Analisis Faktor Kepuasan Pelanggan terhadap Produk Nike dan Adidas dengan *Decision Tree* dan *Random Forest*

Kelly Mae[1], Fareza Ananda Putra[2], Leony Hana Noah Zebua[3], Reuben Ryan Peter[4]
Program Studi Sistem Informasi, Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
kelly.mae@student.umn.ac.id[1], fareza.putra@student.umn.ac.id[2], leony.zebua@student.umn.ac.id[3], reuben.peter@student.umn.ac.id[4]

***Abstrak* — Nike dan Adidas adalah merek tertua di industri sepatu. Kedua merek mewakili kinerja yang sangat baik dalam produk mereka. Moto Nike adalah "*Just Do It*", sedangkan moto Adidas adalah "*Nothing is Impossible*". Kedua merek ini telah memperkenalkan produk ikonik selama bertahun-tahun. Beberapa produk Nike Legendary adalah Jordans, Air Max, Airforce. Di saat yang sama, Adidas juga memiliki produk ikonik seperti Superstars, Ultraboost, NMD dan Hit The Yeezy's terbaru. Dengan menggunakan *dataset* yang disediakan, kita bisa menggali lebih dalam analisis produk-produk dapat berbeda satu sama lain. Oleh karena itu, penelitian menggunakan algoritma *Decision Tree* dan *Random Forest* dapat digunakan untuk menemukan kombinasi data yang tidak terduga.**

Index Terms* — *Nike, Adidas, Classification, Python, CRISP-DM

I. INTRODUCTION (LATAR BELAKANG)

"Nike" dalam mitologi Yunani, yang berarti dewi kemenangan, adalah perusahaan alas kaki, pakaian, dan peralatan olahraga Amerika. Nama Nike sendiri mudah dikenali dan merupakan salah satu merek terbaik di industrinya. Nike dikenal luas karena berbagai alasan sebagai sepatu, pakaian, dan aksesori terbaik untuk berbagai aktivitas olahraga dan kebugaran. Selain itu, Nike juga dikenal memiliki hubungan dengan atlet kelas dunia. Salah satu atlet paling terkenal yang membantu memperkenalkan nama Nike ke dunia adalah Michael Jordan. Kemudian banyak atlet mulai memakai merek tersebut, meski itu bukan bagian dari kontrak Nike. Ini adalah fokus perhatian dunia. Pada saat yang sama, diyakini juga bahwa Nike memiliki gudang teknologi yang memungkinkan perusahaan untuk terus berinovasi, memiliki produk berkualitas tinggi, dan periklanan yang dinamis [1].

Adidas Salomon AG atau yang lebih dikenal dengan Adidas adalah perusahaan alas kaki dan berbagai perlengkapan olahraga lainnya yang berbasis di Herzogenaurach, Jerman. Perusahaan ini pertama kali didirikan oleh Adolf (Adi) Dassler pada 18 Agustus 1949. Nama Adidas sendiri berasal dari nama pendirinya. Produk Adidas pertama kali diperkenalkan ke pasar pada tahun 1950. Adidas didedikasikan untuk produksi peralatan olahraga untuk atlet dalam olahraga "ekstrim" tertentu, seperti lompat jauh, di mana seorang atlet bernama Dick Fosbury melompat dengan sepatu Adidas. Pada final Piala Dunia 1970, ketika Jerman mengalahkan Belanda dengan selisih 21 poin di final, Franz Beckenbauer dari Jerman mengenakan Adidas pada upacara penobatan. Setelah kematian Adi Dassler pada 1980-an, istrinya Kathe dan anak-anak mereka mengambil alih perusahaan tersebut. Pada 1990-an, Adidas tumbuh menjadi perusahaan manufaktur berbasis penjualan di bawah CEO Robert Louis Dreyfus. Pada saat yang sama, pada tahun 1995, Adidas mulai "*go public*", mencatatkan sahamnya untuk pertama kalinya di bursa saham Frankfurt dan Paris [2].

Dari pernyataan sebelumnya, kami menyimpulkan bahwa membangun merek harus memiliki strategi yang baik agar merek tersebut dapat dikenal oleh semua orang dan menjaga keberlangsungan perusahaan untuk waktu yang lama [3]. Oleh karena itu, merek besar seperti Nike dan Adidas juga memiliki strategi yang hebat. Di antara strategi-strategi yang berbeda ini, banyak faktor yang dibutuhkan kedua perusahaan untuk mencapai reputasi yang baik di mata pelanggan dan di panggung dunia. Saat melakukan pembelian, pembeli merasakan kebanggaan khusus atas nilai produk itu sendiri. Pembeli merasa bangga dengan produk yang ada melalui beberapa faktor, yaitu harga jual, *review* orang lain terhadap merek itu sendiri, dan reputasi merek itu sendiri [4]. Oleh karena itu, rumusan masalah dari penelitian ini adalah untuk mengetahui aspek apa saja yang menjadi pertimbangan pelanggan dalam membeli dengan menggunakan algoritma *Decision Tree* dan *Random Forest*. Penelitian ini juga berfokus pada perbandingan dua algoritma untuk menemukan algoritma mana yang lebih cocok untuk diimplementasikan dengan mempelajari akurasinya.

II. LITERATURE REVIEW (TINJAUAN PUSTAKA)

A. *CRISP-DM (Cross Industry Standard Process - Data Mining)*

CRISP-DM (*Cross Industry Standard Process - Data Mining*) adalah suatu model proses *data mining* (*data mining framework*) yang merupakan sebuah metode netral yang dapat digunakan dalam segala lini bisnis dan berbagai *tools*. *CRISP-DM* merupakan sebuah tahapan dalam sebuah proyek yang tiap tahapannya dan penjabaran memberikan sebuah gambaran siklus hidup (*lifecycle*) dari *data mining* jika dilihat sebagai model proses. Adapun 6 tahap dalam *CRISP-DM*, yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment* [5].

Business Understanding

Business understanding merupakan langkah yang berfokus dalam memahami tujuan dan ketentuan dalam proyek yang dilihat dari perspektif suatu bisnis perusahaan, dimana pada akhirnya pengetahuan yang diperoleh akan diubah menjadi definisi dari suatu permasalahan pada data mning serta rencana proyek awal yang didesain untuk mencapai tujuan yang sudah ditentukan. Business understanding merupakan langkah yang memegang peranan penting dalam menentukan tujuan bisnis yang mampu mengarahkan pada proyek selanjutnya.

Data Understanding

Data understanding merupakan langkah dimana data awal akan dikumpulkan untuk meningkatkan pemahaman suatu perusahaan mengenai data tersebut. Dalam data understanding, diperlukan proses identifikasi akan potensi munculnya masalah kualitas data, wawasan terdahulu mengenai data yang bersangkutan, dan kemungkinan hipotesis yang merupakan hasil dari subset data yang mampu menampilkan informasi tersirat dari data tersebut.

Data Preparation

Data preparation merupakan langkah yang melibatkan penugasan akan peran yang spesifik, seperti *data reduction*, *data wrangling* dan *data cleansing*, serta *data transformation* berupa pembuatan variable *dummy* untuk kebutuhan analisis dan pengujian pada langkah selanjutnya.

Modeling

Modeling merupakan langkah yang mencakup seleksi dan pengembangan dari teknik dan model analitik yang digunakan. Dalam *modeling*, beberapa bagian dari dataset seringkali disisihkan untuk kegiatan training dan validation atas model yang dibuat.

Evaluation

Evaluation merupakan langkah dimana perusahaan akan melakukan peninjauan dan interpretasi hasil analisis dari kegiatan *modeling* dalam konteks tujuan bisnis dan indikator akan keberhasilan yang sesuai dengan ketentuan yang telah ditetapkan pada langkah awal.

Deployment

Deployment merupakan langkah akhir dimana hasil pengetahuan yang didapatkan dari data analisis akan diubah ke dalam bentuk serangkaian rekomendasi yang dapat digunakan pada tindakan selanjutnya. Dalam *deployment*, diperlukan pemahaman akan efisiensi dalam menyampaikan hasil analisis terhadap unsur bisnis memegang peran utama dalam keberhasilan suatu proyek analitik.

B. *Introduction to Decision Tree and Random Forest Algorithm*

Decision Tree merupakan suatu pohon di mana pengujian atribut dan simpul daun ditunjukkan pada setiap node untuk memberikan klasifikasi. Dalam kasus klasifikasi, contoh tes dimulai pada simpul akar yang meguji nilai fitur per simpul dan memilah ke cabang yang sesuai hingga mencapai klasfikasi melalui simpul daun [6].

