	30389436	*			0
7	39950721	*			0
8	39872630	*			0
9	39880310	*			0
10	616655		*		1
11	39792742	*			0
12	39802417	*			0
13	40012860	*			0
14	110281		*		1
15	39739034	*			0
16	39742383	*			0
17	40007439	*			0
18	39884232	*			0
19	39884288	*			0
20	639184			*	2

## 86 PENGENALAN ALGORITMA

- Iterasi Kedua

Untuk melanjutkan perhitungan pada iterasi kedua, diperlukan *centroid* baru. *Centroid* baru tersebut dihitung menggunakan cara yang sama dengan yang dilakukan untuk mencari *centroid* pada iterasi kedua dengan persamaan 3.3.

Berikut ini merupakan perhitungan untuk memperoleh *centroid* baru pada iterasi kedua.

- C0

$$\left( \frac{(1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 0.9 + 1 + 1 + 1 + 1 + 1 + 1 + 1)}{17} \right)$$

$$= 0.994117647$$

$$\left( \frac{(0 + 0.25 + 0.25 + 0.25 + 0 + 0.25 + 0 + 0.25 + 0 + 0.25 + 0 + 0.25 + 0 + 0.25 + 0.25 + 0.25 + 0.25)}{17} \right)$$

$$= 0.147058824$$

$$\left( \frac{(0.08 + 0.13 + 0.23 + 0.01 + 0.32 + 0.32 + 0.48 + 0.48 + 0.03 + 0.19 + 0.08 + 0.24 + 0.23 + 0.23 + 0.35 + 0.35 + 0.35)}{17} \right)$$

$$= 0.241176471$$

Sehingga *centroid* C0 yang baru bernilai (0.994117647, 0.147058824, 0.241176471).

- C1

$$\left( \frac{(1 + 1)}{2} \right)$$

$$= 1$$

$$\left( \frac{(0.5 + 0.5)}{2} \right)$$

$$= 0.5$$

$$\left( \frac{(0.2 + 0.24)}{2} \right)$$

$$= 0.22$$

Sehingga *centroid* C1 yang baru bernilai (1, 0.5, 0.22).

- C2

$$\left( \frac{(1)}{1} \right)$$

$$= 1$$

$$\left( \frac{(0.75)}{1} \right)$$

$$= 0.75$$

$$\left( \frac{(0.87)}{17} \right)$$

$$= 0.87$$

Sehingga *centroid* C2 yang baru bernilai (1, 0.75, 0.87).

Setelah *centroid* baru didapat, hitung kembali jarak antara *centroid* baru dengan data, kemudian pilih jarak minimum untuk menentukan data tersebut berada pada *cluster* tertentu.

Tabel 3.8 Hasil Perhitungan Jarak Data Dengan *Centroid*

Pada Iterasi Kedua

NO	NCLI	DC0	DC1	DC2
1	39684298	0.218263	0.51923	1.089312
2	39713960	0.15163	0.265707	0.893085
3	39716635	0.103713	0.2502	0.812158
4	39818227	0.253129	0.326497	0.994786
5	39813816	0.166955	0.509902	0.930054
6	30389436	0.129787	0.269258	0.743303
7	39950721	0.280531	0.56356	0.84534
8	39872630	0.260131	0.360694	0.634114
9	39880310	0.257403	0.534883	1.126099
10	616655	0.355384	0.02	0.715122
11	39792742	0.148573	0.270924	0.849941
12	39802417	0.218263	0.51923	1.089312
13	40012860	0.103116	0.250799	0.804301
14	110281	0.352992	0.02	0.677791
15	39739034	0.1476	0.5001	0.985951
16	39742383	0.103713	0.2502	0.812158
17	40007439	0.18304	0.516624	0.912634
18	39884232	0.149913	0.28178	0.721388
19	39884288	0.149913	0.28178	0.721388
20	639184	0.871201	0.696419	0

Tabel 3.9 Hasil Pemetaan *Centroid* Iterasi Kedua

NO	NCLI	DC0	DC1	DC2
1	39684298	*		
2	39713960	*		
3	39716635	*		
4	39818227	*		
5	39813816	*		
6	30389436	*		
7	39950721	*		
8	39872630	*		
9	39880310	*		
10	616655		*	
11	39792742	*		
12	39802417	*		
13	40012860	*		
14	110281		*	
15	39739034	*		
16	39742383	*		
17	40007439	*		
18	39884232	*		
19	39884288	*		
20	639184			*

Dikarenakan *centroid* pada iterasi kedua sama dengan *centroid* pada iterasi pertama berarti pengulangan perhitungan atau iterasi dihentikan pada iterasi kedua dikarenakan posisi data sudah tidak ada yang berpindah posisi dan sudah berada pada *cluster* nya.

### 3.15 Metode Penentuan K

Nilai K dalam suatu proses *clustering* merupakan sebuah elemen penting. Dikarenakan K ini adalah inisialisasi dari jumlah *cluster* atau kelompok untuk pembagian data yang dilakukan dalam *clustering*. Untuk memperoleh hasil pengelompokan data yang maksimal, penentuan nilai K ini tidak dapat ditentukan secara sembarang. Oleh karena itu digunakanlah sebuah metode yang dapat menentukan nilai K untuk mendapatkan hasil *clustering* yang optimal dan dinamakan dengan metode *Elbow*.

#### 3.15.1 Metode *Elbow*

Metode *Elbow* adalah sebuah metode yang digunakan untuk memperoleh informasi dalam menentukan jumlah *cluster* terbaik dengan cara melihat hasil persentase dari perbandingan antara jumlah *cluster* yang akan membentuk siku dalam suatu titik tertentu. Metode ini memberikan ide/gagasan dengan cara memilih nilai *cluster* dan kemudian menambah nilai *cluster* tersebut untuk dijadikan model data dalam penentuan *cluster* terbaik. Dan selain itu persentase perhitungan yang dihasilkan menjadi pembanding antara jumlah *cluster* yang ditambah. Hasil persentase yang berbeda dari setiap nilai *cluster* dapat ditunjukkan dengan menggunakan grafik sebagai sumber informasinya. Jika nilai *cluster* pertama dengan nilai *cluster* kedua memberikan sudut dalam grafik atau nilainya mengalami penurunan paling besar maka nilai *cluster* tersebut yang terbaik.

Menurut Bholowalia dan Kumar (2014) tahapan metode Elbow dalam menentukan nilai K pada pada K-Means:

1. Menginisialisasi awal nilai k.
2. Menaikan nilai k.
3. Menghitung hasil *Sum of Square Error* dari tiap nilai k.
4. Analisa hasil *Sum of Square Error* dari nilai k yang mengalami penurunan secara drastis.
5. Cari dan tetapkan nilai k yang berbentuk siku.

Pada metode *Elbow* nilai *cluster* terbaik dilihat dengan membandingkan nilai yang akan diambil dari hasil perhitungan *Sum of Square Error* (SSE) dari masing-masing cluster, karena semakin besar jumlah *cluster* K maka nilai dari *Sum of Square Error* (SSE) tersebut akan semakin kecil. Untuk menghitung SSE dapat menggunakan Persamaan 3.4.

$$SSE = \sum_{K=1}^K \sum_{x_i \in S_k} \| X_i - C_k \|^2 \dots \dots \dots \quad (3.4)$$

Dimana:

K = jumlah *cluster*

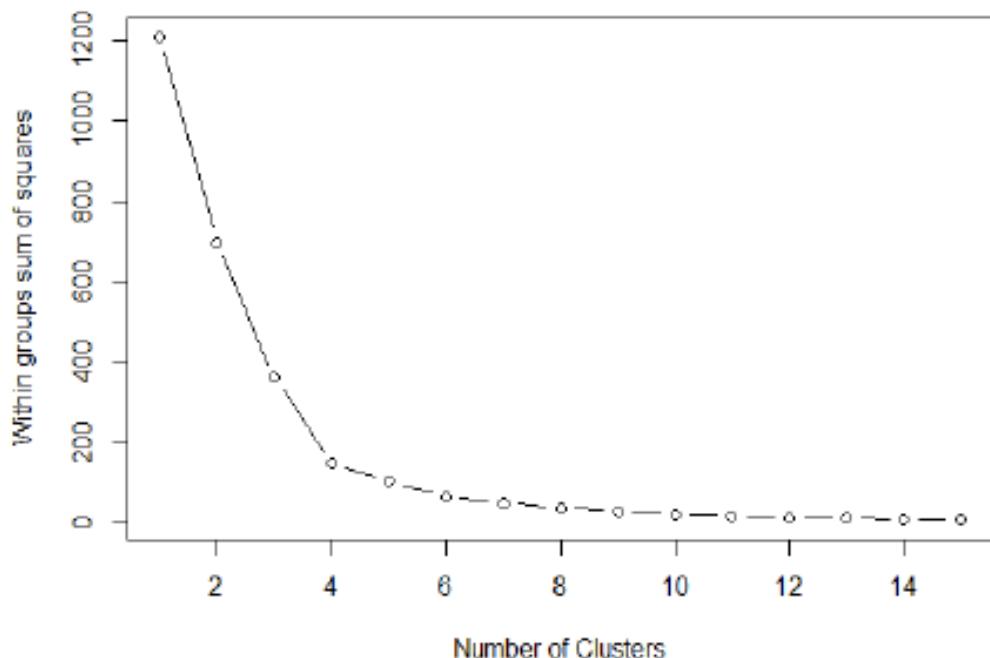
$x_i$  = data ke – i

$C_k$  = centroid *cluster*

*Sum of Square Error* (SSE) merupakan rumus yang digunakan untuk mengukur perbedaan antara data yang diperoleh dengan model perkiraan yang telah dilakukan sebelumnya. SSE sering digunakan sebagai acuan penelitian terkait dalam menentukan optimal *cluster*.

Setelah dilihat hasil dari SSE maka diperoleh beberapa nilai K yang mengalami penurunan yang paling besar dan signifikan dan selanjutnya hasil dari nilai K tersebut akan turun secara perlahan-lahan sampai hasil nilai K tersebut stabil. Misal, nilai *cluster* dengan nilai K = 2 ke *cluster* yang nilai

$K = 3$ , kemudian dari *cluster* dengan nilai  $K = 3$  ke *cluster* dengan nilai  $K = 4$ , dari perubahan jumlah cluster yang berbeda tersebut dapat dilihat penurunan yang signifikan yang nantinya akan membentuk siku pada titik *cluster* dengan nilai  $K = 3$ . Dengan demikian dapat disimpulkan bahwa hasil dari penggunaan metode *elbow* untuk menentukan nilai  $K$  yang terbaik dan diperoleh nilai  $K$  yang terbaik adalah 3 seperti ditunjukkan pada Gambar 3.7.



Gambar 3.9 Grafik Metode Elbow

### 3.16 Metode Evaluasi *Cluster*

Evaluasi dapat diartikan sebagai sebuah proses untuk memeriksa, menilai, membuat suatu keputusan ataupun menyediakan informasi yang berguna terhadap program atau proses yang telah dilaksanakan serta untuk mengukur sejauh mana ketercapaian dari sebuah proses tersebut. Dalam hal ini evaluasi clustering dilakukan untuk mengejui performa dari suatu cluster. Dengan adanya evaluasi *cluster* dapat dilihat dari berbagai aspek bahwa

proses *clustering* yang dilakuakn memiliki hasil yang baik serta telah dilakukan dengan tahapan-tahapan yang tepat.

Berikut ini merupakan beberapa metode yang digunakan untuk melakukan evaluasi *cluster*.

### 3.16.1 Silhouette index

Secara umum, indeks validitas Silhouette menghitung rata-rata nilai setiap titik pada himpunan data. Lebih spesifik, perhitungan nilai setiap titik merupakan selisih dari nilai *separation* dan *compactness* yang dibagi dengan maksimum antara keduanya. Jumlah *cluster* yang terbaik ditunjukkan dengan nilai Silhouette yang semakin mendekati 1 (Rosseeuw, 1987). Misalkan terdapat  $N$  buah titik pada suatu himpunan data, terdapat pula di dalamnya *cluster*  $p$  dan *cluster*  $q$  dengan  $x_i$  adalah titik pada *cluster*  $p$  dan  $y_j$  adalah titik pada *cluster*  $q$ , sehingga  $a_{p,i}$  adalah rata-rata jarak titik  $x_i$  ke setiap titik pada *cluster*  $p$ , dan  $d_{q,i}$  adalah rata-rata jarak titik  $x_i$  ke setiap titik pada *cluster*  $q$ . Maka rumus perhitungan indeks validitas Silhouette dapat dilihat pada Persamaan 3.5.

$$\begin{aligned}
 \text{SIL} &= \frac{1}{N} \sum_{i=0}^N s_{x_i} \\
 s_{x_i} &= \frac{(b_{q,i} - a_{p,i})}{\max \{a_{p,i}, b_{q,i}\}}, p \neq q, \\
 b_{q,i} &= \min d_{q,i} : q = 1, \dots, k, \\
 a_{p,i} &= \frac{1}{n_q} \sum_{j=1}^{n_q} d(x_i, y_j), \\
 a_{p,i} &= \frac{1}{n_p} \sum_{k=1}^{n_p} d(x_i, x_k).
 \end{aligned} \tag{3.5}$$

### 3.16.2 Davies-Bouldin Index

Indeks validitas Davies-Bouldin (DB) menghitung rata-rata nilai setiap titik pada himpunan data. Perhitungan nilai setiap titik adalah jumlah nilai *compactness* yang dibagi dengan jarak antara kedua titik pusat *cluster* sebagai *separation*.

Jumlah *cluster* terbaik ditunjukkan dengan nilai DB yang semakin kecil (Davies & Bouldin, 1979). Misalkan terdapat suatu himpunan data dengan  $k$  buah *cluster*, terdapat  $n_p$  buah titik pada *cluster*  $p$  dan  $n_q$  buah titik pada *cluster*  $q$  dengan titik pusatnya masing-masing adalah  $c_p$  dan  $c_q$ , sehingga  $M_{pq}$  adalah jarak antara titik pusat *cluster*  $p$  dan *cluster*  $q$ ,  $S_p$  dan  $S_q$  berturut-turut merupakan rata-rata jarak setiap titik pada *cluster*  $p$  dan  $q$  ke titik pusatnya pada *cluster* yang terkait, yaitu  $c_p$  dan  $c_q$ , dengan perhitungan indeks validitas DB dapat dilihat pada Persamaan 3.6.

$$DB = \frac{1}{k} \sum_{p=1}^k R_p,$$

$$R_p = \max R_{p,q}, p \neq q,$$

$$R_{p,q} = \frac{(S_p + S_q)}{M_{pq}},$$

$$S_p = \frac{1}{n_p} \sum_{i=1}^{n_p} d(x_i, c_p), \dots \quad (3.6)$$

$$S_q = \frac{1}{n_q} \sum_{j=1}^{n_q} d(y_j, c_q),$$

$$M_{pq} = d(c_p, c_q).$$

### 3.16.3 Calinski Harabasz Index

Indeks validitas Calinski-Harabasz (CH) menghitung perbandingan antara nilai *Sum of Square between cluster* (SSB) sebagai separation dan nilai *Sum of Square within cluster* (SSW) sebagai *compactness* yang dikalikan dengan faktor normalisasi, yaitu selisih jumlah data dengan jumlah *cluster* dibagi dengan jumlah *cluster* dikurang satu. Jumlah *cluster* terbaik ditunjukkan dengan semakin besar nilai CH (Baarsch & Celebi, 2012). Misalkan terdapat suatu himpunan data dengan  $k$  buah *cluster* dan  $N$  buah titik data, misal  $C_l$  adalah *cluster* ke -  $l$  dengan  $x_i$  adalah titik ke -  $i$  pada *cluster* ke -  $l$ ,  $N_l$  adalah jumlah titik pada *cluster* ke -  $l$ , dan  $\bar{x}_l$  adalah titik pusat *cluster* ke -  $l$ , maka perhitungan indeks validitas CH dapat dilihat pada Persamaan 3.7.

$$CH = \frac{trace(SSB)}{trace(SSW)} \times \frac{N - k}{k - 1},$$

$$SSW = \sum_{l=1}^k \sum_{x_i \in C_l} (x_i - \bar{x}_l)(x_i - \bar{x}_l)^T, \quad \dots \dots \dots (3.7)$$

$$SSB = \sum_{l=1}^k N_l (\bar{x}_l - \bar{x})(\bar{x}_l - \bar{x})^T,$$

$$\bar{x}_l = \frac{1}{N_l} \sum_{x_i \in C_l} x_i,$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i.$$

# **BAB 4**

---

## **PENGENALAN PYTHON**

---

### **4.1 Apa itu Python?**

Python merupakan sebuah bahasa pemrograman yang interpretative karena dianggap memiliki kemudahan untuk dipelajari serta memiliki fokus terhadap keterbacaan kode. Dapat dikatakan, bahwa python merupakan bahasa pemrograman yang memiliki kode-kode pemrograman yang jelas, mudah dipahami, serta lengkap.

Python sering disebut sebagai bahasa pemrograman yang multi-paradigma. Dikarenakan python merupakan bahasa pemrograman yang berorientasi objek, cukup imperatif, serta memiliki fungsional yang tinggi.

Bahasa pemrograman python merupakan bahasa pemrograman yang cukup populer dan banyak digunakan oleh kalangan *engineer* yang ada di

seluruh dunia untuk membuat perangkat lunak. Bahkan beberapa python juga dijadikan sebagai bahasa pemrograman yang digunakan oleh berbagai perusahaan besar seperti Google, NASA, Instagram, YouTube, dan Spotify.



Gambar 4.1 Logo Bahasa Pemrograman Python

Kemudian, python juga banyak digunakan untuk membuat berbagai program, baik itu program yang berbasis CLI maupun berbasis GUI. Selain itu python juga dapat digunakan untuk membuat Aplikasi Mobile, Web, IoT, Game, hingga program yang diperuntukan untuk *hacking*.

Python dapat digunakan untuk berbagai sistem operasi di berbagai platform baik itu Windows, Linux, Java Virtual Machine, OS/2, Amiga, Palm,

Symbian, Sun solaris maupun MacOS dan bahkan saat ini bahasa pemrograman python juga tersedia untuk mikroprosesor seperti Raspberry-Pi. Python juga dapat digunakan untuk menangani pekerjaan di berbagai bidang yang memerlukan bantuan bahasa pemrograman seperti untuk berbagai keperluan pengembangan perangkat lunak yang berjalan di berbagai sistem operasi yang telah disebutkan tersebut. Hal tersebut menjadikan distribusi aplikasi yang dibuat dengan menggunakan bahasa pemrograman python sangat luas serta *multi-platform*.

Walaupun python tidak cukup populer seperti bahasa pemrograman lainnya namun dibalik hal tersebut python memiliki berbagai kelebihan serta dapat dijadikan salahsatu bahasa pemrograman yang layak untuk dipelajari. Bahasa pemrograman python juga dijadikan sebagai salahsatu bahasa pemrograman resmi yang digunakan oleh perusahaan Google.

## 4.2 Sejarah Python

Python merupakan sebuah bahasa pemrograman yang dibuat dan dikembangkan pertama kali oleh Guido van Rossum yang merupakan seorang programmer asal Belanda pada tahun 1991 di Stichting Mathematisch Centrum (CWI), Amsterdam yang mana python ini merupakan kelanjutan dari bahasa pemrograman ABC yang telah dibuat sebelumnya. Kemudian pada tahun 1995, Guido pindah ke CNRI di Virginia Amerika dan pengembangan dari python terus dilanjutkan. Nama python diambil bukan dari nama seekor ular, melainkan karena Guido memiliki kecintaan terhadap acara televisi Monty Python's Flying Circus, yang kemudian diambilah nama python untuk bahasa pemrograman yang dibuat oleh Guido tersebut.

Perkembangan python lambat laun semakin populer dengan berbagai kelenihan yang dimiliki serta kemudahan untuk dipelajari. Pengguna pyhton dari tahun ke tahun menunjukkan peningkatan serta perkembangan python dari awal dibuat hingga saat ini sangat signifikan.

Berikut ini merupakan perkembangan *release version* dari python mulai dari awal kemudian python versi 2 hingga python versi 3.

Tabel 4.1 *Release Version Python*

No	Version Name	Release
1	Python 1.4	25 October 1996
2	Python 1.5	17 February 1998
3	Python 1.6	5 September 2000
4	Python 2.0	16 October 2000
5	Python 2.1	15 April 2001
6	Python 2.2	21 December 2001
7	Python 2.3	29 July 2003
8	Python 2.4	30 November 2004
9	Python 2.5	19 September 2006
10	Python 2.6	1 October 2008

Lanjutan Tabel 4.1 *Release Version Python*

No	Version Name	Release
11	Python 2.7	4 July 2004
12	Python 3.0	3 December 2008
13	Python 3.1	27 June 2009
14	Python 3.2	20 February 2011
15	Python 3.3	29 September 2012
16	Python 3.4	16 March 2014
17	Python 3.5	13 September 2015
18	Python 3.6	23 December 2016
19	Python 3.7	27 June 2018
20	Python 3.8	14 October 2019

Versi python yang ditulis pada buku ini hanya sebagian besar untuk memberikan gambaran terkait dengan perkembangan dari bahasa pemrograman python. Untuk mendapatkan informasi yang lebih lengkap terkait dengan *release version* dari python beserta *documentation* dapat dilihat pada website resmi python atau melalui tautan berikut <https://www.python.org/doc/versions/>.

### 4.3 Mengapa Harus Menggunakan Python?

Bahasa pemograman yang ada saat ini secara garis besar dibagi menjadi menjadi 2 kelompok. Kelompok pertama yaitu, bahasa pemrograman dengan *compiled* atau bahasa pemrograman yang bekerja dengan menggunakan

sistem *compile* yaitu proses konversi dari *source code* ke dalam *binary code* agar dapat dimengerti dan diekseskusi oleh mesin seperti bahasa C, C++, VB dan lain sebagainya. Kemudian kelompok yang kedua yaitu bahasa pemrograman dengan *interpreted* atau bahasa pemrograman yang tidak membutuhkan proses *compile* seperti halnya bahasa pemrograman python. Namun, di dalam python proses dilakukan dengan mengubah *source code* ke dalam format dalam bentuk *intermediate* sehingga dapat berjalan di atas platform dari sistem operasi sebuah komputer.

Python di dalam bahasa pemrograman dikategorikan sebagai bahasa pemrograman tingkat tinggi. Hingga saat ini, python merupakan bahasa pemrograman yang sangat direkomendasikan untuk dipelajari, karena python merupakan bahasa pemrograman yang memiliki kehebatan untuk menangani pembuatan berbagai macam aplikasi yang banyak dibutuhkan saat ini yang mencakup berbagai permasalahan baik itu terkait dengan *big data*, *data mining*, *deep learning*, *data science*, maupun *machine learning*.

Hal tersebut menjadikan python merupakan bahasa pemrograman yang memiliki tingkat kesulitan terbilang rendah, sederhana, serta simpel untuk dipelajari serta digunakan dimana beberapa keunggulan tersebut menjadikan python cocok digunakan untuk membuat aplikasi yang berkaitan dengan kecerdasan buatan (*artificial intelligent*). Kemudian python juga memiliki fitur yang menjadikan python ini sebagai bahasa pemrograman yang dinamis yaitu manajemen memori otomatis. Kemudian beberapa alasan lain yang menjadikan bahasa pemrograman python layak dipelajari dan digunakan adalah sebagai berikut.

1. Mudah untuk dipelajari. Perintah dan sintaks yang digunakan dalam python cukup singkat dan sebagian seperti dalam bahasa inggris. Sehingga memudahkan untuk memahami masing-masing perintah yang digunakan dibandingkan bahasa pemrograman lain.

2. *Open source.* Dalam membuat aplikasi menggunakan python kita tidak perlu khawatir dengan lisensi karena lisensi python dapat digunakan oleh siapapun atau *open source*. Untuk lebih lengkap, akan dijelaskan pada subbab berikutnya terkait dengan lisensi dari python.
3. Powerfull. Python merupakan bahasa pemrograman yang dapat digunakan untuk membuat berbagai macam aplikasi dari mulai desktop, network, mobile, hingga website. Selain itu python juga sangat dikenal karena dapat digunakan untuk membuat aplikasi *hacking*.
4. *Portable* serta dapat digunakan di berbagai sistem operasi.
5. Memiliki dukungan komunitas yang aktif sehingga menjadi developer dapat dengan mudah untuk mencari dokumentasi serta bertanya apabila memiliki permasalahan ketika membuat program dengan menggunakan python.
6. Python juga dapat digabungkan dengan bahasa pemrograman lain seperti Visual Basic Net atau VB .Net.

#### 4.4 Lisensi Python

Setiap bahasa pemrograman yang ada pasti memiliki sebuah lisensi. Lisensi merupakan sebuah izin yang harus dimiliki oleh seseorang untuk menggunakan, mendistribusikan, serta memodifikasi suatu bahasa pemrograman tidak terkecuali python. Pada dasarnya, python dapat digunakan serta dimiliki oleh siapa saja secara bebas, bahkan untuk kepentingan komersial. Namun, dalam hal penggunaan *packages* atau modul-modul dari python itu sendiri merupakan hasil pengembangan dari pihak ketiga, oleh karena itu dimungkinkan memiliki lisensi yang berbeda, seperti lisensi yang berbayar.

## 4.5 Kelebihan dan Kekurangan Python

Berbagai bahasa pemrograman pasti memiliki kelebihan dan kekurangan. Hal tersebut tidak terlepas dari bahasa pemrograman python. Berikut ini merupakan kelebihan dan kekurangan dari bahasa pemrograman python.

### Kelebihan Python

1. Python pada saat ini merupakan bahasa pemrograman cukup populer. Per November 2019, python menempati urutan ke-2 bahasa pemrograman yang paling populer di dunia.
2. Python cukup mudah untuk digunakan dan dipelajari disbanding bahasa pemrograman lainnya. Memiliki sintaks yang sederhana, mudah diingat serta dipahami karena hal tersebut diperoleh dari filosofi python itu sendiri yaitu menekankan pada aspek kemudahan untuk dibaca (*readibility*). Dikarenakan *source code* yang dimiliki oleh python memiliki kemudahan dalam penulisan dan pembacaan, sehingga dapat lebih mempermudah melakukan perbaikan apabila terjadi kesalahan dalam proses pembuatan aplikasi dan memudahkan dalam pemeliharaan.
3. Python merupakan bahasa pemrograman yang memiliki berbagai fungsi (multifungsi). Oleh karena itu, bahasa pemrograman python dapat digunakan untuk membuat berbagai macam produk aplikasi baik itu aplikasi website, game, robotika, *data mining*, sampai aplikasi berbasis kecerdasan buatan. Selain itu aplikasi desktop dan mobile pun bias dibuat dengan menggunakan python.
4. Dalam hal penulisan, python memiliki baris kode yang lebih efisien disbanding bahasa pemrograman lain. Penulisan sebuah perintah yang terdiri dari 5 baris kode dalam bahasa pemrograman lain bisa disederhanakan oleh python hanya menjadi 1 baris saja. Hal tersebut menjadikan waktu pembuatan aplikasi menggunakan python menjadi lebih

cepat dan ringkas dibanding dengan menggunakan bahasa pemrograman lain.

5. Python memiliki dukungan *library* (pustaka) standar yang banyak. Kemudian *packages* yang disediakan untuk mendukung kebutuhan pembuatan program pun tersedia sangat banyak baik itu dibuat oleh pengembang *official* dari python maupun pengembang dari pihak ketiga.
6. Dalam penggunaannya python dapat dikolaborasikan dengan bahasa pemrograman lain.
7. Python juga memiliki dukungan yang baik untuk pengembangan ekosistem dari *Internet of Things*. Saat ini banyak sekali sistem yang mengusung konsep *Internet of Things* dan dibangun dengan menggunakan python. Kemudian python juga digunakan sebagai basis dari bahasa pemrograman *board* yang digunakan untuk menjalankan sistem *Internet of Things* termasuk diantaranya yaitu Raspberry-pi.
8. Memiliki sistem pengelolaan memori yang dilakukan secara otomatis seperti halnya pada Java.
9. Python merupakan bahasa pemrograman yang berorientasi baik secara prosedural maupun objek sekaligus.
10. Memiliki sifat modular sehingga mudah untuk dikembangkan dan membuat modul baru.

## Kekurangan Python

1. Terdapat beberapa tugas yang tidak dapat dilakukan dan berada diluar jangkauan kemampuan dari python. Walaupun dikatakan python merupakan bahasa pemrograman yang dinamis, namun python tidak secepat atau efisien sebagai statis.
2. Python dapat digunakan untuk pembuatan aplikasi mobile. Namun untuk pengembangan kedepan, penggunaan python cukup buruk untuk aplikasi mobile.

3. Python bertindak sebagai interperete, bukai sebagai alat bantu terbaik untuk pengantar komponen kinerja kritis. Sehingga python bukan merupakan pilihan yang baik untuk melakukan tugas-tugas intensif memori.
4. Python memiliki keterbatasan dalam membuat game, sehingga hampir mustahil membuat game 3D dengan spesifikasi grafis yang tinggi dengan menggunakan python.
5. Terbatasnya akses terhadap basis data.
6. Python tidak bisa digunakan untuk dasar sebuah bahasa pemrograman implementasi yang digunakan untuk beberapa komponen, namun masih dapat digunakan dengan baik sebagai bagian depan *script interface*.
7. Tingkat efisiensi dan *flexibility trade off* tidak diberikan secara menyeluruh pada python.

## 4.6 Cakupan Aplikasi Python

Aplikasi yang dibuat dengan menggunakan bahasa pemrograman python dapat mencakup atau dapat digunakan untuk melaksanakan tugas yang berbeda-beda.

Berikut ini beberapa cakupan atau area penggunaan aplikasi yang dibuat dengan menggunakan bahasa pemrograman python.

1. Perangkat bantu shell.
2. Melaksanakan tugas-tugas yang biasa dilakukan oleh sistem administrator.
3. Program dalam bentuk extension.
4. Digunakan sebagai *interface* untuk *library C/C++*.
5. Pembuatan sistem aplikasi dan prototype yang membutuhkan waktu penggerjaan yang cepat.
6. Untuk prototype yang fleksibel, sehingga dapat dimodifikasi baik itu dikurangi atau ditambah fitur sesuai dengan permintaan.
7. Pembuatan modul sesuai dengan bahasa pemrograman.

8. Sebagai pengganti untuk penulisan parse khusus.
9. Untuk membuat sebuah API berbasis GUI yang sederhana namun canggih.
10. Pengkasesan terhadap database.
11. Aplikasi pemrograman yang terdistribusi.
12. Penerapan untuk mekanisme API *client-server* yang terintegrasi.
13. Penggunaan untuk *script* internet CGI, antarmuka HTTP, Aplet dan lain sebagainya.

## 4.7 Instalasi Python

Instalasi merupakan sebuah proses pemasangan sebuah program atau *software*, setiap perangkat lunak yang akan digunakan dalam sebuah komputer harus diinstal terlebih dahulu untuk bisa digunakan di dalam suatu sistem operasi. Selain itu fungsi lain dari instalasi adalah untuk penyesuaian antara program dengan peralatan atau komponen yang terdapat dalam komputer. Python dapat diperoleh secara gratis. Ketika buku ini ditulis versi python terbaru yang tersedia berada pada versi 3.8. pada buku ini sistem operasi yang digunakan adalah Windows, sehingga semua *interface* dan langkah-langkah instalasi python akan disesuaikan dengan lingkungan Windows. Dalam melakukan instalasi python ada beberapa hal yang perlu dipersiapkan diantaranya adalah sebagai berikut.

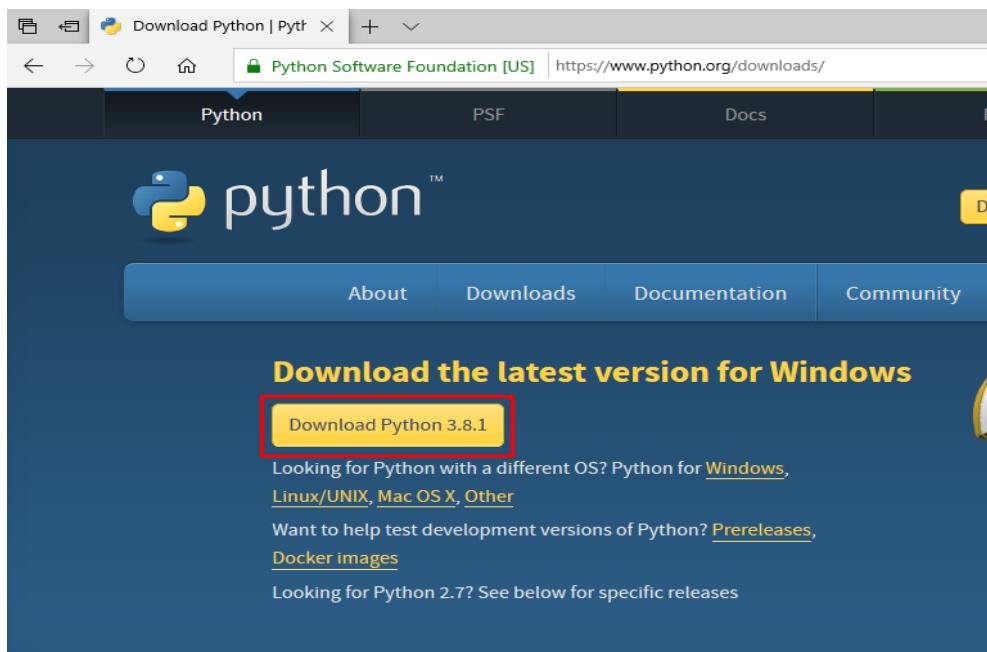
1. Python yang merupakan interpreter untuk menerjemahkan bahasa pemrograman python ke dalam bahasa mesin sehingga program dapat dijalankan.
2. Teks editor atau IDE yang nantinya akan digunakan untuk menulis kode dalam bahasa pemrograman python

Proses instalasi python yang dilakukan pada sistem operasi Windows cukup mudah, dikarenakan cara instalasi nya hampir sama dengan cara menginstall software pada umumnya di Windows.

Ada beberapa cara yang dapat dilakukan untuk melakukan instalasi python. Berikut ini langkah-langkah yang dapat dilakukan untuk menginstal python.

