BAB 3
METODOLOGI PENELTIAN

3.1 Objek dan Subjek Penelitian

Objek penelitian ini adalah popularitas gim indi pada *steam platform* dan fitur yang mempengaruhinya, Sedangkan Subjek pada penelitian ini adalah gim indi pada *steam platform* yang rilis mulai dari tahun 2012 hingga pertengahan tahun 2023.

3.2 Alat dan Bahan Penelitian

3.2.1 Alat Penelitian

Alat yang akan digunakan pada penelitian ini meliputi perangkat keras dan perangkat lunak serta beberapa teori dan standar. Berikut adalah detail dari alat-alat yang akan digunakan :

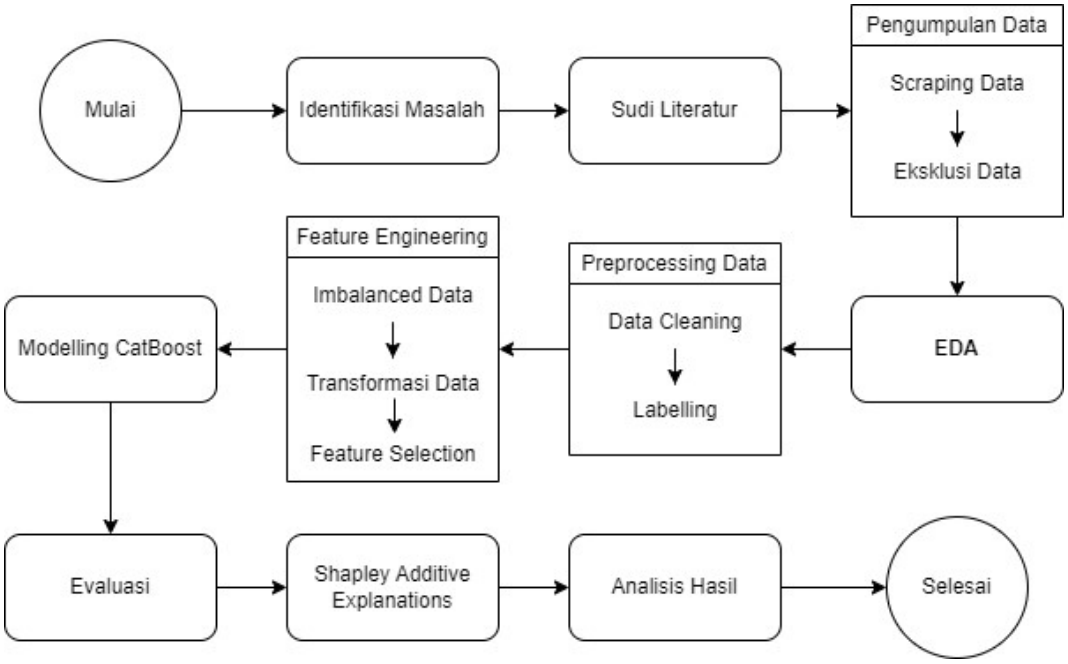
1. Perangkat Keras
   a) Processor Intel® Core i5-11400
   b) Besar memori RAM 16GB
   c) Perangkat keras laptop standar
2. Perangkat Lunak
   a) Sistem operasi *Windows 11*
   b) Dokumen editor *Microsoft Office*
   c) Software pengolahan data *Jupyter Notebook*
   d) Sistem *Mendeley* untuk keperluan sitasi

3.2.2 Bahan Penelitian

Bahan penelitian yang akan digunakan pada penelitian ini adalah hasil scraping data dari *steam platform* yang berisikan daftar dan informasi gim dari tahun 2012 hingga pertengahan 2023. Fitur yang akan digunakan adalah yang bertipe kategorikal dan numerik.

3.3 Diagram Alir Penelitian

Penelitian akan dilakukan mengacu kepada diagram alir, agar tahapan penelitian dilakukan secara urut dan tidak menggangu tahapan penelitian lainnya. Diagram alir ditampilkan pada Gambar 3.1 berikut ;



Gambar 3.1 Diagram Alir Penelitian

3.3.1 Identifikasi Masalah

Identifikasi masalah merupakan tahap awal dari penelitian, dengan melakukan identifikasi masalah peneliti mendapatkan gambaran mengenai tujuan dan manfaat penelitian dari prediksi popularitas gim indi.

3.3.2 Studi Literatur

Penelitian dilanjutkan dengan melakukan studi literatur daripada penelitian sebelumnya yang memiliki keterkaitan objek atau metode yang digunakan. Studi literatur merupakan tahap awal yang penting dilakukan pada saat penelitian untuk mendapatkan informasi dan gambaran dari objek atau metode yang akan digunakan, sebagai pijakan atau rujukan penelitian.

3.3.3 Pengumpulan Data

3.3.3.1 Scraping Data

Data yang digunakan pada penelitian ini adalah data primer berbentuk ekstensi json, hasil dari proses scraping web *steam* menggunakan python. Lalu disimpan dengan nama games.json. Teknik pengumpulan data yang digunakan pada penelitian ini adalah Scraping data yang merupakan teknik pengumpulan data dari website dengan cara mengambil data dari halaman tampilan web menggunakan wget! dan menyimpannya dalam format yang dapat diakses oleh komputer dan diberi nama games.json.