Random forest merupakan sebuah kemungkinan entitas menggunakan sekumpulan algoritma *CART* untuk melakukan prediksi. *Rondom forest* melibatkan pembentukan subset dari data pelatihan melalui proses pertukaran metode *bagging*. Dalam hal ini, datanya akan nilai yang sama dapat dipilih lebih dari satu kali, sedangkan tanggal lain juga tidak bisa dipilih [7].

C. Penggunaan *Decision Tree* dan *Random Forest*

Ada beberapa penggunaan algoritma *decision tree*. *Decision tree* dapat membagi beberapa dataset menjadi kelas yang terpisah. *Decision tree* digunakan untuk semua jenis variabel target; Namun, ini terutama digunakan untuk yang berbentuk kategoris. *Decision Tree* melibatkan penggunaan metode perolehan informasi untuk melakukan data membelah dan menghitung dengan metode*Entropi* atau *Gini Index* dalam pemisahan data untuk menemukan homogenitas dalam kumpulan datanya [8]. Selain itu, *decision tree* juga mampu melakukan prediksi hasil untuk laporan masa depan. Hasil dari, *decision tree* adalah salah satu metode yang paling efektif untuk penambangan data.

Sebagai perbandingan, ada beberapa penggunaan Algoritma *random forest*. *Random forest* memiliki Metode "*Bagging*" yang lebih umum dan memiliki kemampuan untuk mengintegrasikan berbagai jenis fitur. *Random forest* memiliki pendekatan suara mayoritas yang dapat meminimalkan adanya kesalahan klasifikasi efektif [9]. *Random Forest* dapat menentukan peringkat signifikansi terkait untuk setiap prediktor yang didasarkan pada kesalahan prediksi regresi pada *Out-Of-Bag* atau *OOB* [10].

D. Keunggulan dan Kelemahan Algoritma *Decision Tree*

Algoritma *Decision tree* memiliki keunggulan dalam mengklasifikasikan catatan yang tidak diketahui secara cepat. *Decision tree* sangat bagus dengan adanya redundan atribut dan sedikit tegas di hadapan kebisingan jika metode *overfitting* diberikan. Namun, *decision tree* memiliki kelemahan yaitu tidak ada data yang berlaku memiliki efek buruk dalam konstruksi keputusan pohon. Pada kesempatan ini, setiap perubahan kecil pada data dapat mempengaruhi keseluruhan tampilan pohon keputusan. Selain itu, sub-pohon di pohon keputusan dapat menjadi diproduksi dalam jumlah banyak [11].

E. Keunggulan dan Kelemahan Algoritma *Random Forest*

Algoritma *random forest* memiliki beberapa keuntungan dalam menangkap interaksi yang rumit di antara fitur yang digunakan dengan memiliki pengetahuan tentang kombinasi nonlinier dari mereka. *Random forest* adalah mampu beroperasi secara merata dengan baik dengan kontinu, diskrit, atau nilai yang hilang dengan sedikit atau tanpa modifikasi. *Random forest* relatif kuat untuk *outlier*, seperti sebagai contoh berisik dan label berisik. Namun demikian, *random forest* memiliki kelemahan dimana skema yang dapat ditindaklanjuti terbukti secara tidak langsung mengingat contoh. Akibatnya, hasil model adalah secara tidak langsung dapat diinterpretasikan oleh manusia [12].

F. Training and Testing

Data *training* dan *testing* hanya ada pada klasifikasi salah satu jenis algoritma *Machine Learning* yaitu *Supervised Learning*. *Data training* biasa disiapkan untuk melatih algoritma untuk mencari model yang sesuai sehingga bisa mendapatkan hasil yang maksimal. Berbeda dengan *data training*, *data testing* digunakan untuk menguji dan mengetahui peforma model yang didapatkan pada tahapan *testing*.

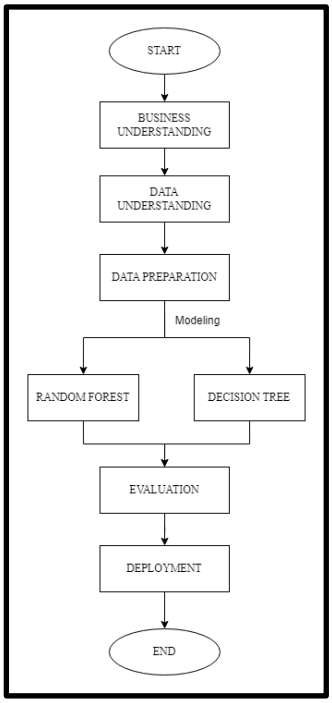
G. Confusion Matrix

Confusion Matrix adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah yang merupakan representasi hasil proses klasifikasi pada confusion matrix yaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative*. *True Positive* adalah ketika melakukan prediksi positif dan yang di prediksi benar. *True Negative* adalah ketika melakukan prediksi negatif dan yang dipreksi itu benar. *False Positive* adalah ketika melakukan prediksi positif tetapi hasil prediksi salah. *False Negative* adalah ketika melakukan prediksi negatif tetapi yang diprediksi salah.

III. METHODOLOGY (METODOLOGI PENELITIAN)

Penelitian ini menggunakan data pembelian sepatu Nike dan Adidas pada tahun 2021. Data penjualan sepatu yang berasal dari *website* penjualan online, sehingga kepuasan pelanggan ditentukan oleh beberapa faktor berupa harga jual, nama *brand*, *discount*, *review*, rating dan lain-lain. Data yang menjadi fokus penelitian ini adalah gambaran tentang kepuasan pelanggan. Pengumpulan data bersumber sekunder dari situs web penghimpun data, yaitu Kaggle (https://www.kaggle.com/).

Metode penelitian yang diterapkan dalam penelitian ini adalah metode deskriptif kuantitatif (*descriptive quantitative*). Metode ini tepatnya digunakan untuk menjelaskan suatu kondisi, fenomena, ataupun variabel penelitian berdasarkan observasi yang telah dilakukan. Algoritma yang diimplementasikan dalam penelitian ini adalah *Decision Tree* dan *Random Forest.* Penggunaan algoritma ini tepat karena mayoritas variable merupakan variabel numerik dan algoritma-algoritma yang digunakan ini juga tepat untuk menyelesaikan masalah klasifikasi. Adapun alur yang digunakan dalam penelitian ini, yaitu sebagai berikut.



Gambar 1. Alur Penelitian

Berikut berupa tahap yang dijalankan untuk memperoleh hasil dari data yang dipakai.

1. Memanggil semua *library* yang diperlukan untuk algoritma Decision Tree, *Random Rorest.*

2. Mengimport dataset yang digunakan yaitu dataset Penjualan Nike dan Adidas.
3. Mempelajari dataset dan memeriksa nilai *missing value* pada *dataset*.
4. Membuat visualisasi berisi dengan variabel prediktor terhadap variabel target.
5. Membuat plot korelasi dan *pair plots* untuk melihat hubungan antar variabel-variabel prediktor.
6. Membagi dataset menjadi data *training* dan data *testing*, dengan komposisi 70:30.
7. Menerapkan algoritma *Decision Tree* sebelum dan sesudah proses *pruning*.
8. Mengevaluasi model *Decision Tree* menggunakan *Classification Report* dan *Confusion Matrix*.
9. Menerapkan algoritma *Random Forest*.
10. Mengevaluasi model *Random Forest* menggunakan *Confusion Matrix*.
11. Membandingkan kedua model yang telah dirancang dan mencari model dengan akurasi terbaik.
12. Membuat kesimpulan dari penelitian yang dilakukan.
13. Membuat saran dan limitasi dari penelitian yang dilakukan.

Adapun penggunaan *framework Cross-Industry Standard Process for Data Mining* (*CRISP-DM*) berupa gambaran mengenai siklus hidup dari proyek yang diperoleh dari *data mining* serta tahapan gambarannya. Terdapat 6 tahapan pada proses *CRISP-DM* yaitu, *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment*. 6 tahapan tersebut dijelaskan sebagai berikut:

A. Business Understanding

Hal yang diperlukan yang sesuai dengan kebutuhan serta tujuan dari sudut pandang bisnis selanjutnya dimana hal ini diartikan pengetahuan ke dalam bentuk pendefinisian masalah pada metode *data mining* dimana hal ini akan mempengaruhi strategi untuk mencapai tujuan yang diinginkan pada proses *data mining*. Memahami tujuan dan persyaratan proyek dari perspektif bisnis, dimana hal ini mengubah pengetahuan ini menjadi definisi masalah serta *data mining* dan juga rencana awal yang dirancang untuk mencapai tujuan perusahaan.