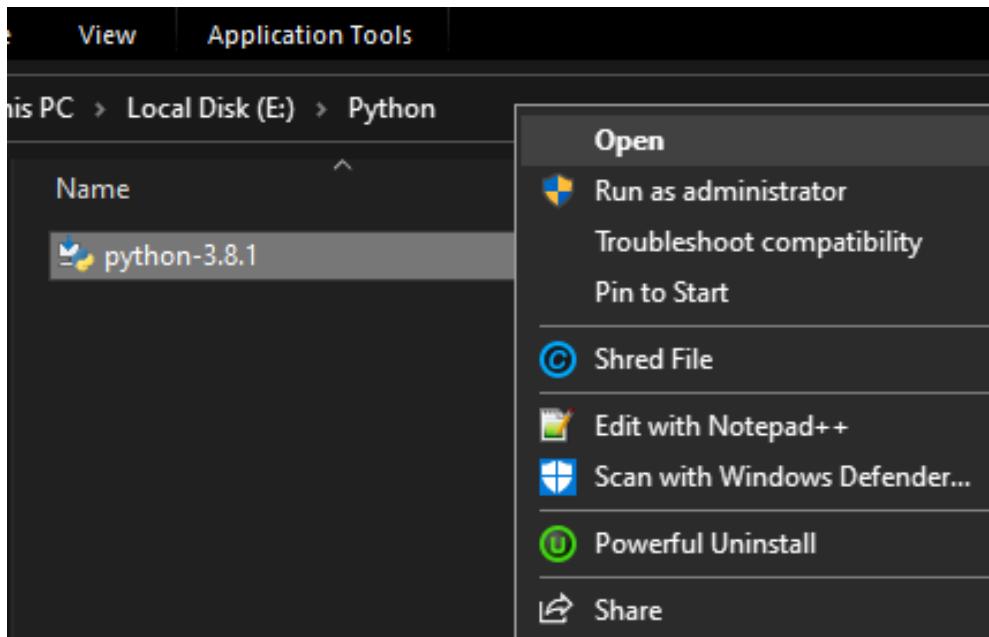
### 1. Install python secara langsung.

- Langkah awal yang dilakukan bila melakukan instalasi python dengan cara pertama adalah dengan mendownload file installer terlebih dari website *official* python yaitu <https://www.python.org/downloads/>, kemudian pilih versi dari python atau dapat menggunakan python dengan versi terbaru serta sesuaikan dengan sistem operasi dimana python tersebut dipasang, karena pada buku ini sistem operasi yang digunakan adalah Windows maka python yang dipilih merupakan python yang diperlukan untuk sistem operasi Windows.



Gambar 4.2 Download Python

- Setelah file *installer* python berhasil di download, kemudian jalankan dengan cara klik kanan pada file *installer* kemudian klik *Open* atau *double click* pada file *installer* nya.



Gambar 4.3 Eksekusi *Installer* Python

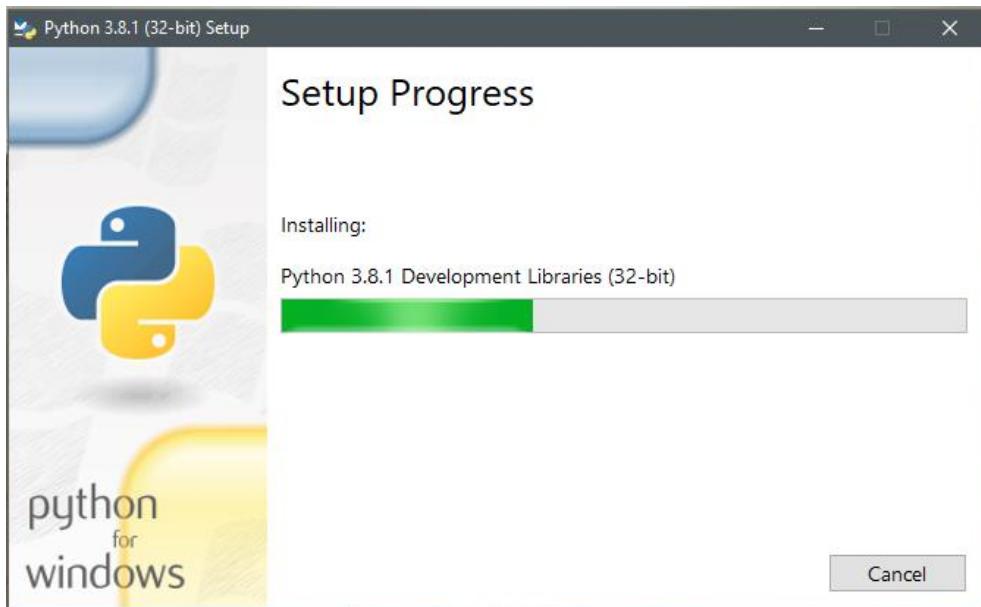
- Setelah file *installer* dieksekusi, maka akan muncul tampilan awal untuk melakukan instalasi, dalam tampilan awal yang ditunjukkan oleh Gambar 4.4 ada beberapa opsi terkait dengan setting yang menurut penulis direkomendasikan untuk diinstall atau dicentang pada opsi nya seperti opsi **Install launcher for all users** yang berfungsi untuk membuat sebuah peluncur atau *shortcut* pada menu program sehingga mempermudah saat akan menjalankan python. Kemudian opsi **Add Python 3.8 to PATH** yang berfungsi agar direktori python didaftarkan pada Environment Variables dari sistem operasi Windows sehingga direktori tersebut dapat dipanggil atau dieksekusi melalui command prompt atau CMD. Kemudian untuk melanjutkan instalasi dilakukan dengan klik pada opsi **Install Now** secara *default*, namun apabila tidak ingin menginstall secara keseluruhan dari

python opsi yang dipilih untuk melanjutkan instalasi adalah **Customize installation.**



Gambar 4.4 Tampilan Awal Installer Python

- Setelah melanjutkan instalasi, tunggu hingga proses instalasi selesai. Pada tahap ini akan dilakukan pemasangan python pada sistem operasi beserta dengan library standar yang tersedia pada setiap paket instalasi, serta penambahan direktori python pada Environment Variables.



Gambar 4.5 Proses Instalasi Python

- Apabila instalasi telah berhasil dilakukan, maka akan muncul tampilan seperti pada Gambar 4.6 beserta notifikasi yang memberitahukan bahwa proses instalasi python telah berhasil dilakukan. Selain itu terdapat pula informasi apabila membutuhkan tutorial *online* dari python beserta dengan dokumentasinya bisa mengunjungi halaman *official* python dengan cara klik pada kalimat yang di garis bawahi. Kemudian klik opsi Close untuk mengakhiri proses instalasi.



Gambar 4.6 Proses Instalasi Python Berhasil

- Setelah python berhasil diinstal maka dapat dilakukan beberapa pengujian seperti mengecek versi dari python melalui CMD.

```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 10.0.18363.592]
(c) 2019 Microsoft Corporation. All rights reserved.

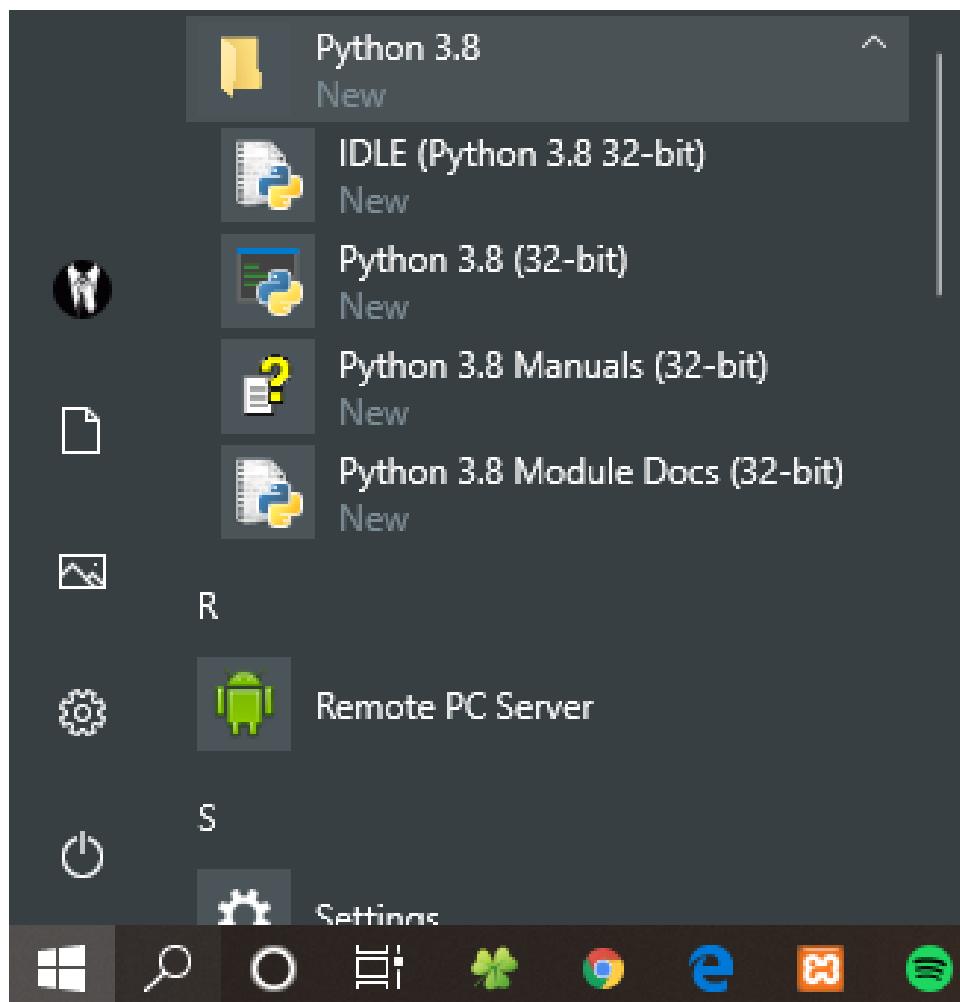
C:\Users\fikri>python
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916
32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.

>>>
```

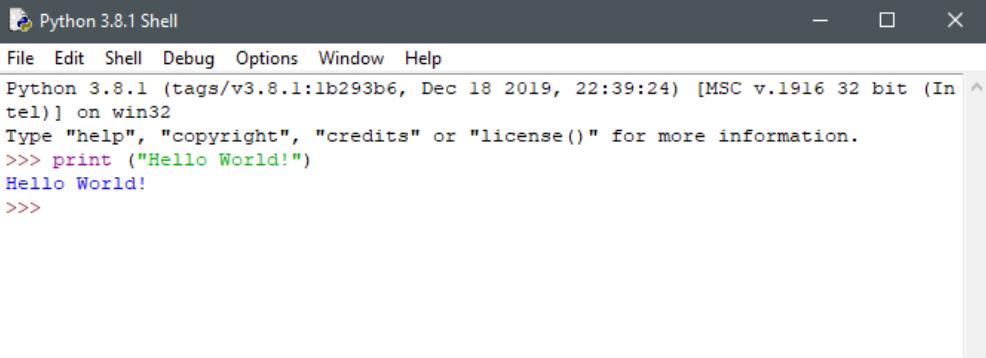
Gambar 4.7 Pengecekan Python Melalui CMD

- Setelah melakukan pengecekan melalui CMD, mengindikasikan bahwa direktori python telah ditambahkan ke PATH sehingga python dapat dipanggil dan dapat menjalankan file berekstensi python melalui CMD.
- Pengujian juga dapat dilakukan dengan membuka python shell. Dengan cara klik Start, kemudian pada all program cari folder python sesuai dengan

versinya dan pilih IDLE (Python 3.8 32-bit). Tunggu hingga jendela python shell muncul.



Gambar 4.8 Pengecekan Python Melalui Python Shell



The screenshot shows the Python 3.8.1 Shell window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the Python version information: Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (In tel)] on win32. It also shows the standard help message: Type "help", "copyright", "credits" or "license()" for more information. A user has entered the command >>> print ("Hello World!") followed by a carriage return, resulting in the output Hello World!.

Gambar 4.9 Pengujian Python Melalui Python Shell

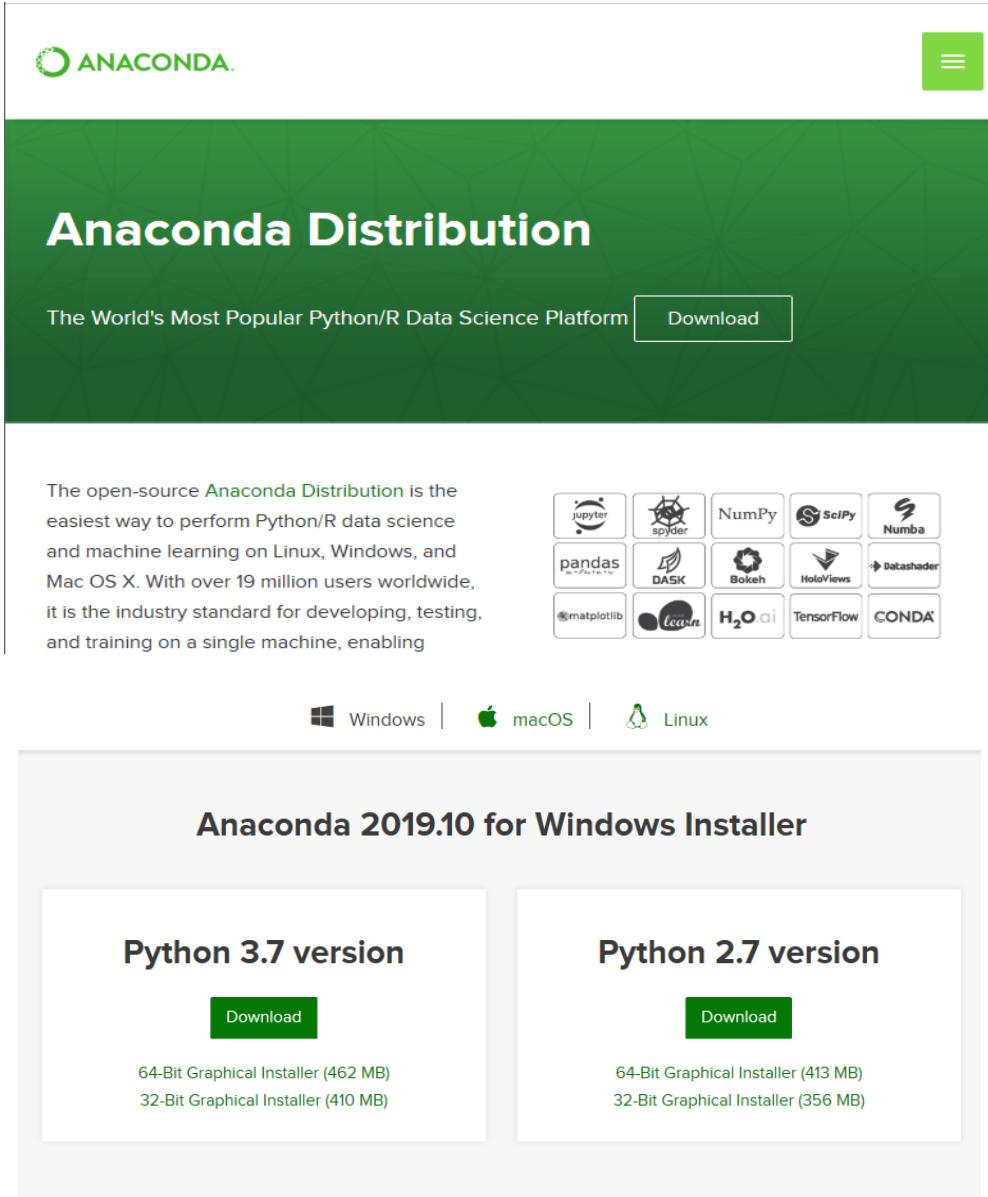
Python shell merupakan sebuah terminal yang disediakan oleh python untuk melakukan operasi sederhana seperti menampilkan teks, melakukan perhitungan sederhana dan lain sebagainya. Pada gambar 4.9 dilakukan percobaan untuk menampilkan teks dengan perintah print (“Hello World!”) kemudian dibawah perintah tersebut akan muncul hasil dari eksekusi perintahnya. Hal tersebut menandakan bahwa python yang diinstal telah siap untuk digunakan.

- Setelah python berhasil diinstal, python dapat digunakan untuk melakukan tugas seperti membuat sebuah program. Pembuatan program dapat dilakukan dengan python shell, namun untuk lebih leluasa dalam melakukan pengkodean serta melakukan pemeliharaan terhadap kode yang dibuat diharapkan untuk menginstall teks editor yang umum digunakan untuk menulis kode menggunakan bahasa pemrograman python secara terpisah dikarenakan editor tersebut tidak disediakan dalam paket instalasi dari python.
- 2. Instalasi dengan menggunakan paket distribusi yang didalamnya sudah tersedia lengkap baik itu python maupun *library* beserta editornya dengan menggunakan paket anaconda yang dapat diperoleh dari website officialnya. Anaconda sendiri tidak hanya diperuntukan untuk sistem

operasi Windows saja melainkan juga tersedia untuk sistem operasi lain seperti Linux mapun MacOS.

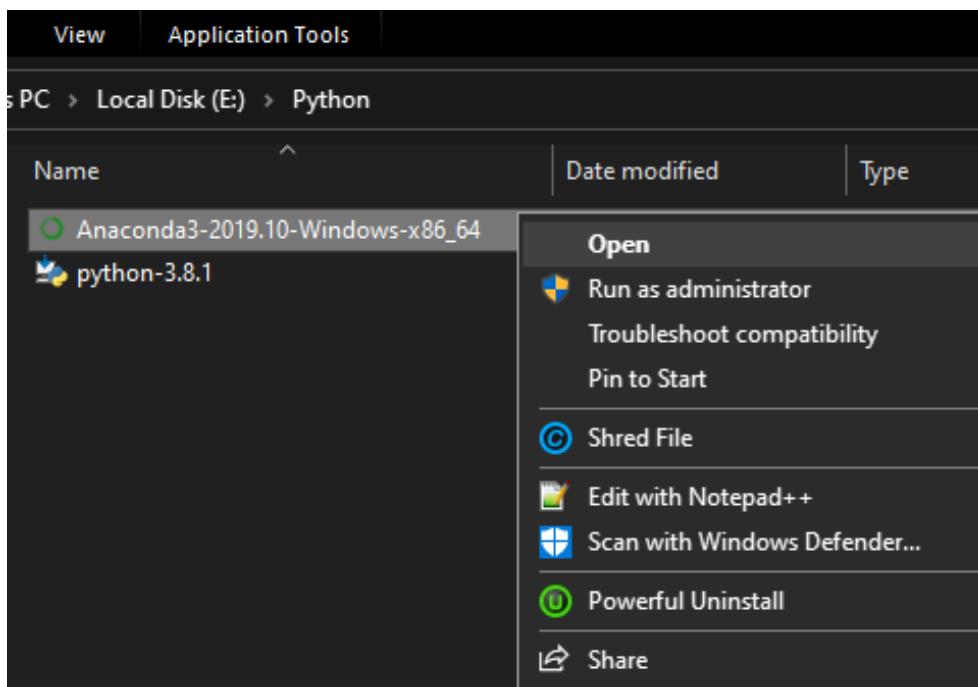
Penulis menggunakan cara kedua untuk melakukan instalasi python dikarenakan cara kedua ini memiliki banyak kelebihan dibanding cara pertama. Dengan cara kedua ini pembaca tidak harus mengintal editor secara terpisah dikarenakan editor untuk menulis kode sudah disertakan dalam paket instalasi. Anaconda merupakan sebuah paket aplikasi yang didalamnya tidak hanya berisi aplikasi python melainkan juga terdapat didalamnya editor untuk menulis kode seperti jupyter notebook atau lebih dikenal dengan istilah jupyter. Selain itu juga terdapat editor lain yang tersedia seperti spyder, sehingga memberikan keleluasan kepada pembaca untuk menulis kode menggunakan editor sesuai dengan keinginan pembaca. Anaconda juga biasanya digunakan oleh seorang *data scientist*, *it professionals*, serta kalangan *executives*. Hal tersebut dikarenakan pada anaconda terdapat banyak sekali *library* yang dapat digunakan untuk mendukung serta mempercepat mereka dalam melakukan pekerjaannya. Berikut ini merupakan cara instalasi dari anaconda.

- Download terlebih dahulu anaconda dari website *officialnya* yaitu <https://www.anaconda.com/distribution/>. Kemudian pilih versi anaconda yang akan digunakan sesuai dengan sistem operasinya. Penulis disini menggunakan anaconda dengan versi python 3.7 dan versi yang dipilih merupakan aplikasi untuk sistem operasi Windows serta sesuaikan juga versi nya dengan versi dari sistem operasi nya yaitu 32 bit ataupun 64 bit.



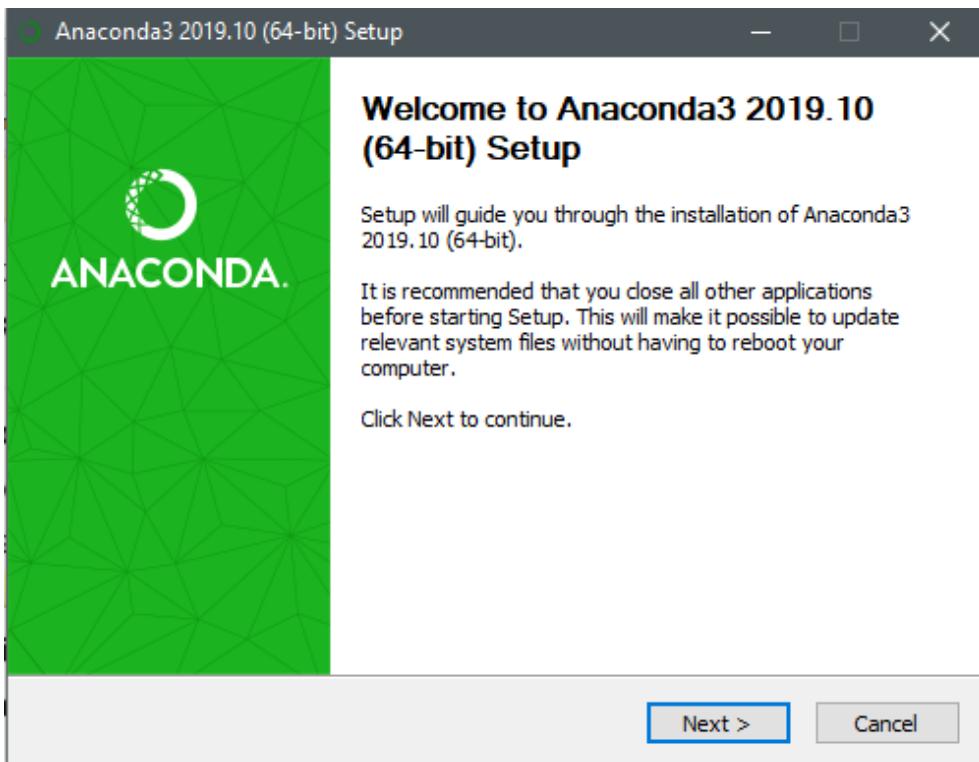
Gambar 4.10 Download Anaconda

- Setelah *installer* anaconda berhasil didownload, eksekusi *installer* tersebut dengan cara klik kanan pada file *installernya* kemudian klik *Open* atau *double click* pada file *installernya*.



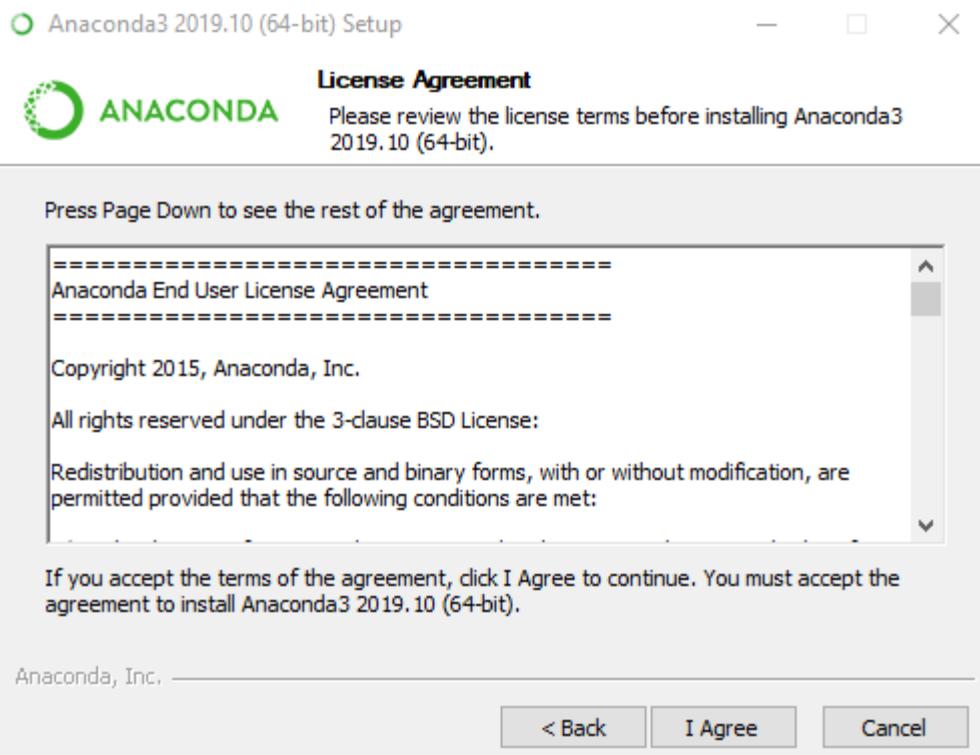
Gambar 4.11 Eksekusi *Installer* Anaconda

- Tunggu hingga muncul tampilan awal untuk melakukan instalasi anaconda.
- Setelah muncul tampilan awal proses instalasi, akan ada beberapa informasi yang tertera terkait dengan proses instalasi yang dilakukan seperti direkomendasikan untuk menutup program lain yang sedang berjalan sebelum memulai instalasi. Untuk melanjutkan proses instalasi dapat dilakukan dengan memilih opsi Next.



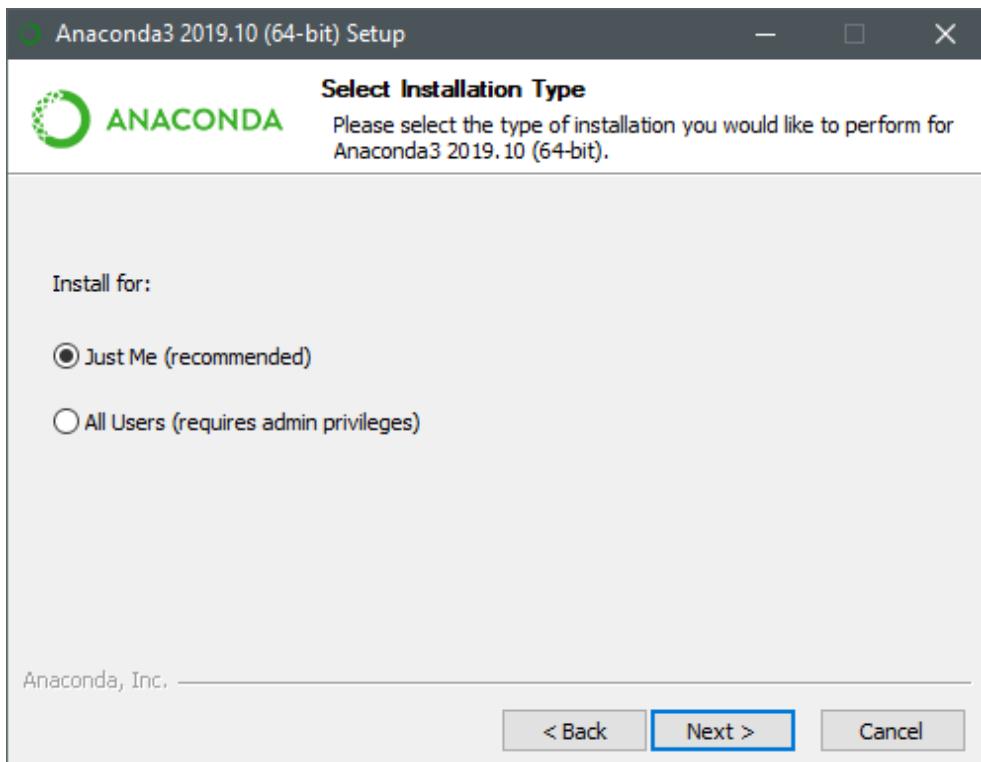
Gambar 4.12 Tampilan Awal Instalasi Anaconda

- Setelah beranjak dari tampilan awal, selanjutnya akan dihadapkan pada tampilan License Agreement. Untuk menginstall anaconda kita harus menyetujui serta mengikuti perjajian tersebut dengan cara klik **I Agree** untuk menandakan kita telah menyetujui dan akan melanjutkan proses instalasi.



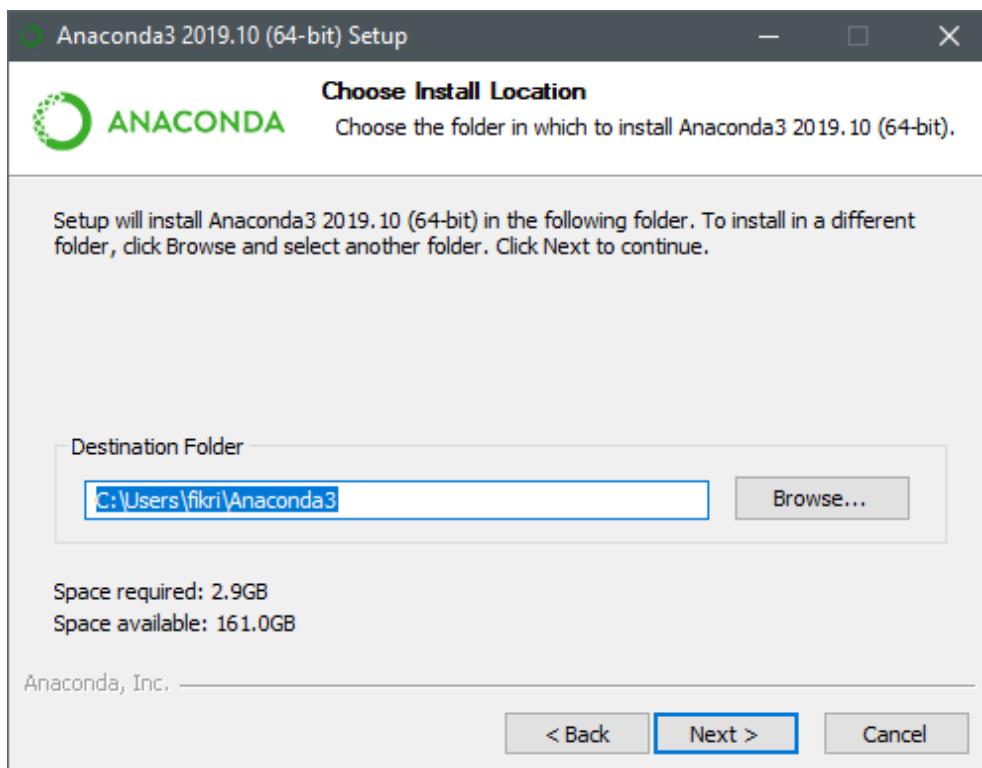
Gambar 4.13 License Agreement Anaconda

- Proses selanjutnya dari tahapan instalasi yaitu memilih tipe instalasi. Penulis disini memilih opsi **Just Me** sesuai dengan opsi yang direkomendasikan oleh anaconda. Namun apabila komputer yang akan diinstal anaconda merupakan komputer yang digunakan secara bersama-sama oleh beberapa orang yang masing-masing pengguna tersebut memiliki user login ke komputer yang berbeda serta sama-sama menggunakan anaconda maka dapat dipilih opsi All Users, agar semua pengguna yang memiliki akun login yang berbeda tersebut dapat melakukan pekerjaan dengan anaconda serta merubah konfigurasi dari anaconda sesuai dengan kebutuhan dalam pengerajan tugasnya.



Gambar 4.14 Pemilihan Tipe Instalasi Anaconda

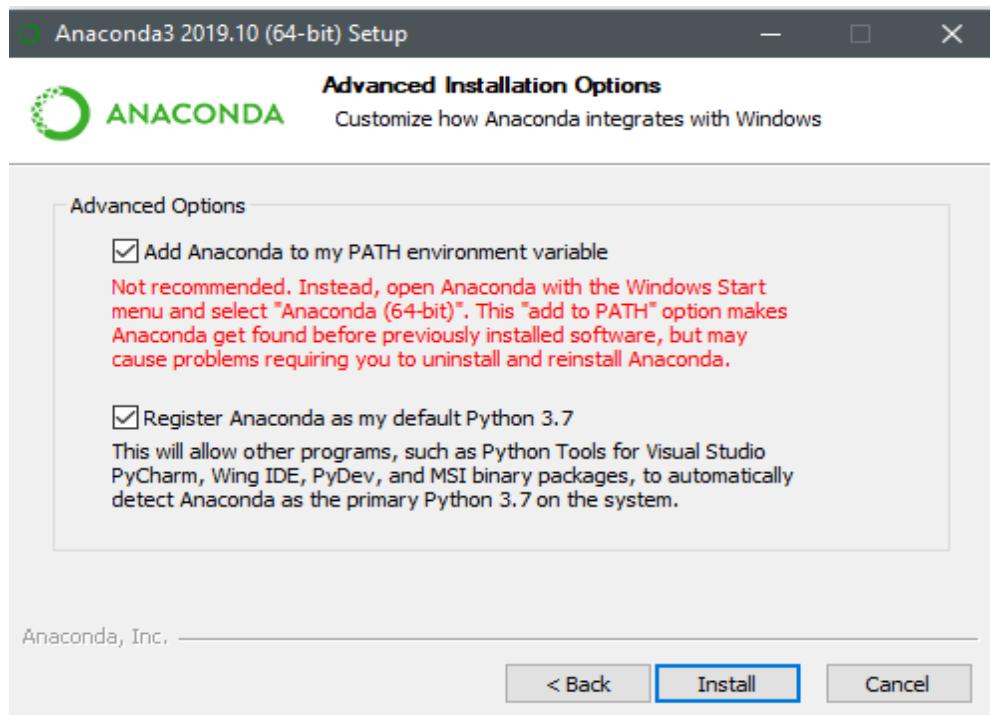
- Setelah melalui proses pemilihan tipe instalasi, selanjutnya adalah memilih lokasi untuk menyimpan hasil instalasi dari anaconda serta diinformasikan pula berapa penyimpanan yang dibutuhkan untuk menginstal anaconda serta kapasitas yang tersedia dari lokasi penyimpanan yang akan dijadikan sasaran untuk menyimpan hasil instalasi. Lokasi penyimpanan hasil instalasi yang digunakan dapat dipilih pada direktori mana saja yang masih memiliki ruang penyimpanan sesuai dengan kebutuhan dari instalasi anaconda. Pada buku ini penulis akan menyimpan hasil instalasi sesuai dengan rekomendasi dari anaconda yaitu pada local disk C pada direktori Users. Apabila telah selesai menentukan lokasi dari penyimpanan hasil instalasi klik opsi Next untuk melanjutkan.