3.3.3.2 Eksklusi Data

Data yang terkumpul akan dilakukan pensortiran untuk mendapatkan data yang hanya berisi gim indi dari tahun 2012. Selanjutnya dijadikan data baru, data inilah yang akan digunakan untuk penelitian, ditunjukan pada Gambar 3.2

Gambar 3.2 Eksklusi gim indi

Sedangkan fitur yang akan digunakan untuk tahap selanjutnya hanya fitur yang berjenis numerikal dan kategorikal saja sehingga hanya menggunakan total 13 fitur, yaitu *name, release_date, required_age, price, windows, mac, linux, achievements, category, genre, supported_languages, full_audio_languages,* dan *Estimated_owners*. Fitur *Estimated_owners* akan digunakan sebagai variabel

target pada penelitian ini karena menggambarkan tingkat popularitas dari suatu gim pada data tersebut.

3.3.4 *Exploratory Data Analysis (EDA)*

Tahap *Exploratory Data Analysis (EDA)* dilakukan untuk memahami data yang akan digunakan , seperti struktur data, identifikasi *missing value,* dan melihat distribusi data untuk mengidentifikasi *outlier*. Sebelum membangun model prediktif atau melakukan analisis lebih lanjut.

EDA dapat membantu dalam memahami struktur data yang digunakan seperti jumlah baris dan fitur, tipe data setiap fitur, serta apakah ada data yang hilang atau data duplikat, contoh pemahaman struktur data ditunjukan pada Gambar 3.3 :