B. Data Understanding

Beberapa hal yang dilakukan pada tahap ini berupa memahami kebutuhan serta tujuan dari sudut pandang bisnis selanjutnya mengartikan pengetahuan ke dalam bentuk pendefinisian masalah pada *data mining* dan juga menentukan *planning* untuk mencapai tujuan *data mining*. Memahami tujuan dan persyaratan proyek dari perspektif bisnis, dan kemudian mengubah pengetahuan ini menjadi definisi masalah data mining dan rencana awal yang dirancang untuk mencapai tujuan yang diinginkan.

C. Data Preparation

Saat semua aktivitas sudah dilakukan maka tahap selanjutnya yang perlu untuk dilakukan adalah menyiapkan data yang akan digunakan, dari data mentah awal. Dimana hal tersebut termasuk tabel, kasus, dan pemilihan atribut serta transformasi dan pembersihan data untuk alat pemodelan. Berikut merupakan ada beberapa hal yang akan dilakukan mencakup melakukan pembersihan data, melakukan pemilihan data, record dan atribut-atribut, dan juga melakukan transformasi terhadap data atau transformasi data, untuk dijadikan masukan dalam tahap pemodelan.

D. Modeling

Penerapan dari penggunaan teknik dari pemodelan dengan parameter yang telah disesuaikan dengan teknik *data mining*. Oleh karena itu, sering diperlukan sebuah persiapan dikarenakan beberapa data memiliki tahap persiapan yang khusus untuk setiap tahap-tahapnya.

E. Evaluation

Evaluasi model ini mencakupi seluruh, dan juga setiap langkah yang dijalankan untuk menyusun model dimana hal ini digunakan untuk memastikan model tersebut agar sesuai dengan tujuan bisnis yang telah ditetapkan dari awal. Tentu terdapat beberapa masalah bisnis, dimana hal ini penting untuk dipertimbangkan. Pada akhir fase ini, keputusan tentang penggunaan hasil data mining tercapai.

F. Deployment

Pada tahap ini hal yang dilakukan adalah untuk membuat artikel ataupun jurnal untuk digunakan pada model yang dihasilkan.

IV. Results and Analysis (Hasil dan Analisis)

A. Business Understanding

Penelitian ini melibatkan produk dari 2 *brand* sepatu terkenal, yaitu Nike dan Adidas sebagai objek penelitian. Penelitian ini menggunakan *dataset* yang dikumpulkan secara manual. *Dataset* ini diambil dari situs web *dataset*, yaitu Kaggle.com. Penelitian ini diharapkan dapat membawa hasil analisis dan model yang mampu memprediksi kepuasan pelanggan lebih lanjut dalam penggunaan produk Nike dan Adidas.

B. Data Understanding

Dataset yang diperoleh dari situs web Kaggle terdiri dari 3.268 data yang digunakan untuk menggambarkan faktor kepuasan pelanggan terhadap produk Nike dan Adidas. *Dataset* ini terdiri atas 9 variabel prediktor dan 1 variabel target berupa variabel *Rating* yang kemudian akan diolah dengan menggunakan bahasa pemrograman Python. Berikut merupakan variabel yang akan digunakan dalam

memprediksi apakah pelanggan puas dengan produk sepatu Nike maupun Adidas (*Rating*):

1. *Product Name*
2. *Product ID*
3. *Listing Price*
4. *Sale Price*
5. *Discount*
6. *Brand*
7. *Description*
8. *Reviews*
9. *Last Visited*

Data Exploration and Understanding

```
import pandas as pd # Data Analysis and Data Manipulation
import matplotlib.pyplot as plt # Data Visualization
import seaborn as sns # Data Visualization

from sklearn.preprocessing import StandardScaler # Data Prepocessing
from sklearn.model_selection import train_test_split # Data Splitting
from sklearn.tree import DecisionTreeClassifier # Decision Tree
from sklearn.ensemble import RandomForestClassifier # Random Forest
from sklearn.tree import plot_tree # Decision Tree Plot
from sklearn import metrics # Evaluation Metrics
from sklearn.metrics import classification_report, plot_confusion_matrix, accuracy_score # Evaluation Metrics
```

Gambar 2. *Import Libraries*

Gambar 2 menunjukkan kegiatan *import* semua *library* yang dibutuhkan untuk penelitian ini guna memanggil fungsi yang akan digunakan. Adapun *library* pertama berupa *Pandas* (*Python for Data Analysis*) yang secara umum berfokus pada proses analisis data dalam Python. Adapun *library* Matplotlib yang digunakan untuk kebutuhan *plot* dalam bahasa pemrograman Pythton. *Library Seaborn* merupakan *library* visual dari Python yang berdasar pada *Matplotlib* untuk kebutuhan visualisasi data secara statistik. Selain itu, terdapat *library* yang ditujukan untuk kegiatan pemodelan berupa *sklearn* (*Scikit-Learn*) dimana *library* ini dapat digunakan untuk kebutuhan pemrosesan data, pembagian data menjadi data *training* dan data *testing*, pembuatan model *Decision Tree* dan *Random Forest*, hingga melakukan evaluasi atas model yang telah dibuat dengan *metrics* dan *classification report*.

	Product Name	Product ID	Listing Price	Sale Price	Discount	Brand	Description	Rating	Reviews	Last Visited
0	Women's adidas Originals NMD_Racer Primeknit S...	AH2430	14999	7499	50	Adidas Adidas ORIGINALS	Channeling the streamlined look of an '80s rac...	4.8	41	2020-04-13T15:06:14
1	Women's adidas Originals Sleek Shoes	G27341	7599	3799	50	Adidas ORIGINALS	A modern take on adidas sport heritage, tailor...	3.3	24	2020-04-13T15:06:15
2	Women's adidas Swim Puka Slippers	CM0081	999	599	40	Adidas CORE / NEO	These adidas Puka slippers for women's come wi...	2.6	37	2020-04-13T15:06:15
3	Women's adidas Sport Inspired Questar Ride Shoes	B44832	6999	3499	50	Adidas CORE / NEO	Inspired by modern tech runners, these women's...	4.1	35	2020-04-13T15:06:15
4	Women's adidas Originals Taekwondo Shoes	D98205	7999	3999	50	Adidas ORIGINALS	This design is inspired by vintage Taekwondo s...	3.5	72	2020-04-13T15:06:15
...	...	...	...	...	...	...	...	...	...	...
3263	Air Jordan 6 Retro	CI1236-100	15995	12797	0	Nike	The Air Jordan 6 Retro recaptures the memorabl...	5.0	1	2020-04-13T15:41:01
3264	Nike Phantom Venom Club IC	AO0578-717	4995	3497	0	Nike	The Nike Phantom Venom Club IC is engineered f...	0.0	0	2020-04-13T15:41:03
3265	Nike Mercurial Superfly 7 Academy TF	AT7978-414	8495	5947	0	Nike	The soft upper of the Nike Mercurial Superfly ...	5.0	1	2020-04-13T15:41:07
3266	Nike Air Max 98	AH6799-300	0	16995	0	Nike	The Nike Air Max 98 features the OG design lin...	4.0	4	2020-04-13T15:41:19
3267	Nike P-6000 SE	CJ0585-600	8995	6297	0	Nike	A mash-up of Pegasus' past, the Nike P-6000 SE...	0.0	0	2020-04-13T15:42:57

3268 rows × 10 columns

Gambar 3. *Dataset* Adidas *Vs* Nike

Gambar 3 menunjukkan pemanggilan *dataset* yang digunakan dengan menggunakan fungsi pd.read_csv(). *Dataset* ini berisi seluruh variabel beserta observasinya yang akan digunakan untuk pembuatan model *Machine Learning* yang berbasis klasifikasi.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3268 entries, 0 to 3267
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Product Name   3268 non-null   object
 1   Product ID     3268 non-null   object
 2   Listing Price  3268 non-null   int64
 3   Sale Price     3268 non-null   int64
 4   Discount       3268 non-null   int64
 5   Brand          3268 non-null   object
 6   Description    3265 non-null   object
 7   Rating         3268 non-null   float64
 8   Reviews        3268 non-null   int64
 9   Last Visited   3268 non-null   object
dtypes: float64(1), int64(4), object(5)
memory usage: 255.4+ KB
```

Gambar 4. Struktur Data Adidas *Vs* Nike

Gambar 4 menunjukkan struktur data dalam *dataset* Adidas *Vs* Nike dengan fungsi info(). Terdapat 10 variabel dalam *dataset* tersebut, yaitu *Product Name*, *Product ID*, *Listing Price*, *Sale Price*, *Discount*, *Brand*, *Description*, *Rating*, *Reviews*, dan *Last Visited* dengan tipe data yang beragam berupa *float*, *integer*, dan *object*.

	count	mean	std	min	25%	50%	75%	max
Rating	3268.0	3.242105	1.428856	0.0	2.6	3.5	4.4	5.0

Gambar 5. *Summary* Variabel Target *Rating*

Gambar 5 menunjukkan *summary* dari variabel target *Rating* yang mencakup nilai *count*, *mean*, *std* (*Standard Deviation*), *min*, 25% (*Q1*), 50% (*Q2*), 75% (*Q3*), dan *max*.