Gambar 4.15 Pemilihan Lokasi Instalasi Anaconda

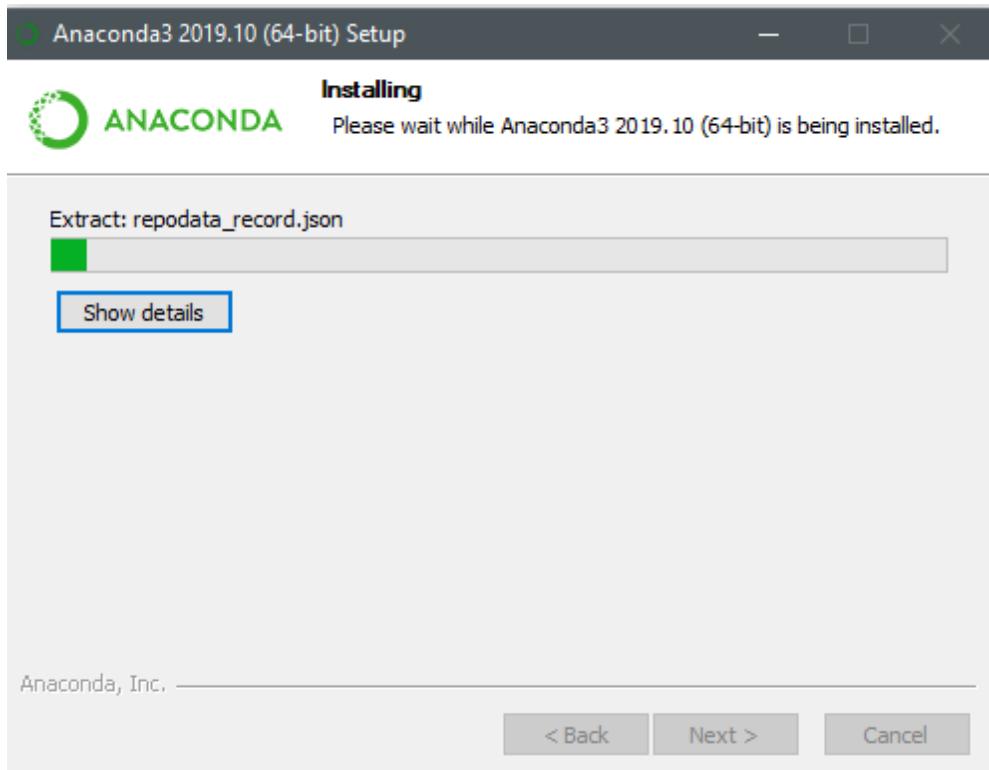
- Proses selanjutnya adalah tahapan pemilihan opsi lanjutan dari instalasi. Dimana pada tahap akan diberikan pilihan pengaturan lanjutan untuk bagaimana kustomisasi dari anaconda agar dapat terintegrasi dengan Windows. Akan terdapat beberapa opsi yaitu **Add anaconda to my PATH environment variable**, opsi ini menjadikan anaconda akan dapat dipanggil serta dieksekusi melalui CMD. Kemudian opsi kedua yaitu **Register Anaconda as my default Python 3.7**, opsi ini akan mengijinkan berbagai macam program lain seperti tools yang digunakan untuk menulis kode menggunakan bahasa pemrograman python seperti Visual Studio, PyCharm, Wing IDE, PyDev, dan MSI binary packages untuk secara otomatis mendeteksi anaconda sebagai opsi utama dari Python 3.7 yang ada dalam sistem apabila terdapat beberapa versi python dalam sistem operasi yang sama. Kedua opsi tersebut dapat dipilih ataupun tidak sesuai

dengan kebutuhan pembaca. Selanjutnya pilih opsi install untuk memulai proses pemasangan anaconda.



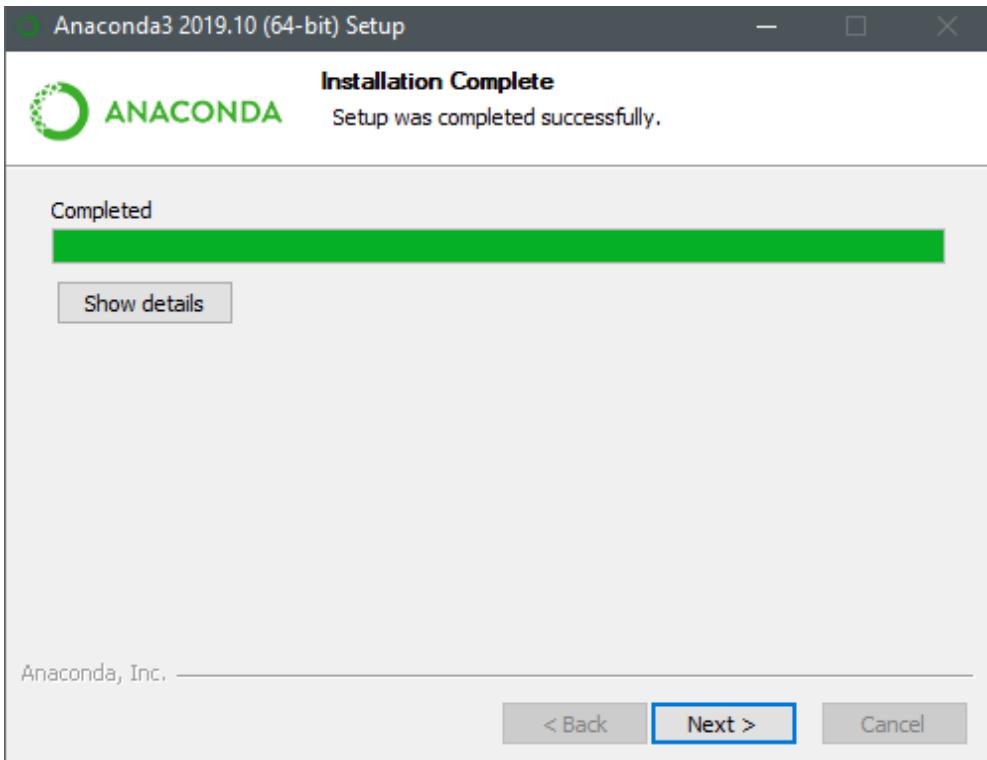
Gambar 4.16 Pemilihan Opsi Lanjutan Instalasi Anaconda

- Proses instalasi akan berjalan dan memakan waktu agak lama dikarenakan pada tahap ini terdapat proses yang dilakukan untuk melakukan penyesuaian antara fitur-fitur yang tersedia dalam aplikasi anaconda yang dipasang dengan komponen yang tersedia pada komputer. Tunggu hingga proses selesai yang ditandai dengan munculnya notifikasi instalasi berhasil dan pastikan tidak menutup jendela instalasi untuk menghindari resiko ketidakberhasilan dari proses instalasi yang sedang dilakukan.



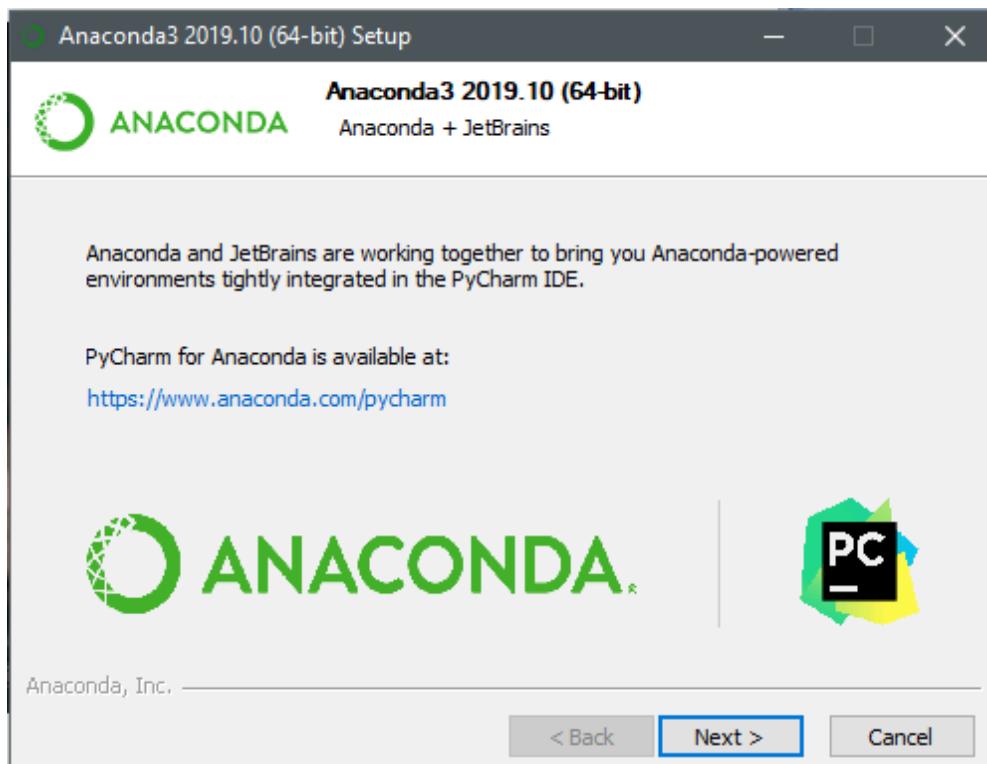
Gambar 4.17 Proses Instalasi Anaconda

- Gambar 4.18 menunjukan bahwa instalasi dari anaconda telah berhasil dan diapasang secara lengkap. Kemudian klik opsi Next untuk melanjutkan.



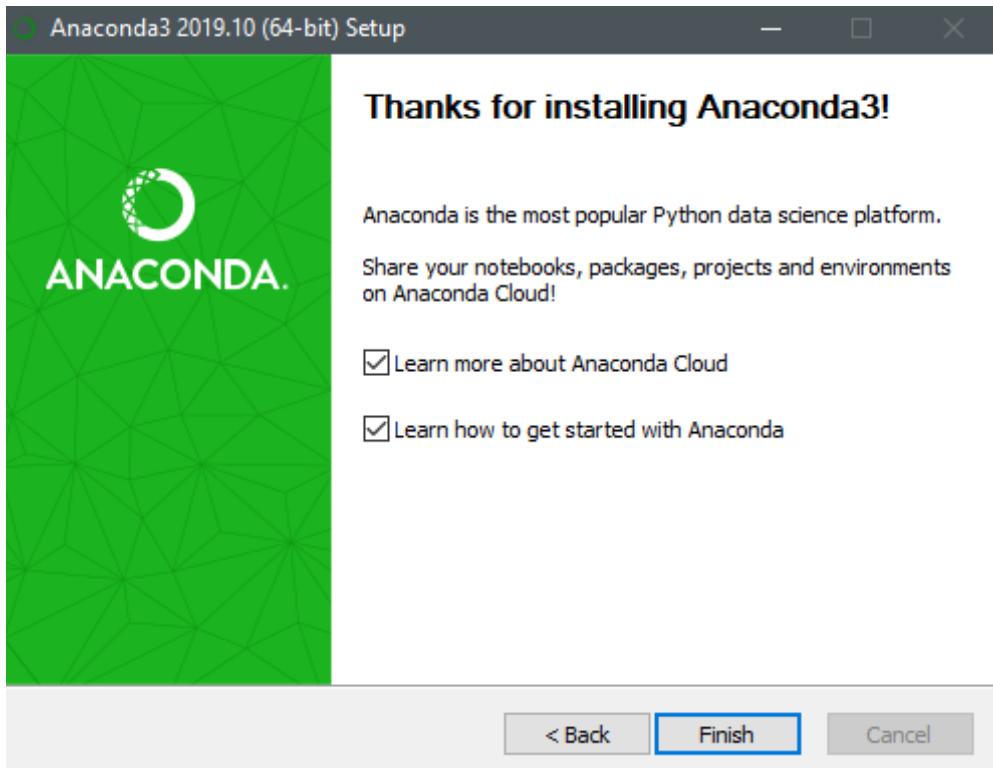
Gambar 4.18 Notifikasi Proses Instalasi Anaconda Selesai

- Berikut ini merupakan pemberitahuan mengenai perusahaan Anaconda dan Jetbrains melakukan kerjasama dalam hal membangun sebuah environment yang dapat dikombinasikan antara anaconda dengan editor untuk menulis kode menggunakan bahasa pemrograman python yaitu PyCharm IDE sehingga menjadikan pekerjaan yang dilakukan menjadi lebih powerful. PyCharm untuk anaconda dapat diperoleh dengan mengunjungi website dari anaconda atau melalui tautan berikut <https://www.anaconda.com/pycharm>. Untuk melanjutkan, pilih opsi Next.



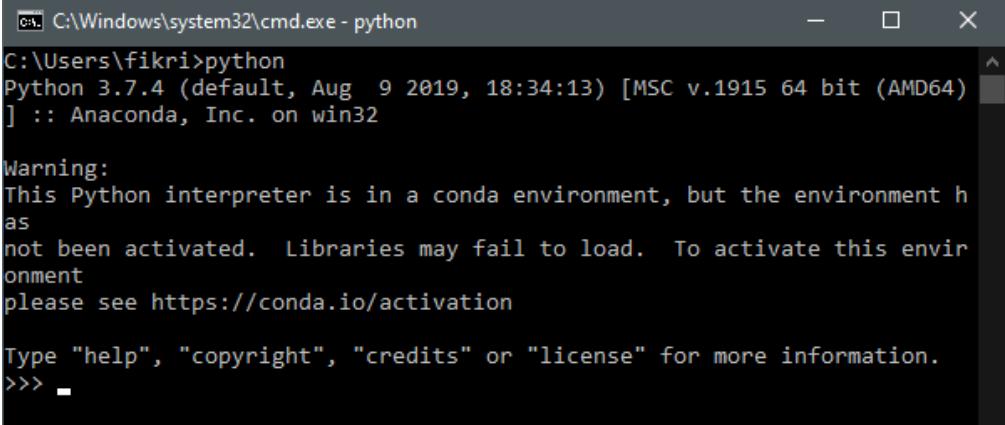
Gambar 4.19 Notifikasi Rekomendasi Editor Untuk Menulis Kode

- Pada tahap ini merupakan tahap akhir dari seluruh rangkaian instalasi anaconda, pada halaman terdapat informasi ucapan terimakasih karena telah menginstal anaconda. Kemudian terdapat pula pernyataan bahwa anaconda merupakan platform python yang sangat populer untuk digunakan dalam *data science*. Untuk mengakhiri proses instalasi dapat dilakukan dengan klik pada opsi Finish.



Gambar 4.20 Ucapan Terimakasih Telah Menginstal Anaconda

- Tahap selanjutnya setelah proses instalasi adalah melakukan pengecekan apakah anaconda telah berhasil diinstall dan python dapat dipanggil melalui CMD. Pada CMD akan terlihat informasi versi dari python yang digunakan serta versi dari anaconda.



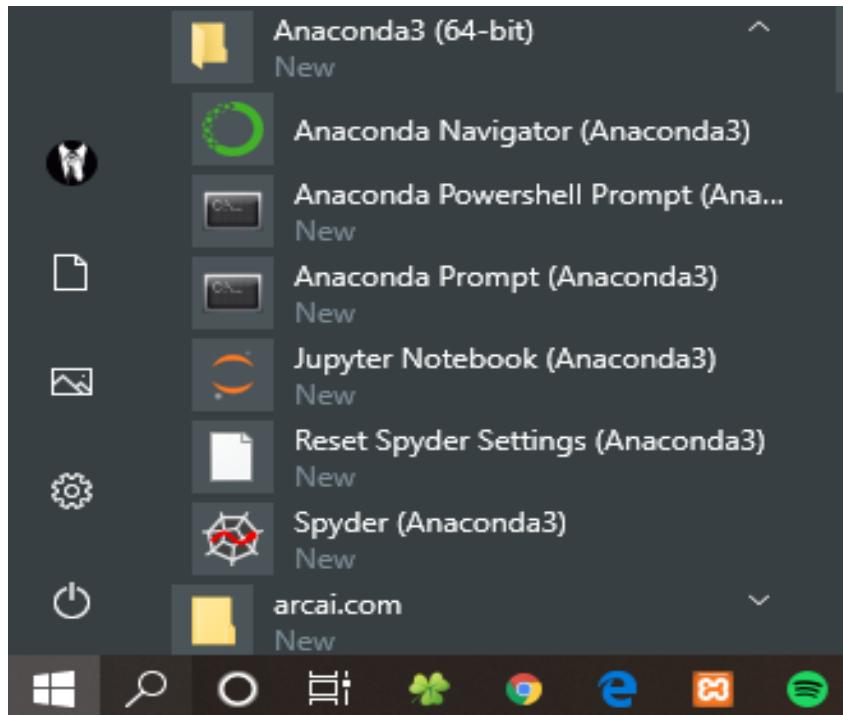
```
C:\Windows\system32\cmd.exe - python
C:\Users\fikri>python
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>>
```

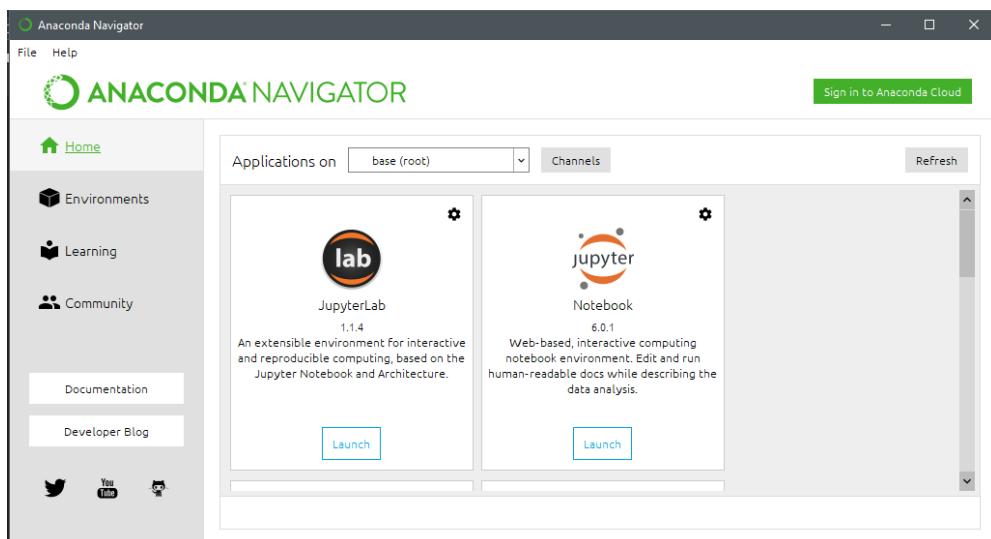
Gambar 4.21 Pengecekan Anaconda Melalui CMD

- Untuk mengakses navigator dari anaconda dapat dilakukan dengan klik pada start, kemudia pada all program cari folder Anaconda3 kemudian pilih Anaconda Navigator.



Gambar 4.22 Cara Menampilkan Anaconda Navigator

- Berikut ini merupakan tampilan dari anaconda navigator. Pada Anaconda Navigator ini merupakan area dari lingkungan kerja anaconda dimana dalam paket instalasi tersedia editor untuk menulis kode dengan menggunakan bahasa pemrograman python yaitu jupyter notebook dan spyder yang dapat diakses salahsatunya dengan melalui anaconda navigator selain dengan melalui launcher secara langsung yang tersedia pada folder hasil instalasi anaconda yang dapat ditemui pada all program.



Gambar 4.23 Tampilan Awal Anaconda Navigator

## 4.8 Jupyter Notebook

Penjelasan terkait dengan salahsatu editor untuk menulis kode menggunakan bahasa pemrograman python yaitu jupyter notebook dibuat dalam bahasan yang terpisah, guna lebih memahami apa itu jupyter notebook serta bagaimana penggunaannya dikarenakan pada bab selanjutnya akan dibahas bagaimana cara pemodelan segmentasi pelanggan produk digital pada perusahaan telekomunikasi dengan bahasa pemrograman python dan editor yang digunakan yaitu jupyter notebook.

Jupyter merupakan sebuah organisasi non-profit yang bergerak di bidang pengembangan software interktif untuk berbagai bahasa pemrograman. Notebook merupakan salahsatu dari software buatan jupyter.

Jupyter notebook merupakan sebuah aplikasi editor untuk menulis kode dengan menggunakan bahasa pemograman python dalam bentuk website yang berada pada localhost di komputer kita dimana *source code* dan penjelasan dari masing-masing perintah dapat dibuat dalam satu halaman yang memungkinkan untuk interaksi secara interaktif, kemudian hasil pekerjaan yang telah dilakukan pada jupyter notebook tersebut disimpan dalam bentuk yang menarik dengan file berekstensi ipynb.

Jupyter notebook merupakan pengembangan dari IPython yang lahir pada tahun 2014 yang memiliki lisensi gratis atau digunakan dan dirilis oleh siapa saja namun haru berada atau mengikuti persyaratan dari lisensi modifikasi BSD, yang mana IPython bertindak sebagai kernel sedangkan Jupyter menggunakan antarmuka Notebook untuk interface dari aplikasi editornya. Jupyter mengembangkan produknya secara terbuka di *platform* github yang didukung oleh komunitas jupyter sehingga pengembangan dari jupyter sangat cepat, selain itu dukungan dari komunitas tersebut menjadikan dokumentasi dari jupyter itu sangat banyak sehingga memudahkan untuk mencari solusi untuk permasalah yang terjadi apabila sedang menggunakan jupyter. Tidak hanya digunakan untuk menulis kode, jupyter notebook juga dapat digunakan untuk menulis *rich text element* seperti sebuah paragraph, persamaan matematika, bahkan bisa juga menampilkan gambar dan tautan. Walaupun ditulis menggunakan bahasa pemrograman python namun jupyter notebook juga dapat digunakan untuk menulis kode dengan menggunakan bahasa pemrograman lain seperti bash, C, C++, C#, bahkan Java serta masih banyak bahas pemrograman lainnya, kemampuan tersebut diimplementasikan secara modular dalam bentuk kernel, diman pada saat ini sudah terdapat 130 kernel lebih yang tersedia serta mendukung hampir 100 bahasa pemrograman

sehingga untuk menggunakan bahasa pemrograman lain dengan menggunakan jupyter notebook hanya perlu menginstal kernel dari bahasa pemrograman tertentu. Untuk informasi lebih lengkap terkait kernel bahasa pemrograman selain python yang telah mendukung untuk diterapkan pada jupyter notebook dapat dilihat pada akun github dari jupyter atau pada tautan <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>. Fitur utama yang tersedia pada jupyter notebook adalah sebagai berikut.

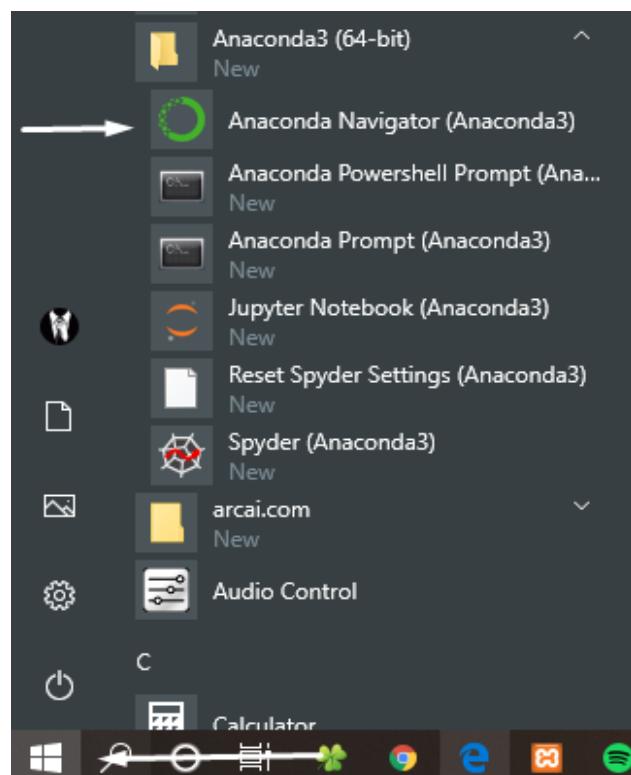
1. Mendukung pengeditan kode yang dapat dilakukan secara langsung pada browser dengan disertakan juga *highlight* terhadap sintaks, indentasi ataupun *tab completion* yang dilakukan secara otomatis.
2. Memiliki kemampuan untuk mengeksekusi secara langsung pada tempat untuk menulis serta mengeditnya yaitu web browser yang mana hasil dari eksekusinya dilampirkan dibawah setiap perintah yang dijalankan.
3. Menampilkan hasil komputasi menggunakan representasi media yang menarik, seperti dalam bentuk HTML, LaTeX, PNG, SVG atauun yang lainnya. Misalkan, hasil komputasi berupa gambar ataupun grafik hasil render dengan menggunakan *library* matplotlib yang berada dalam satu cell yang sama.
4. Untuk pengeditan dalam bentuk teks dibuat dalam bentuk markdown sehingga memudahkan penyediaan teks yang difungsikan sebagai komentar atau penjelasan dari kode dimana teks yang digunakan tidak terbatas pada teks tertentu saja.
5. Dengan adanya sistem markdown cells, memudahkan untuk memasukan persamaan matematika dalam dalam bentuk LaTex dan diterjemahkan menggunakan MathJax sehingga mengurangi resiko kesalahan penulisan.

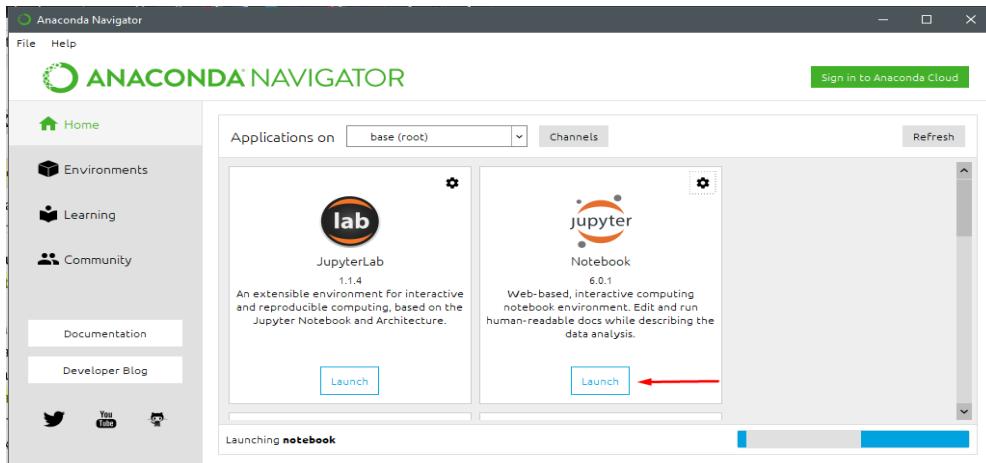
Aplikasi jupyter notebook sudah tersedia pada saat menginstal anaconda bersama dengan aplikasi lain seperti spyder serta berbagai macam *library* yang dapat mendukung untuk berbagai tugas, seperti *library* untuk pengolahan data dan visualisasi data yang biasa digunakan dalam bidang *data science*.

Terdapat beberapa cara yang dapat dilakukan untuk menjalankan jupyter notebook, seperti yang akan dijelaskan berikut ini.

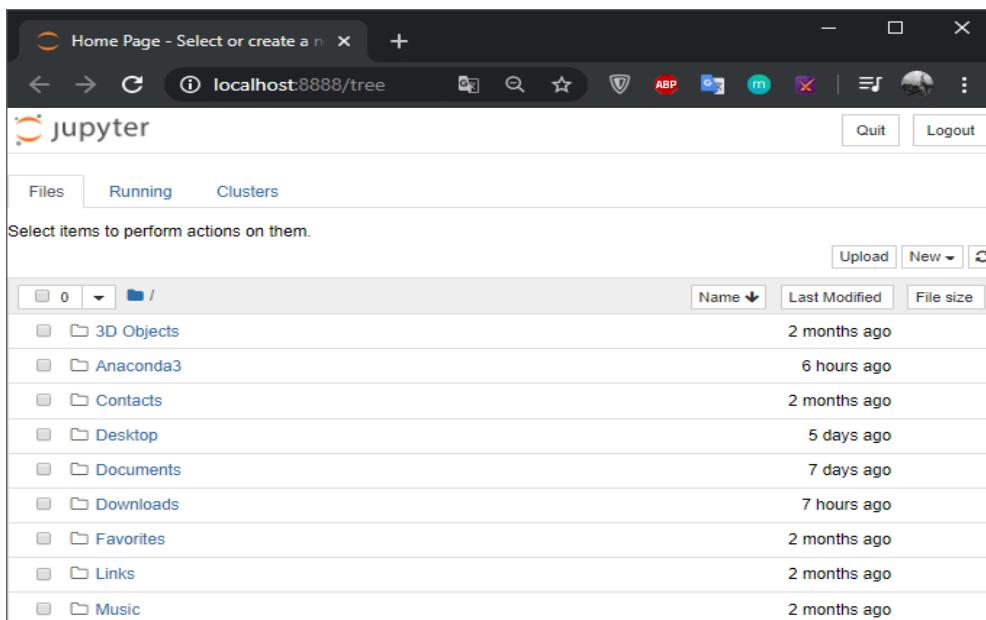
1. Menjalankan jupyter notebook melalui anaconda navigator

- Untuk melakukan cara yang pertama, buka terlebih dahulu anaconda navigator sampai muncul tampilan awal dari anaconda navigator. Dengan cara klik start kemudian, cari folder Anaconda3 kemudian klik Anconda Navigator, setelah muncul tampilan dari anaconda navigator klik launch pada bagian jupyter notebook, tunggu hingga muncul tampilan yang secara otomatis akan terbuka dalam browser.





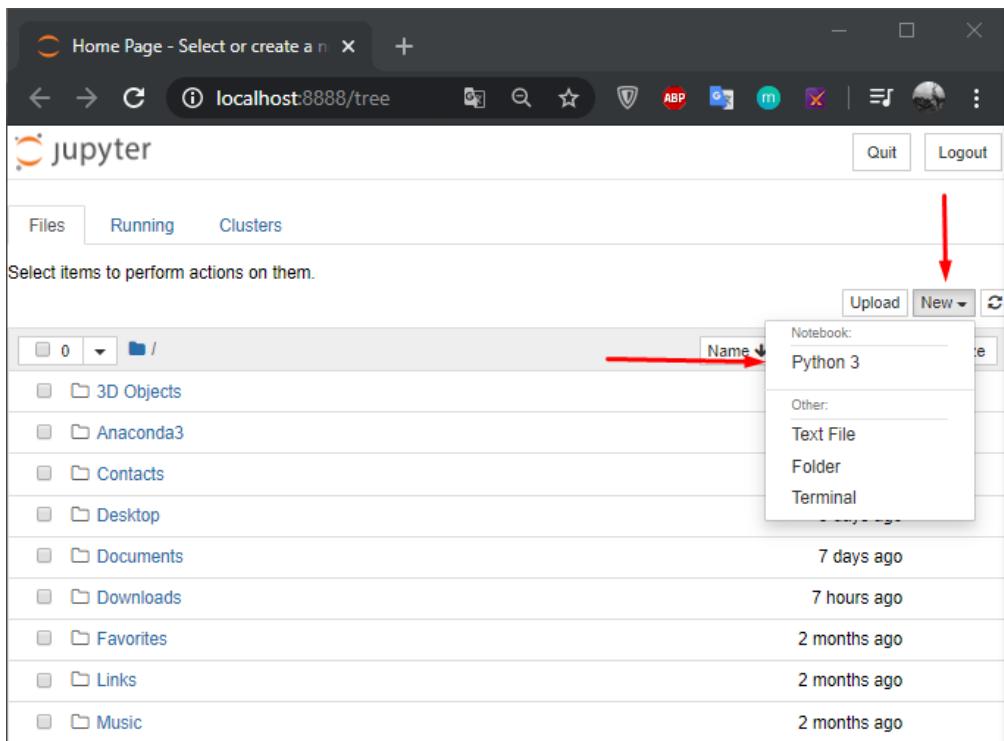
Gambar 4.24 Menjalankan Jupyter Notebook Melalui Anaconda Navigator



Gambar 4.25 Tampilan Awal Jupyter Notebook

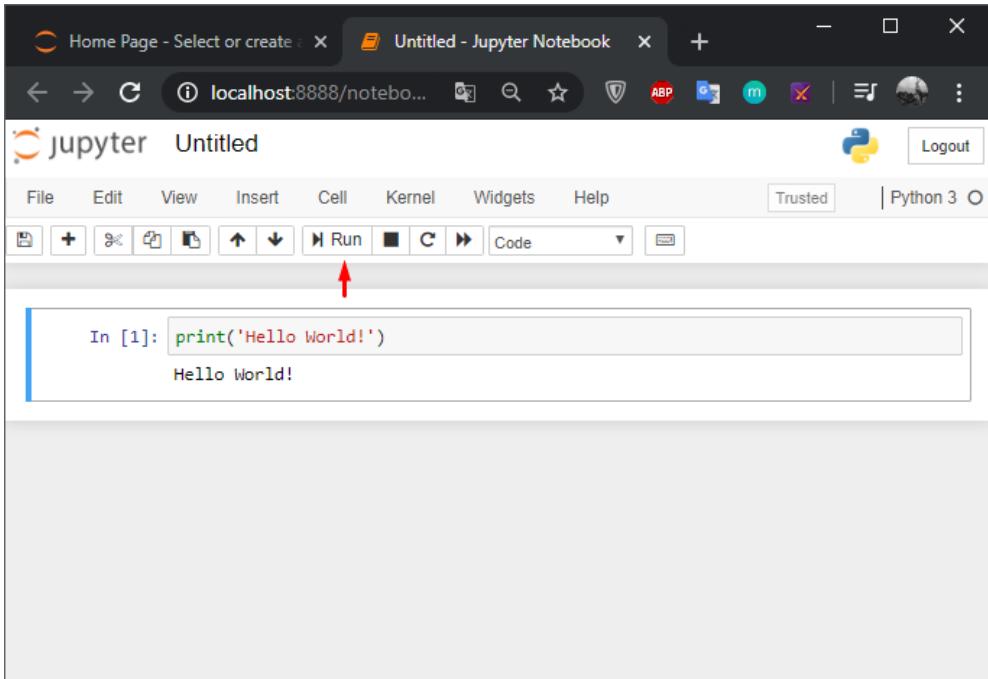
- Setelah muncul tampilan awal dari jupyter notebook, untuk membuat file baru dapat dilakukan dengan cara klik New, kemudian pilih tipe file atau bentuk editor yang akan dibuat. Disini penulis menggunakan file notebook

dengan yang kompatibel dengan menggunakan bahasa pemrograman python versi 3.