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 47924 entries, 0 to 70209
Data columns (total 17 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   name                  47924 non-null  object
 1   release_date          47924 non-null  object
 2   required_age          47924 non-null  int64
 3   price                 47924 non-null  float64
 4   website               47924 non-null  object
 5   support_url           47924 non-null  object
 6   support_email         47924 non-null  object
 7   windows               47924 non-null  bool
 8   mac                   47924 non-null  bool
 9   linux                 47924 non-null  bool
 10  achievements          47924 non-null  int64
 11  supported_languages   47924 non-null  object
 12  full_audio_languages  47924 non-null  object
 13  developers            47924 non-null  object
 14  categories            47924 non-null  object
 15  genres                47924 non-null  object
 16  estimated_owners      47924 non-null  object
dtypes: bool(3), float64(1), int64(2), object(11)
memory usage: 5.6+ MB
```

Gambar 3.3 Struktur data

Gambar 3.3 menunjukan nama, *missing value,* dan tipe dari fitur-fitur yang ada pada data. Selain untuk mengetahui struktur data, statistik deskriptif data juga dilakukan untuk mengetahui distribusi data. Informasi seperti jumlah data, rata-rata, standar deviasi, *Q1, Q2, Q3,* nilai minimal, dan nilai maksimal pada fitur nnumerik akan muncul saat melakukan statistik deskriptif ditunjukan pada Tabel 3.1 :

Tabel 3.1 Statistik deskriptif

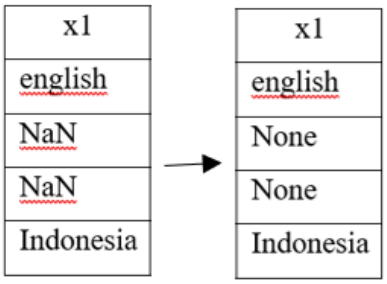
	required_age	price
count	22426.000000	22426.000000
mean	0.210827	6.882547
std	1.861769	7.7072547
min	0.000000	0.000000
25%	0.000000	1.000000
50%	0.000000	5.000000
75%	0.000000	10.000000
max	21.000000	200.000000

Pada Tabel 3.1 dengan menggunakan statistik deskriptif diketahui jumlah data, rata-rata, standar deviasi, *Q1, Q2, Q3,* nilai minimal, dan nilai maksimal dari fitur numerik *required_age,* dan *price*.

3.3.5 *Preprocessing Data*

3.3.5.1 *Data Cleaning*

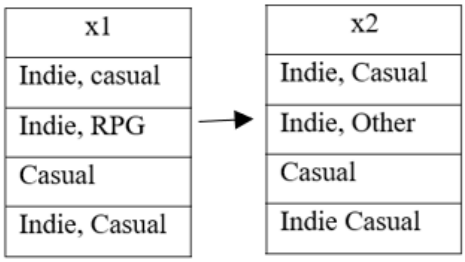
Data yang tidak berkualitas akan menghasilkan model yang tidak maksimal. Keputusan yang berkualitas pasti juga berasal dari data yang berkualitas. Tahapan *data cleaning* yang dilakukan pertama adalah mengatasi *missing value*, dan menggabungkan data minoritas fitur kategorikal pada data.