```
Product Name      0
Product ID        0
Listing Price     0
Sale Price        0
Discount          0
Brand             0
Description       3
Rating            0
Reviews           0
Last Visited      0
dtype: int64
```

Gambar 6. Pengecekan Nilai *Missing Value*

Berdasarkan Gambar 6, adapun pengecekan nilai yang mengandung *missing value* pada *dataset* Adidas *Vs* Nike dengan menggunakan fungsi *is.null()*. Hasil pengecekan tersebut menunjukkan bahwa tidak terdapat nilai *missing value* dalam *dataset* Adidas *Vs* Nike.

C. *Data Preparation*

Data Filtering and Cleansing

```
# Brand Grouping
brandGroup = nikeAdidas.groupby('Brand')

# Nike
nike = brandGroup.get_group('Nike')

# Adidas
originals = brandGroup.get_group('Adidas ORIGINALS')
neo = brandGroup.get_group('Adidas CORE / NEO')
sports = brandGroup.get_group('Adidas SPORT PERFORMANCE')
type = [originals, neo, sports]

adidas = pd.concat(type)
adidas.reset_index(inplace = True, drop = True)
```

Gambar 7. Pengelompokkan Produk berdasarkan *Brand*

Gambar 7 menunjukkan pengelompokkan produk berdasarkan *brand* Adidas dan Nike menggunakan fungsi *gropby*() serta pemerolehan grup dengan fungsi *get_group()*. Adapun penggabungan data dari berbagai produk Adidas yang disimpan ke dalam variabel Adidas dengan fungsi *concat()* dan penghapusan indeks dengan fungsi *reset_index()*.

```
# Filter Product Rating
def ratingCategory(x):
    if x <= 4:
        return "Bad"
    else:
        return "Good"

nikeAdidas["Rating"] = nikeAdidas["Rating"].apply(ratingCategory)
```

Gambar 8. *Filter* Variabel Target *Rating*

Pada Gambar 8 di atas, terdapat proses *data filtering* pada variabel target *Rating* dengan mengacu pada nilai *75%* (*Q3*). Proses tersebut dilakukan sebagai bentuk penyerdehanaan data oleh tipe datanya yang bersifat numerikal guna mempermudah proses analisa data. Oleh karena itu, data dapat diserdehanakan dengan label "*Good*" dan "*Bad*" dengan menggunakan fungsi yang telah dibuat. Fungsi tersebut mencakup kondisi apabila nilai pada variabel target *Rating* berada di atas 4, maka akan diberikan label "*Good*", sedangkan nilai pada variabel target *Rating* yang berada di bawah dan sama dengan 4 termasuk dalam label "*Bad*".

```
# Drop Unneccessary Columns
nikeAdidas = nikeAdidas.drop(columns=["Brand","Description","Last Visited","Product Name","Product ID"])
```

Gambar 9. *Data Cleansing*

Adapun Gambar 9 berupa proses *data cleansing* dengan fungsi *drop()* yang mencakup penghapusan variabel yang tidak digunakan dalam proses analisa data, seperti variabel *Brand*, *Description*, *Last Visited*, *Product Name*, dan *Product ID*.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3268 entries, 0 to 3267
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Listing Price  3268 non-null   int64
 1   Sale Price     3268 non-null   int64
 2   Discount       3268 non-null   int64
 3   Rating         3268 non-null   object
 4   Reviews        3268 non-null   int64
dtypes: int64(4), object(1)
memory usage: 127.8+ KB
```

Gambar 10. Struktur Data setelah *Data Cleansing*

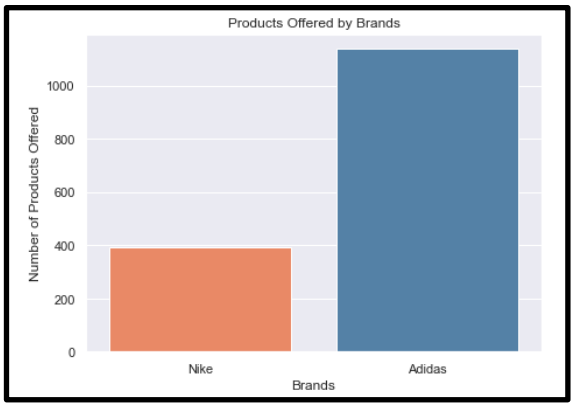
Gambar 10 menunjukkan struktur data dalam *dataset* Adidas *Vs* Nike setelah melakukan *data cleansing* dengan fungsi info(). Terdapat 5 variabel yang akan digunakan untuk proses analisis data, yaitu *Listing Price*, *Sale Price*, *Discount*, *Rating*, dan *Reviews* dengan tipe data yang beragam berupa *integer*, dan *object*.

	Listing Price	Sale Price	Discount	Rating	Reviews
0	14999	7499	50	Good	41
1	7599	3799	50	Bad	24
2	999	599	40	Bad	37
3	6999	3499	50	Good	35
4	7999	3999	50	Bad	72
5	4799	1920	60	Bad	45
6	4799	2399	50	Good	2
7	999	599	40	Bad	7
8	5599	2799	50	Good	16
9	6599	3959	40	Bad	39

Gambar 11. Tampilan 10 Baris Pertama

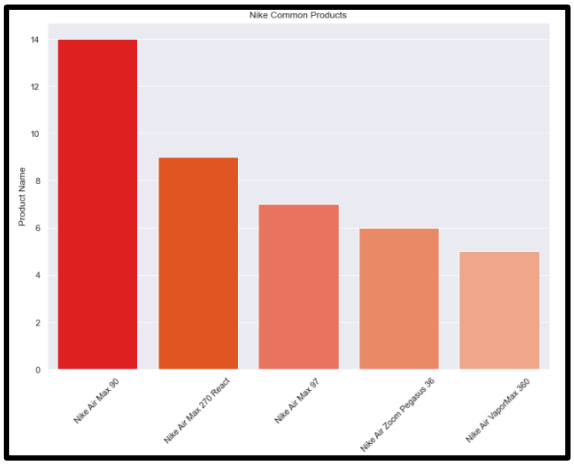
Gambar 11 menunjukkan tampilan 10 baris pertama pada *dataset* Adidas *Vs* Nike setelah melalui proses *data filtering* dan *cleansing*.

Data Visualization



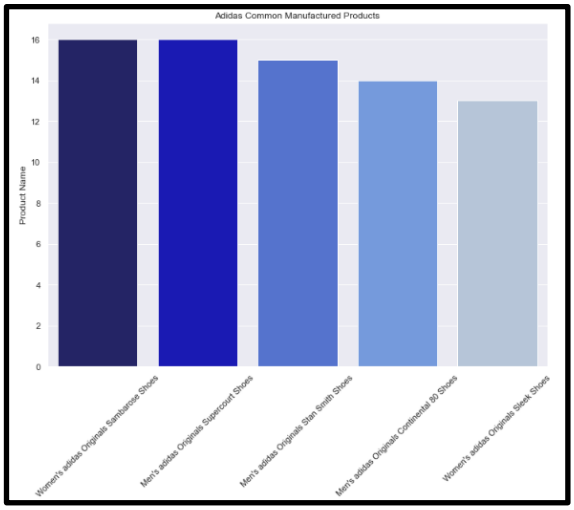
Gambar 12. Distribusi *Products Offered by Brands*

Gambar 12 menunjukkan distribusi akan produk yang ditawarkan berdasarkan *brand* (*Products offered by brands*), yaitu Nike dan Adidas. Berdasarkan *bar plot* di atas, dapat terlihat bahwa produk yang ditawarkan oleh Adidas lebih tinggi daripada produk yang ditawarkan oleh Nike.