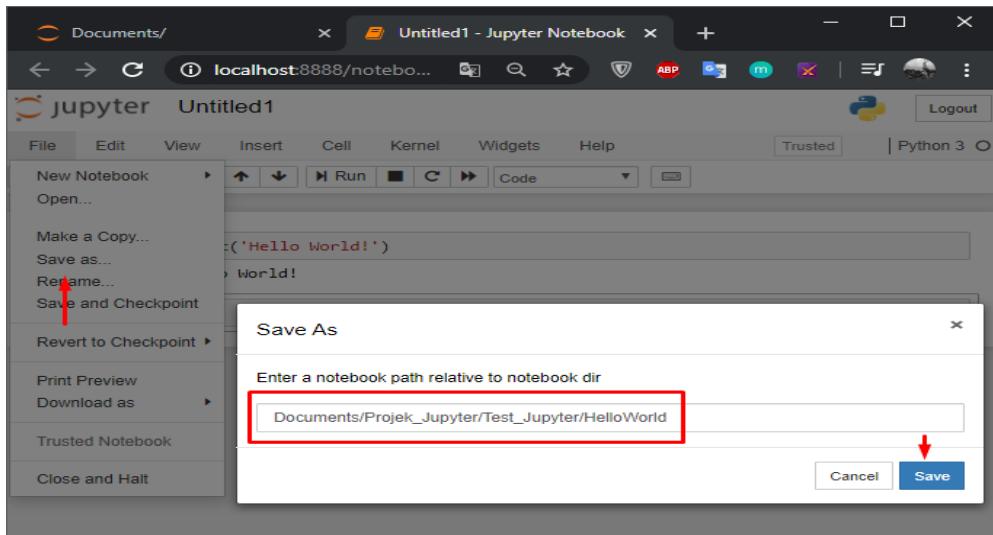
Gambar 4.26 Membuat File Baru Pada Jupyter Notebook

- Selanjutnya akan muncul tab baru yang dapat kita gunakan menulis kode. Pada tampilan ini terdapat berbagai fitur yang dapat dapat digunakan untuk membantu dalam penulisan kode seperti untuk menyimpan file beserta cek point, menambah cell, copy, cut, dan paste cell tertentu, memindahkan posisi cel, bahkan untuk menjalankan perintah pada masing-masing cell. Penulis akan memberi contoh sederhana pengeksekusian perintah untuk menampilkan teks di dalam jupyter notebook, yaitu dengan cara isi terlebih dahulu cell pada baris pertama dengan perintah tertentu, kemudian klik Run, maka hasil eksekusi akan langsung muncul secara otomatis dibawah baris perintah. Hal tersebut berlaku untuk menjalankan cell berikutnya.

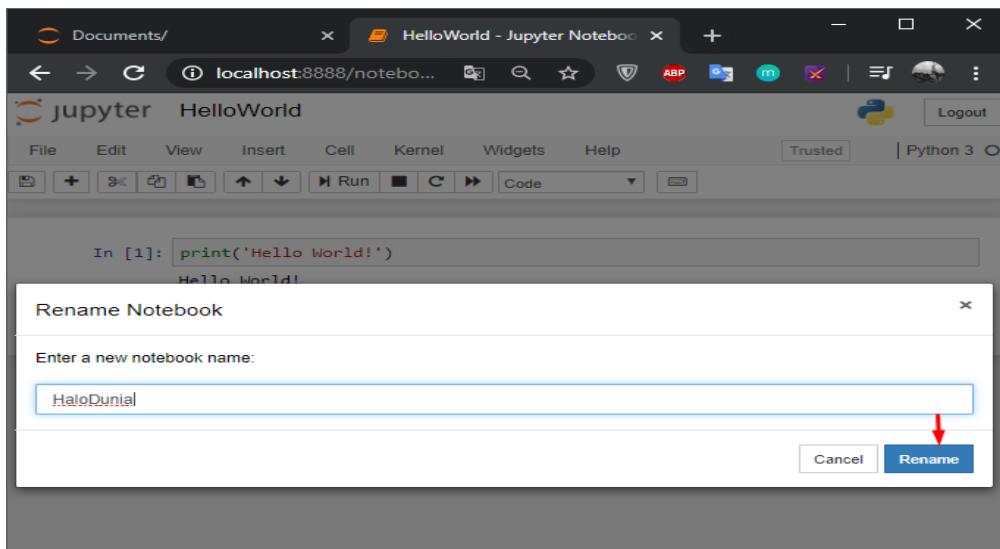


Gambar 4.27 Mengeksekusi Baris Perintah Pada Jupyter Notebook

- Untuk menyimpan file dapat dilakukan dengan klik File kemudian Save as, kemudian masukan direktori yang akan dijadikan sebagai tempat penyimpanan kemudian klik Save. Nama file akan secara otomatis dibuat oleh sistem yaitu Untitled, namun selanjutnya kita mengganti nama file tersebut dengan cara klik kembali pada file kemudian masukan nama baru untuk file tersebut.



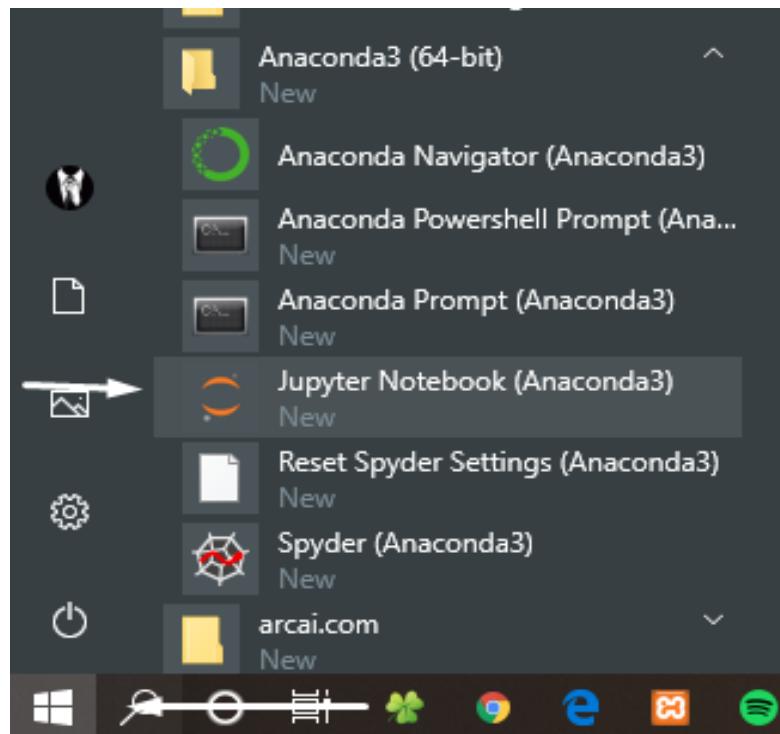
Gambar 4.28 Menyimpan File Pada Jupyter Notebook



Gambar 4.29 Mengganti Nama File Pada Jupyter Notebook

## 2. Menjalankan jupyter notebook melalui *shortcut*

- Untuk cara kedua ini dapat dilakukan dengan klik pada start, kemudian cari folder dari Anaconda3, kemudian klik *shortcut* dari jupyter notebook dan kemudian akan tampil halaman awal dari jupyter notebook pada browser seperti pada cara pertama.



Gambar 4.30 Menjalankan Jupyter Notebook Melalui Shortcut

## 4.9 *Library* Python

Setiap bahasa pemrograman pasti memiliki sebuah *library*. *Library* itu sendiri merupakan sebuah kumpulan fungsi atau perintah yang digunakan untuk menjalankan tugas tertentu dalam suatu bahasa pemrograman agar dapat memenuhi kebutuhan dalam membuat program. *Library* yang tersedia pada bahasa pemrograman python ada yang merupakan *library* standar dan ada juga yang dibangun serta dikembangkan oleh pihak ketiga bahkan oleh komunitas. Berikut ini beberapa *library* standar yang tersedia dalam setiap paket instalasi python, dimana masing-masing *library* tersebut dikelompokan sesuai dengan tugas serta fungsinya.

- **Fungsi bawaan**

1. Truth value testing.
2. Boolean operations.
3. Comparisons.
4. Numeric types.
5. Interator types.
6. Sequence types.
7. Text sequence type.
8. Binary sequence types.
9. Set types.
10. Mapping types.

- **Layanan pemrosesan teks**

1. string – operasi string umum.
2. re – operasi ekspresi reguler.
3. difflib – pembantu untuk delta komputasi.
4. textwrap – pembungkus dan pengisian teks.
5. unicodedata – basis data unicode.
6. stringprep – persiapan string internet.
7. readline – antarmuka readline GNU.
8. rlcompleter – fungsi penyelesaian untuk readline GNU.

- **Layanan data biner**

1. struct – untuk menafsirkan byte sebagai data biner yang dapat dikemas.
2. codecs – untuk registry codec dan kelas dasar.

- **Modul numerik dan matematika**

1. numbers – kelas dasar abstrak numerik.
2. math – fungsi matematika.
3. cmath – fungsi matematika untuk bilangan kompleks.
4. decimal – aritmatika titik tetap decimal dan floating point.
5. fractions – angka rasional.
6. random – untuk menghasilkan angka secara acak.
7. statistics – untuk fungsi statistic dalam matematika.

- **Modul pemrograman fungsional**

1. itertools – fungsi membuat iterator untuk perulangan yang efisien.
2. functools – fungsi dan operasi tingkat tinggi pada objek yang dapat dipanggil.
3. operator – operator standar yang digunakan sebagai fungsi.

- **Akses file dan direktori**

1. pathlib – jalur sistem file berorientasi objek.
2. os.path – manipulasi pathname umum.
3. fileinput – iterate over lines dari beberapa input stream.
4. stat – menafsirkan stat() hasil.
5. filecmp – perbandingan file dan direktori.
6. tempfile – menghasilkan file dan direktori sementara.
7. glob – perluasan pola pathname style Unix.
8. fnmatch – pencocokan pola nama file Unix.
9. linecache – akses acak ke baris teks.
10. shutil – operasi file tingkat tinggi.

- **Tipe data**

1. datetime – tipe tanggal dan waktu dasar.
2. calendar – fungsi terkait kalender umum.
3. collections – tipe data container.
4. array – tipe data array yang efisien dari suatu nilai numerik.

- **Kompresi dan pengarsipan data**

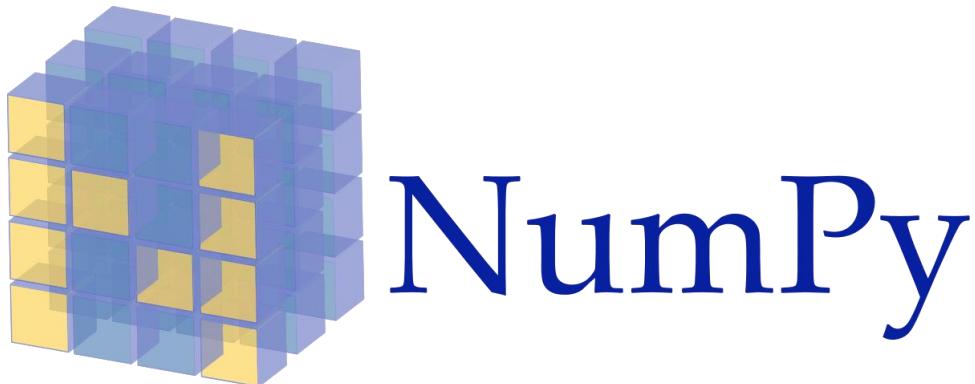
1. zlib – kompresi yang kompatibel dengan gzip.
2. gzip – dukungan untuk file gzip.
3. bz2 – dukungan untuk kompresi bzip2.
4. zipfile – untuk menangani arsip zip.
5. tarfile – untuk membaca dan menulis file pada arsip tar.

*Library* standar tersebut dapat digunakan untuk melakukan operasi yang ada dalam bahasa pemrograman python. Penjelasan serta daftar *library* standar yang ada dalam bahasa pemrograman python dapat dilihat pada tautan <https://docs.python.org/3/library/index.html> untuk python 3 dan <https://docs.python.org/2/library/index.html> untuk python 2.

Selain *library* standar yang tersedia dalam setiap instalasi python, terdapat pula *library* yang dibuat untuk membantu melaksanakan tugas dalam bidang tertentu serta mempercepat pelaksanaan tugas dari masing tahapan yang dilalui agar hasilnya dapat diperoleh dengan cepat serta mengurangi resiko kesalahan dalam melaksanakan tugas terutama bila tugas tersebut berkaitan dengan pengolahan data seperti *data science*.

Berikut ini akan dijelaskan beberapa *library* yang dapat digunakan untuk membantu pelaksanaan tugas dalam *data science*.

### 4.9.1 Numpy



Gambar 4.31 Logo Library Numpy

Numpy merupakan singkatan dari *Numerical Python*. Numpy adalah sebuah *library* untuk bahasa pemrograman python, dengan menambahkan dukungan yang besar untuk menangani multi dimensional array dan matriks, bersama dengan koreksi fungsi *high-level* matematika untuk memproses nilainya dalam bentuk array. Numpy diciptakan pada tahun 2005 oleh Travis Oliphant dan memiliki banyak *contributor*. Kemudian *library* ini juga dapat digunakan untuk berbagai keperluan seperti melakukan manipulasi *array*, menghasilkan *random value*, serta mengembalikan nilai array dengan spasi yang sama untuk setiap elemen sesuai dengan interval dan lainnya.

Numpy dapat berisi class, object, fungsi, dan kode program yang dapat digunakan untuk memecahkan permasalahan model matematika komputer dengan tingkat kompleksitas yang tinggi dan tidak dapat diselesaikan hanya dengan menggunakan python saja. Salahsatu contoh model tersebut adalah objek array multidimensi yang memiliki kinerja yang tinggi dimana memiliki struktur data yang kompleks untuk sebuah perhitungan array dan matriks secara efisien.

#### 4.9.2 Pandas



Gambar 4.32 Logo Library Pandas

Pandas adalah salah satu *library* di python yang *open source*, berlisensi BSD yang menyediakan performansi tinggi, struktur data yang *easy-to-use* dan merupakan *tools* analisis data untuk bahas pemrograman python. Pandas merupakan proyek yang disponsori oleh NumFOCUS. Pandas memiliki *behavior* khusus dalam hal menambahkan fungsi *groupby*, yang dikenal sebagai “*name aggregation*” untuk penamaan kolom dari output ketika mengaplikasikan fungsi agregasi ke kolom yang spesifik.

Pandas Series adalah sebuah *one-dimensional array* yang mampu menampung berbagai tipe data (integer, string, float, python objects, *etc*), pandas series tidak lain adalah sebuah kolom seperti pada *sheet excel*. Pandas series akan tercipta dengan me-*load datasets* dari *storage* yang ada, *storage* dapat berupa SQL *database*, berkas CSV, dan berkas excel.

### 4.9.3 Seaborn



Gambar 4.33 Logo Library Seaborn

Seaborn merupakan *library* visualisasi data pada lingkungan Python bersifat sumber terbuka yang berlisensi BSD dan dibangun di atas *library* matplotlib. Seaborn akan mempermudah kita sebagai analis data untuk memproduksi visualisasi yang indah tanpa kostumisasi rumit seperti yang kita hadapi pada matplotlib. Seaborn awalnya diperkenalkan Michael askom, yang kala itu merupakan seorang mahasiswa doktoral neurosains di Universitas Stanford, untuk memvisualisasikan data untuk analisis jaringan saraf pada awal tahun 2014 silam. Saat ini seaborn telah memasuki versi stabil 0.9.0, dan akan terus dikembangkan oleh komunitasnya yang cukup besar.

Seaborn memiliki berbagai bentuk visualisasi yang lebih menarik, beberapa model diantaranya yaitu model implot, barplot, kdeplot, scatterplot, distplot, lineplot, facetgrid, relplot, dan lain sebagainya. Untuk melihat lengkap berbagai contoh visualisasinya, dapat dilihat pada *example gallery* yang disediakan pada website official dari seaborn.

#### 4.9.4 Scikit-learn



Gambar 4.34 Logo Library Scikit-learn

Scikit-learn (sklearn) adalah *library machine learning* yang dibuat untuk Bahasa python. *Library* ini menyediakan kelas-kelas untuk proses klasifikasi, regresi, *clustering*, dan juga beberapa kelas-kelas lain untuk melakukan pembacaan data, praproses data, dan lain-lain. Scikit-learn menyediakan beberapa kelas yang berguna untuk proses kategorisasi dokumen, seperti CountVectorizer dan TfidfTransformer. Kelas CountVectorizer berfungsi untuk mengubah dokumen menjadi representasi vektornya dengan pembobotan TF. Sedangkan TfidfTransformer berfungsi untuk mentransformasi vektor dengan pembobotan TF menjadi vektor dengan pembobotan TF-IDF. Untuk membantu membangun classifier, *library* ini menyediakan kelas-kelas seperti MultinomialNB (implementasi classifier Multinomial Naïve Bayes), DecisionTreeClassifier (implementasi classifier Decision Tree), Algoritma Regresi, Algoritma Clustering, Parameter Tuning, Data Preprocessing Tool, Export/Import Model, Machine Learning Pipeline dan lain-lain.

#### 4.9.5 Matplotlib



Gambar 4.35 Logo Library Matplotlib

Matplotlib adalah *library* plotting Python 2D yang menghasilkan gambar publikasi dapat berupa plot, histogram, power spectra, grafik batang, grafik error, scatterplot, dll. Matplotlib dapat digunakan di dalam script Python, shell Python dan ipython, matlab, server aplikasi web, dan GUI. Dalam beberapa tahun terakhir, library ini telah menyebar luas dan banyak digunakan di kalangan ilmiah dan teknik. Di antara semua fitur yang telah menjadikannya alat yang paling banyak digunakan dalam representasi grafis data, ada beberapa yang menonjol diantaranya adalah sebagai berikut.

1. Kesederhanaan dalam penggunaannya.
2. Pengembangan bertahap dan visualisasi data interaktif.
3. Ekspresi dan teks dalam Latex.
4. Kontrol yang lebih besar atas elemen grafis.
5. Ekspor ke banyak format, seperti PNG, PDF, SVG, dan EPS.

## **BAB 5**

---

# **PENERAPAN SEGMENTASI PELANGGAN MENGGUNAKAN PYTHON**

---

### **5.1 Mengapa Perlu Dilakukan Segmentasi Pelanggan?**

Dalam setiap bisnis apapun baik itu dalam bidang jasa ataupun manufaktur, yang menjadi salahsatu faktor penting yang dapat berpengaruh terhadap kemajuan suatu perusahaan adalah pemasaran. Hal tersebut akan mempengaruhi kegiatan sebuah perusahaan agar bisa bersaing dengan perusahaan-perusahaan lain untuk menguasai dan mempertahankan pasar potensial yang ada. Pemasaran bukan saja hanya pengembangan produk dan jasa yang dibutuhkan tetapi segmentasi pelanggan juga harus diperhitungkan.

Perusahaan telekomunikasi di Indonesia sekarang sedang melakukan banyak inovasi untuk melakukan persaingan yang semakin ketat, karena

pelanggan berhak memilih dari banyaknya penyedia layanan yang ada. Persaingan ini merupakan hal yang penting dan tidak boleh diabaikan oleh perusahaan karena dapat mempengaruhi pendapatan yang diperoleh oleh perusahaan.

Persaingan tersebut mengakibatkan beralihnya pelanggan ke perusahaan telekomunikasi lain atau bisa disebut *Churn*. *Churn* adalah pemutusan jasa suatu perusahaan oleh pelanggan karena pelanggan tersebut lebih memilih menggunakan layanan jasa perusahaan kompetitor. Pada persaingan pasar ini, dilihat dari pengalaman bahwa setiap tahunnya sekitar 30-35% laju *churn* dan membutuhkan 5-10 kali usaha serta biaya untuk menambah pelanggan baru daripada mempertahankan yang sudah ada. Untuk itu industri telekomunikasi lebih memilih untuk mempertahankan pelanggan.

Dengan keadaan yang demikian maka perusahaan membutuhkan suatu strategi yang tepat dan berkelanjutan untuk menjangkau kebutuhan sasaran pasar dalam dunia telekomunikasi khususnya. Maka dari itu, segmentasi pelanggan perlu diperhatikan bagi perusahaan jasa seperti pada perusahaan telekomunikasi, karena hal tersebut merupakan suatu hal terpenting dan menjadi langkah prioritas utama bagi kelangsungan penjualan produk perusahaan, menghadapi persaingan dan harus mampu mempertahankan eksistensinya di era pemasaran modern saat ini.

Pada saat ini beberapa perusahaan telekomunikasi terkemuka melakukan kegiatan strategi pemasaran dengan cara publisitas produk pada sosial media, *personal selling*, serta periklanan.

Namun pada kegiatan pemasaran tersebut dirasa belum efektif dikarenakan masih ada beberapa produk yang kurang diminati. Fokus utama perusahaan untuk bersaing dengan kompetitornya adalah pelanggan. Permasalahannya adalah belum ada informasi tentang karakteristik pelanggan yang dimiliki oleh perusahaan yang mana informasi

tersebut dapat dijadikan sebagai langkah awal untuk melakukan analisa karakteristik pelanggan, informasi karakteristik pelanggan tersebut juga dapat digunakan sebagai dasar bagi penetapan segmentasi dan penentuan target pasar yang dilayani.

Seperti yang telah disebutkan pada pembahasan sebelumnya, segmentasi pelanggan itu sendiri adalah suatu cara untuk mengelompokkan pelanggan ke dalam beberapa *cluster* (kelompok) dan setiap *cluster* memiliki beberapa anggota dengan karakteristik yang sama. Usaha untuk mengklasifikasikan objek-objek dengan suatu kesamaan ke dalam satu grup tersebut juga dapat dikatakan sebagai *clustering*. Analisis *cluster* akan membangun suatu *cluster* yang baik ketika setiap anggota dari *cluster* memiliki derajat kesamaan yang tinggi (homogen internal) (Grawe, 1999; Castro, 2002). Tujuan dari dilakukannya segmentasi pelanggan adalah untuk mengetahui perilaku serta karakteristik dari masing-masing pelanggan serta untuk menentukan strategi pemasaran yang tepat sehingga dapat meningkatkan keuntungan bagi pihak perusahaan. Selain dari itu, proses *marketing* (komunikasi, produk/jasa, program) yang dilakukan pun dapat menjadi lebih terfokus dikarenakan masing-masing segmen atau kelompok memang sudah memiliki kemiripan, baik dari segi kebutuhan maupun perilakunya. Pengelompokan tersebut dilakukan dengan menggunakan teknik data mining, oleh karena itu salahsatu implementasi atau pengaplikasian dari teknik data mining dapat digunakan untuk menyelesaikan permasalahan terkait dengan segmentasi pelanggan.

Salahsatu teknik data mining tersebut adalah *clustering*. *Clustering* memiliki peran yang penting dalam data mining, dimana teknik ini akan membagi data kedalam beberapa *cluster* sesuai dengan kemiripannya. Adapun manfaat lain dari *clustering* diantaranya adalah *clustering* merupakan metode segmentasi data yang sangat berguna dalam prediksi dan analisis masalah

bisnis tertentu seperti segmentasi pasar, *marketing* dan pemetaan zonasi wilayah.

*Clustering* memiliki banyak algoritma yang dapat digunakan dalam menyelesaikan berbagai permasalahan, baik itu secara hirarki maupun non-hirarki. Dalam buku ini, penulis menggunakan teknik *clustering* yang sangat populer dan sering digunakan yaitu K-Means untuk menyelesaikan permasalahan segmentasi pelanggan. K-Means adalah salahsatu algoritma *clustering* yang dapat mempartisi, atau membagi data yang ada ke dalam satu atau lebih kelompok. Algoritma ini membagi data ke dalam suatu kelompok berdasarkan karakteristik dan kesamaan dari suatu data.

Untuk dapat melakukan segmentasi pelanggan menggunakan algoritma K-Means maka diperlukan suatu data. Data yang akan dianalisis merupakan data pelanggan yang diperoleh dari arsip perusahaan. Adapun proses pengelompokkan data dilakukan dengan mengambil 3 jenis atribut diantaranya yaitu lama berlangganan, jumlah paket yang diambil, dan jumlah tagihan yang dibayar oleh pelanggan.

Setelah dianalisis data mana saja yang digunakan, selanjutnya data tersebut akan diproses untuk menentukan segmentasi pelanggan yang tepat dengan membuat *clustering* data pelanggan menggunakan algoritma K-means kemudian dibuatkan juga dalam suatu model. Hasil dari pemodelan data tersebut akan mengelompokkan pelanggan kedalam sejumlah *cluster* dan menentukan profil konsumen (*customer profiling*). Pengelompokan tersebut akan menghasilkan karakteristik dari masing-masing pelanggan pada tiap *cluster* yang dapat dijadikan salah satu acuan untuk pengambilan keputusan perusahaan.

## 5.2 Penerapan CRISP-DM Pada Segmentasi Pelanggan

Dalam proses segmentasi pelanggan, diperlukan adanya suatu metodologi agar setiap pendekatan yang dilakukan dalam untuk menyelesaikan permasalahan dapat berjalan secara sistematis. Metodologi yang digunakan dalam buku ini dan banyak digunakan dalam data mining yaitu CRISP-DM (*Cross Standart Industries for Data Mining*). Berikut ini merupakan langkah-langkah yang dilakukan dalam melakukan segmentasi pelanggan yang telah disesuaikan dengan metodologi CRISP-DM.

### 5.2.1 Analisis Permasalahan

Analisis permasalahan dilakukan berdasarkan penelitian pada salahsatu perusahaan telekomunikasi yan ada di Indonesia. Dimana terdapat beberapa permasalahan yaitu diantaranya mengenai strategi pemasaran, hal tersebut dirasa belum efektif dikarenakan masih ada beberapa produk yang kurang diminati. Maka dari itu perlunya mengetahui karakteristik pelanggan yang mana hal tersebut merupakan langkah awal untuk menganalisa data pelanggan yang diperlukan untuk penetapan segmentasi dan penentuan target pasar yang dilayani.

Untuk mengetahui segmentasi pelanggan tersebut maka digunakan teknik data mining. Data yang dianalisis merupakan data pelanggan yang didapat dari arsip perusahaan. Kemudian dari data tersebut diambil 3 atribut yaitu lama berlangganan, jumlah paket yang diambil, dan jumlah tagihan yang dibayar oleh pelanggan. Data tersebut mewakili karakteristik pelanggan yang akan diketahui guna untuk penetapan segmentasi pelanggan dan customer proifiling. Proses segmentasi data pelanggan dilakukan dengan menggunakan metode K-means clustering.

### 5.2.2 Pengumpulan Data

Pengumpulan data dilakukan dengan mengambil data dari arsip perusahaan yang berupa data pelanggan yang berada dalam suatu wilayah operasional. Data tersebut berisikan informasi yang diperoleh perusahaan mengenai pelanggan selama berlangganan layanan perusahaan dari mulai bulan Januari hingga Oktober 2019. Informasi tersebut beberapa diantaranya nomor pelanggan, alamat, tanggal registrasi, tanggal layanan dapat digunakan, tanggal berlangganan, layanan *add on* yang digunakan, serta harga dari masing-masing layanan *add on*. Keseluruhan data berjumlah 4640 record akan tetapi pada studi kasus ini diambil sampel datanya 30 % dari jumlah keseluruhan yaitu 1392 record.

### 5.2.3 Pengolahan Data

Pengolahan data dilakukan apabila semua data yang akan digunakan untuk proses segmentasi telah dikumpulkan dan dipahami bahwa data tersebut sesuai kebutuhan. Pada proses pengolahan data, data pelanggan yang diperoleh dari arsip perusahaan memiliki bermacam-macam atribut serta dilakukan proses pemilihan atribut. Pemilihan atribut tersebut dilakukan untuk memilih atribut mana saja yang cocok digunakan untuk proses segmentasi serta dapat mewakili identitas dari masing-masing pelanggan seperti pada Tabel 5.1.

Tabel 5.1 Dataset Pelanggan

NCLI	LAMA_LANGGANAN	JUMLAH_LAYANAN	JUMLAH_TAGIHAN_HARUS_DIBAYAR
39684298	11	1	418000
39713960	11	2	511500
39716635	11	2	698500
39818227	11	2	291500
68228	11	2	517000
39937697	11	1	352000
40011078	11	2	847000
39817407	11	1	621500
39864058	11	2	1100000
39952623	11	1	902000
39891161	11	1	352000
40033755	11	2	291500
39786171	11	2	2090000
39995005	11	2	621500
32745610	11	2	814000
380513	1	2	1358500
39765624	11	1	286000
39600770	11	1	275000
39999904	11	1	275000
39892083	11	1	495000

#### 5.2.4 Pemodelan

Model dapat diartikan sebagai representasi dari sebuah objek ataupun ide yang direalisasikan dalam bentuk yang sederhana. Sebuah model dapat berisi informasi-informasi terkait dengan suatu sistem yang akan dibuat dengan memiliki tujuan untuk mempelajari sistem yang sebenarnya. Model juga dapat dijadikan sebagai sebuah tiruan dari suatu sistem ataupun kejadian yang sebenarnya yang didalamnya memiliki sebuah informasi penting yang dapat dianalisis untuk penelitian lebih lanjut.

Pemodelan adalah proses untuk membuat, membangun serta membentuk sebuah model dari suatu sistem nyata yang digunakan untuk sebuah simulasi yang bertujuan untuk memperkecil kesalahan pengembangan dan hasil dari model dalam implementasi sistem dikemudian hari. Model simulasi merupakan salah satu bentuk model matematis yang bersifat deskriptif atau prediktif yang menggambarkan suatu hubungan dari sebab dan

akibat dari sebuah sistem yang terdapat pada model komputer yang diharapkan dapat menggambarkan perilaku yang terjadi secara nyata. Model simulasi dapat digunakan sebagai indikator untuk mengetahui satu atau lebih hal yang terjadi terhadap komponen atau variable yang digunakan. Simulasi dapat diartikan sebagai kumpulan dari berbagai metode dan aplikasi yang digunakan untuk membuat tiruan atau representasi dari perilaku dari suatu sistem secara nyata, yang biasanya dilakukan pada komputer dengan menggunakan perangkat lunak tertentu. Tujuan suatu pemodelan adalah untuk menganalisa dan memberi prediksi yang dapat mendekati kenyataan sebelum sistem di terapkan di lapangan.

Aplikasi yang digunakan dalam pemodelan adalah jupyter notebook dengan menggunakan bahasa pemograman python. Berikut ini merupakan pemodelan segmentasi pelanggan menggunakan jupyter notebook.