Gambar 3.4 *Handle Missing Value*

Gambar 3.4 menunjukan perlakuan yang diberikan untuk mengatasi *missing value* yaitu dengan mengubahnya menjadi "*None*". Karena kosongnya nilai pada variabel data *steam* bukan merupakan suatu kesalahan data melainkan gim

tersebut memang tidak memiliki kriteria pada suatu fitur, misal *audio* atau *subtitle.*



Gambar 3.5 Penggabungan minoritas fitur kategorikal

Gambar 3.5 menunjukan perlakuan yang diberikan untuk menggabung minoritas pada fitur kategorikal pada data dengan menghapus yang tidak termasuk 10 terbanyak dari tiap fitur. Setelah itu dilakukan penghapusan pada data duplikat agar setiap data gim memiliki informasi unik disetiap fiturnya

3.3.5.2 *Labeling*

Tahap preprocessing yang dilakukan selanjutnya adalah pemberian label. Menentukan kelas berdasarkan ketentuan tersendiri. Fitur yang menjadi label pada penelitian ini adalah fitur *estimated_owner* dan diberi perlakuan transformasi data nilai 0 untuk level dibawah 20000, dan 1 untuk level diatas 20000 berdasarkan *value* dari fitur *estimated_owners*, ditunjukan pada Tabel 3.3 :

Tabel 3.2 labelling data

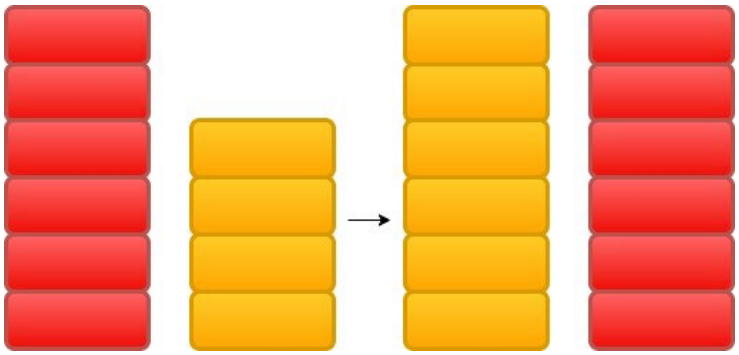
estimated_owners	label
0 - 0	0
0 - 20000	0
20000 - 50000	1
50000 - 100000	1
100000 - 200000	1
200000 - 500000	1
500000 - 1000000	1
1000000 - 2000000	1
2000000 - 5000000	1
5000000 - 10000000	1
10000000 - 20000000	1
20000000 - 50000000	1

Nilai pada fitur *Estimated_owners* tidak ditentukan oleh peneliti melainkan nilai asli dari hasil *scraping*. Ambang batas popularitas suatu gim yang peneliti terapkan didasari pada hasil *Exploratory Data Analysis* untuk menghindari ketidakseimbangan terhadap kelas yang berlebih pada data.

3.3.6 *Feature Engineering*

3.3.6.1 *Imbalanced Data*

Menangani ketidakseimbangan distribusi kelas pada dataset. Ketidakseimbangan ini dapat mempengaruhi kinerja model yang akan di bangun, oleh karena itu *oversampling* diterapkan untuk mencapai keseimbangan antar kelas. dilakukan menggunakan fungsi *random oversampling*, ditunjukan pada Gambar 3.6:



Gambar 3.6 *Random Oversampling*

3.3.6.2 Transformasi Data

Transformasi Data dilakukan untuk mengubah atau memodifikasi fitur-fitur yang ada dalam *dataset*. Proses ini *one-hot encoding* pada fitur kategorikal yang bertujuan untuk memastikan data siap untuk pemodelan, ditunjukan pada Gambar 3.7 :

Original Data

Team	Points
A	25
A	12
B	15
B	14
B	19
B	23
C	25