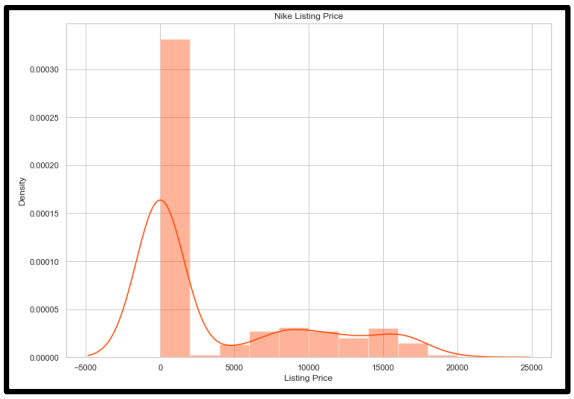
Gambar 13. Distribusi Nike *Common Products*

Gambar 13 menunjukkan distribusi *common products* dari Nike (Nike *common products*). Berdasarkan *bar plot* di atas, dapat terlihat bahwa Nike Air Max 90 merupakan *common products* dari Nike dengan frekuensi tertinggi, sedangkan Nike Air VaporMax 360 memiliki frekuensi terendah dari *common products* lainnya milik Nike.



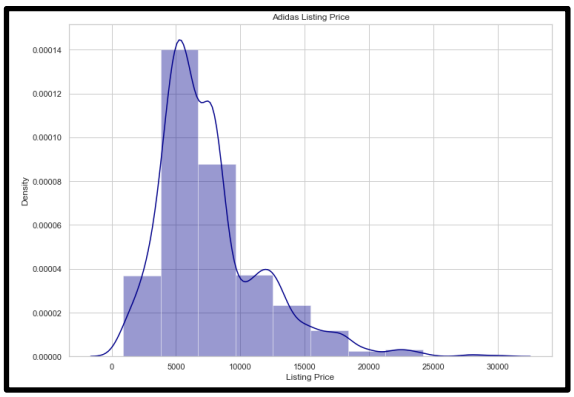
Gambar 14. Distribusi Adidas *Common Products*

Gambar 14 menunjukkan distribusi *common products* dari Adidas (Adidas *common products*). Berdasarkan *bar plot* di atas, dapat terlihat bahwa Women's adidas Originals Sambarose Shoes dan Men's adidas Originals Supercourt Shoes memegang frekuensi tertinggi, sedangkan Women's adidas Originals Sleek Shoes memiliki frekuensi terendah dari *common products* lainnya milik Adidas.



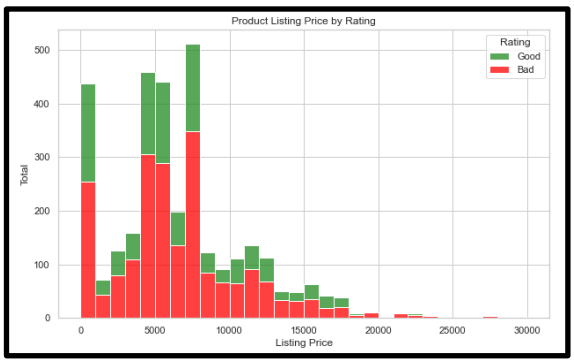
Gambar 15. Distribusi Nike *Listing Price*

Gambar 15 menunjukkan distribusi Nike *Listing Price*, yaitu harga yang tercantum pada produk Nike. Berdasarkan *histogram* di atas, dapat terlihat bahwa semakin rendah *listing price* pada produk Nike, maka semakin banyak pelanggan yang membeli produk Nike tersebut (*high-density* pada 0).



Gambar 16. Distribusi Adidas *Listing Price*

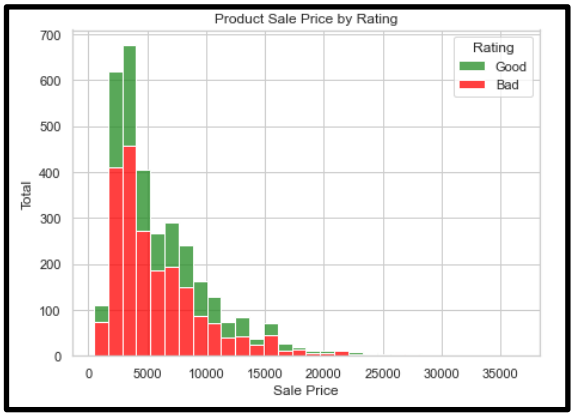
Gambar 16 menunjukkan distribusi Adidas *Listing Price*, yaitu harga yang tercantum pada produk Adidas. Berdasarkan *histogram* di atas, dapat terlihat bahwa pelanggan lebih banyak membeli produk Adidas dengan *listing price* yang berkisar 50.00 hingga 100.00 USD (*high-density* pada 5000).



Gambar 17. Distribusi *Product Listing Price by Rating*

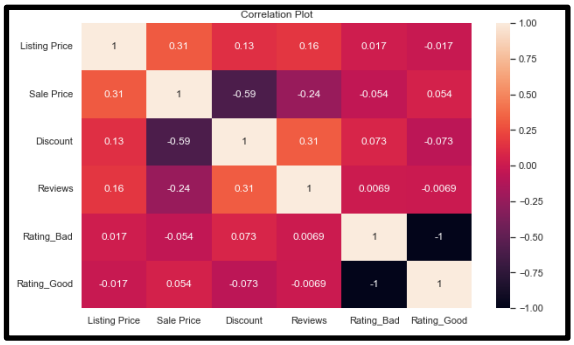
Gambar 17 menunjukkan distribusi *product listing price by rating*. Berdasarkan *histogram* di

atas, dapat terlihat bahwa produk dengan *listing price* sekitar 70.00 USD memiliki *rating* "*Good*" dan "*Bad*" terbanyak yang diikuti oleh produk dengan *listing price* sekitar 40.00 hingga 50.00 USD, dimana *listing price* pada produk tersebut didominasi oleh *rating* "*Good*" (*high-density* pada 7000).



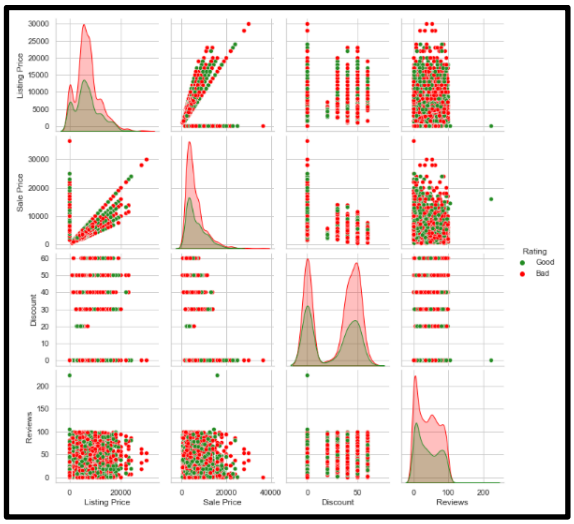
Gambar 18. Distribusi *Product Sale Price by Rating*

Gambar 18 menunjukkan distribusi *product sale price by rating*. Berdasarkan *histogram* di atas, dapat terlihat bahwa produk dengan *sale price* sekitar 20.00 hingga 30.00 USD memiliki *rating* "*Good*" dan "*Bad*" terbanyak, dimana *sale price* pada produk tersebut didominasi oleh *rating* "*Good*" (*high-density* pada 3000).



Gambar 19. *Correlation Plot*

Gambar 19 menunjukkan *correlation plot* dimana terdapat hubungan searah yang lemah pada vairabel *listing price* dengan variabel *sale price*. Selain itu, tidak terdapat hubungan yang kuat antara variabel prediktor dengan variabel target *rating* yang dipecah menjadi variabel *rating_good* dan *rating_bad*.



Gambar 20. *Pair Plots*

Gambar 20 menunjukkan *pair plots* berupa keseluruhan grafik, dimana terdapat distribusi setiap variabel dan hubungan antarvariabel. Berdasarkan *pair plots* di atas, dapat terlihat bahwa setiap penetapan *listing price*, *sale price*, *discount*, serta pemberian *review* pelanggan terhadap suatu produk tertentu dapat membawa *rating* yang beragam, baik *rating* "*Good*" maupun "*Bad*".

Data Splitting

```
x = nikeAdidas.drop("Rating",axis = 1)
y = nikeAdidas["Rating"]
```

Gambar 20. Penetapan Variabel Independen dan Variabel Dependen

Adapun tahap *data splitting* yang melibatkan pendefinisian variabel x sebagai variabel independen yang berisi semua variabel yang memengaruhi variabel *Rating* sebagai variabel y atau variabel dependen sebagaimana disajikan dalam Gambar 20.