#### **5.2.4.1 Import Module**

*Import Module* merupakan langkah awal yang dilakukan dalam pemodelan yang ada dalam studi kasus segmentasi pelanggan. Pada tahap ini penulis melakukan *import module* atau *library* yang akan digunakan untuk pemanggilan fungsi yang digunakan selama proses pemodelan, hal tersebut dilakukan karena jupyter notebook merupakan salahsatu aplikasi yang dapat digunakan untuk meenulis kode menggunakan bahasa pemrograman python sehingga *library* yang digunakan merupakan *library* yang berjalan pada bahasa pemrograman python seperti Numpy, Pandas, Sklearn, Matplotlib dan Seaborn untuk visualisasi data. Pada Gambar 5.1 merupakan cara *import module* atau *library*, serta *library* yang digunakan dalam pemodelan.

```
In [1]: %matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn import metrics
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import silhouette_score
from sklearn.metrics import davies_bouldin_score
from sklearn.metrics import pairwise_distances
```

Gambar 5.1 Import Module atau Library

#### 5.2.4.2 Baca Data

Pada tahap ini dilakukan proses pembacaan terhadap data atau biasa dikenala dengan istilah *load data* pelanggan. Perintah yang digunakan untuk *load data* adalah pd.read\_csv dimana pd merupakan inisialisasi dari *library* Pandas. Data yang dibaca adalah sebuah file excel dengan format ekstensi csv. Kemudian untuk menampilkan hasil pembacaan dari data tersebut dapat digunakan perintah data\_cust.head() seperti ditunjukan pada Gambar 5.2.

```
In [2]: data_cust = pd.read_csv('DATASET_MENTAH.csv')
data_cust.head()
```

Out[2]:

	NCLI	LAMA_LANGGANAN	JUMLAH_LAYANAN	JUMLAH_TAGIHAN_HARUS_DIBAYAR
0	39684298	11	1	418000
1	39713960	11	2	511500
2	39716635	11	2	698500
3	39818227	11	2	291500
4	68228	11	2	517000

Gambar 5.2 Baca Data atau *Load Data*

### 5.2.4.3 Normalisasi Data

Pada tahap ini penulis melakukan proses normalisasi data, dimana normalisasi itu sendiri merupakan sebuah transformasi untuk merubah nilai data atau untuk menyamakan skala atribut data kedalam *range* yang lebih spesifik yang lebih kecil yaitu antara 0-1. Normalisasi dilakukan apabila terdapat perbedaan *range* nilai dari masing-masing atribut yang akan diolah. Berikut ini merupakan cara normalisasi data yang ditunjukan pada Gambar 5.3.

```
In [3]: A = data_cust[['LAMA_LANGGANAN', 'JUMLAH_LAYANAN', 'JUMLAH_TAGIHAN_HARUS_DIBAYAR']]
scaler = MinMaxScaler(feature_range=(0, 1))
rescaled_data_cust = scaler.fit_transform(A)

np.set_printoptions(precision=2)
B = pd.DataFrame(rescaled_data_cust, columns=['LAMA_LANGGANAN', 'JUMLAH_LAYANAN', 'JUMLAH_TAGIHAN_HARUS_DIBAYAR'])
C = B.round(2)
C
```

	LAMA_LANGGANAN	JUMLAH_LAYANAN	JUMLAH_TAGIHAN_HARUS_DIBAYAR
0	1.0	0.00	0.08
1	1.0	0.25	0.13
2	1.0	0.25	0.23
3	1.0	0.25	0.01
4	1.0	0.25	0.13
...	...	...	...
1387	1.0	0.00	0.04
1388	1.0	0.00	0.04
1389	1.0	0.00	0.13
1390	1.0	0.00	0.13
1391	1.0	0.25	0.13

```
In [4]: D = data_cust[['NCLI']]
E = pd.DataFrame(D)

F = pd.concat([D, C], axis=1, sort=False)
data_cust2 = F
data_cust2
```

	NCLI	LAMA_LANGGANAN	JUMLAH_LAYANAN	JUMLAH_TAGIHAN_HARUS_DIBAYAR
0	39684298	1.0	0.00	0.08
1	39713960	1.0	0.25	0.13
2	39716635	1.0	0.25	0.23
3	39818227	1.0	0.25	0.01
4	68228	1.0	0.25	0.13
...	...	...	...	...
1387	39805580	1.0	0.00	0.04
1388	39786958	1.0	0.00	0.04
1389	39910134	1.0	0.00	0.13
1390	39907704	1.0	0.00	0.13
1391	39930270	1.0	0.25	0.13

Gambar 5.3 Normalisasi Data

#### 5.2.4.4 Split Data

Pada tahap ini penulis melakukan proses *split data* atau proses membagi data, dimana data pelanggan yang telah di *load* dibagi kedalam dua kategori yaitu *train data* dan *test data*. Dalam studi kasus ini, *split data* dilakukan untuk membagi data menjadi 3 simulasi, yang mana dari masing-masing simulasi tersebut memiliki persentase yang berbeda yaitu untuk simulasi pertama data dibagi menjadi 80% *train data* dan 20% *test data*, untuk simulasi kedua data dibagi menjadi 70% *train data* dan 30% *test data*, kemudian untuk simulasi ketiga data dibagi menjadi 50% *train data* dan 50% *test data*.

```
In [9]: train_data, test_data = train_test_split(data_cust2, test_size=.2)
print(train_data.shape, test_data.shape)
```

(1113, 4) (279, 4)

Train Data

```
In [10]: train_data
```

Out[10]:

	NCLI	LAMA_LANGGANAN	JUMLAH_LAYANAN	JUMLAH_TAGIHAN_HARUS_DIBAYAR
863	39840291	1.0	0.50	0.35
722	40004208	1.0	0.00	0.14
411	39940204	1.0	0.25	0.35
584	39777762	1.0	0.00	0.10
1029	39908311	1.0	0.00	0.04
...	...	...	...	...
422	39802282	1.0	0.25	0.35
240	39802075	1.0	0.00	0.08
435	39751182	1.0	0.25	0.35
606	39912123	1.0	0.00	0.10
1117	40051249	1.0	0.25	0.17

1113 rows × 4 columns

## 156 PENERAPAN SEGMENTASI PELANGGAN MENGGUNAKAN PYTHON

```
In [11]: old_colname = train_data.columns.tolist()
old_colname
Out[11]: ['NCLI', 'LAMA_LANGGANAN', 'JUMLAH_LAYANAN', 'JUMLAH_TAGIHAN_HARUS_DIBAYAR']

In [12]: replaced_colname = ['id_plg','lama_langganan','jumlah_layanan','total_tagihan']

In [13]: dict_colname=dict(zip(old_colname,replaced_colname))
dict_colname
Out[13]: {'NCLI': 'id_plg',
'LAMA_LANGGANAN': 'lama_langganan',
'JUMLAH_LAYANAN': 'jumlah_layanan',
'JUMLAH_TAGIHAN_HARUS_DIBAYAR': 'total_tagihan'}

In [14]: train_data2=train_data.rename(columns=dict_colname)

In [15]: train_data2
Out[15]:
   id_plg  lama_langganan  jumlah_layanan  total_tagihan
0  863    39840291        1.0            0.50         0.35
1  722    40004208        1.0            0.00         0.14
2  411    39940204        1.0            0.25         0.35
3  584    39777762        1.0            0.00         0.10
4  1029   39908311        1.0            0.00         0.04
.. ...
9  422    39802282        1.0            0.25         0.35
10 240    39802075        1.0            0.00         0.08
11 435    39751182        1.0            0.25         0.35
12 606    39912123        1.0            0.00         0.10
13 1117   40051249        1.0            0.25         0.17
1113 rows × 4 columns
```

Test Data

```
In [16]: test_data
Out[16]:
   NCLI  LAMA_LANGGANAN  JUMLAH_LAYANAN  JUMLAH_TAGIHAN_HARUS_DIBAYAR
0  498    39965774        1.0            0.00         0.14
1  685    39979908        1.0            0.00         0.08
2  1068   39983392        1.0            0.00         0.04
3  939    39880550        1.0            0.00         0.08
4  765    39825884        1.0            0.25         0.13
.. ...
9  699    40044082        1.0            0.00         0.08
10 86     334979          1.0            0.25         0.17
11 792    39895179        1.0            0.00         0.13
12 41     39745365        1.0            0.25         0.13
13 542    39891247        1.0            0.25         0.17
279 rows × 4 columns
```

```
In [17]: old_colname = test_data.columns.tolist()
old_colname
```

```
Out[17]: ['NCLI', 'LAMA_LANGGANAN', 'JUMLAH_LAYANAN', 'JUMLAH_TAGIHAN_HARUS_DIBAYAR']
```

```
In [18]: replaced_colname = ['id_plg','lama_langganan','jumlah_layanan','total_tagihan']
```

```
In [19]: dict_colname=dict(zip(old_colname,replaced_colname))
dict_colname
```

```
Out[19]: {'NCLI': 'id_plg',
'LAMA_LANGGANAN': 'lama_langganan',
'JUMLAH_LAYANAN': 'jumlah_layanan',
'JUMLAH_TAGIHAN_HARUS_DIBAYAR': 'total_tagihan'}
```

```
In [20]: test_data2=test_data.rename(columns=dict_colname)
```

```
In [21]: test_data2
```

```
Out[21]:
   id_plg  lama_langganan  jumlah_layanan  total_tagihan
0  498    39965774        1.0          0.00      0.14
1  685    39979908        1.0          0.00      0.08
2  1068   39983392        1.0          0.00      0.04
3  939    39880550        1.0          0.00      0.08
4  765    39825884        1.0          0.25      0.13
...
6  699    40044082        1.0          0.00      0.08
7  86     334979          1.0          0.25      0.17
8  792    39895179        1.0          0.00      0.13
9  41     39745365        1.0          0.25      0.13
10 542    39891247        1.0          0.25      0.17
```

279 rows × 4 columns

Gambar 5.4 *Split Data* Simulasi Pertama

Pada Gambar 5.4 menjelaskan bagaimana cara melakukan *split data* untuk simulasi pertama. Perintah untuk melakukan *split data* yaitu `train_test_split(data_cust2, test_size=.2)` dimana `train_test_split` merupakan nama fungsi, sedangkan `data_cust2` merupakan nama dataset yang digunakan, sedangkan `test_size=.2` adalah perintah membagi *test data* menjadi 20% dari total data sedangkan sisa data yang memiliki persentasi 80% akan secara otomatis menjadi *train data*. Kemudian untuk melihat jumlah data yang telah dibagi dapat dilakukan dengan perintah `print(train_data.shape, test_data.shape)`, maka akan muncul jumlah data dari *train data* dan *test data*

yang telah dibagi beserta jumlah kolomnya, dan untuk melihat data mana saja yang termasuk ke dalam *train data* atau *test data* dapat memanggil data dengan perintah `train_data` atau `test_data` pada cell baru. Pada tahap ini juga dapat dilakukan proses penggantian nama atribut apabila terdapat nama atribut yang cukup panjang atau dirasa memiliki penamaan menggunakan singkatan yang sulit dipahami, agar mudah untuk diingat maka nama atribut tersebut diganti menjadi nama atribut tidak terlalu panjang, prosesnya dimulai dengan menggabungkan nama atribut yang merupakan nama kolom yang akan diganti dalam sebuah list dari variable `train_data` yang telah di *split* dengan perintah `train_data.columns.tolist()`. Kemudian nama yang baru dideklarasikan menggunakan perintah `replaced_colname = [nama atribut yang baru]`, setelah itu nama lama dan nama yang baru digabungkan dalam sebuah variable yaitu `dict_colname`. Dan untuk mengganti nama dilakukan dengan perintah `train_data.rename(columns=dict_colname)`. Hal tersebut berlaku untuk penggantian nama kolom pada *test data* apabila diperlukan penggantian nama.

```
In [5]: train_data, test_data = train_test_split(data_cust2, test_size=.3)
print(train_data.shape, test_data.shape)

(974, 4) (418, 4)
```

```
In [6]: train_data
```

```
Out[6]:
```

	NCLI	LAMA_LANGGANAN	JUMLAH_LAYANAN	JUMLAH_TAGIHAN_HARUS_DIBAYAR
651	39816572	1.0	0.00	0.01
968	39853868	1.0	0.25	0.17
910	39710602	1.0	0.00	0.08
1228	39903951	1.0	0.25	0.23
1264	39794427	1.0	0.00	0.10
...	...	...	...	...
219	39937171	1.0	0.00	0.01
403	39896100	1.0	0.25	0.35
1258	39783910	1.0	0.00	0.10
84	39838968	1.0	0.25	0.17
856	39864124	1.0	0.25	0.35

974 rows × 4 columns

```
In [7]: old_colname = train_data.columns.tolist()
old_colname
```

```
Out[7]: ['NCLI', 'LAMA_LANGGANAN', 'JUMLAH_LAYANAN', 'JUMLAH_TAGIHAN_HARUS_DIBAYAR']
```

```
In [8]: replaced_colname = ['id_plg','lama_langganan','jumlah_layanan','total_tagihan']
```

```
In [9]: dict_colname=dict(zip(old_colname,replaced_colname))
dict_colname
```

```
Out[9]: {'NCLI': 'id_plg',
'LAMA_LANGGANAN': 'lama_langganan',
'JUMLAH_LAYANAN': 'jumlah_layanan',
'JUMLAH_TAGIHAN_HARUS_DIBAYAR': 'total_tagihan'}
```

```
In [10]: train_data2=train_data.rename(columns=dict_colname)
```

```
In [11]: train_data2
```

```
Out[11]:
```

	id_plg	lama_langganan	jumlah_layanan	total_tagihan
651	39816572	1.0	0.00	0.01
968	39853868	1.0	0.25	0.17
910	39710602	1.0	0.00	0.08
1228	39903951	1.0	0.25	0.23
1264	39794427	1.0	0.00	0.10
...	...	...	...	...
219	39937171	1.0	0.00	0.01
403	39896100	1.0	0.25	0.35
1258	39783910	1.0	0.00	0.10
84	39838968	1.0	0.25	0.17
856	39864124	1.0	0.25	0.35

974 rows × 4 columns

## 160 PENERAPAN SEGMENTASI PELANGGAN MENGGUNAKAN PYTHON

Test Data

```
In [12]: test_data
```

```
Out[12]:
```

	NCLI	LAMA_LANGGANAN	JUMLAH_LAYANAN	JUMLAH_TAGIHAN_HARUS_DIBAYAR
79	357131	1.0	0.25	0.17
348	65128	1.0	0.25	0.23
145	630352	1.0	0.25	0.19
525	39756218	1.0	0.25	0.17
433	40003963	1.0	0.25	0.35
...	...	...	...	...
606	39912123	1.0	0.00	0.10
149	39839013	1.0	0.25	0.30
587	39945692	1.0	0.00	0.10
548	39700561	1.0	0.00	0.04
586	39858225	1.0	0.00	0.10

418 rows × 4 columns

```
In [13]: old_colname = test_data.columns.tolist()
```

```
Out[13]: ['NCLI', 'LAMA_LANGGANAN', 'JUMLAH_LAYANAN', 'JUMLAH_TAGIHAN_HARUS_DIBAYAR']
```

```
In [14]: replaced_colname = ['id_plg','lama_langganan','jumlah_layanan','total_tagihan']
```

```
In [15]: dict_colname=dict(zip(old_colname,replaced_colname))
```

```
Out[15]: {'NCLI': 'id_plg',
          'LAMA_LANGGANAN': 'lama_langganan',
          'JUMLAH_LAYANAN': 'jumlah_layanan',
          'JUMLAH_TAGIHAN_HARUS_DIBAYAR': 'total_tagihan'}
```

```
In [16]: test_data2=test_data.rename(columns=dict_colname)
```

```
In [17]: test_data2
```

```
Out[17]:
```

	id_plg	lama_langganan	jumlah_layanan	total_tagihan
79	357131	1.0	0.25	0.17
348	65128	1.0	0.25	0.23
145	630352	1.0	0.25	0.19
525	39756218	1.0	0.25	0.17
433	40003963	1.0	0.25	0.35
...	...	...	...	...
606	39912123	1.0	0.00	0.10
149	39839013	1.0	0.25	0.30
587	39945692	1.0	0.00	0.10
548	39700561	1.0	0.00	0.04
586	39858225	1.0	0.00	0.10

418 rows × 4 columns

Gambar 5.5 *Split Data Simulasi Kedua*

Pada Gambar 5.5 menjelaskan bagaimana cara melakukan *split data* untuk simulasi kedua. Perintah untuk melakukan *split data* yaitu `train_test_split(data_cust2, test_size=.3)` dimana `train_test_split` merupakan nama fungsi, sedangkan `data_cust2` merupakan nama dataset yang digunakan, sedangkan `test_size=.3` adalah perintah membagi *test data* menjadi 30% dari total data sedangkan sisa data yang memiliki persentasi 70% akan secara otomatis menjadi *train data*. Kemudian untuk melihat jumlah data yang telah dibagi dapat dilakukan dengan perintah `print(train_data.shape, test_data.shape)`, maka akan muncul jumlah data dari *train data* dan *test data* yang telah dibagi beserta jumlah kolomnya, dan untuk melihat data mana saja yang termasuk ke dalam *train data* atau *test data* dapat memanggil data dengan perintah `train_data` atau `test_data` pada cell baru. Untuk perintah penggantian nama atribut dalam simulasi kedua memiliki penjelasan serta proses yang sama dengan yang telah dijelaskan sebelumnya pada simulasi pertama. Penggantian nama ini bersifat opsional dan tidak harus dilakukan pada setiap kali membuat sebuah model.

## 162 PENERAPAN SEGMENTASI PELANGGAN MENGGUNAKAN PYTHON

```
In [5]: train_data, test_data = train_test_split(data_cust2, test_size=.5)
print(train_data.shape, test_data.shape)

(696, 4) (696, 4)
```

Train Data

```
In [151]: train_data

Out[151]:
   NCLI  LAMA_LANGGANAN  JUMLAH_LAYANAN  JUMLAH_TAGIHAN_HARUS_DIBAYAR
0  979    536013           1.0            0.25                  0.17
1  813    39807943          1.0            0.25                  0.23
2  1010   39806602          1.0            0.00                  0.10
3  532    39904098          1.0            0.25                  0.17
4  133    39664550          1.0            0.00                  0.10
...
5  ...    ...
6  448    39854605          1.0            0.00                  0.35
7  1173   39955999          1.0            0.00                  0.01
8  846    39783631          1.0            0.50                  0.24
9  176    39947897          1.0            0.00                  0.01
10 270    39538364          1.0            0.25                  0.32
```

696 rows × 4 columns

```
In [152]: old_colname = train_data.columns.tolist()
old_colname

Out[152]: ['NCLI', 'LAMA_LANGGANAN', 'JUMLAH_LAYANAN', 'JUMLAH_TAGIHAN_HARUS_DIBAYAR']

In [153]: replaced_colname = ['id_plg','lama_langganan','jumlah_layanan','total_tagihan']

In [154]: dict_colname=dict(zip(old_colname,replaced_colname))
dict_colname

Out[154]: {'NCLI': 'id_plg',
'LAMA_LANGGANAN': 'lama_langganan',
'JUMLAH_LAYANAN': 'jumlah_layanan',
'JUMLAH_TAGIHAN_HARUS_DIBAYAR': 'total_tagihan'}
```

```
In [155]: train_data2=train_data.rename(columns=dict_colname)
```

```
In [156]: train_data2
```

```
Out[156]:
   id_plg  lama_langganan  jumlah_layanan  total_tagihan
0  979    536013           1.0            0.25      0.17
1  813    39807943          1.0            0.25      0.23
2  1010   39806602          1.0            0.00      0.10
3  532    39904098          1.0            0.25      0.17
4  133    39664550          1.0            0.00      0.10
...
5  ...    ...
6  448    39854605          1.0            0.00      0.35
7  1173   39955999          1.0            0.00      0.01
8  846    39783631          1.0            0.50      0.24
9  176    39947897          1.0            0.00      0.01
10 270    39538364          1.0            0.25      0.32
```

696 rows × 4 columns

Test Data

```
In [157]: test_data
```

	NCLI	LAMA_LANGGANAN	JUMLAH_LAYANAN	JUMLAH_TAGIHAN_HARUS_DIBAYAR
800	39732412	1.0	0.50	0.24
1227	39898433	1.0	0.25	0.24
1279	39822334	1.0	0.25	0.11
1115	40017136	1.0	0.25	0.17
773	39827352	1.0	0.50	0.14
...	...	...	...	...
141	39710293	1.0	0.00	0.38
363	39882478	1.0	0.25	0.24
505	39704813	1.0	0.25	0.01
748	39792907	1.0	0.25	0.13
191	39749623	1.0	0.00	0.01

696 rows × 4 columns

```
In [158]: old_colname = test_data.columns.tolist()
old_colname
```

```
Out[158]: ['NCLI', 'LAMA_LANGGANAN', 'JUMLAH_LAYANAN', 'JUMLAH_TAGIHAN_HARUS_DIBAYAR']
```

```
In [159]: replaced_colname = ['id_plg','lama_langganan','jumlah_layanan','total_tagihan']
```

```
In [160]: dict_colname=dict(zip(old_colname,replaced_colname))
dict_colname
```

```
Out[160]: {'NCLI': 'id_plg',
'LAMA_LANGGANAN': 'lama_langganan',
'JUMLAH_LAYANAN': 'jumlah_layanan',
'JUMLAH_TAGIHAN_HARUS_DIBAYAR': 'total_tagihan'}
```

```
In [161]: test_data2=test_data.rename(columns=dict_colname)
```

```
In [162]: test_data2
```

```
Out[162]:
```

	id_plg	lama_langganan	jumlah_layanan	total_tagihan
800	39732412	1.0	0.50	0.24
1227	39898433	1.0	0.25	0.24
1279	39822334	1.0	0.25	0.11
1115	40017136	1.0	0.25	0.17
773	39827352	1.0	0.50	0.14
...	...	...	...	...
141	39710293	1.0	0.00	0.38
363	39882478	1.0	0.25	0.24
505	39704813	1.0	0.25	0.01
748	39792907	1.0	0.25	0.13
191	39749623	1.0	0.00	0.01

696 rows × 4 columns

Gambar 5.6 Split Data Simulasi Ketiga

Pada Gambar 5.6 menjelaskan bagaimana cara melakukan *split data* untuk simulasi ketiga. Perintah untuk melakukan *split data* yaitu `train_test_split(data_cust2, test_size=.5)` dimana `train_test_split` merupakan nama fungsi, sedangkan `data_cust2` merupakan nama dataset yang digunakan, sedangkan `test_size=.5` adalah perintah membagi *test data* menjadi 50% dari total data sedangkan sisa data yang memiliki persentasi 50% akan secara otomatis menjadi *train data*. Kemudian untuk melihat jumlah data yang telah dibagi dapat dilakukan dengan perintah `print(train_data.shape, test_data.shape)`, maka akan muncul jumlah data dari *train data* dan *test data* yang telah dibagi beserta jumlah kolomnya, dan untuk melihat data mana saja yang termasuk ke dalam *train data* atau *test data* dapat memanggil data dengan perintah `train_data` atau `test_data` pada cell baru. Untuk perintah penggantian nama atribut dalam simulasi kedua memiliki penjelasan serta proses yang sama dengan yang telah dijelaskan sebelumnya pada simulasi pertama. Penggantian nama ini bersifat opsional dan tidak harus dilakukan pada setiap kali membuat sebuah model.

#### 5.2.4.5 Penentuan Jumlah *Cluster* Terbaik

Pada tahap ini penulis menentukan jumlah *cluster* terbaik dengan cara menghitung nilai *Sum Square Error* (SSE) kemudian memvisualisasikannya dengan menggunakan metode Elbow dimana hasil perhitungan SSE digambarkan dalam bentuk siku lalu nilai yang memiliki penurunan secara drastis merupakan jumlah K yang optimal. Pada Gambar 5.7 ditunjukkan jumlah K yang optimal untuk simulasi pertama dan simulasi kedua yaitu 3 dikarenakan pada posisi tersebut membentuk siku.

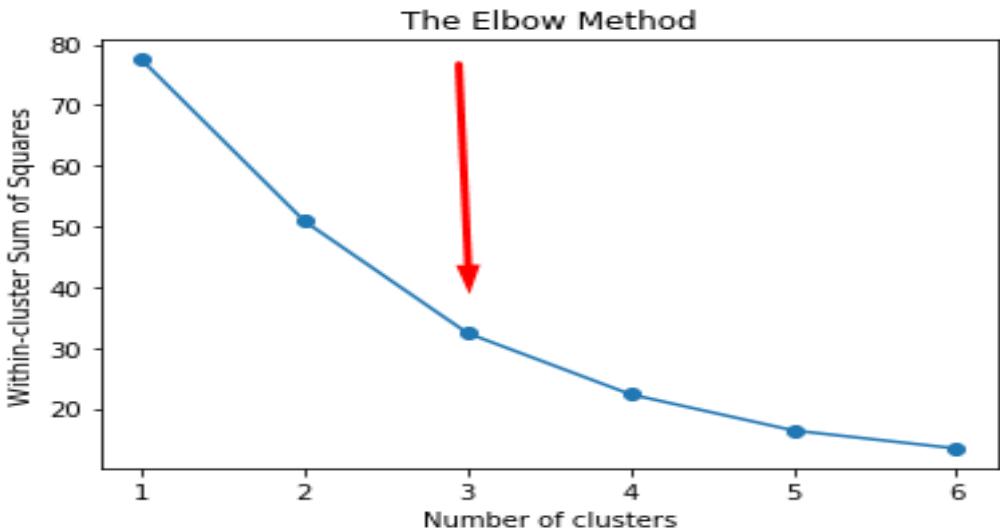
```
In [23]: wcss = []
for i in range(1,7):
    kmeans = KMeans(i)
    kmeans.fit(train_data_x)
    wcss_iter = kmeans.inertia_
    wcss.append(wcss_iter)

print(wcss)

number_clusters = range(1,7)
plt.plot(number_clusters,wcss, marker='o')
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('Within-cluster Sum of Squares')

[77.52515956873316, 50.879166957796585, 32.42201391310743, 22.32995652515045, 16.37458478337056, 13.395513311526756]

Out[23]: Text(0, 0.5, 'Within-cluster Sum of Squares')
```



Gambar 5.7 Penentuan Jumlah *Cluster* Terbaik

#### 5.2.4.6 K-Means *Clustering*

##### 5.2.4.6.1 Simulasi Pertama

Pada simulasi pertama penulis menggunakan persentase pembagian *train data* dan *test data* 80% 20%. Dimana masing-masing dari *train data* dan *test data* akan dilakukan proses *clustering* menggunakan algoritma K-Means.

- *Train Data*

1. Inisialisasi data

Proses ini dilakukan untuk menginisialisasi data menggunakan *train data* serta pemilihan atribut yang digunakan untuk proses *clustering*.

Pada Gambar 5.8 dilakukan inisialisasi *train\_data2* dengan nilai yang berada pada kolom *lama\_langganan*, *jumlah\_layanan*, dan *total\_tagihan* pada variabel X.

```
In [19]: X = train_data2[['lama_langganan', 'jumlah_layanan', 'total_tagihan']]
```

Gambar 5.8 Inisialisasi *Train data* Simulasi Pertama

2. Model *Clustering*

Proses ini merupakan proses pembuatan model *clustering* ditandai dengan adanya perintah *km.fit(X)* dan K yang digunakan untuk proses *clustering* merupakan hasil penentuan jumlah *cluster* terbaik yang telah dibahas pada sub bab sebelumnya yaitu berjumlah 3 seperti yang ditunjukan pada Gambar 5.9.

```
In [20]: km = KMeans(n_clusters=3)
km.fit(X)

Out[20]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                 n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
                 random_state=None, tol=0.0001, verbose=0)
```

Gambar 5.9 Model *Clustering Train Data* Simulasi Pertama

3. Menampilkan Centroid

Proses ini dilakukan untuk melihat centroid atau titik pusat yang digunakan selama proses *clustering* untuk *train data*. Pada gambar 5.10 perintah yang digunakan untuk menentukan centroid yaitu *km.clusters\_center\_* yang mana fungsi ini telah disediakan oleh *library*

sklearn.clusters untuk operasi yang dibuthkan oleh algoritma K-Means salahsatunya yaitu penentuan centroid. Kemudian hasil penentuan centroid tersebut diinisialisasikan pada variabel centroid\_train\_data dan ditampilkan dengan menggunakan perintah print(centroid\_train\_data).

```
In [21]: centroid_train_data = km.cluster_centers_
print(centroid_train_data)

[[9.94363257e-01 4.16333634e-16 1.00083507e-01]
 [9.96869852e-01 2.98187809e-01 2.00214168e-01]
 [1.55555556e-01 2.12962963e-01 1.99629630e-01]]
```

Gambar 5.10 Centroid *Train Data* Simulasi Pertama

#### 4. Proses *Clustering*

Pada tahap ini proses *clustering* dilakukan dimana data akan di *predict* sesuai dengan model model *clustering* X seperti yang dapat dilihat pada Gambar 5.11, kemudian data yang telah di predict tersebut diberikan label menggunakan perintah km.labels\_ yang diinisialisasikan pada variabel cluster\_train\_labels. Selain itu pada tahap ini juga ditampilkan label *cluster* dari masing-masing data dengan perintah print(cluster\_train\_labels).

```
In [22]: train_data2['cluster'] = km.predict(X)

In [23]: cluster_train_labels = km.labels_
print(cluster_train_labels)

[1 0 1 ... 1 2 0]
```

Gambar 5.11 Proses *Clustering Train Data* Simulasi Pertama

In [24]: train_data2.head(1113)					
Out[24]:					
	id_plg	lama_langganan	jumlah_layanan	total_tagihan	cluster
1323	39854780	1.0	0.25	0.17	1
1284	39869493	1.0	0.00	0.10	0
363	39882478	1.0	0.25	0.24	1
726	39986673	1.0	0.25	0.56	1
214	39902991	1.0	0.00	0.01	0
...	...	...	...	...	...
606	39912123	1.0	0.00	0.10	0
384	39727301	1.0	0.50	0.24	1
1125	33507802	1.0	0.25	0.17	1
182	40035380	0.3	0.00	0.01	2
1262	39799153	1.0	0.00	0.10	0

1113 rows × 5 columns

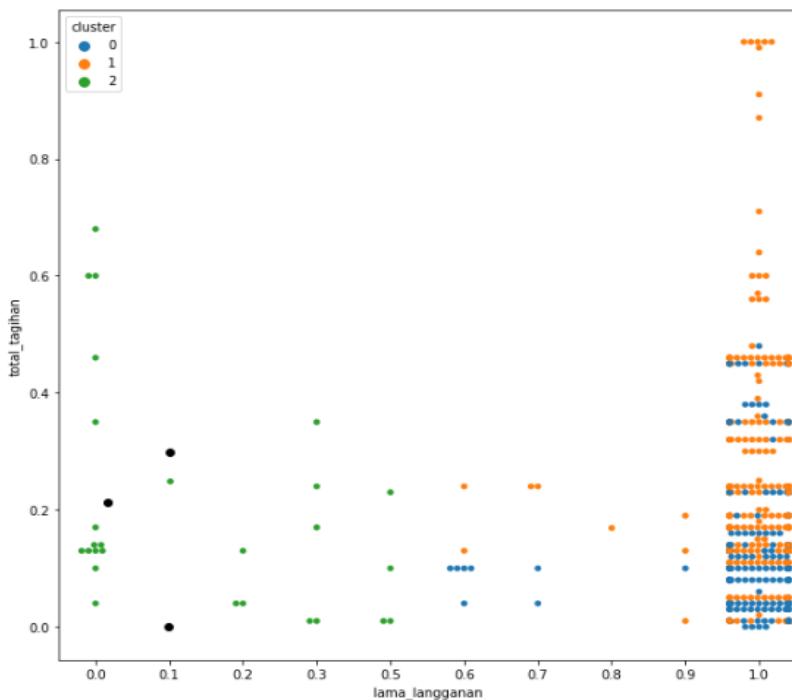
Gambar 5.12 Hasil *Clustering Train Data* Simulasi Pertama

## 5. Visualisasi

Pada tahap ini dilakukan visualisasi penyebaran data dari masing-masing *cluster*, dikarenakan visualisasi dilakukan dengan grafik 2 dimensi sehingga hanya memiliki 2 sumbu dimana sumbu tersebut merupakan pasangan atribut yang digunakan selama *clustering* yaitu lama\_langganan dan total\_tagihan, lama\_langganan dan jumlah\_layanan, serta jumlah\_layanan dan total tagihan. Dalam visualisasi ini, digunakan *library* seaborn dengan pola swarmplot supaya bila ada titik yang berhimpit masih bisa dilihat dengan jelas. Untuk perintah visualisasi yang lebih jelas bisa dilihat pada gambar 5.13.

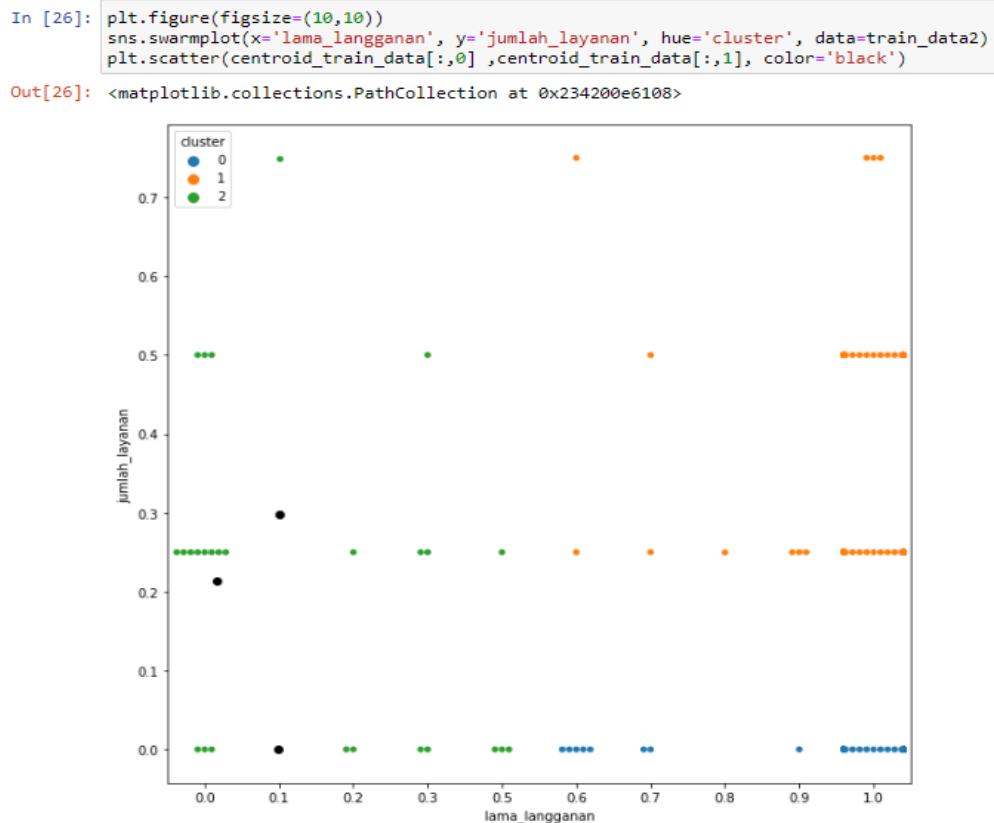
```
In [25]: plt.figure(figsize=(10,10))
sns.swarmplot(x='lama_langganan', y='total_tagihan', hue='cluster', data=train_data2)
plt.scatter(centroid_train_data[:,0], centroid_train_data[:,1], color='black')

Out[25]: <matplotlib.collections.PathCollection at 0x2342009bf08>
```



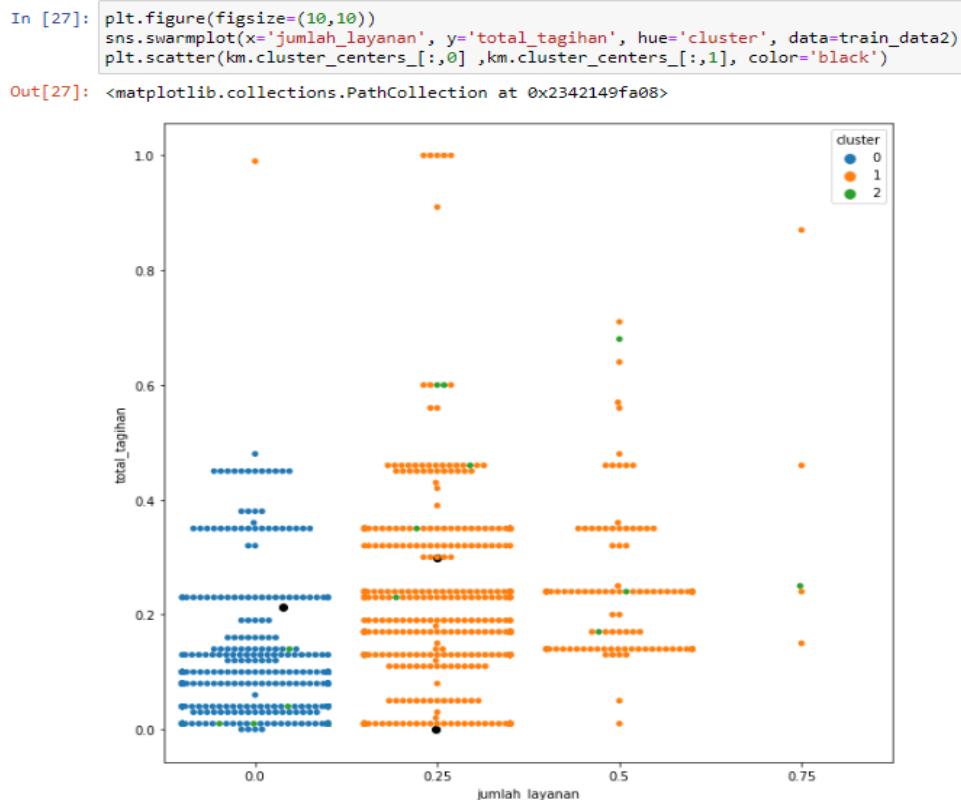
Gambar 5.13 Visualisasi Atribut lama\_langganan dan total\_tagihan Pada *Train Data* Simulasi Pertama

Pada Gambar 5.13 menjelaskan bahwa titik yang mewakili pelanggan yang berada pada posisi semakin ke kanan dari sumbu x merupakan pelanggan yang memiliki jumlah lama langganan semakin lama dan kemudian bila titik tersebut semakin ke atas dari sumbu y maka pelanggan tersebut memiliki jumlah total tagihan yang semakin tinggi. Kemudian untuk centroid ditandai dengan titik warna hitam.