C	29

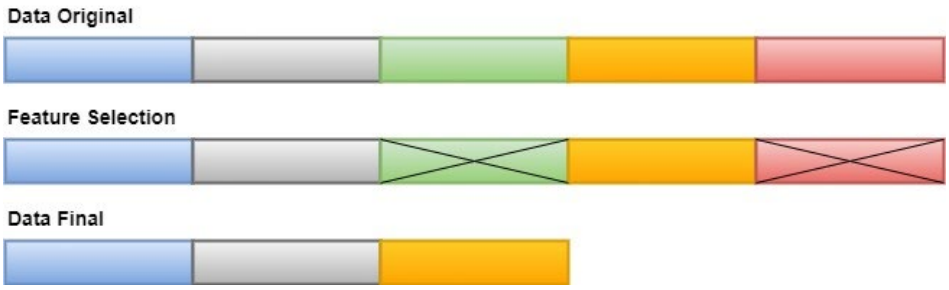
One-Hot Encoded Data

Team_A	Team_B	Team_C	Points
1	0	0	25
1	0	0	12
0	1	0	15
0	1	0	14
0	1	0	19
0	1	0	23
0	0	1	25
0	0	1	29

Gambar 3.7 Contoh penggunaan *one-hot encoding* [35]

3.3.6.3 *Feature Selection*

Feature Selection merupakan salah satu tahap yang penting dilakukan dalam mengidentifikasi fitur-fitur yang akan digunakan model terhadap target yang ingin diprediksi. Dengan menghapus fitur yang tidak digunakan pada dataset. Peneliti dapat memastikan bahwa dataset yang digunakan telah diolah dengan baik dan menjadi landasan yang kuat untuk masuk ke proses pemodelan, ditunjukan pada Gambar 3.8 :



Gambar 3.8 *Feature Selection*

3.3.7 *Modelling CatBoost*

Dalam tahap ini, peneliti akan menggunakan algoritma *CatBoost* untuk membuat model prediksi popularitas gim indi menggunakan data yang sudah diolah langkah sebelumnya. Algoritma *CatBoost* dipilih karena memiliki kemampuan yang lebih baik dalam memprediksi khususnya dengan data yang memiliki fitur kategorikal daripada algoritma lain yang digunakan pada penelitian sebelumnya. Algoritma *CatBoost* mengimplementasikan *Gradient Boosting*

Decision Tree (GBDT) dalam memprediksi. *Gradien Boosting* sendiri merupakan teknik *ensemble* yang berarti model akhir dibentuk dengan menggabungkan beberapa model individual dengan tujuan untuk mengoptimalkan *loss function*.

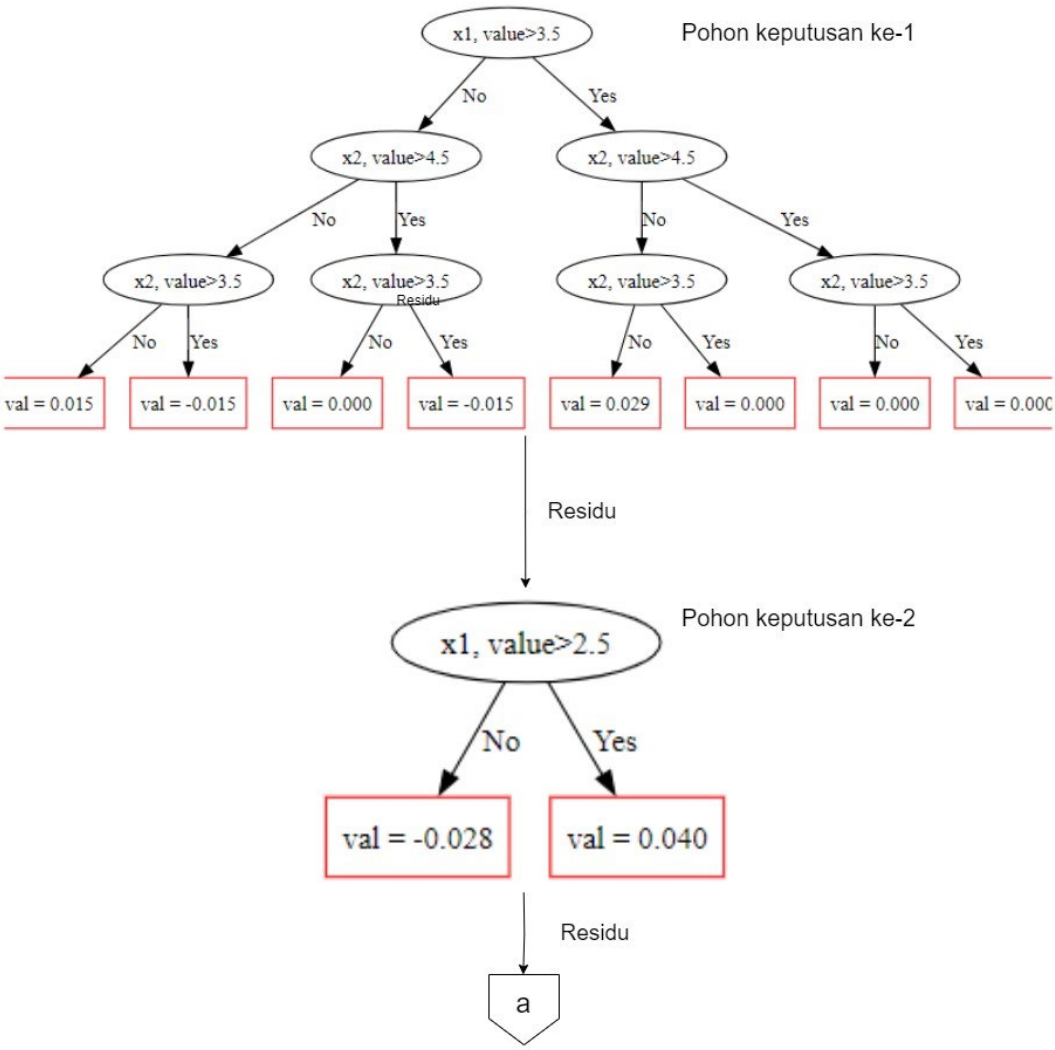
Pada Tabel 3.3 menunjukan sebuah data dengan fitur x1 dan x2 sebagai variabel independen (x) dan Target sebagai variabel dependen.

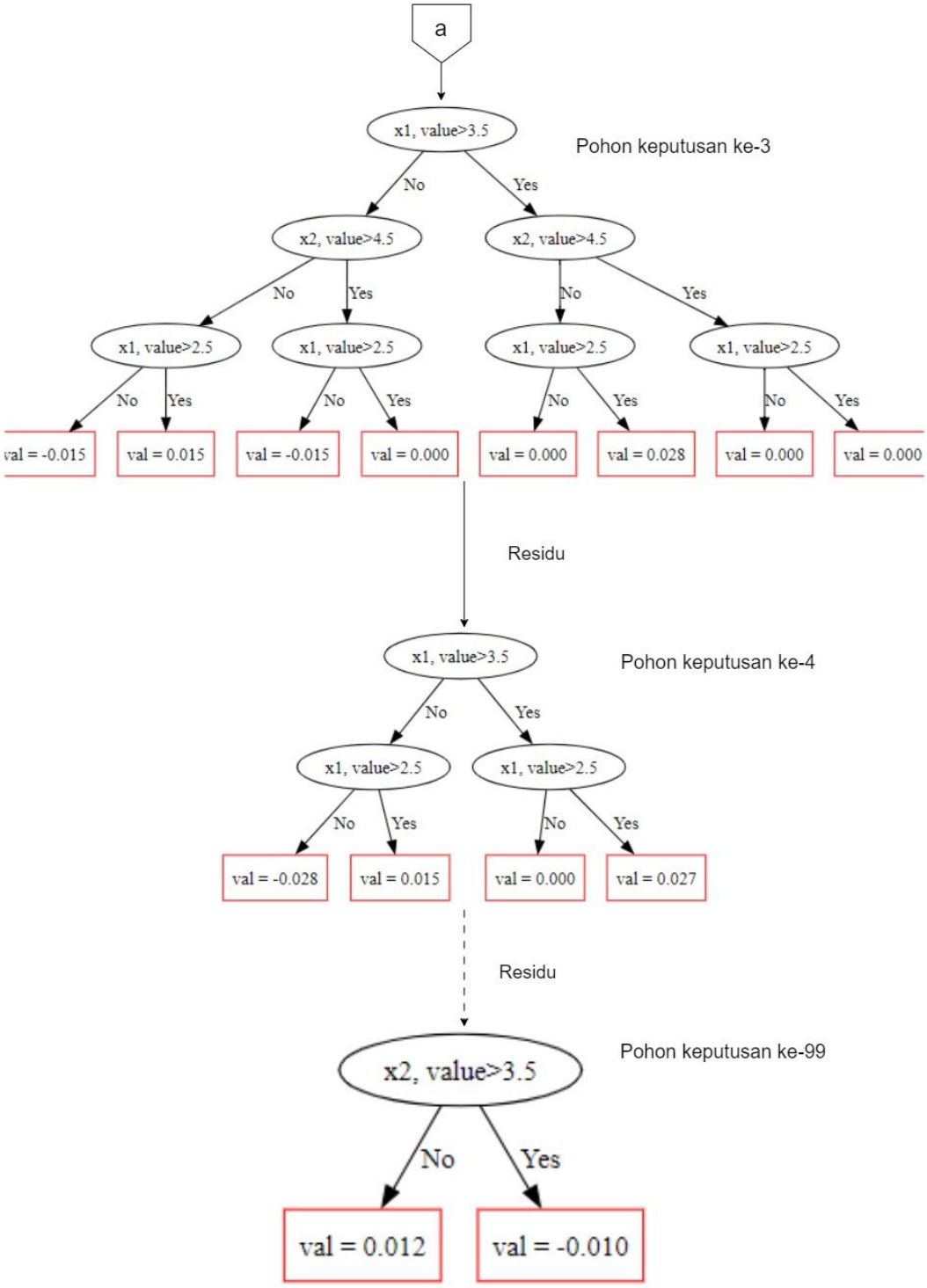
Tabel 3.3 Contoh data

x1	x2	Target
1	5	0
2	4	0
3	3	1
4	2	1
5	1	1

Berdasarkan data Tabel 3.3 langkah-langkah cara kerja *CatBoost* menggunakan *Gradient Boosting :*

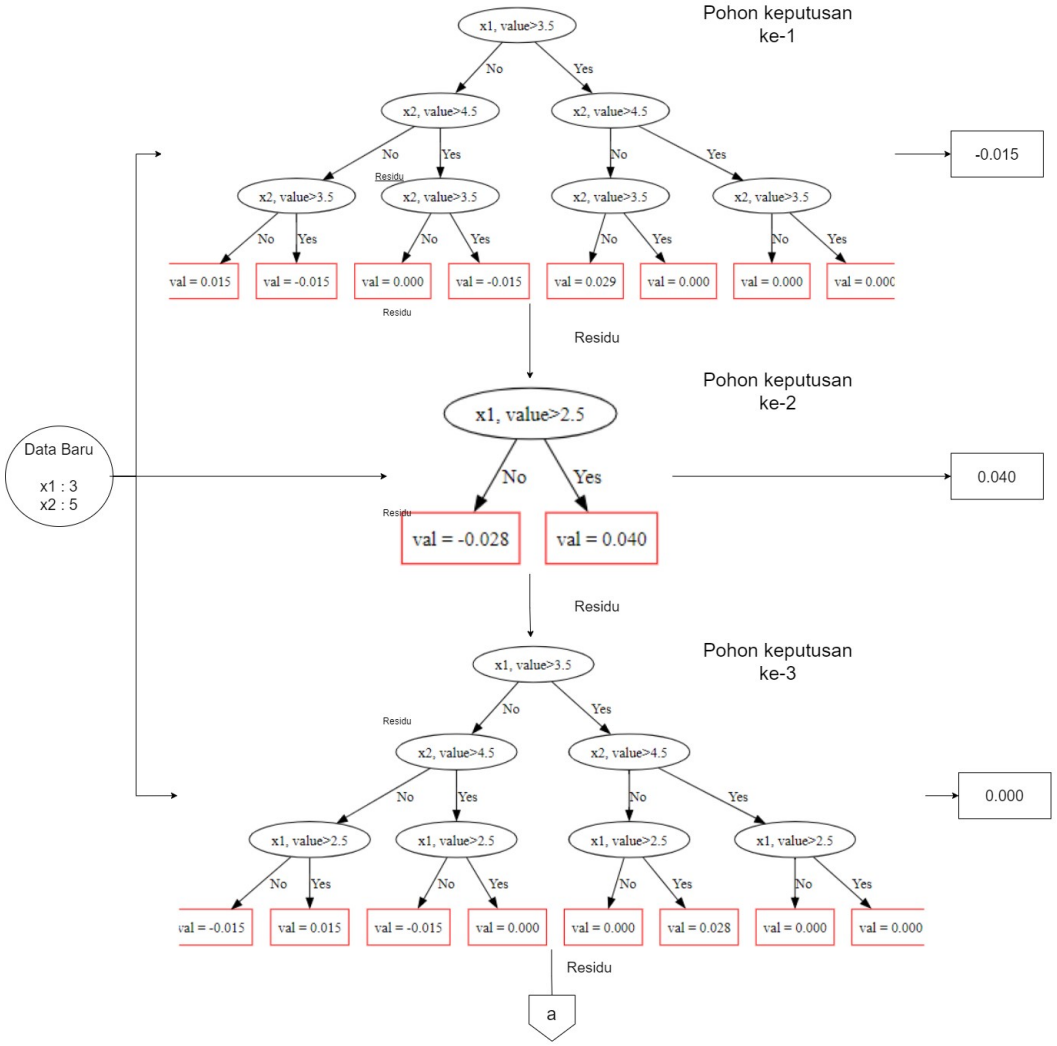
1. Membuat pohon keputusan berdasarkan data Tabel 3.3, yang menghasilkan prediksi baru menggunakan persamaan 2.1
2. Membuat pohon keputusan baru yang memprediksi residu antara nilai prediksi dan nilai aktual dari pohon keputusan sebelumnya
3. Melakukan iterasi langkah 1 dan 2 hingga iterasi ke-n yang ditentukan

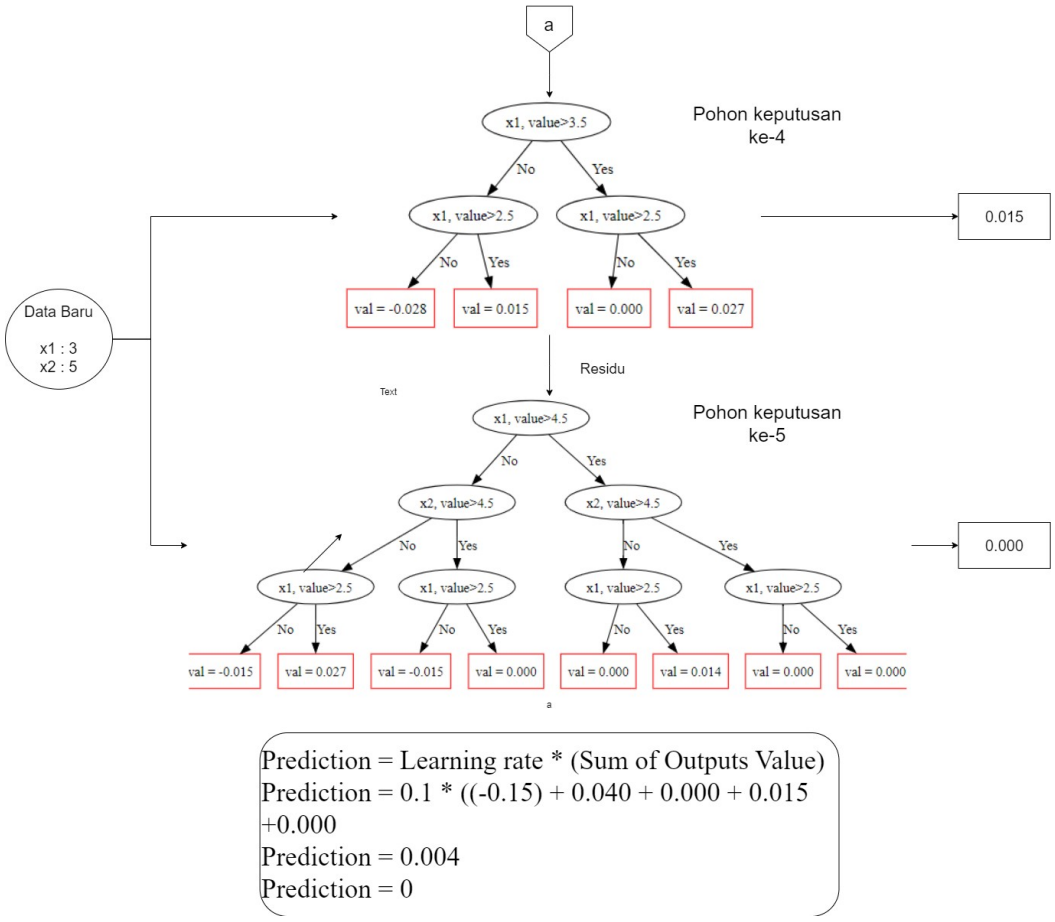

x1, value>3.5
Pohon keputusan ke-1
No
Yes
x2, value>4.5
x2, value>4.5
No
Yes
No
Yes
x2, value>3.5
x2, value>3.5
x2, value>3.5
x2, value>3.5
Residu
No
Yes
No
Yes
No
Yes
No
Yes
val = 0.015
val = -0.015
val = 0.000
val = -0.015
val = 0.029