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.30, random_state = 42)
```

Gambar 21. *Data Splitting*

Gambar 21 menggambarkan proses *data splitting* dengan proporsi 70:30, dimana 70% data akan diambil menjadi data *training* dan 30% data akan dijadikan sebagai data *testing*. Hal ini dapat terlihat dari *test_size* = 0.3 yang menandakan bahwa komposisi yang digunakan untuk kebutuhan *testing* sebanyak 30% dari keseluruhan data.

```
print("Training Data :", len(x_train))
print("Testing Data :", len(x_test))

Training Data : 2287
Testing Data : 981
```

Gambar 22. Jumlah *Training Data* dan *Testing Data*

Gambar 22 menggambarkan hasil komposisi data atas *data splitting*, dimana diperoleh data *training* sebanyak 2.287 observasi dan data *testing* sebanyak 981 observasi.

Data Standardization

```
scaler = StandardScaler()

scaled_x_train = scaler.fit_transform(x_train)
scaled_x_test = scaler.transform(x_test)
```

Gambar 23. *Data Standardization*

Gambar 23 menggambarkan proses standardisasi data (*data standardization*) dimana nilai skala pada variabel x dalam data *training* dan data *testing* akan disamakan dengan fungsi *StandardScaler()*.

D. Modeling

1. Decision Tree Model

```
# Decision Tree Model Fit
dt = DecisionTreeClassifier(random_state = 42)
dt.fit(scaled_x_train, y_train)
```

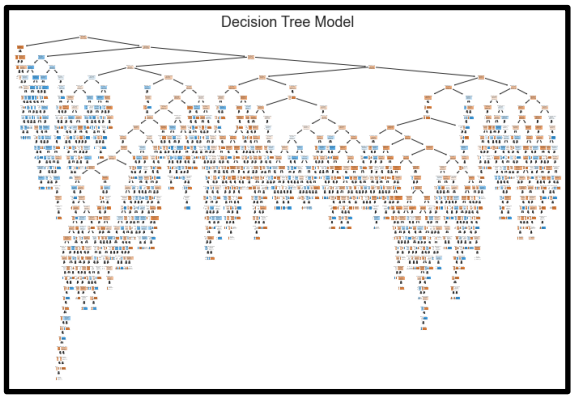
Gambar 24. *Decision Tree Model Fit*

Gambar 24 menggambarkan proses pembuatan model *Decision Tree* atas data *training* yang telah distandarisasi menggunakan fungsi *DecisionTreeClassifier()*.

```
# Decision Tree Model Plot
plt.figure(figsize = (12, 8))
plot_tree(dt, filled = True, feature_names = x_train.columns)
plt.title("Decision Tree Model", fontsize = 18)
plt.show()
```

Gambar 25. *Decision Tree Model Plotting*

Gambar 25 menyajikan pembuatan *plot* untuk model *Decision Tree* untuk melihat bentuk *node* pada *plot* tersebut.



Gambar 26. *Decision Tree Model Plot*

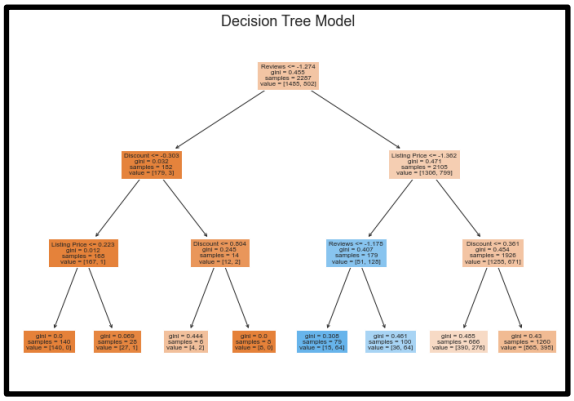
Gambar 26 menyajikan *tree plot* yang dibuat untuk model *Decision Tree*. Berdasarkan *tree plot* di atas, terdapat data *training* yang digunakan dalam jumlah yang banyak sehingga *node* yang dihasilkan juga menjadi banyak. Oleh sebab itu, visual yang akan ditampilkan pada *tree plot* menjadi kurang jelas.

```
# Prunned Decision Tree Model Plot
simpleDt = DecisionTreeClassifier(max_depth = 3)
simpleDt.fit(scaled_x_train, y_train)

plt.figure(figsize = (12, 8))
plot_tree(simpleDt, filled = True, feature_names = x_train.columns, fontsize = 8)
plt.title("Decision Tree Model", fontsize = 18)
plt.show()
```

Gambar 27. *Decision Tree Model Plot Pruning*

Gambar 27 menunjukkan teknik *pruning* yang diterapkan guna menyederhanakan pembuatan *tree plot* untuk model *Decision Tree* sehingga pada akhirnya dapat memberikan tampilan *node* yang lebih jelas.



Gambar 28. *Decision Tree Model Plot after Pruning*

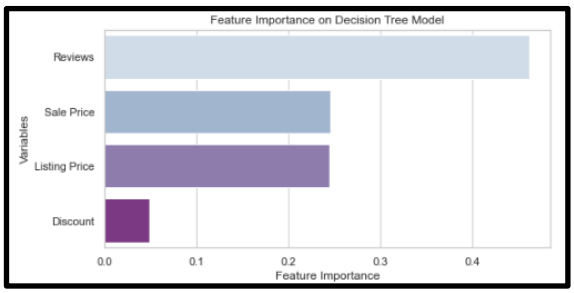
Setelah dilakukan teknik *pruning* pada *plot* untuk model *Decision Tree*, maka akan menampilkan *tree plot* yang lebih sederhana dengan tampilan *node* yang jelas sebagaimana diilustrasikan pada Gambar 28.

```
# Feature Importance on Decision Tree Model
dt_feature = pd.DataFrame({"Variables": x_train.columns, "Feature Importance": dt.feature_importances_})
dt_feature = dt_feature.sort_values(['Feature Importance'], ascending = False)

plt.figure(figsize = (8, 4))
sns.barplot(data = dt_feature, x = "Feature Importance", y = "Variables", palette = "BuPu")
plt.title("Feature Importance on Decision Tree Model")
plt.show()
```

Gambar 29. *Feature Importance on Decision Tree Model*

Gambar 29 menunjukkan pembuatan *feature importance plot* untuk model *Decision Tree* yang menunjukkan seberapa penting variabel prediktor terhadap prediksi model tersebut.



Gambar 30. *Feature Importance Plot on Decision Tree Model*

Gambar 30 menunjukkan *feature importance plot* untuk model *Decision Tree*, dimana variabel *Reviews* memiliki tingkat *importance* tertinggi, sedangkan variabel *Discount* memiliki tingkat *importance* terhadap variabel *Rating*.

```
# Decision Tree Model Prediction
dt_predict = dt.predict(scaled_x_test)

print(classification_report(y_test, dt_predict))
#print("Accuracy:", metrics.accuracy_score(y_test, dt_predict))
```

```
              precision    recall  f1-score   support

         Bad       0.69      0.69      0.69       634
        Good       0.44      0.44      0.44       347

    accuracy                           0.60       981
   macro avg       0.56      0.56      0.56       981
weighted avg       0.60      0.60      0.60       981
```

Gambar 31. *Classification Report in Decision Tree Model*

Gambar 31 menunjukkan kegiatan prediksi dengan menggunakan data *testing* pada model *Decision Tree* yang menghasilkan suatu laporan klasifkasi (*classification report*). Berdasarkan *classification report* di atas, diperoleh nilai akurasi sebesar 0.60 atau 60% pada data *testing*.

	Actual	Predicted
0	Bad	Bad
1	Good	Good
2	Good	Good
3	Bad	Good
4	Bad	Good
...	...	...
976	Bad	Bad
977	Good	Bad
978	Bad	Bad
979	Bad	Good
980	Bad	Bad

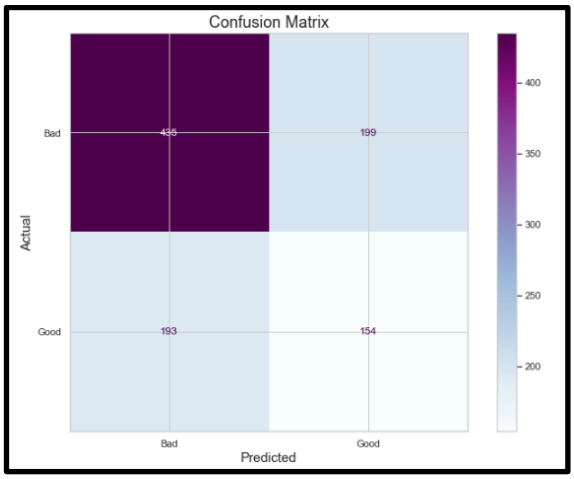
Gambar 32. *Actual vs Predicted in Decision Tree Model*

Gambar 32 menampilkan *dataframe* yang berisi variabel *actual* dan *predicted* dengan nilai tersendiri, dimana nilai-nilai tersebut memberikan dampak terhadap nilai akurasi pada model *Decision Tree*.