Gambar 5.14 Visualisasi Atribut lama\_langganan dan jumlah\_layanan Pada *Train Data* Simulasi Pertama

Pada Gambar 5.14 menjelaskan bahwa titik yang mewakili pelanggan yang berada pada posisi semakin ke kanan dari sumbu x merupakan pelanggan yang memiliki jumlah lama langganan semakin lama dan kemudian bila titik tersebut semakin ke atas dari sumbu y maka pelanggan tersebut memiliki total jumlah layanan yang semakin banyak. Kemudian untuk centroid ditandai dengan titik warna hitam.



Gambar 5.15 Visualisasi Atribut jumlah\_layanan dan total\_tagihan Pada *Train Data* Simulasi Pertama

Pada Gambar 5.14 menjelaskan bahwa titik yang mewakili pelanggan yang berada pada posisi semakin ke kanan dari sumbu x merupakan pelanggan yang memiliki jumlah layanan semakin banyak dan kemudian bila titik tersebut semakin ke atas dari sumbu y maka pelanggan tersebut memiliki jumlah total tagihan yang semakin tinggi. Kemudian untuk centroid ditandai dengan titik warna hitam.

## 6. Evaluasi *Cluster*

Evaluasi *cluster* dilakukan untuk melihat *performance* dari *clustering test data*. Adapun proses evaluasi *cluster* dilakukan dengan metode Silhouette Score, Davies Bouldin Index, dan Calinski Harabasz Index.

### Train Data

```
In [45]: silhouette_score(X,cluster_train_labels)
```

```
Out[45]: 0.5350956648481786
```

```
In [46]: silhouette_score(X,cluster_train_labels == 0)
```

```
Out[46]: 0.4820330999095601
```

```
In [47]: silhouette_score(X,cluster_train_labels == 1)
```

```
Out[47]: 0.500247647380693
```

```
In [48]: silhouette_score(X,cluster_train_labels == 2)
```

```
Out[48]: 0.7034674179192978
```

Gambar 5.16 Silhouette Score Pada *Train Data* Simulasi Pertama

**Train Data**

```
In [53]: davies_bouldin_score(X, cluster_train_labels)
Out[53]: 0.6704737639961774

In [54]: davies_bouldin_score(X, cluster_train_labels == 0)
Out[54]: 0.8478574492536206

In [55]: davies_bouldin_score(X, cluster_train_labels == 1)
Out[55]: 0.964327590303711

In [56]: davies_bouldin_score(X, cluster_train_labels == 2)
Out[56]: 0.6025750787520832
```

Gambar 5.17 Davies Bouldin Index Pada *Train Data* Simulasi Pertama**Train Data**

```
In [61]: metrics.calinski_harabasz_score(X, cluster_train_labels)
Out[61]: 806.1070246452166

In [62]: metrics.calinski_harabasz_score(X, cluster_train_labels == 0)
Out[62]: 598.2878797485031

In [63]: metrics.calinski_harabasz_score(X, cluster_train_labels == 1)
Out[63]: 567.4546738270112

In [64]: metrics.calinski_harabasz_score(X, cluster_train_labels == 2)
Out[64]: 360.7280546541717
```

Gambar 5.18 Calinski Harabasz Index Pada *Train Data* Simulasi Pertama

- *Test Data*

1. Inisialisasi data

Proses ini dilakukan untuk menginisialisasi data menggunakan *test data* serta pemilihan atribut yang digunakan untuk proses *clustering*. Dimana semua *test data* yang berada pada kolom lama\_langganan, jumlah\_layanan, dan total\_tagihan diinisialisasikan pada variabel Z seperti yang ditunjukkan pada Gambar 5.19.

```
In [32]: Z = test_data2[['lama_langganan', 'jumlah_layanan', 'total_tagihan']]
```

Gambar 5.19 Inisialisasi *Test Data* Simulasi Pertama

2. Model *Clustering*

Gambar 5.20 menjelaskan proses pembuatan model *clustering* ditandai dengan adanya perintah km.fit(Z) dan K yang digunakan selama proses *clustering* merupakan hasil penentuan jumlah *cluster* terbaik yang telah dibahas pada sub bab sebelumnya yaitu berjumlah 3 serta jumlah K yang sama dengan yang ada pada model *clustering train data*.

```
In [33]: km1 = KMeans(n_clusters=3)
km1.fit(Z)

Out[33]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                 n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
                 random_state=None, tol=0.0001, verbose=0)
```

Gambar 5.20 Model *Clustering Test Data* Simulasi Pertama

3. Menampilkan Centroid

Proses ini yang dilakukan pada Gambar 5.21 menjelaskan cara bagaimana melihat centroid atau titik pusat yang digunakan selama proses *clustering* dari *test data*.

```
In [34]: centroid_test_data = km1.cluster_centers_
print(centroid_test_data)

[[9.5092025e-01 2.99079755e-01 2.18466258e-01]
 [1.00000000e+00 3.88578059e-16 1.06574074e-01]
 [6.25000000e-02 1.56250000e-01 1.56250000e-01]]
```

Gambar 5.21 Centroid *Test Data* Simulasi Pertama

#### 4. Proses *Clustering*

Pada tahap ini proses *clustering* dilakukan dimana *test data* dari simulasi pertama akan di *predict* sesuai dengan model *model clustering Z*. Selain itu pada tahap ini juga ditampilkan label *cluster* dari masing-masing data seperti yang ditunjukkan pada Gambar 5.22.

Gambar 5.22 Proses *Clustering Test Data* Simulasi Pertama

In [37]:	test_data2.head(279)					
Out[37]:		id_plg	lama_langganan	jumlah_layanan	total_tagihan	cluster
	716	39973425	1.0	0.25	0.32	0
	432	39877222	1.0	0.25	0.35	0
	383	39714012	1.0	0.50	0.24	0
	559	39884237	1.0	0.00	0.04	1
	252	40006611	1.0	0.00	0.08	1
	...	...	...	...	...	...
	534	34273476	1.0	0.25	0.17	0
	76	39926616	1.0	0.25	0.17	0
	442	39916158	1.0	0.25	0.35	0
	987	39780306	1.0	0.00	0.10	1
	518	39504146	1.0	0.75	0.02	0

279 rows × 5 columns

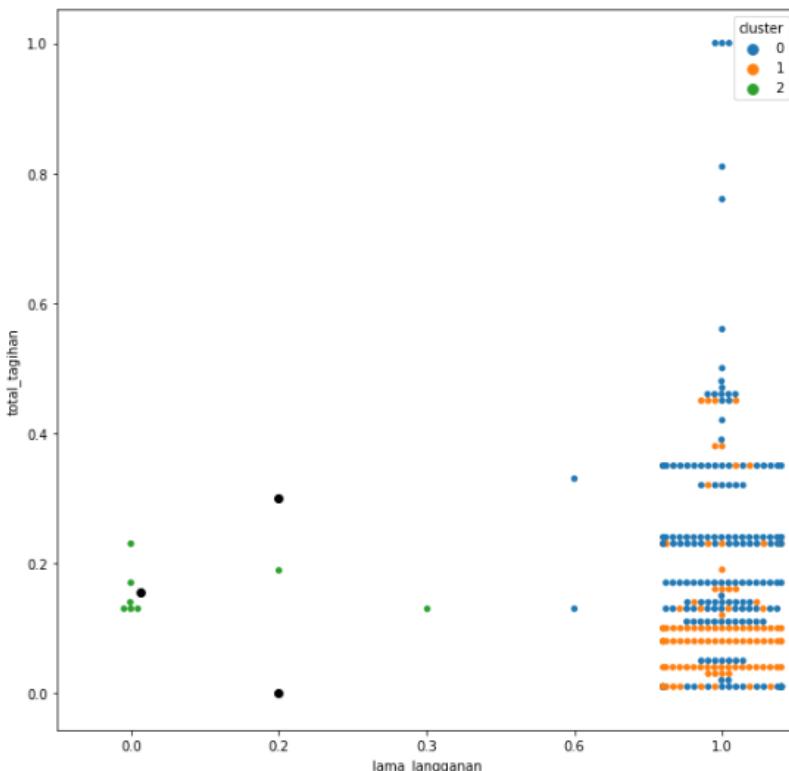
Gambar 5.23 Hasil *Clustering Test Data* Simulasi Pertama

## 5. Visualisasi

Pada tahap ini dilakukan visualisasi penyebaran data dari masing-masing *cluster*, dikarenakan visualisasi dilakukan dengan grafik 2 dimensi sehingga hanya memiliki 2 sumbu dimana sumbu tersebut merupakan atribut yang digunakan selama *clustering* yaitu *lama\_langganan* dan *total\_tagihan*, *lama\_langganan* dan *jumlah\_layanan*, serta *jumlah\_layanan* dan *total tagihan*.

```
In [38]: plt.figure(figsize=(10,10))
sns.swarmplot(x='lama_langganan', y='total_tagihan', hue='cluster', data=test_data2)
plt.scatter(centroid_test_data[:,0] ,centroid_test_data[:,1], color='black')

Out[38]: <matplotlib.collections.PathCollection at 0x2342150d3c8>
```

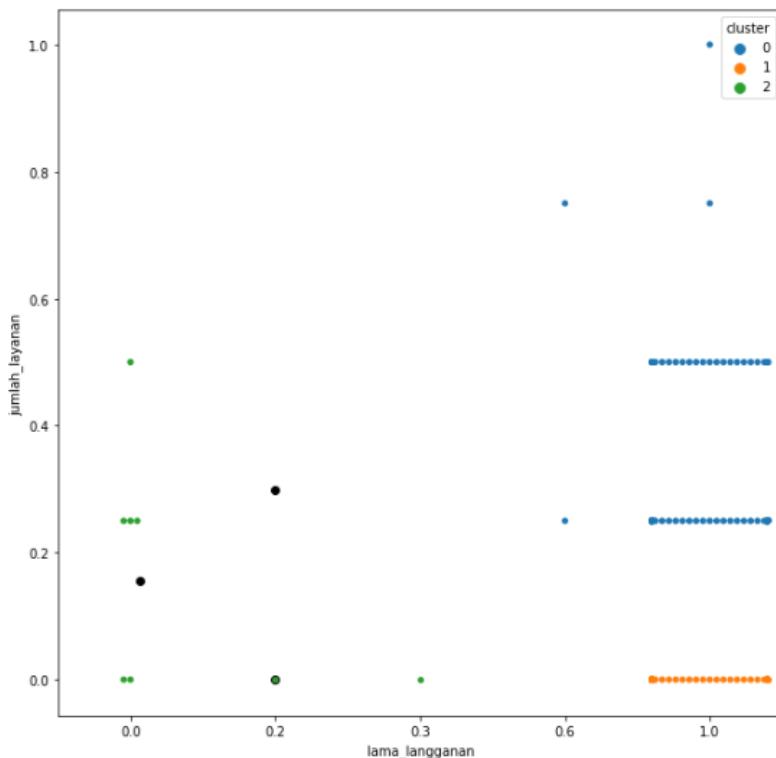


Gambar 5.24 Visualisasi Atribut lama\_langganan dan total\_tagihan Pada *Test Data* Simulasi Pertama

Pada Gambar 5.24 merupakan hasil visualisasi dari dua atribut yaitu lama\_langganan dan total\_tagihan. Jika titik dari data berada pada sumbu x dengan posisi semakin berada di kanan maka, masa waktunya berlangganan semakin lama dan bila titik berada di sumbu y dengan posisi semakin ke atas maka total tagihan yang dimiliki pelanggan semakin tinggi. Kemudian titik yang berwarna hitam mewakili centroid yang digunakan selama proses clustering.

```
In [39]: plt.figure(figsize=(10,10))
sns.swarmplot(x='lama_langganan', y='jumlah_layanan', hue='cluster', data=test_data2)
plt.scatter(centroid_test_data[:,0] ,centroid_test_data[:,1], color='black')

Out[39]: <matplotlib.collections.PathCollection at 0x23421894ac8>
```

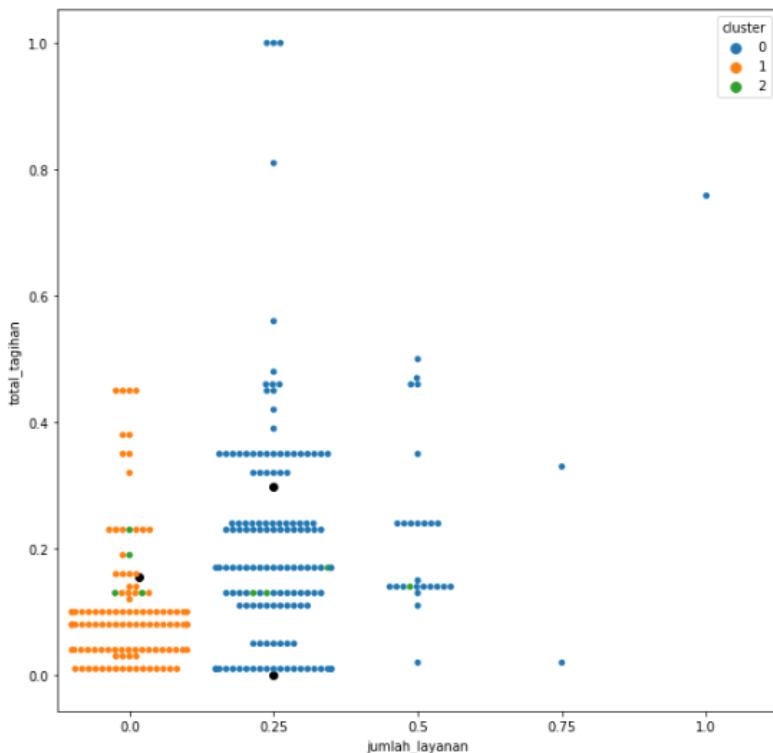


Gambar 5.25 Visualisasi Atribut lama\_langganan dan jumlah\_layanan Pada *Test Data* Simulasi Pertama

Pada Gambar 5.25 merupakan hasil visualisasi dari dua atribut yaitu lama\_langganan dan jumlah\_layanan. Jika titik dari data berada pada sumbu x dengan posisi semakin berada di kanan maka, masa waktunya berlangganan semakin lama dan bila titik berada di sumbu y dengan posisi semakin ke atas maka jumlah layanan yang digunakan pelanggan semakin banyak. Kemudian titik yang berwarna hitam mewakili centroid yang digunakan selama proses clustering.

```
In [40]: plt.figure(figsize=(10,10))
sns.swarmplot(x='jumlah_layanan', y='total_tagihan', hue='cluster', data=test_data)
plt.scatter(centroid_test_data[:,0] ,centroid_test_data[:,1], color='black')

Out[40]: <matplotlib.collections.PathCollection at 0x2342190a508>
```



Gambar 5.26 Visualisasi Atribut jumlah\_layanan dan total\_tagihan Pada *Test Data* Simulasi Pertama

## 6. Evaluasi *Cluster*

Evaluasi *cluster* dilakukan untuk melihat *performance* dari *clustering test data*. Adapun proses evaluasi *cluster* dilakukan dengan metode Silhouette Score, Davies Bouldin Index, dan Calinski Harabasz Index.

Test Data

```
In [49]: silhouette_score(Z,cluster_test_labels)
Out[49]: 0.5076609088001013

In [50]: silhouette_score(Z,cluster_test_labels == 0)
Out[50]: 0.4700153277634574

In [51]: silhouette_score(Z,cluster_test_labels == 1)
Out[51]: 0.4327521015524476

In [52]: silhouette_score(Z,cluster_test_labels == 2)
Out[52]: 0.7167927673208508
```

Gambar 5.27 Silhouette score pada *Test Data* Simulasi Pertama

Test Data

```
In [57]: davies_bouldin_score(Z, cluster_test_labels)
Out[57]: 0.6297372027289669

In [58]: davies_bouldin_score(Z, cluster_test_labels == 0)
Out[58]: 1.0529206838064291

In [59]: davies_bouldin_score(Z, cluster_test_labels == 1)
Out[59]: 0.8960073898002592

In [60]: davies_bouldin_score(Z, cluster_test_labels == 2)
Out[60]: 0.42555450351861807
```

Gambar 5.28 Davies Bouldin Index pada *Test Data* Simulasi Pertama

#### Test Data

```
In [65]: metrics.calinski_harabasz_score(Z, cluster_test_labels)
Out[65]: 198.52238679062506

In [66]: metrics.calinski_harabasz_score(Z, cluster_test_labels == 0)
Out[66]: 114.97859050345302

In [67]: metrics.calinski_harabasz_score(Z, cluster_test_labels == 1)
Out[67]: 113.25878359780417

In [68]: metrics.calinski_harabasz_score(Z, cluster_test_labels == 2)
Out[68]: 117.95595010741032
```

Gambar 5.29 Calinski Harabasz Index pada *Test Data* Simulasi Pertama

#### 5.2.4.6.2 Simulasi Kedua

Pada simulasi kedua, penulis melakukan proses *clustering* dengan menggunakan persentase *train data* dan *test data* sebanyak 70% dan 30%.

- *Train Data*

1. Inisialisasi data

Proses ini dilakukan untuk menginisialisasi data menggunakan *train data* serta pemilihan atribut yang digunakan untuk proses *clustering* pada simulasi kedua. Dalam tahap inisialisasi data, semua *train data* yang akan diproses dalam *clustering* ditampung dalam sebuah variabel yang diberi nama X, seperti terlihat pada Gambar 5.30.

```
In [18]: X = train_data2[['lama_langganan', 'jumlah_layanan', 'total_tagihan']]
```

Gambar 5.30 Inisialisasi *Train Data* Simulasi Kedua

## 2. Model *Clustering*

Proses yang ditunjukan pada Gambar 5.31 merupakan proses pembuatan model *clustering* ditandai dengan adanya perintah `km.fit(X)` dan K yang digunakan untuk proses *clustering* merupakan hasil penentuan jumlah *cluster* terbaik yang telah dibahas pada sub bab sebelumnya yaitu berjumlah 3.

```
In [19]: km = KMeans(n_clusters=3)
km.fit(X)

Out[19]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                 n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
                 random_state=None, tol=0.0001, verbose=0)
```

Gambar 5.31 Model *Clustering* Simulasi Kedua

## 3. Menampilkan Centroid

Proses ini dilakukan untuk melihat centroid atau titik pusat yang digunakan selama proses *clustering*. Perintah yang digunakan untuk menentukan centroid yaitu `km.cluster_centers_` yang kemudian diinisialisasikan dalam variabel `centroid_train_data`. Hal tersebut dilakukan untuk mempermudah yang menampilkan centroid, yaitu hanya dengan mengetikan perintah `print` yang diikuti nama dari variabelnya seperti yang terlihat pada Gambar 5.32.

```
In [20]: centroid_train_data = km.cluster_centers_
print(centroid_train_data)

[[9.96411483e-01 5.55111512e-17 9.72248804e-02]
 [9.96219282e-01 2.98676749e-01 2.06030246e-01]
 [1.11111111e-01 2.03703704e-01 1.89629630e-01]]
```

Gambar 5.32 Centroid *Train Data* Simulasi Kedua

#### 4. Proses *Clustering*

Pada proses *clustering* yang dilakukan pada Gambar 5.33, ditunjukkan bahwa data yang akan di *predict* adalah data yang telah diinisialisasikan sesuai dengan model *clustering* X. Selain itu pada tahap ini juga ditampilkan label *cluster* dari masing-masing data.

```
In [21]: train_data2['cluster'] = km.predict(X)

In [22]: cluster_train_labels = km.labels_
print(cluster_train_labels)

[0 1 0 1 1 1 1 0 0 1 2 0 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 0 0 1 1 2 1 1 0
 1 0 0 1 0 0 1 0 0 0 0 1 1 1 0 1 1 1 0 0 0 1 1 1 1 0 1 1 0 1 1 0 0 1 0 0 1
 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0
 1 1 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 1 1 1 0
 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 0 0 1 0 0
 0 1 1 1 1 1 0 1 0 1 0 1 0 1 1 0 2 0 1 1 0 1 0 0 1 1 1 0 1 0 1 1 0 1 1 0 0
 1 0 1 0 0 1 1 0 1 1 1 1 0 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1 0 1 0
 1 1 0 1 1 1 1 0 1 1 0 1 0 1 1 1 1 0 0 1 2 1 1 0 1 0 1 2 1 1 0 1 0 1 0
 1 1 0 0 1 1 1 1 0 1 1 1 0 0 0 1 0 1 0 2 1 0 1 0 0 1 0 0 1 0 1 0 1 0 1 0
 0 0 0 0 0 0 0 0 1 1 1 2 1 0 1 0 1 1 0 0 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 1
 0 0 1 0 0 1 1 1 0 1 1 1 0 1 1 1 0 0 0 1 1 1 0 0 0 0 1 1 1 0 1 1 0 1 1 0
 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 1 1 2 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1
 0 1 1 1 0 0 1 0 1 0 1 0 1 1 0 2 1 1 1 1 0 0 0 1 2 2 2 1 1 0 1 1 0 1 0
 0 1 1 0 0 0 1 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1
 1 0 0 1 0 1 0 0 1 1 0 0 1 1 1 0 1 0 1 0 0 0 1 1 1 0 0 1 1 0 1 0 1 0 0 1
 1 0 1 1 0 0 1 0 0 2 0 0 0 1 0 1 0 1 1 1 1 1 0 1 1 1 0 0 1 0 1 0 0 1 1 1
 1 1 0 1 1 0 0 1 0 0 1 0 0 1 1 1 0 0 0 1 1 1 0 1 0 0 2 1 1 1 1 1 0 0 0
 1 1 1 1 2 0 1 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 2 1 0 0 1 1 1 1 0 1 1 0 1
 1 1 0 0 0 0 0 1 0 1 1 1 1 1 1 1 0 1 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0
 1 1 1 1 2 0 0 1 1 0 1 0 1 1 1 1 0 2 0 1 1 0 1 1 1 1 1 1 0 1 1 0 1 0 1 1 0
 1 1 1 1 1 1 1 0 0 0 0 1 0 0 1 1 1 1 1 1 1 0 1 2 1 1 1 1 0 1 0 0 1 1 0 0
 1 0 2 0 0 0 1 0 0 1 1 0 0 1 0 1 1 1 0 1 1 0 1 0 0 1 1 2 0 1 1 0 1 1 1 1 1
 1 0 2 1 1 0 1 1 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 1 1 1 0 1 0 1 0 0 1 0 0 1 0 2
 1 1 1 0 1 0 0 0 0 0 1 1 0 2 0 1 0 1 1 1 1 1 0 0 0 1 2 0 0 1 0 1 1 1 0 1
 0 1 1 1 1 1 1 0 0 0 0 1 0 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 1
 1 0 1 1 0 0 1 1 1 0 0 1 0 1 1 1 0 0 1 1 1 0 1 0 1 0 0 1 1 0 1 0 0 1 0 1 0 0 1
 0 2 0 1 0 1 0 1 1 0 0 1]
```

Gambar 5.33 Proses *Clustering Train Data* Simulasi Kedua

Kemudian setelah melewati proses clustering dan diberikan label pada masing-masing data, dapat dilihat bahwa data set untuk *train data* telah memiliki label yang menunjukkan data berada pada cluster mana seperti ditunjukkan pada Gambar 5.34.

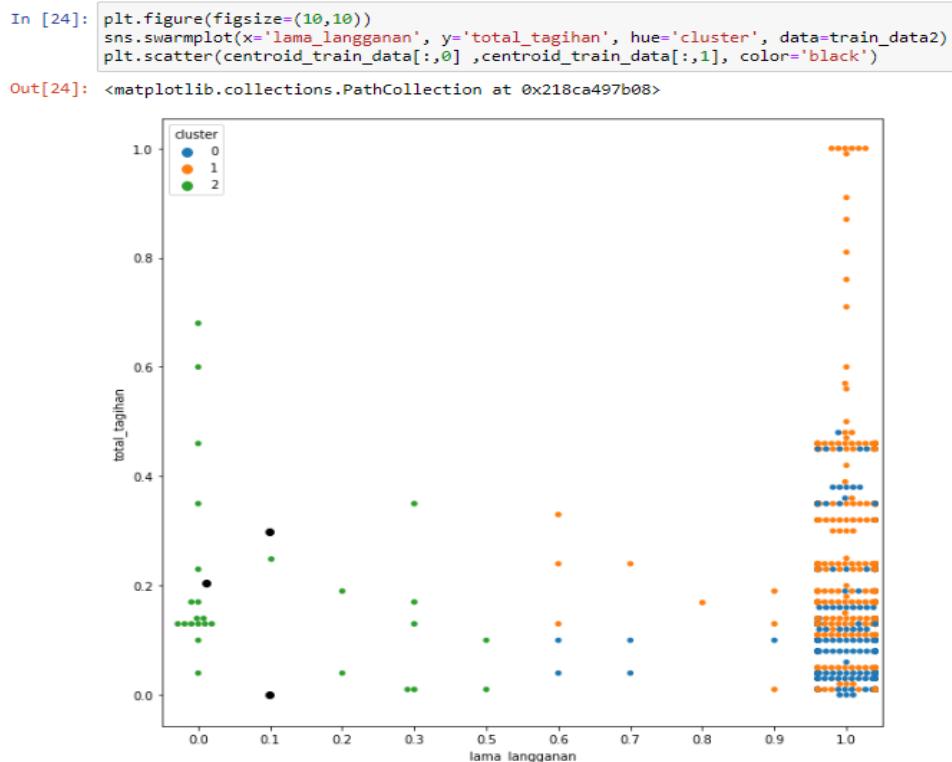
In [23]: train_data2.head(974)					
Out[23]:					
	id_plg	lama_langganan	jumlah_layanan	total_tagihan	cluster
257	40017030	1.0	0.00	0.08	0
492	39921776	1.0	0.25	0.46	1
125	39728660	1.0	0.00	0.10	0
526	39756118	1.0	0.25	0.17	1
1317	39971395	1.0	0.25	0.13	1
...	...	...	...	...	...
506	39699772	1.0	0.25	0.01	1
1231	39936577	1.0	0.25	0.24	1
1160	39856035	1.0	0.00	0.08	0
986	39992603	1.0	0.00	0.10	0
1204	39871209	1.0	0.25	0.24	1

974 rows × 5 columns

Gambar 5.34 Hasil *Clustering Train Data* Simulasi Kedua

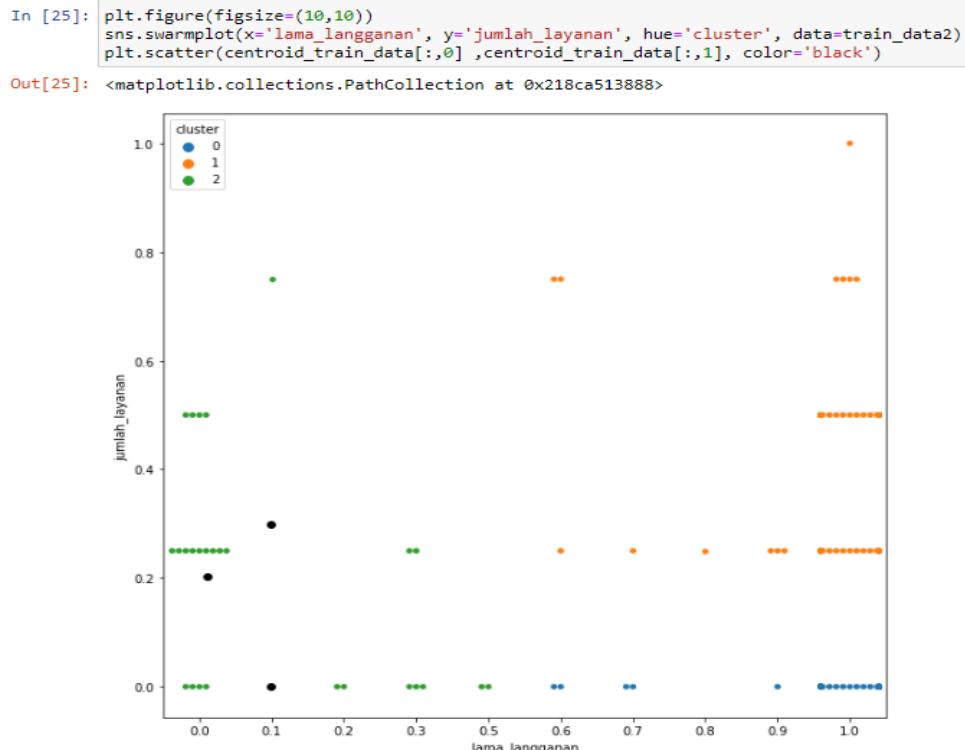
## 5. Visualisasi

Pada tahap ini dilakukan visualisasi penyebaran data dari masing-masing *cluster*, dikarenakan visualisasi dilakukan dengan grafik 2 dimensi sehingga hanya memiliki 2 sumbu dimana sumbu tersebut merupakan pasangan atribut yang digunakan selama *clustering* yaitu *lama\_langganan* dan *total\_tagihan*, *lama\_langganan* dan *jumlah\_layanan*, serta *jumlah\_layanan* dan *total tagihan*. *Library* yang digunakan untuk visualisasi adalah *library* seaborn dengan bentuk grafik yaitu swarmplot. Seaborn dipilih karena memiliki visualisasi yang lebih menarik serta memiliki pola grafik yang berbeda. Walaupun menggunakan seaborn, namun *library* matplotlib harus diload dikarenakan seaborn berada dibawah *library* matplotlib.



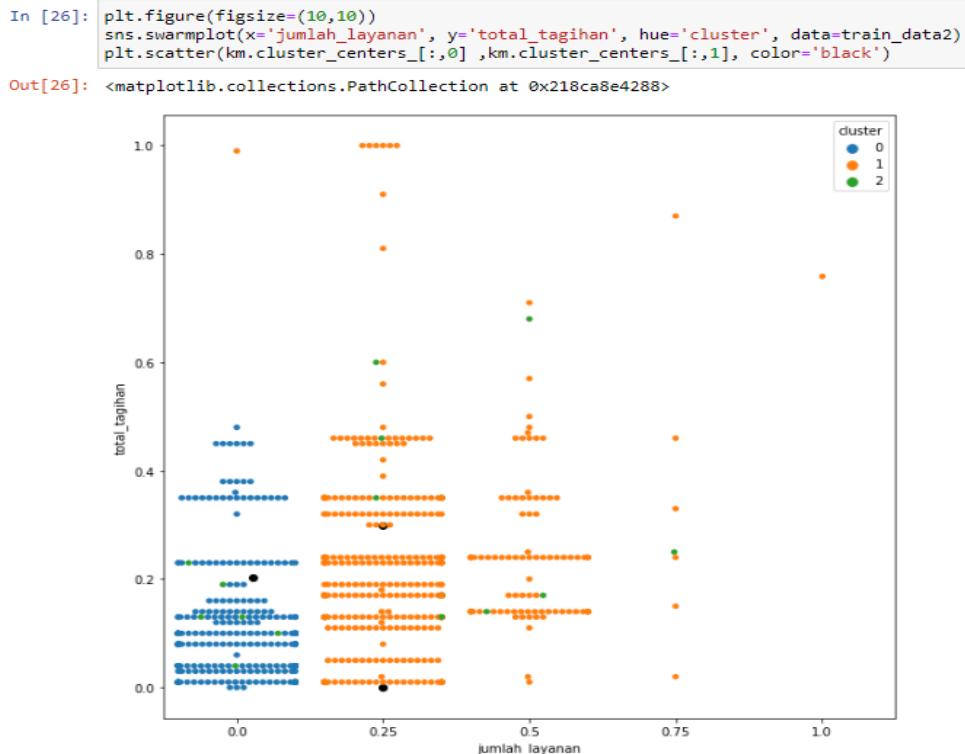
Gambar 5.35 Visualisasi Atribut *lama\_langganan* dan *total\_tagihan* Pada *Train Data* Simulasi Kedua

Visualisasi yang ditunjukkan pada Gambar 5.35 merupakan visualisasi dalam bentuk grafik dari hasil proses *clustering* menggunakan *train data* yang dilakukan pada simulasi kedua. Jika posisi titik berada semakin ke kanan dari sumbu x maka waktu lama langganan dari pelanggan semakin lama. Sedangkan bila posisi titik berada semain kea atas dari sumbu y maka total tagihan dari pelanggan semakin tinggi. Titik yang berada grafik memiliki warna yang berbeda sesuai dengan clusternya seperti yang terlihat pada keterangan gambar, kemudian untuk titik yang berwarna hitam merupakan centroid yang digunakan selama proses clustering.



Gambar 5.36 Visualisasi Atribut *lama\_langganan* dan *jumlah\_layanan* Pada *Train Data* Simulasi Kedua

Pada Gambar 5.36 merupakan visualisasi dalam bentuk grafik dari hasil proses *clustering* menggunakan *train data* yang dilakukan pada simulasi kedua. Jika posisi titik berada semakin ke kanan dari sumbu x maka waktu lama langganan dari pelanggan semakin lama. Sedangkan bila posisi titik berada semakin ke atas dari sumbu y maka jumlah layanan yang digunakan pelanggan semakin banyak. Titik yang berada grafik memiliki warna yang berbeda sesuai dengan clusternya seperti yang terlihat pada keterangan gambar, kemudian untuk titik yang berwarna hitam merupakan centroid yang digunakan selama proses clustering.