val = 0.000
val = 0.000
val = 0.000
Residu
Pohon keputusan ke-2
x1, value>2.5
No
Yes
val = -0.028
val = 0.040
Residu
a



Gambar 3.9 Cara kerja Algoritma *CatBoost*

Gambar 3.9 menjelaskan cara kerja algoritma *CatBoost* pada data Tabel 3.3. Algoritma *CatBoost* sendiri mengimplementasikan algoritma *Gradient Boosting* yang menggabungkan banyak pohon keputusan simetris dalam membuat prediksi. Setiap pohon keputusan yang dibuat akan memprediksi residu antara hasil prediksi model sebelumnya dengan data actual. Iterasi akan terus dilakukan bergantung dari parameter iterasi yang ditentukan.





Gambar 3.10 Contoh penggunaan *CatBoost* pada data baru

Pada Gambar 3.10 dengan menginputkan data baru untuk dilakukan prediksi. Menggunakan persamaan 2.1 algoritma *CatBoost* memprediksi bahwa data tersebut masuk ke kelas 0 setelah melewati 5 iterasi pohon keputusan.

Selain performanya yang baik dalam melakukan prediksi dikarenakan mengimplementasikan *Gradient Boosting*. Kemampuannya dalam mengatasi data heterogen khususnya data kategorikal juga menjadi alasan utama kenapa penelitian ini menggunakan algoritma *CatBoost*. Berbeda dengan algoritma lain yang memerlukan Teknik *one-hot encoding* atau *label encoder* terlebih dahulu sebelum melakukan pemodelan dan penggunaan metode target boosting yang menyebabkan terjadinya *data leakage.* Algoritma ini memiliki cara sendiri dalam mengubah data kategorikal menjadi numerik. Cara kerja *CatBoost* dalam mengubah data kategorikal menjadi numerik pada data ditunjukan sebagai berikut.

Tabel 3.4 Contoh data kategorikal

x1	Target
kategori_a	0
kategori_b	1
kategori_a	0
kategori_a	1
kategori_b	1
kategori_c	1

Pada Tabel 3.4 menunjukan sebuah data dengan fitur x1 sebagai variabel independent (x) dan Target sebagai variabel dependent. *CatBoost* akan melakukan transformasi terhadap fitur x1 yang bertipe kategorikal menjadi numerik dengan menggunakan persamaan 2.3, dan ditunjukan pada perhitungan dibawah :

$$ctr_1 = \frac{0 + 0.5}{0 + 1} = 0.5$$

$$ctr_2 = \frac{0 + 0.5}{0 + 1} = 0.5$$

$$ctr_3 = \frac{1 + 0.5}{1 + 1} = 0.75$$

$$ctr_4 = \frac{0 + 0.5}{2 + 1} = 0.25$$

$$ctr_5 = \frac{1 + 0.5}{1 + 1} = 0.75$$

$$ctr_6 = \frac{0 + 0.5}{0 + 1} = 0.5$$

maka akan didapatkan fitur x1 yang sudah diubah dari kategorikal menjadi numerik ditunjukan pada tabel 3.5 :

Tabel 3.5 Hasil perubahan fitur kategorikal

x1
0.5
0.5
0.75
0.25
0.75