```
# Confusion Matrix in Decision Tree Model
plot_confusion_matrix(dt, scaled_x_test, y_test, cmap="BuPu")
plt.title('Confusion Matrix', fontsize = 18)
plt.xlabel('Predicted', fontsize = 15)
plt.ylabel('Actual', fontsize = 15)
plt.show()
```

Gambar 33. *Confusion Matrix Plotting in Decision Tree Model*

Gambar 33 menampilkan pembuatan *plot* untuk *confusion matrix* pada model *Decision Tree* yang mampu mengukur performa atau kinerja untuk masalah klasifikasi.



Gambar 34. *Confusion Matrix Plot on Decision Tree Model*

Gambar 34 menyajikan hasil *confusion matrix plot* yang telah dibuat untuk model *Decision Tree*. Berdasarkan *plot* di atas, maka diperoleh hasil sebanyak 154 *True Positive* (*TP*), 435 *True Negative* (*TN*), 199 *False Positive* (*FP*), dan 193 *False Negative* (*FN*).

2. Random Forest

```
# Finding the Optimal Number of Trees
rf_error = []

for n in range(1 , 41):
    rf = RandomForestClassifier(n_estimators = n, random_state = 42)
    rf.fit(scaled_x_train, y_train)
    rf_predict = rf.predict(scaled_x_test)
    rf_error.append(1 - accuracy_score(rf_predict, y_test))
```

Gambar 35. *Finding the Optimal Number of Trees*

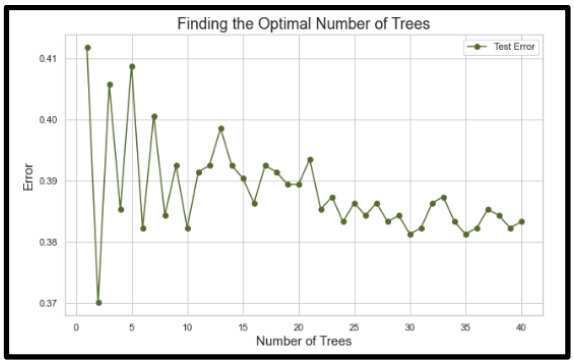
Gambar 35 menampilkan pencarian jumlah pohon yang paling optimal dengan nilai *error* terkecil untuk model *Random Forest*. Pencarian ini melibatkan penggunaan fungsi dengan *for looping* untuk membandingkan nilai *error* dari jumlah *tree* 1 hingga 40. Pencarian ini dilakukan karena *Random Forest* merupakan hutan dari *Decision Tree* sehingga jumlah *tree* lebih banyak daripada *Decision Tree*.

```
# Finding the Optimal Number of Trees Plot
plt.figure(figsize=(10,6))

plt.plot(range(1, 41), rf_error, label = 'Test Error', color = "darkolivegreen", marker = "o")
plt.title("Finding the Optimal Number of Trees", fontsize = 18)
plt.xlabel("Number of Trees", fontsize = 15)
plt.ylabel("Error", fontsize = 15)
plt.legend()
plt.show()
```

Gambar 36. *Finding the Optimal Number of Trees Plot*

Gambar 36 menampilkan pembuatan *plot* atas pencarian jumlah pohon yang paling optimal.



Gambar 37. *Finding the Optimal Number of Trees Plot*

Gambar 37 menunjukkan *plot* yang mampu menunjukkan jumlah *tree* yang optimal. Berdasarkan *plot* di atas, dapat terlihat bahwa jumlah *tree* 2 memiliki nilai *error* terkecil sehingga jumlah *tree* tersebut akan digunakan untuk model *Random Forest*. Oleh karena *Random Forest* menampilkan *tree* dalam jumlah yang banyak, maka hal ini menjadi kurang efektif dalam menampilkan visualnya dikarenakan akan memakan waktu yang lama.

```
# Random Forest Model Fit
rf = RandomForestClassifier(n_estimators = 2,random_state = 42)
rf.fit(scaled_x_train, y_train)
```

Gambar 38. *Random Forest Model Fit*

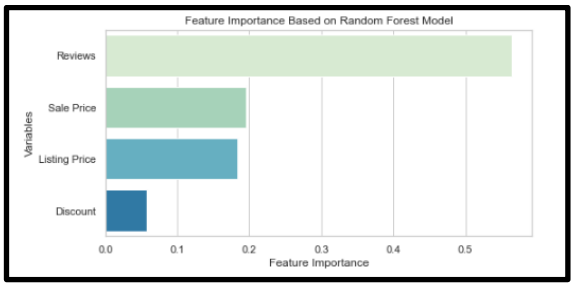
Gambar 38 menggambarkan proses pembuatan model *Random Forest* atas data *training* yang telah distandarisasi menggunakan fungsi *RandomForestClassifier()* dengan jumlah pohon yang optimal, yaitu 2.

```
# Feature Importance on Random Forest Model
rf_feature = pd.DataFrame({"Variables": x_train.columns, "Feature Importance": rf.feature_importances_})
rf_feature = rf_feature.sort_values(['Feature Importance'],ascending = False)

plt.figure(figsize = (8,4))
sns.barplot(data = rf_feature, x = "Feature Importance", y ="Variables", palette = "GnBu")
plt.title("Feature Importance Based on Random Forest Model")
plt.show()
```

Gambar 39. *Feature Importance on Random Forest Model*

Gambar 39 menunjukkan pembuatan *feature importance plot* untuk model *Random Forest* yang menunjukkan seberapa penting variabel prediktor terhadap prediksi model tersebut.



Gambar 40. *Feature Importance Plot on Decision Tree Model*

Gambar 40 menunjukkan *feature importance plot* untuk model *Random Forest*, dimana variabel *Reviews* memiliki tingkat *importance* tertinggi, sedangkan variabel *Discount* memiliki tingkat *importance* terhadap variabel *Rating*.

```
# Random Forest Model Prediction
rf_predict = rf.predict(scaled_x_test)

print(classification_report(y_test, rf_predict))
#print("Accuracy:", metrics.accuracy_score(y_test, rf_predict))
```

```
              precision    recall  f1-score   support

         Bad       0.68      0.81      0.74       634
        Good       0.47      0.31      0.37       347

    accuracy                           0.63       981
   macro avg       0.57      0.56      0.55       981
weighted avg       0.60      0.63      0.61       981
```

Gambar 41. *Classification Report in Random Forest Model*

Gambar 41 menunjukkan kegiatan prediksi dengan menggunakan data *testing* pada model *Random Forest* yang menghasilkan suatu laporan klasifkasi (*classification report*). Berdasarkan *classification report* di atas, diperoleh nilai akurasi sebesar 0.63 atau 63% pada data *testing*.

	Actual	Predicted
0	Bad	Bad
1	Good	Bad
2	Good	Bad
3	Bad	Good
4	Bad	Bad
...	...	...
976	Bad	Bad
977	Good	Bad
978	Bad	Bad
979	Bad	Bad
980	Bad	Bad

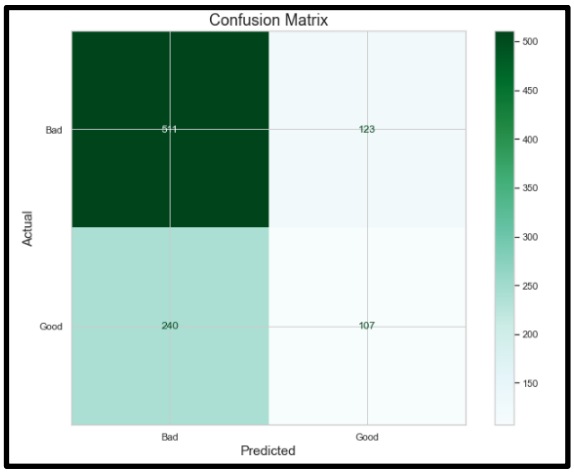
Gambar 42. *Actual vs Predicted in Random Forest Model*

Gambar 32 menampilkan *dataframe* yang berisi variabel *actual* dan *predicted* dengan nilai tersendiri, dimana nilai-nilai tersebut memberikan dampak terhadap nilai akurasi pada model *Random Forest*.