Gambar 5.37 Visualisasi Atribut jumlah\_layanan dan total\_tagihan Pada *Train Data* Simulasi Kedua

Grafik yang ditunjukkan pada Gambar 5.36 merupakan visualisasi dari hasil proses *clustering* menggunakan *train data* yang dilakukan pada simulasi kedua. Jika posisi titik berada semakin ke kanan dari sumbu x jumlah layanan yang digunakan oleh pelanggan semakin banyak. Sedangkan bila posisi titik berada semakin ke atas dari sumbu y maka menunjukkan total tagihan yang dibayar pelanggan semakin tinggi. Titik yang berada grafik memiliki warna yang berbeda sesuai dengan clusternya seperti yang terlihat pada keterangan gambar, kemudian untuk titik yang berwarna hitam merupakan centroid yang digunakan selama proses clustering.

## 6. Evaluasi *Cluster*

Evaluasi *cluster* dilakukan untuk melihat *performance* dari *clustering* darta Test. Adapun proses evaluasi *cluster* dilakukan dengan metode Silhouette Score, Davies Bouldin Index, dan Calinski Harabasz Index.

### Train Data

```
In [38]: silhouette_score(X,cluster_train_labels)
Out[38]: 0.5392900264263295

In [39]: silhouette_score(X,cluster_train_labels == 0)
Out[39]: 0.4735719632745629

In [40]: silhouette_score(X,cluster_train_labels == 1)
Out[40]: 0.4977341627871874

In [41]: silhouette_score(X,cluster_train_labels == 2)
Out[41]: 0.7158037727704025
```

Gambar 5.38 Silhouette Score Pada *Train Data* Simulasi Kedua

### Train Data

```
In [46]: davies_bouldin_score(X, cluster_train_labels)
Out[46]: 0.6405603202094682

In [47]: davies_bouldin_score(X, cluster_train_labels == 0)
Out[47]: 0.8529308406684851

In [48]: davies_bouldin_score(X, cluster_train_labels == 1)
Out[48]: 0.9883456250748196

In [49]: davies_bouldin_score(X, cluster_train_labels == 2)
Out[49]: 0.5428227526954669
```

Gambar 5.39 Davies Bouldin Index pada *Train Data* Simulasi Kedua

## Train Data

```
In [54]: metrics.calinski_harabasz_score(X, cluster_train_labels)
Out[54]: 751.766981512962

In [55]: metrics.calinski_harabasz_score(X, cluster_train_labels == 0)
Out[55]: 474.07023741236895

In [56]: metrics.calinski_harabasz_score(X, cluster_train_labels == 1)
Out[56]: 451.29929328033074

In [57]: metrics.calinski_harabasz_score(X, cluster_train_labels == 2)
Out[57]: 384.504254023288
```

Gambar 5.40 Calinski Harabasz Index Pada *Train Data* Simulasi Kedua• *Test Data*

## 1. Inisialisasi data

Proses ini dilakukan untuk menginisialisasi data menggunakan *test data* serta pemilihan atribut yang digunakan untuk proses *clustering*. *Test data* yang akan di *cluster* yang berada pada kolom *lama\_langganan*, *jumlah\_layanan*, dan *total\_tagihan* diinisialisasikan pada sebuah variabel yang diberi nama Z untuk mempermudah membuat model *cluster* serta saat proses *clustering*. Cara inisialisasi data ke dalam sebuah variabel bisa dilihat pada Gambar 5.41.

```
In [28]: Z = test_data2[['lama_langganan', 'jumlah_layanan', 'total_tagihan']]
```

Gambar 5.41 Inisialisasi *Test Data* Simulasi Kedua2. Model *Clustering*

Proses ini merupakan proses pembuatan model *clustering* ditandai dengan adanya perintah *km.fit(Z)* dan K yang digunakan selama proses *clustering* merupakan hasil penentuan jumlah *cluster* terbaik yaitu berjumlah 3 yang mana jumlah K tersebut sama dengan yang ada

pada model *clustering train data*. Perintah untuk membuat model clustering ditunjukan pada Gambar 5.42.

```
In [29]: km1 = KMeans(n_clusters=3)
km1.fit(z)

Out[29]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                 n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
                 random_state=None, tol=0.0001, verbose=0)
```

Gambar 5.42 Model *Clustering Test Data* Simulasi Kedua

### 3. Menampilkan Centroid

Proses ini dilakukan untuk melihat centroid atau titik pusat yang digunakan selama proses *clustering* dengan menggunakan perintah print kemudian diikuti nama variabel seperti ditunjukan pada Gambar 5.43. Namun sebelum ditampilkan, sentroid harus terlebih dahulu dibangkitkan atau ditentukan, perintah untuk membangkitkan centroid yaitu km1.cluster\_centers\_ kemudian centroid tersebut diinisialisasikan pada sebuah variabel agar mempermudah pemanggilan saat ditampilkan.

```
In [30]: centroid_test_data = km1.cluster_centers_
print(centroid_test_data)

[[ 9.92899408e-01 -4.99600361e-16  1.11301775e-01]
 [ 9.98750000e-01  2.97916667e-01  2.00083333e-01]
 [ 2.55555556e-01  1.94444444e-01  1.83333333e-01]]
```

Gambar 5.43 Centroid *Test Data* Simulasi Kedua

### 4. Proses *Clustering*

Pada proses *clustering* yang ditunjukan pada Gambar 5.44 data akan *predict* merupakan data yang sesuai dengan model yang telah dibuat pada tahap pembuatan model *clustering* yaitu Z. Selain itu pada tahap

ini juga ditampilkan label *cluster* dari masing-masing data, seperti yang ditunjukkan pada Gambar 5.45.

```
In [31]: test_data2['cluster']=km1.predict(Z)

In [32]: cluster_test_labels = km1.labels_
print(cluster_test_labels)

[1 1 0 1 1 0 0 1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0
 0 1 1 0 1 1 0 1 1 0 0 0 1 0 0 1 0 1 2 1 0 1 1 1 1 0 1 0 1 1 1 0 1 0 1 1 1 0 1 0 1 1 1
 1 1 0 1 1 0 1 1 0 1 0 0 0 1 0 1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 2 1 1 1
 0 0 0 1 0 0 0 0 0 1 1 1 1 1 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 0 1 1 0 1 1 1 0 0 0 1
 1 1 0 1 1 1 0 1 0 0 1 1 0 1 1 0 0 0 1 0 1 0 1 0 1 0 1 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 1
 1 0 1 0 0 0 0 0 1 0 1 1 1 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0 1 1 1 1 1 0 0 1 1 1 0 0 0 1
 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 0 1 1 0 2 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 0 1 1 1
 1 1 1 0 2 1 1 0 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 0 2 1 1 1 1 0 1 0 1 1 0 1 1 1 0
 1 0 1 1 1 1 0 0 0 1 0 0 0 1 1 1 1 0 1 0 1 2 1 1 1 1 0 0 0 1 0 1 0 0 1 0 1 1 1 2
 0 1 1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 1 0 1
 0 0 1 0 1 1 0 0 0 1 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 2 0 1 1 1 1 0 2 1 1
```

Gambar 5.44 Proses *Clustering Test Data* Simulasi Kedua

In [36]:	test_data2.head(418)				
Out[36]:	id_plg	lama_langganan	jumlah_layanan	total_tagihan	cluster
	249	39985558	1.0	0.00	0.08
	1238	40018693	1.0	0.25	0.23
	389	39771255	1.0	0.25	0.23
	974	637640	1.0	0.25	0.17
	693	40001566	1.0	0.00	0.08
	...	...	...	...	...
	1173	39955999	1.0	0.00	0.01
	1036	39972989	0.7	0.00	0.04
	756	39791100	1.0	0.50	0.14
	589	33716704	1.0	0.00	0.10
	528	39842088	1.0	0.25	0.17

Gambar 5.45 Hasil *Clustering Test Data* Simulasi Kedua

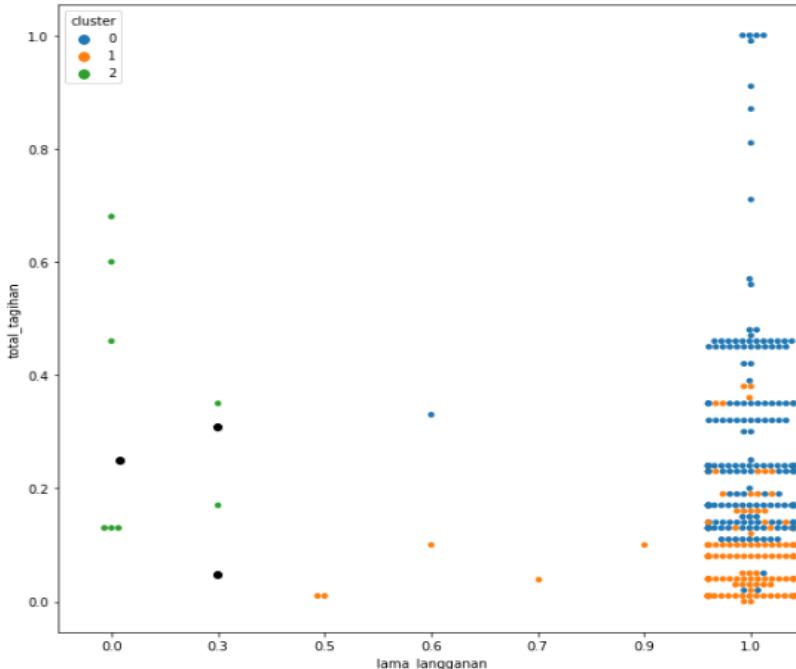
## 5 Visualisasi

Pada tahap ini dilakukan visualisasi penyebaran data dari masing-masing *cluster*, dikarenakan visualisasi dilakukan dengan grafik 2 dimensi sehingga hanya memiliki 2 sumbu dimana sumbu tersebut

merupakan atribut yang digunakan selama *clustering* yaitu lama\_langganan dan total\_tagihan, lama\_langganan dan jumlah\_layanan, serta jumlah\_layanan dan total tagihan.

```
In [37]: plt.figure(figsize=(10,10))
sns.swarmplot(x='lama_langganan', y='total_tagihan', hue='cluster', data=test_data2)
plt.scatter(centroid_test_data[:,0], centroid_test_data[:,1], color='black')

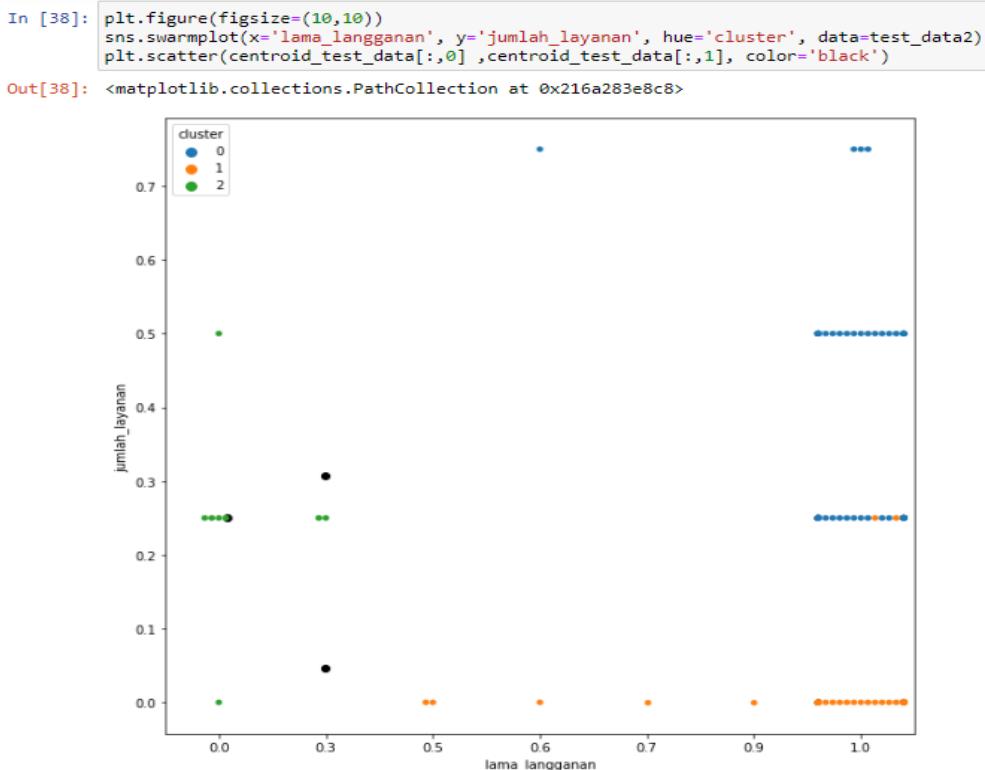
Out[37]: <matplotlib.collections.PathCollection at 0x216a24c2848>
```



Gambar 5.46 Visualisasi Atribut lama\_langganan dan total\_tagihan Pada *Test Data* Simulasi Kedua

Pada Gambar 5.46 menunjukkan visualisasi dari hasil proses *clustering* menggunakan *test data* pada simulasi kedua. Visualisasi tersebut dibuat dalam bentuk grafik dimana dari masing-masing titik memiliki warna sesuai dengan keberadaan data di dalam *cluster*. Kemudian untuk titik yang berwarna hitam merupakan centroid yang digunakan pada proses clustering. Titik yang berada semakin ke kanan pada sumbu x menunjukan bahwa waktu langganan yang dimiliki pelanggan semakin lama, sedangkan titik yang berada semakin ke

atas pada sumbu y menunjukkan bahwa total tagihan yang dibayar pelanggan semakin tinggi.

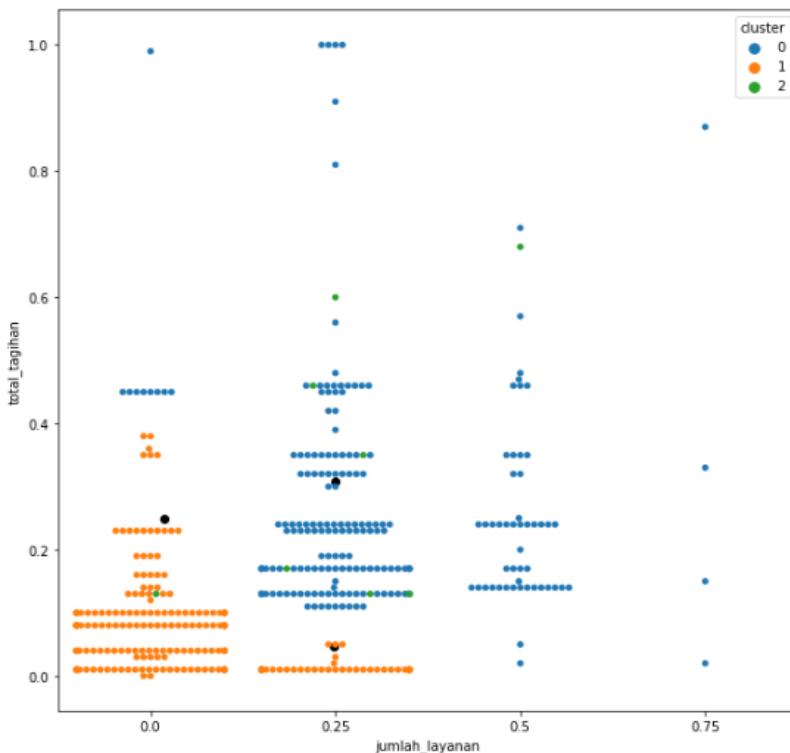


Gambar 5.47 Visualisasi Atribut *lama\_langganan* dan *jumlah\_layanan* Pada *Test Data* Simulasi Kedua

Visualisasi hasil clustering yang ditunjukkan pada Gambar 5.47 menunjukkan posisi dari data sesuai dengan posisinya yang disimbolkan dengan titik dan memiliki warna yang berbeda sesuai dengan *cluster* dimana data berada. Data yang berada semakin ke kanan dari sumbu x meunjukkan bahwa waktu langganan dari pelanggan semakin lama sedangkan data yang berada semakin keatas dari sumbu y menunjukkan semakin banyaknya jumlah layanan yang digunakan oleh pelanggan.

```
In [39]: plt.figure(figsize=(10,10))
sns.swarmplot(x='jumlah_layanan', y='total_tagihan', hue='cluster', data=test_data2)
plt.scatter(centroid_test_data[:,0] ,centroid_test_data[:,1], color='black')

Out[39]: <matplotlib.collections.PathCollection at 0x216a2addac8>
```



Gambar 5.48 Visualisasi Atribut jumlah\_layanan dan total\_tagihan Pada *Test Data* Simulasi Kedua

Pada Gambar 5.48, merupakan visualisasi hasil *clustering* dengan atribut jumlah layanan dan total tagihan. Titik yang ditunjukkan pada grafik berada pada posisi yang berbeda, bila titik berada semakin ke kana pada sumbu x maka jumlah layanan yang digunakan oleh pelanggan semakin banyak sedangkan bila titik berada semakin keatas dari sumbu y maka jumlah total tagihan yang dimiliki oleh pelanggan semakin tinggi.

## 6. Evaluasi *Cluster*

Evaluasi *cluster* dilakukan untuk melihat performance dari *clustering test data*. Adapun proses evaluasi *cluster* dilakukan dengan metode silhouette score, davies bouldin index, dan calinski harabasz.

```
Test Data

In [42]: silhouette_score(Z,cluster_test_labels)
Out[42]: 0.5077949467251326

In [43]: silhouette_score(Z,cluster_test_labels == 0)
Out[43]: 0.46968551122863267

In [44]: silhouette_score(Z,cluster_test_labels == 1)
Out[44]: 0.4839117931730241

In [45]: silhouette_score(Z,cluster_test_labels == 2)
Out[45]: 0.6579458973613102
```

Gambar 5.49 Silhouette Score Pada *Test Data* Simulasi Kedua

```
Test Data

In [50]: davies_bouldin_score(Z, cluster_test_labels)
Out[50]: 0.731339910380838

In [51]: davies_bouldin_score(Z, cluster_test_labels == 0)
Out[51]: 0.8796048393222466

In [52]: davies_bouldin_score(Z, cluster_test_labels == 1)
Out[52]: 0.9766217121225618

In [53]: davies_bouldin_score(Z, cluster_test_labels == 2)
Out[53]: 0.6768949365623108
```

Gambar 5.50 Davies Bouldin Index Pada *Test Data* Simulasi Kedua

```
Test Data
```

```
In [58]: metrics.calinski_harabasz_score(Z, cluster_test_labels)
Out[58]: 251.63458203459965
```

```
In [59]: metrics.calinski_harabasz_score(Z, cluster_test_labels == 0)
Out[59]: 236.4587794361536
```

```
In [60]: metrics.calinski_harabasz_score(Z, cluster_test_labels == 1)
Out[60]: 231.58088299115602
```

```
In [61]: metrics.calinski_harabasz_score(Z, cluster_test_labels == 2)
Out[61]: 93.7111634522696
```

Gambar 5.51 Calinski Harabasz Index Pada *Test Data* Simulasi Kedua

#### 5.2.4.6.3 Simulasi Ketiga

Pada simulasi ketiga penulis melakukan proses *clustering* dengan menggunakan persentase *train data* dan *test data* sebanyak 50% dan 50%. Proses clustering yang dilakukan pada simulasi ketiga memiliki kesamaan dengan simulasi pertama dan kedua, hanya memiliki perbedaan dari persentase data yang digunakan serta deklarasi variabel dan model pada proses clustering.

- *Train Data*

1. Inisialisasi data

Proses yang dilakukan pada Gambar 5.52 ini yaitu proses inisialisasi data menggunakan *train data* serta pemilihan atribut yang digunakan untuk proses *clustering* pada simulasi ketiga.

```
In [18]: X = train_data2[['lama_langganan', 'jumlah_layanan', 'total_tagihan']]
```

Gambar 5.52 Inisialisasi *Train Data* Simulasi Ketiga

## 2. Model *Clustering*

Gambar 5.53 merupakan proses pembuatan model *clustering* ditandai dengan adanya perintah `km.fit(X)` dan K yang digunakan untuk proses *clustering* merupakan hasil penentuan jumlah *cluster* terbaik yaitu berjumlah 3.

```
In [19]: km = KMeans(n_clusters=3)
km.fit(X)

Out[19]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                 n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
                 random_state=None, tol=0.0001, verbose=0)
```

Gambar 5.53 Model *Clustering* Simulasi Ketiga

## 3. Menampilkan Centroid

Proses ini dilakukan untuk melihat centroid atau titik pusat yang digunakan selama proses *clustering*. Sebelum centroid dapat ditampilkan maka terlebih dahulu centroid tersebut harus dibangkitkan, perintah untuk membangkitkan centroid yaitu `km.cluster_centers_` kemudian setelah dibangkitkan, centroid tersebut ditampung pada sebuah variabel yang diberi nama `centroid_train_data`, kemudian barulah centroid tersebut ditampilkan dengan perintah `print` kemudian diikuti dengan nama variabel tersebut seperti ditunjukkan pada Gambar 5.54.

```
In [20]: centroid_train_data = km.cluster_centers_
print(centroid_train_data)

[[9.95454545e-01 2.97979798e-01 1.97979798e-01]
 [9.94755245e-01 5.27355937e-16 1.01573427e-01]
 [1.28571429e-01 2.32142857e-01 2.23571429e-01]]
```

Gambar 5.54 Centroid *Train Data* Simulasi Ketiga

#### 4. Proses *Clustering*

Proses *clustering* yang dilakukan pada Gambar 5.55 yaitu dimana *train data* dari simulasi ketiga akan di *predict* sesuai dengan model model *clustering* X.

Selain itu pada tahap ini juga ditampilkan label *cluster* dari masing-masing data seperti ditunjukkan pada Gambar 5.56.

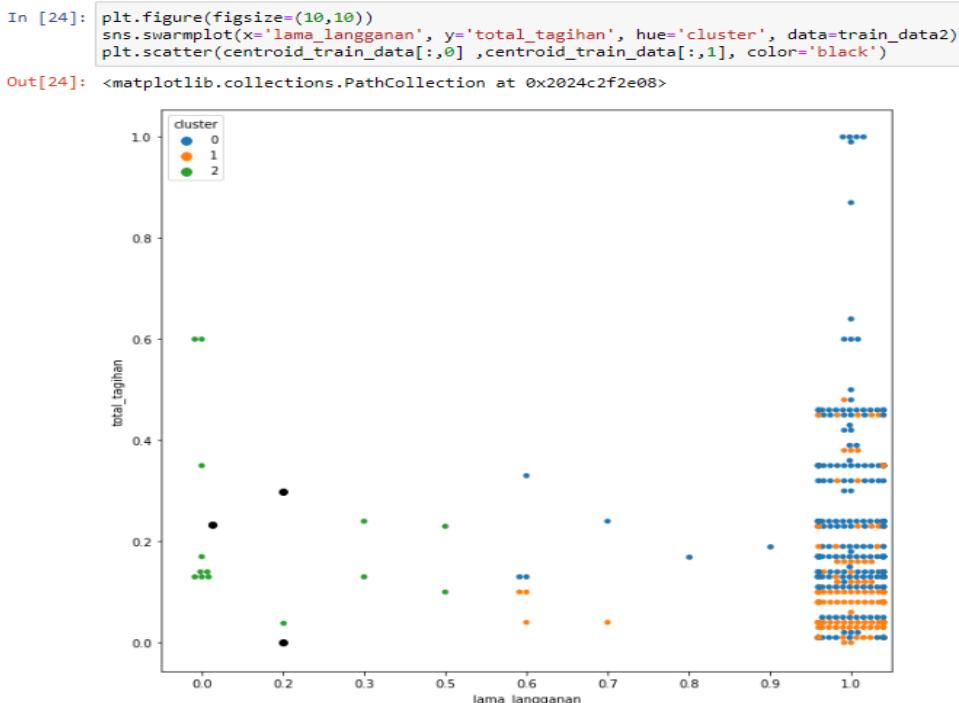
Gambar 5.55 Proses *Clustering Train Data* Simulasi Ketiga

In [23]:	train_data2.head(696)				
Out[23]:	id_plg	lama_langganan	jumlah_layanan	total_tagihan	cluster
1307	39897042	1.0	0.50	0.14	0
896	39852481	1.0	0.25	0.45	0
759	39808281	1.0	0.50	0.14	0
986	39992603	1.0	0.00	0.10	1
719	40046677	1.0	0.50	0.32	0
...	...	...	...	...	...
1118	39962332	1.0	0.00	0.16	1
1322	39827034	1.0	0.25	0.17	0
980	39717894	1.0	0.25	0.17	0
683	39972125	1.0	0.00	0.08	1
265	40025363	1.0	0.25	0.32	0

Gambar 5.56 Hasil *Clustering Train Data* Simulasi Ketiga

## 5. Visualisasi

Pada tahap ini dilakukan visualisasi yang bertujuan untuk menampilkan penyebaran data dari masing-masing *cluster*, dikarenakan visualisasi dilakukan dengan grafik 2 dimensi sehingga hanya memiliki 2 sumbu dimana sumbu tersebut merupakan pasangan atribut yang digunakan selama *clustering* yaitu *lama\_langganan* dan *total\_tagihan*, *lama\_langganan* dan *jumlah\_layanan*, serta *jumlah\_layanan* dan *total tagihan*.

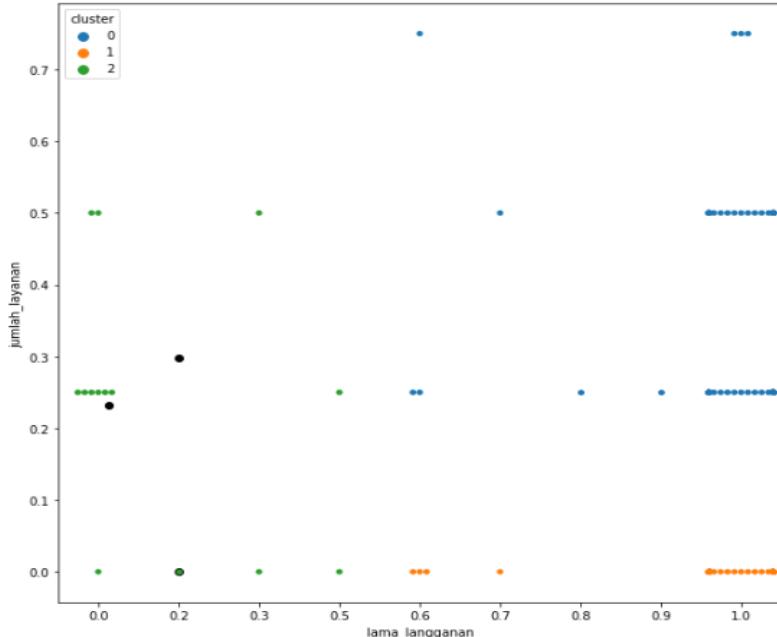


Gambar 5.57 Visualisasi Atribut *lama\_langganan* dan *total\_tagihan* Pada *Train Data* Simulasi Ketiga

Pada Gambar 5.57 menunjukkan visualisasi hasil menggunakan *train data* pada simulasi ketiga yang mana, persebaran data disimbolkan dengan titik dan pada titik tersebut diberikan warna yang berbeda sesuai dengan masing-masing cluster. Persebaran data berada pada sumbu x dan y, bila

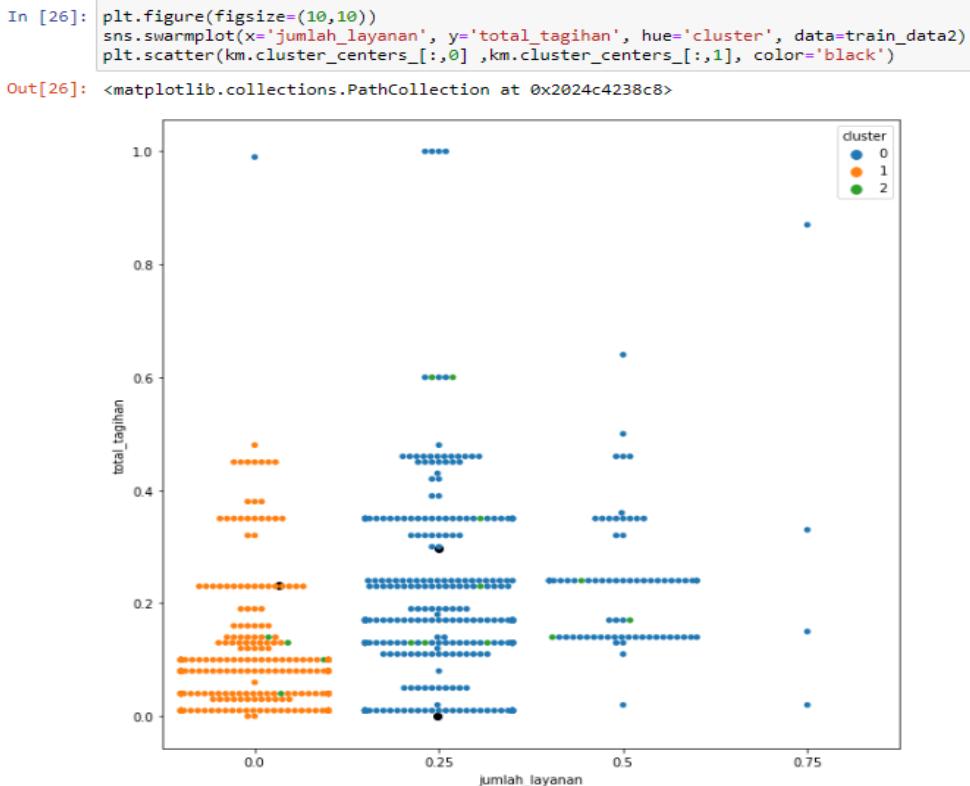
posisi data berada semakin ke kanan dari sumbu x maka waktu langganan yang dimiliki pelanggan semakin lama, sedangkan posisi data yang berada semakin keatas dari sumbu y menunjukan bahwa jumlah total tagihan yang dimiliki pelanggan semakin tinggi.

```
In [25]: plt.figure(figsize=(10,10))
sns.swarmplot(x='lama_langganan', y='jumlah_layanan', hue='cluster', data=train_data2)
plt.scatter(centroid_train_data[:,0] ,centroid_train_data[:,1], color='black')
Out[25]: <matplotlib.collections.PathCollection at 0x2024c3afac8>
```



Gambar 5.58 Visualisasi Atribut *lama\_langganan* dan *jumlah\_layanan* Pada *Train Data* Simulasi Ketiga

Visualisasi penyebaran data yang ditunjukan pada Gambar 5.58 menunjukan bahwa data berada diantara sumbu x dan sumbu y dari grafik yang ada. Data yang berada pada sumbu x dan memiliki posisi di kanan atau semakin ke kanan menunjukan bahwa waktu langganan yang dimiliki oleh pelanggan semakin lama, sedangkan data yang berada pada sumbu y dan memiliki posisi semakin ke atas menunjukan bahwa jumlah layanan yang digunakan oleh pelanggan semakin banyak.



Gambar 5.59 Visualisasi Atribut jumlah\_layanan dan total\_tagihan Pada  
Train Data Simulasi Ketiga

Pada Gambar 5.59 menunjukkan visualiasi penyebaran data hasil *clustering* dengan atribut jumlah layanan dan total tagihan. Atribut tersebut masing-masing berada pada sumbu x dan y. Bila posisi data yang berada pada sumbu x semakin ke kanan maka menunjukan bahwa jumlah layanan yang digunakan okeh oleh pelanggan semakin banyak, sedangkan data yang berada pada sumbu y semakin ke atas menunjukan bahwa total tagihan yang dimiliki oleh pelanggan semakin tinggi.

## 6. Evaluasi *Cluster*

Evaluasi *cluster* dilakukan untuk melihat performance dari *clustering test data*. Adapun proses evaluasi *cluster* dilakukan dengan metode silhouette score, davies bouldin index, dan calinski harabasz.