0.5

Metode transformasi ini dianggap mampu mengatasi dimensi variabel yang berlebihan yang diakibatkan *one-hot encoding* dan mampu mengantisipasi terjadinya *data leakage.* Metode transformasi ini membuat algoritma *CatBoost* pintar dalam memahami *impact* dari fitur kategorikal terhadap target.

Model *CatBoost* juga memiliki beberapa hyperparameter tuning yang dapat digunakan untuk meningkatkan performa model, diantaranya adalah *objective* untuk menentukan tujuan pemodelan untuk klasifikasi atau regresi, *learning_rate* untuk menentukan tingkat pembelajaran yang digunakan model setiap iterasi, *depth* untuk mengatur kedalaman pohon, *l2_leaf_reg* untuk menghindari terjadinya *overfit, iterations* untuk mengatur seberapa banyak iterasi dilakukan. *task_type* untuk menentukan perangkat keras yang digunakan untuk melatihmodel dilatih.

3.3.8 Evaluasi

Peneliti akan mengevaluasi kinerja model menggunakan metrik konfusi, *classification report* dan Kurva *AUC* untuk mengukur sejauh mana model mampu melakukan prediksi popularitas gim indi dengan tepat dan efisien.

3.3.9 *Shapley Additive Explanations (SHAP)*

Peneliti akan menggunakan metode *Shapley Additive Explanations* (*SHAP*) untuk memberikan wawasan yang berharga tentang bagaimana masing-masing fitur berkontribusi terhadap hasil prediksi menggunakan *shapley value* dari tiap instansi fitur, sehingga memungkinkan pemahaman yang lebih baik tentang faktor-faktor yang mempengaruhi prediksi popularitas gim indi. Langkah-langkah *SHAP* dalam menentukan nilai *shapley* pada fitur :

1. Mencari probabilitas satu instansi fitur masuk setelah instansi fitur lainnya.
2. Mencari *marginal* kontribusi dari satu instansi fitur terhadap instansi fitur lainnya

3. Menjumlahkan probabilitas dikalikan dengan *marginal* kontribusi instansi fitur tersebut pada hasil prediksi

Tabel 3.6 Contoh hasil prediksi model

x1	x2	x3	$\hat{f}$
1	1	1	1
0	0	0	0
1	1	0	0.75
1	0	1	0.75
0	1	1	0.5
1	0	0	0.5
0	1	0	0.5
0	0	1	0

Pada Tabel 3.6 menunjukan sebuah data dengan fitur x1, x2, dan x3 sebagai variabel independent (x) dan $\hat{f}$ sebagai hasil prediksi model. Untuk mengetahui nilai *shapley* pada instansi suatu fitur, terhadap instansi suatu fitur lainnya, maka langkah-langkah *SHAP* dalam menentukan nilai *shapley* adalah :

1. Mencari *marginal* kontribusi dari setiap instansi pada fitur terhadap hasil prediksi menggunakan persamaan 2.4 :