```
# Confusion Matrix in Random Forest Model
plot_confusion_matrix(rf, scaled_x_test, y_test, cmap = "BuGn")
plt.title('Confusion Matrix', fontsize = 18)
plt.xlabel('Predicted', fontsize = 15)
plt.ylabel('Actual', fontsize = 15)
plt.show()
```

Gambar 43. *Confusion Matrix Plotting in Random Forest Model*

Gambar 43 menampilkan pembuatan *plot* untuk *confusion matrix* pada model *Random Forest* yang mampu mengukur performa atau kinerja untuk masalah klasifikasi.



Gambar 44. *Confusion Matrix Plot on Random Forest Model*

Gambar 44 menyajikan hasil *confusion matrix plot* yang telah dibuat untuk model *Random Forest*. Berdasarkan *plot* di atas, maka diperoleh hasil sebanyak 107 *True Positive* (*TP*), 511 *True Negative* (*TN*), 123 *False Positive* (*FP*), dan 240 *False Negative* (*FN*).

E. Evaluation

```
accuracy = []

for i in [dt_predict, rf_predict]:
    accuracy.append(accuracy_score(y_test,i))

print("============== ACCURACY ==============\n")
print(f"Decision Tree Model Accuracy\t: {round(accuracy[0] * 100)}%")
print(f"Random Forest Model Accuracy\t: {round(accuracy[1] * 100)}%")
```

Gambar 45. *Accuracy Score*

Adapun tahap evaluasi yang melibatkan perbandingan antara model *Decision Tree* dan *Random Forest*. Dalam hal ini, adapun penggabungan nilai akurasi pada model *Decision Tree* dan *Random Forest* dengan fungsi *append()* sebagaimana dijelaskan dalam Gambar 45.

```
============== ACCURACY ==============

Decision Tree Model Accuracy    : 60%
Random Forest Model Accuracy    : 63%
```

Gambar 46. *Accuracy Score Comparison*

Adapun tampilan nilai akurasi pada model *Decision Tree* dan *Random Forest* sebagaimana dijelaskan dalam Gambar 46. Berdasarkan hasil akurasi pada kedua model tersebut, dapat disimpulkan bahwa model *Random Forest* memiliki tingkat akurasi tertinggi sebesar 63%.

F. Deployment

Setelah model terbaik telah diperoleh dalam tahap evaluasi, maka dapat melanjutkan tahap *deployment* berupa implementasi atas model terbaik. Dalam hal ini, adapun penyesuaian akan model yang digunakan sehingga dapat memeroleh suatu hasil yang sesuai dengan target awal dari tahap *CRISP-DM*, yaitu tujuan bisnis yang dijalankan dari Adidas dan Nike berupa meningkatkan kepuasan pelanggan dalam menggunakan produknya.

V. CONCLUSION

A. Conclusion

Kedua metode *data mining* memberikan tingkat akurasi yang berbeda tergantung pada pendekatan yang berbeda dari masing-masing metode. Memilih *random forest* dan algoritma *decision tree* merupakan langkah yang digunakan karena hutan acak memiliki akar yang tidak baik untuk pengambilan keputusan. Oleh karena itu, kami menggunakan metode lain, yaitu *decision tree*, dan menemukan akurasi yang lebih baik. *Random forest* dijalankan dengan membangun banyak *trees* selama pelatihan dan menerapkan setiap mod kelas sebagai hasil dari setiap pohon. *Decision tree* menggunakan *node* untuk menggambarkan model yang terbentuk. Meskipun masing-masing model memiliki kelebihan dan kekurangan, penelitian ini menggunakan dataset yang sama dengan akurasi yang tidak dapat diprediksi. Prediksi yang akurat membutuhkan model dengan hasil yang lebih akurat untuk menghindari kesalahan peramalan. Hasil penelitian ini menunjukkan bahwa algoritma *decision tree* lebih akurat dalam memprediksi kepuasan pelanggan terhadap suatu merek. Faktor penting yang mempengaruhi kepuasan pelanggan terhadap merek berdasarkan algoritma *decision tree* adalah *rating*, harga jual, dan daftar harga. Pasalnya, berdasarkan data yang ada, rating merupakan kolom yang jarang bernilai nol, artinya hampir semua pembeli memberikan *rating* pembelian untuk setiap pembelian, baik daftar maupun harga jual memiliki dampak yang cukup besar pada pembelian. Kedua variabel ini secara langsung mempengaruhi daya beli pembeli.

B. Suggestion

Analisis data dan pemodelan diperlukan untuk memprediksi faktor-faktor yang mempengaruhi kepuasan pelanggan. Dengan menggunakan kumpulan data yang lengkap, dapat dilakukan pengamatan terhadap variabel target untuk menentukan faktor mana yang secara signifikan mempengaruhi kepuasan konsumen merek. Dalam penelitian ini, model *decision tree* merupakan model yang paling cocok untuk menarik kesimpulan tentang faktor-faktor apa saja yang mempengaruhi kepuasan konsumen terhadap pembelian merek Nike dan Adidas.

C. Limitation

Penelitian ini membutuhkan variabel target yang jelas dimana variabel target yang berbeda dapat digunakan pada setiap kasus. Akibatnya, berbagai

jenis variabel target dapat mempengaruhi hasil pemodelan.

REFERENCES (REFERENSI)

[1] H. A. A. Mahdi, M. Abbas and T. I. Mazar, "A Comparative Analysis of Strategies and Business Models of Nike, Inc. and Adidas Group with special reference to Competitive Advantage in the context of a Dynamic and Competitive Environment," *International Journal of Business Management and Economic Research*, vol. 6, no. 3, pp. 167-177, 2015.

[2] Merdeka.com, "Putus hubungan dengan Adidas diklaim tak bakal rugikan Milan," Merdeka.coms, 25 October 2017. [Online]. Available: https://www.merdeka.com/sepakbola/bolanet/putus-hubungan-dengan-adidas-diklaim-tak-bakal-rugikan-milan.html. [Accessed 6 December 2022].

[3] V. Matović, M. Milenko Stanić and I. Igor Drinić, "IMPACT BRANDING ON CONSUMER PREFERENCE TOWARDS BUYING A CERTAIN PRODUCT: COMPARATIVE ANALYSIS OF BRANDS NIKE AND ADIDAS," *ЕКОНОМИКА,* vol. 65, no. 3, pp. 35-44, 2019.

[4] H. C. Lim, K. Kim and Y. Cheong, "Factors affecting sportswear buying behavior: A comparative analysis of luxury sportswear," *Journal of Business Research*, vol. 69, no. 12, pp. 5793-5800, December 2016.

[5] S. Jaggia, A. Kelly, K. Lertwachara and L. Chen, "Applying the CRISP-DM Framework for Teaching Business Analytics," *Decision Sciences Journal of Innovative Education*, vol. 18, no. 4, pp. 612-634, 2020.

[6] S. Gavankar and S. Sawarkar , "A Novel EagerDT Complexity Approach to Deal with Missing Values in Decision," *International Journal of Simulation -- Systems, Science & Technology*, vol. 19, no. 6, pp. 1-5, December 2018.

[7] M. Belgiu and L. Drăguţ, "Random Forest in Remote Sensing: A Review of Applications and Future Directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24-31, 2016.

[8] P. Gulati, A. Sharma and M. Gupta, "Theoretical Study of Decision Tree Algorithms to Identify Pivotal Factors for Performance Improvement: A Review," *International Journal of Computer Applications*, vol. 141, no. 14, pp. 19-25, May 2016.

[9] X. Mei, R. Wang, W. Yang, F. Qian, X. Ye, L. Zhu, Q. Chen, B. Han, T. Deyer, J. Zeng, X. Dong, W. Gao and W. Fang, "Predicting malignancy of pulmonary ground-glass nodules and their invasiveness by random forest," *Journal of thoracic disease*, vol. 10, no. 1, pp. 458-463, 2018.

[10] Y. Everingham, J. Sexton, D. Skocaj and G. Inman-Bamber, "Accurate prediction of sugarcane yield using a random forest algorithm," *Agronomy for sustainable development*, vol. 36, no. 27, 2016.

[11] M. Somvanshi, P. Chavan, S. Tambade and S. V. Shinde, "A review of machine learning techniques using decision tree and support vector machine," in *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, Pune, 2016.

[12] M. Ibrahim, "An empirical comparison of random forest-based and other learning-to-rank algorithms," *Pattern Analysis and Applications*, vol. 23, pp. 1133-1155, 2020.

VI. GROUP MEMBERS AND ROLES

No.	Name	NIM	Roles
1	Kelly Mae	00000051428	Project Manager - Code, Report, PPT
2	Fareza Ananda Putra	00000051475	Code, Report. PPT
3	Leony Hana Noah Zebua	00000042544	Code, Report, PPT
4	Reuben Ryan Peter	00000043424	Code, Report, PPT