Train Data

```
In [38]: silhouette_score(X,cluster_train_labels)
Out[38]: 0.518688628052643

In [39]: silhouette_score(X,cluster_train_labels == 0)
Out[39]: 0.49137220298228895

In [40]: silhouette_score(X,cluster_train_labels == 1)
Out[40]: 0.47334858879859115

In [41]: silhouette_score(X,cluster_train_labels == 2)
Out[41]: 0.7055374015382756
```

Gambar 5.60 Silhouette Score Pada *Train data* Simulasi Ketiga

Train Data

```
In [46]: davies_bouldin_score(X, cluster_train_labels)
Out[46]: 0.6762327210253868

In [47]: davies_bouldin_score(X, cluster_train_labels == 0)
Out[47]: 0.9692466697456035

In [48]: davies_bouldin_score(X, cluster_train_labels == 1)
Out[48]: 0.8464337013330451

In [49]: davies_bouldin_score(X, cluster_train_labels == 2)
Out[49]: 0.566221894586707
```

Gambar 5.61 Davies Bouldin Index Pada *Train data* Simulasi Ketiga

### Train Data

```
In [54]: metrics.calinski_harabasz_score(X, cluster_train_labels)
Out[54]: 449.04618398585325

In [55]: metrics.calinski_harabasz_score(X, cluster_train_labels == 0)
Out[55]: 347.0110079304546

In [56]: metrics.calinski_harabasz_score(X, cluster_train_labels == 1)
Out[56]: 370.73241368436254

In [57]: metrics.calinski_harabasz_score(X, cluster_train_labels == 2)
Out[57]: 195.83736836610748
```

Gambar 5.62 Calinski Harabasz Index Pada *Train data* Simulasi Ketiga

### • *Test Data*

#### 1. Inisialisasi data

Proses yang dilakukan pada Gambar 5.63 adalah untuk menginisialisasi data menggunakan *test data* pada simulasi ketiga serta pemilihan atribut yang digunakan untuk proses *clustering*.

```
In [28]: Z = test_data2[['lama_langganan', 'jumlah_layanan', 'total_tagihan']]
```

Gambar 5.63 Inisialisasi *Test Data* Simulasi Ketiga

#### 2. Model *Clustering*

Proses ini merupakan proses pembuatan model *clustering* ditandai dengan adanya perintah `km.fit(Z)` dan `K` yang digunakan selama proses *clustering* merupakan hasil penentuan jumlah *cluster* terbaik yaitu berjumlah 3 serta jumlah `K` yang sama dengan yang ada pada model *clustering train data*. Perintah untuk pembuatan model *clustering* dapat lebih jelas dilihat pada Gambar 5.64.

```
In [29]: km1 = KMeans(n_clusters=3)
km1.fit(z)

Out[29]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                 n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
                 random_state=None, tol=0.0001, verbose=0)
```

Gambar 5.64 Model *Clustering Test Data* Simulasi Ketiga

### 3. Menampilkan Centroid

Proses ini dilakukan untuk melihat centroid atau titik pusat yang digunakan selama proses *clustering*. Pada Gambar 5.65, sebelum ditampilkan terlebih dahulu centroid tersebut harus ditentukan, untuk menentukan centroid dapat digunakan perintah `km1.cluster_centers_` kemudian dimasukan atau ditampung dalam sebuah variabel yang diberi nama `centroid_test_data`. Setelah itu barulah centroid dapat ditampilkan dengan menggunakan perintah `print` kemudian diikuti dengan nama variabelnya.

```
In [30]: centroid_test_data = km1.cluster_centers_
print(centroid_test_data)

[[ 9.97593583e-01  2.98796791e-01  2.10534759e-01]
 [ 9.96013289e-01 -4.44089210e-16  1.00996678e-01]
 [ 1.38095238e-01  1.78571429e-01  1.67142857e-01]]
```

Gambar 5.65 Centroid *Test Data* Simulasi Ketiga

### 4. Proses *Clustering*

Pada Gambar 5.66, proses *clustering* dilakukan dengan menggunakan *test data* yang kemudian akan di *predict* sesuai dengan model *clustering Z* yang telah dibuat sebelumnya. Selain itu pada tahap ini juga ditampilkan label *cluster* dari masing-masing data seperti ditunjukkan pada Gambar 5.67.

```
In [31]: test_data2['cluster']=km1.predict(z)

In [32]: cluster_test_labels = km1.labels_
print(cluster_test_labels)

[0 1 1 1 0 1 1 0 1 2 0 0 1 0 1 0 0 0 1 0 0 1 0 1 0 1 1 0 1 1 0 0 0 1 1 0 0
 0 0 1 0 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0 0 1 0 1 0 0 0 0 1 1 0
 1 1 0 0 0 1 1 1 0 1 0 1 1 0 1 1 1 0 1 0 2 1 0 0 0 1 2 0 1 0 0 0 2 0 0 1
 0 0 1 2 0 1 1 0 1 1 1 0 2 0 0 1 1 0 0 1 1 1 0 1 0 0 1 0 0 0 1 0 1 2 1 0 1 0
 1 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 1 0 1 0 1 0 0 1 0 0 0 1 1 0 0 1
 0 1 2 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 1 0 1 0 0 0 1 0 1 0 0
 0 1 0 0 0 1 1 0 1 1 0 0 1 1 1 1 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1
 0 0 0 0 2 1 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 0 2
 1 0 1 0 0 1 1 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 1 1 0 1 0 1 1 1 1 0
 1 1 0 0 1 0 1 1 0 1 0 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 1 1 1 0 1 0 0
 0 1 1 0 2 1 0 0 2 0 1 0 0 2 0 0 0 1 0 2 1 0 0 1 1 0 1 0 1 0 0 0 1 1 0 1 0 0 1 1
 0 1 0 0 0 1 2 0 0 0 0 1 1 0 1 1 2 1 1 0 1 0 2 0 1 0 1 0 0 0 0 1 0 1 0 0 1 1 1
 0 0 1 0 1 0 0 0 0 1 1 0 0 0 1 1 0 1 0 0 1 1 0 0 1 1 0 1 0 1 0 0 1 1 1 0 1 0 1
 0 1 0 1 1 0 1 1 0 1 0 1 1 0 0 0 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
 0 1 1 1 0 1 0 1 1 0 0 0 0 1 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 1 1 1 1 0 0
 1 1 1 0 0 0 1 0 1 0 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0
 1 1 1 0 1 0 1 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0
 1 1 1 0 1 0 1 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0
 0 0 1 1 1 0 0 1 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 1 1 1 0 0 0 0 0 0
 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 1 0 0 1 1 1 2 2]
```

Gambar 5.66 Proses *Clustering Test Data* Simulasi Ketiga

test_data2.head(696)					
	id_plg	lama_langganan	jumlah_layanan	total_tagihan	cluster
497	39148968	1.0	0.25	0.19	0
574	39808639	1.0	0.00	0.04	1
260	40022607	1.0	0.00	0.08	1
233	40003509	1.0	0.00	0.01	1
52	30857311	1.0	0.25	0.13	0
...	...	...	...	...	...
137	39906981	1.0	0.00	0.19	1
554	39714748	1.0	0.00	0.04	1
652	39840392	1.0	0.00	0.01	1
557	38407685	0.0	0.00	0.04	2
782	39875200	0.2	0.25	0.13	2

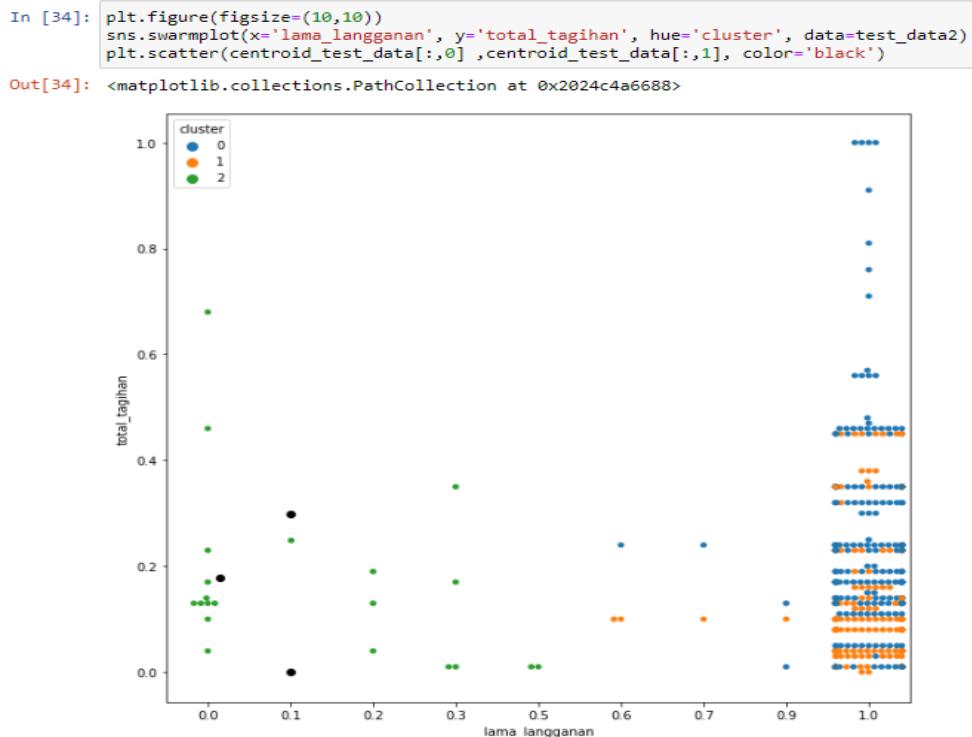
696 rows × 5 columns

Gambar 5.67 Hasil *Clustering Test Data* Simulasi Ketiga

## 5. Visualisasi

Pada tahap ini dilakukan visualisasi yang bertujuan untuk melihat penyebaran data dari masing-masing *cluster*, dikarenakan visualisasi dilakukan dengan grafik 2 dimensi sehingga hanya memiliki 2 sumbu dimana sumbu tersebut merupakan atribut yang digunakan selama

*clustering* yaitu lama\_langganan dan total\_tagihan, lama\_langganan dan jumlah\_layanan, serta jumlah\_layanan dan total tagihan.

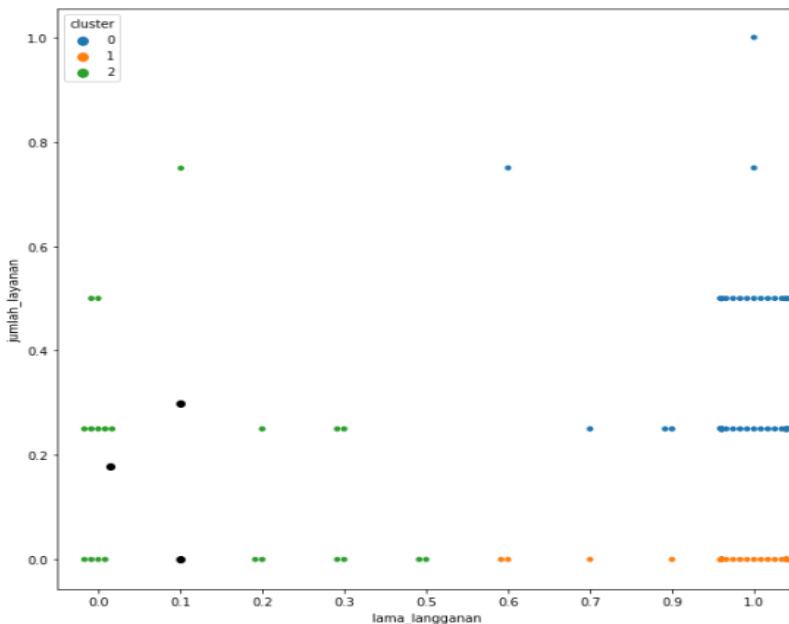


Gambar 5.68 Visualisasi Atribut lama\_langganan dan total\_tagihan Pada *Test Data* Simulasi Ketiga

Pada Gambar 5.68, visualisasi dari hasil *clustering* menggunakan *test data* yang menunjukkan penyebaran data dengan atribut lama langganan dan total tagihan. Data yang berada pada sumbu x dan bila posisinya semakin ke kanan maka waktu langganan yang dimiliki oleh pelanggan semakin lama sedangkan data yang berada pada sumbu y dan posisinya berada semakin ke atas maka jumlah total tagihan yang dimiliki oleh pelanggan semakin tinggi.

```
In [35]: plt.figure(figsize=(10,10))
sns.swarmplot(x='lama_langganan', y='jumlah_layanan', hue='cluster', data=test_data2)
plt.scatter(centroid_test_data[:,0], centroid_test_data[:,1], color='black')

Out[35]: <matplotlib.collections.PathCollection at 0x2024c6dd108>
```

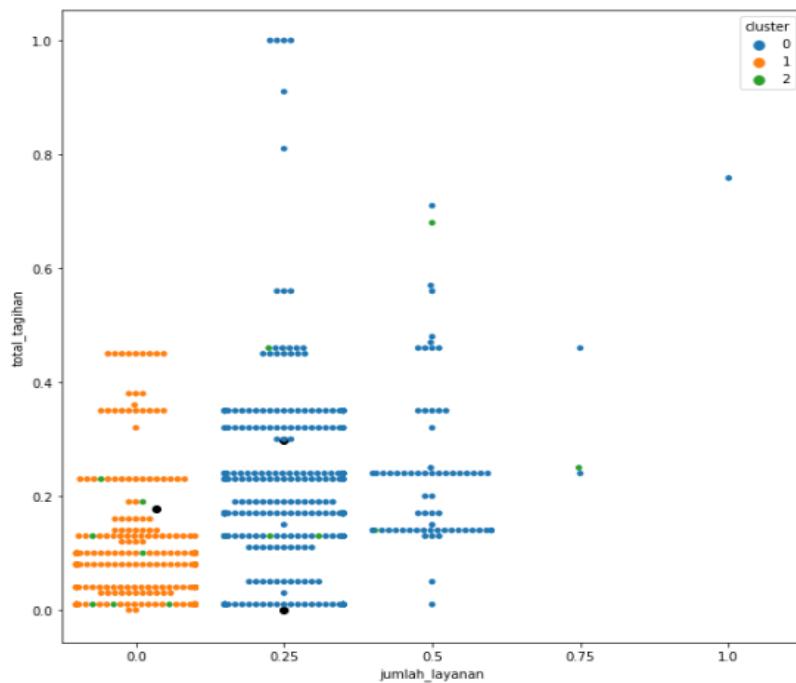


Gambar 5.69 Visualisasi Atribut lama\_langganan dan jumlah\_layanan Pada *Test Data* Simulasi Ketiga

Pada Gambar 5.69, visualisasi dari hasil *clustering* menggunakan *test data* yang menunjukkan penyebaran data dengan atribut lama langganan dan jumlah layanan. Data yang berada pada sumbu x dan bila posisinya semakin ke kanan maka waktu langganan yang dimiliki oleh pelanggan semakin lama sedangkan data yang berada pada sumbu y dan posisinya berada semakin ke atas maka jumlah layanan yang digunakan oleh pelanggan semakin banyak.

```
In [36]: plt.figure(figsize=(10,10))
sns.swarmplot(x='jumlah_layanan', y='total_tagihan', hue='cluster', data=test_data2)
plt.scatter(centroid_test_data[:,0] ,centroid_test_data[:,1], color='black')

Out[36]: <matplotlib.collections.PathCollection at 0x2024c7117c8>
```



Gambar 5.70 Visualisasi Atribut jumlah\_layanan dan total\_tagihan Pada *Test Data* Simulasi Ketiga

Visualisasi yang ditunjukkan pada Gambar 5.69, merupakan hasil *clustering* menggunakan *test data* yang menunjukkan penyebaran data dengan atribut jumlah layanan dan total tagihan. Data yang berada pada sumbu x dan bila posisinya semakin ke kanan maka jumlah layanan yang digunakan oleh pelanggan semakin banyak sedangkan data yang berada pada sumbu y dan posisinya berada semakin ke atas maka jumlah total tagihan yang dimiliki oleh pelanggan semakin tinggi.

## 6. Evaluasi *Cluster*

Evaluasi *cluster* dilakukan untuk melihat *performance* dari *clustering test data*. Adapun proses evaluasi *cluster* dilakukan dengan metode Silhouette Score, Davies Bouldin Index, dan Calinski Harabasz Index.

### Test Data

```
In [42]: silhouette_score(Z,cluster_test_labels)
Out[42]: 0.5408710133310518

In [43]: silhouette_score(Z,cluster_test_labels == 0)
Out[43]: 0.49636316378032064

In [44]: silhouette_score(Z,cluster_test_labels == 1)
Out[44]: 0.47205070937414934

In [45]: silhouette_score(Z,cluster_test_labels == 2)
Out[45]: 0.7078966261069265
```

Gambar 5.71 Silhouette Score pada *Test Data* Simulasi Ketiga

### Test Data

```
In [50]: davies_bouldin_score(Z, cluster_test_labels)
Out[50]: 0.644807448229311

In [51]: davies_bouldin_score(Z, cluster_test_labels == 0)
Out[51]: 0.9905539774327192

In [52]: davies_bouldin_score(Z, cluster_test_labels == 1)
Out[52]: 0.8705282262772778

In [53]: davies_bouldin_score(Z, cluster_test_labels == 2)
Out[53]: 0.5576044592780304
```

Gambar 5.72 Davies Bouldin Index pada *Test Data* Simulasi Ketiga

### Test Data

```
In [58]: metrics.calinski_harabasz_score(Z, cluster_test_labels)
Out[58]: 555.8627507477783

In [59]: metrics.calinski_harabasz_score(Z, cluster_test_labels == 0)
Out[59]: 331.71694124011754

In [60]: metrics.calinski_harabasz_score(Z, cluster_test_labels == 1)
Out[60]: 336.02928961860454

In [61]: metrics.calinski_harabasz_score(Z, cluster_test_labels == 2)
Out[61]: 283.4543153039207
```

Gambar 5.73 Calinski Harabasz Index pada *Test Data* Simulasi Ketiga

#### **5.2.4.7 Pengkajian Pemodelan dan Evaluasi**

##### **5.2.4.7.1 Pengkajian Pemodelan**

Pada tahap ini, penulis akan mengkaji terkait dengan pemodelan yang dilakukan. Tujuan dari studi kasus yang dilakukan adalah untuk melakukan segmentasi data pelanggan serta melakukan customer profiling dengan menggunakan algoritma K-Means, dimana data pelanggan diperoleh dari arsip perusahaan. Proses penelitian serta pengolahan data dilakukan dengan mengikuti tahapan dari metodologi penelitian CRISP-DM.

Data pelanggan yang telah diperoleh tersebut kemudian dimodelkan menggunakan bahasa pemrograman python pada aplikasi jupyter notebook. Dalam proses pemodelan dilakukan 3 simulasi dan pada simulasi tersebut data pelanggan dibagi menjadi *training data* dan *testing data* untuk melakukan proses *clustering* seperti yang telah dijelaskan pada pembahasan sebelumnya.

Proses *clustering* yang dilakukan menggunakan simulasi pertama membagi data menjadi 80% *training data* atau 1113 *record* dan *testing data* 20% atau 279 *record* ke dalam 3 *cluster*. Hasil dari proses *clustering* menggunakan *train data* membagi data ke dalam *cluster* 0 berjumlah 479 *record*, *cluster* 1 berjumlah 607 *record* dan *cluster* 2 berjumlah 27 *record*.

Sedangkan *test data* terbagi ke dalam cluster 0 sebanyak 163 record, *cluster 1* sebanyak 108 record, dan *cluster 2* sebanyak 8 record.

Selanjutnya proses *clustering* yang dilakukan menggunakan simulasi kedua membagi data menjadi 70% *training data* atau 974 record dan *testing data* 30% atau 418 record, dimana hasil *clustering* menggunakan *training data* membagi data ke dalam *cluster 0* sebanyak 532 record, *cluster 1* sebanyak 417 record, dan *cluster 2* sebanyak 25 record. Kemudian hasil *clustering* menggunakan *testing data* membagi data ke dalam *cluster 0* sebanyak 202 record, *cluster 1* sebanyak 208 record, *cluster 2* sebanyak 8 record.

Pada simulasi ketiga, data pelanggan dibagi menjadi 50% *training data* atau 696 record dan *testing data* 50% atau 696 record. Kemudian hasil *clustering* dengan menggunakan *training data* membagi data ke dalam *cluster 0* sebanyak 396 record, *cluster 1* sebanyak 286 record, dan *cluster 2* sebanyak 14 record. Selain itu hasil *clustering* menggunakan *testing data* membagi data kedalam *cluster 0* sebanyak 374 record, *cluster 1* sebanyak 301 record, dan *cluster 2* sebanyak 21 record.

Hasil *clustering* yang telah terbentuk dievaluasi *performance* nya untuk menentukan simulasi mana yang paling tepat digunakan untuk *clustering* data pelanggan tersebut. Evaluasi dilakukan dengan menghitung *score* dari Silhouette Index (SI), Davies-Bouldin Index dan Calinski Harabasz Index (CHI) dimana hasil dari evaluasi tersebut dapat dilihat pada Tabel 5.2.

Tabel 5.2 Hasil Evaluasi *Performance* K-Means Simulasi 1

CLUSTER	SIMULASI 1					
	DATA TRAIN (80%)			DATA TEST (20%)		
	SI	DBI	CHI	SI	DBI	CHI
Global	0.535	0.670	806.107	0.508	0.630	198.522
0	0.482	0.848	598.288	0.470	1.053	114.979
1	0.500	0.964	567.455	0.433	0.896	113.259
2	0.703	0.603	360.728	0.717	0.426	117.956

Pada Tabel 5.2 simulasi 1 terdapat *train data* (80%) dan *test data* (20%) untuk mengetahui simulasi terbaik maka dilakukan validasi menggunakan

indeks SI, DBI dan CHI. Dipilih *cluster* global sebagai pembanding antar simulasi dimana, nilai dari masing-masing indeks pada *train data* dan *test data* kemudian dihitung nilai rata-ratanya dengan nilai sebagai berikut 0.521, 0.650, 502.315.

Tabel 5.3 Hasil Evaluasi *Performance* K-Means Simulasi 2

CLUSTER	SIMULASI 2					
	DATA TRAIN (70%)			DATA TEST (30%)		
	SI	DBI	CHI	SI	DBI	CHI
Global	0.539	0.641	751.767	0.508	0.731	251.635
0	0.474	0.853	474.070	0.470	0.880	236.459
1	0.498	0.988	451.299	0.484	0.977	231.581
2	0.716	0.543	384.504	0.658	0.677	93.711

Pada Tabel 5.3 simulasi 2 terdapat *train data* (70%) dan *test data* (30%) untuk mengetahui simulasi terbaik maka dilakukan validasi menggunakan indeks SI, DBI dan CHI. Dipilih *cluster* global sebagai pembanding antar simulasi dimana, nilai dari masing-masing indeks pada *train data* dan *test data* kemudian dihitung nilai rata-ratanya dengan nilai sebagai berikut 0.524, 0.686, 501.701.

Tabel 5.4 Hasil Evaluasi *Performance* K-Means Simulasi 3

CLUSTER	SIMULASI 3					
	DATA TRAIN (50%)			DATA TEST (50%)		
	SI	DBI	CHI	SI	DBI	CHI
Global	0.519	0.676	449.046	0.541	0.645	555.863
0	0.491	0.969	347.011	0.496	0.991	331.717
1	0.473	0.846	370.732	0.472	0.871	336.029
2	0.706	0.566	195.837	0.708	0.558	283.454

Pada Tabel 6.3 simulasi 2 terdapat *train data* (50%) dan *test data* (50%) untuk mengetahui simulasi terbaik maka dilakukan validasi menggunakan indeks SI, DBI, dan CH. Dipilih *cluster* global sebagai pembanding antar simulasi dimana, nilai dari masing-masing indeks pada *train data* dan *test data* kemudian dihitung nilai rata-ratanya dengan nilai sebagai berikut 0.530, 0.661, 502.454.

Untuk menentukan simulasi mana yang memiliki *performance* optimal dapat dilihat dari hasil evaluasi tersebut, dimana *score* dari Silhouette Index

nya semakin mendekati 1, *score* dari Davies-Bouldin Index nya kecil dan memiliki *score* Calinski Harabasz Index yang besar.

#### 5.2.4.7.2 Evaluasi

Pada tahap evaluasi akan dijelaskan mengenai hasil dari pengkajian dimana terdapat 3 model simulasi yaitu simulasi 1 (80%, 20%), simulasi 2 (70%, 30%), simulasi 3 (50%, 50%). Untuk menentukan simulasi yang memiliki *performance* yang optimal dilihat dari tiga indeks validitas dengan kriteria *relative*, yaitu indeks Silhouette, indeks Davies-Bouldin, dan Indeks Calinski-Harabasz. Pada indeks silhouette (SI) simulasi yang terbaik ditunjukkan dengan nilai Silhouette yang semakin mendekati 1, Indeks validitas Davies-Bouldin simulasi terbaik ditunjukkan dengan nilai DBI yang semakin kecil, sedangkan untuk Indeks validitas Calinski-Harabasz (CHI) simulasi terbaik ditunjukkan dengan semakin besar nilai CHI.

Dari hasil evaluasi *performance* disimpulkan bahwa simulasi yang memiliki performance optimal merupakan simulasi 3 dikarenakan simulasi tersebut telah memenuhi kriteria validitas *relative* dimana terdapat 2 kriteria yang sesuai yaitu memiliki nilai Silhouette indeks mendekati 1 dan nilai CH indeks dengan nilai yang semakin besar.

Setelah diketahui simulasi mana yang telah memenuhi kriteria dengan performance optimal, barulah dapat ditentukan karakteristik anggota dari masing-masing cluster atau dikenal dengan istilah profiling. *Customer profiling* data pelanggan dilakukan dengan cara menganalisis anggota dari masing-masing *cluster* seperti berikut.

##### 1. Train Data:

- *Cluster 0* merupakan pelanggan yang memiliki waktu berlangganan cukup lama dari mulai 7 – 11 bulan dengan jumlah layanan yang diambil mulai dari 1 – 4 layanan dengan total tagihan yang cukup variatif berkisar antara 286000 – 2090000. Sehingga pelanggan pada *cluster 0*

dikategorikan sebagai pelanggan yang memberikan keuntungan terbesar bagi perusahaan.

- *Cluster 1* merupakan pelanggan yang memiliki waktu berlangganan 7, 8, dan 11 bulan dengan 1 jumlah layanan yang diambil dan dengan total tagihan paling rendah yaitu 275000 sedangkan total tagihan paling tinggi yaitu 1144000.

Sehingga pelanggan pada *cluster 1* dikategorikan sebagai pelanggan yang tanpa disadari memiliki potensi besar dalam memberikan keuntungan bagi perusahaan.

- *Cluster 2* merupakan pelanggan yang memiliki waktu berlangganan 1, 3, 4, dan 6 bulan dengan jumlah layanan yang diambil sebanyak 1 – 3 layanan dan dengan total tagihan paling rendah yaitu 286000 dan total tagihan yang paling tinggi yaitu 907500. Sehingga pelanggan pada *cluster 2* dikategorikan sebagai pelanggan yang memberikan keuntungan lebih sedikit daripada biaya untuk memberikan pelayanan.

## 2. Test Data:

- *Cluster 0* merupakan pelanggan yang memiliki waktu berlangganan cukup lama dari mulai 7, 8, 10, dan 11 bulan dengan jumlah layanan yang diambil mulai dari 2 – 5 layanan dengan total tagihan yang cukup variatif berkisar antara 275000 – 2090000. Sehingga pelanggan pada *cluster 0* dikategorikan sebagai pelanggan yang memberikan keuntungan terbesar bagi perusahaan.
- *Cluster 1* merupakan pelanggan yang memiliki waktu berlangganan 7, 8, 10 dan 11 bulan dengan 1 jumlah layanan yang diambil dan dengan total tagihan paling rendah yaitu 275000 sedangkan total tagihan paling tinggi yaitu 1094500. Sehingga pelanggan pada *cluster 1* dikategorikan

sebagai pelanggan yang tanpa disadari memiliki potensi besar dalam memberikan keuntungan bagi perusahaan.

- *Cluster 2* merupakan pelanggan yang memiliki waktu berlangganan 1, 2, 3, 4, dan 6 bulan dengan jumlah layanan yang diambil sebanyak 1 – 4 layanan dan dengan total tagihan paling rendah yaitu 286000 dan total tagihan yang paling tinggi yaitu 1518000. Sehingga pelanggan pada *cluster 2* dikategorikan sebagai pelanggan yang memberikan keuntungan lebih sedikit daripada biaya untuk memberikan pelayanan.



## DAFTAR PUSTAKA

---

- [1] B. E. Adiana, I. Soesanti, and A. E. Permanasari, “Analisis Segmentasi Pelanggan Menggunakan Kombinasi RFM Model dan Teknik Clustering,” *Jutei*, vol. 2, no. 2, pp. 23–32, 2018.
- [2] F. E. M. Agustin, A. Fitria, and H. A. S, “Implementasi Algoritma K-Means Untuk Menentukan Kelompok Pengayaan Materi Mata Pelajaran Ujian Nasional (Studi Kasus: Smp Negeri 101 Jakarta),” *J. Tek. Inform.*, vol. 8, no. 1, pp. 73–78, 2015.
- [3] E. I. M. Faisal, *Advanced Clustering: Teori Dan Aplikasi*. Deepublish, 2015.
- [4] D. Agustina, D. Pujotomo, and D. Puspitasari, “Pengembangan Strategi Hubungan Pelanggan Berdasarkan Segmentasi Pelanggan Menggunakan Data Mining,” vol. 8, no. 7, pp. 117–121, 2012.
- [5] V. A. Angelie, “Segmentasi Pelanggan Menggunakan Clustering K-Means Dan Model Rfm (Studi Kasus : Pt . Bina Adidaya Surabaya),” 2017.
- [6] S. K. M. K. D. A. N. A. P. S. K. M. K. Amril Mutoi Siregar, *DATA MINING: Pengolahan Data Menjadi Informasi dengan RapidMiner*. CV Kekata Group.
- [7] Asroni and R. Adrian, “Penerapan Metode K-Means Untuk Clustering Mahasiswa Berdasarkan Nilai Akademik Dengan Weka Interface Studi Kasus Pada Jurusan Teknik Informatika UMM Magelang,” *Ilm. Semesta Tek.*, vol. 18, no. 1, pp. 76–82, 2015.

- [8] S. Adinugroho and Y. A. Sari, *Implementasi Data Mining Menggunakan Weka*. Universitas Brawijaya Press, 2018.
- [9] T. E. Balqiah and Hapsari Setyowardhani, “Pemahaman Konsep dan Studi Perilaku Konsumen dalam Pengembangan Strategi Pemasaran,” pp. 1–49.
- [10] R. Kurniawan, *Analisis Regresi*. Prenada Media, 2016.
- [11] A. A. Caraka, H. Haryanto, D. P. Kusumaningrum, S. Astuti, F. I. Komputer, and U. D. Nuswantoro, “Logika Fuzzy Menggunakan Metode Tsukamoto,” *Techno.COM*, vol. 14, no. 4, pp. 255–265, 2015.
- [12] S. E. M. M. Nugroho J. Setiadi, *Perilaku Konsumen*. Kencana, 2015.
- [13] A. R. Condribimo, A. V. D. Sano, and H. Nindito, “The Application Of K-Means Algorithm For LQ45 Index on Indonesia Stock Exchange,” *ComTech Comput. Math. Eng. Appl.*, vol. 7, no. 2, p. 151, 2016.
- [14] E. luthfi and U. Amikom, *Algoritma Data Mining*. Penerbit Andi.
- [15] W. M. P. Dhuhita, “Clustering Menggunakan Metode K-Mean Untuk Menentukan Status Gizi Balita,” *J. Inform. Darmajaya*, vol. 15, no. 2, pp. 160–174, 2015.
- [16] M. R. Faisal, *Seri Belajar Pemrograman: Pengenalan Bahasa Pemrograman R. M Reza Faisal*, 2016.
- [17] A. K. Ekasaria and M. Edwar, “Analisis Faktor Yang Mempengaruhi Perilaku Konsumen Dalam Keputusan Menggunakan Listrik Prabayar,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.
- [18] K. Handoko, “Penerapan Data Mining Dalam Meningkatkan Mutu Pembelajaran Pada Instansi Perguruan Tinggi Menggunakan Metode K-Means Clustering (Studi Kasus Di Program Studi Tkj Akademi Komunitas Solok Selatan),” *J. Teknol. dan Sist. Inf.*, vol. 02, no. 03, pp. 31–40, 2016.
- [19] D. N. G. W. Nurcahyo, *Algoritma Data Mining Dan Pengujian*. Deepublish, 2015.
- [20] J. Suntoro, *Data Mining: Algoritma dan Implementasi dengan Pemrograman PHP*. Elex Media Komputindo, 2019.

- [21] S. Manai, *60 Menit Belajar Python: Python for everyone*. Syamsudin M, 2019.
- [22] A. F. Khairati, A. A. Adlina, G. F. Hertono, and B. D. Handari, “Kajian Indeks Validitas pada Algoritma K-Means Enhanced dan K-Means MMCA,” *Pros. Semin. Nas. Mat.*, vol. 2, pp. 161–170, 2019.
- [23] L. Sitorus, *Algoritma dan Pemrograman*. Penerbit Andi.
- [24] A. H. Lubis, “Model Segmentasi Pelanggan Dengan Kernel K-Means Clustering Berbasis Customer Relationship Management,” *J. Penelit. Tek. Inform.*, vol. 1, pp. 36–41, 2016.
- [25] A. Kadir, *Langkah Mudah Pemrograman R*. Elex media komputindo, 2019.
- [26] I. I. Marbun, R. Ginting, and Emalisa, “Analisis Faktor-Faktor Yang Mempengaruhi Perilaku Konsumen Kopi Luwak Bermerek Di Kota Medan,” pp. 1–14, 2014.
- [27] A. T. Rahman, Wiranto, and A. Rini, “Coal Trade Data Clustering Using K-Means (Case Study Pt. Global Bangkit Utama),” *ITSMART J. Teknol. dan Inf.*, vol. 6, no. 1, pp. 24–31, 2017.
- [28] J. Enterprise, *Python untuk Programmer Pemula*. Elex media komputindo, 2019.
- [29] M. G. Sadewo, A. P. Windarto, and A. Wanto, “Penerapan Algoritma Clustering Dalam Mengelompokkan Banyaknya Desa/Kelurahan Menurut Upaya Antisipasi/ Mitigasi Bencana Alam Menurut Provinsi Dengan K-Means,” *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 2, no. 1, pp. 311–319, 2018.
- [30] D. B. Saputra and E. Riksakomara, “Implementasi Fuzzy C-Means dan Model RFM untuk Segmentasi Pelanggan (Studi Kasus : PT. XYZ),” *J. Tek. ITS*, vol. 7, no. 1, pp. 1–6, 2018.
- [31] Y. Wicaksono, “Segmentasi Pelanggan Bisnis Dengan Multi Kriteria Menggunakan K-Means,” *Indones. J. Bus. Intell.*, vol. 1, no. 2, p. 45, 2019.
- [32] J. Enterprise, *Aplikasi Face Detector dan Digital Imaging dengan Python*. Elex Media Komputindo, 2018.

220 DAFTAR PUSTAKA

- [33] A. P. Windarto, “Penerapan Datamining Pada Ekspor Buah-Buahan Menurut Negara Tujuan Menggunakan K-Means Clustering Method,” *Techno.Com*, vol. 16, no. 4, pp. 348–357, 2017.
- [34] Wardana, *Belajar Pemrograman dan Hacking Menggunakan Python*. Elex Media Komputindo, 2019.
- [35] G. F. Wulandari, “Segmantasi Pelanggan Menggunakan Algoritma K-Means Untuk Customer Relationship Management (CRM) Pada Hijab Miulan,” *Ind. Mark. Manag.*, vol. I, no. segmentasi pelanggan, p. 7, 2014.

# **SEGMENTASI PELANGGAN MENGGUNAKAN PYTHON**

Dalam bisnis apapun baik dalam bidang jasa maupun manufaktur salah satu faktor yang mempengaruhi kemajuan sebuah perusahaan adalah pemasaran. Pemasaran bukan saja hanya pengembangan produk dan jasa yang dibutuhkan tetapi segmentasi pelanggan juga harus diperhitungkan. Penulisan buku ini dimaksudkan untuk memberikan arahan atau sarana pembelajaran kepada pembaca mengenai data mining serta penerapan data mining dalam kehidupan sehari-hari. Data mining merupakan suatu proses penggalian data untuk mencari suatu informasi, salah satunya adalah pada penentuan segmentasi pelanggan. Kemudian proses penggalian data tersebut dilakukan dengan menggunakan tools bahasa pemrograman python. Python merupakan sebuah bahasa pemrograman yang interpretative karena dianggap memiliki kemudahan untuk dipelajari serta memiliki fokus terhadap keterbacaan kode. Dapat dikatakan, bahwa python merupakan salah satu bahasa pemrograman yang memiliki kode-kode pemrograman yang jelas, mudah dipahami, lengkap serta tepat digunakan dalam menyelesaikan permasalahan data mining.

**ISBN 978-602-53897-6-4**