   a) Instansi 0 (Nol) pada fitur x1

$$C_{123} - C_{23} = 0 - 0.5 = -0.5$$
$$C_{12} - C_{2} = 0 - 0.75 = -0.75$$
$$C_{13} - C_{3} = 0.5 - 0.75 = -0.25$$
$$C_{1} - C_{0} = 0.5 - 1 = -0.5$$

   b) Instansi 1 (Satu) pada fitur x1

$$C_{123} - C_{23} = 1 - 0.5 = 0.5$$
$$C_{12} - C_{2} = 0.75 - 0.5 = 0.25$$
$$C_{13} - C_{3} = 0.75 - 0 = 0.75$$
$$C_{1} - C_{0} = 0.5 - 0 = 0.5$$

   c) Instansi 0 (Nol) pada fitur x2

$$C_{123} - C_{13} = 0 - 0.5 = -0.5$$

$$C_{12} - C_1 = 0 - 0.5 = -0.5$$
$$C_{23} - C_3 = 0.5 - 0.75 = -0.25$$
$$C_2 - C_0 = 0.75 - 1 = -0.25$$

d) Instansi 1 (Satu) pada fitur x2

$$C_{123} - C_{13} = 1 - 0.75 = 0.25$$
$$C_{12} - C_1 = 0.75 - 0.5 = 0.25$$
$$C_{23} - C_3 = 0.5 - 0 = 0.5$$
$$C_2 - C_0 = 0.5 - 0 = 0.5$$

e) Instansi 0 (Nol) pada fitur x3

$$C_{123} - C_{12} = 1 - 0.75 = -0.25$$
$$C_{23} - C_2 = 0.5 - 0.75 = -0.25$$
$$C_{13} - C_1 = 0.5 - 0.5 = 0$$
$$C_3 - C_0 = 0.75 - 1 = -0.25$$

f) Instansi 1 (Satu) pada fitur x3

$$C_{123} - C_{12} = 1 - 0.75 = 0.25$$
$$C_{23} - C_2 = 0.5 - 0.5 = 0$$
$$C_{13} - C_1 = 0.75 - 0.5 = 0.25$$
$$C_3 - C_0 = 0 - 0 = 0$$

2. Mencari probabilitas kombinasi fitur (*weight)* yang mungkin terbentuk pada *marginal* kontribusi tertentu menggunakan persamaan 2.4 :

$$C_{abc} - C_{bc} = \frac{2!\,1!}{3!} = \frac{1}{3}$$
$$C_{ab} - C_b = \frac{1!\,1!}{3!} = \frac{1}{6}$$
$$C_{ac} - C_c = \frac{1!\,1!}{3!} = \frac{1}{6}$$
$$C_a - C_0 = \frac{1}{3}$$

3. Dilakukan penjumlahan dari *weight* dan *marginal* kontribusi yang ditemukan dengan persamaan 2.4 :

   a) Instansi 0 (Nol) pada fitur x1

$$\phi_i = \frac{1}{3} \times (-0.5) + \frac{1}{6} \times (-0.75) + \frac{1}{6} \times (-0.5) + \frac{1}{3} \times (-0.25)$$

$$\phi_i = -0.45$$

   b) Instansi 1 (Satu) pada fitur x1

$$\phi_i = \frac{1}{3} \times 0.5 + \frac{1}{6} \times 0.25 + \frac{1}{6} \times 0.75 + \frac{1}{3} \times 0.5$$

$$\phi_i = 0.5$$

   c) Instansi 0 (Nol) pada fitur x2

$$\phi_i = \frac{1}{3} \times (-0.5) + \frac{1}{6} \times (-0.5) + \frac{1}{6} \times (-0.25) + \frac{1}{3} \times (-0.25)$$

$$\phi_i = -0.375$$

   d) Instansi 1 (Satu) pada fitur x2

$$\phi_i = \frac{1}{3} \times 0.25 + \frac{1}{6} \times 0.25 + \frac{1}{6} \times 0.5 + \frac{1}{3} \times 0.5$$

$$\phi_i = 0.375$$

   e) Instansi 0 (Nol) pada fitur x3

$$\phi_i = \frac{1}{3} \times (-0.25) + \frac{1}{6} \times (-0.25) + \frac{1}{6} \times 0 + \frac{1}{3} \times (-0.25)$$

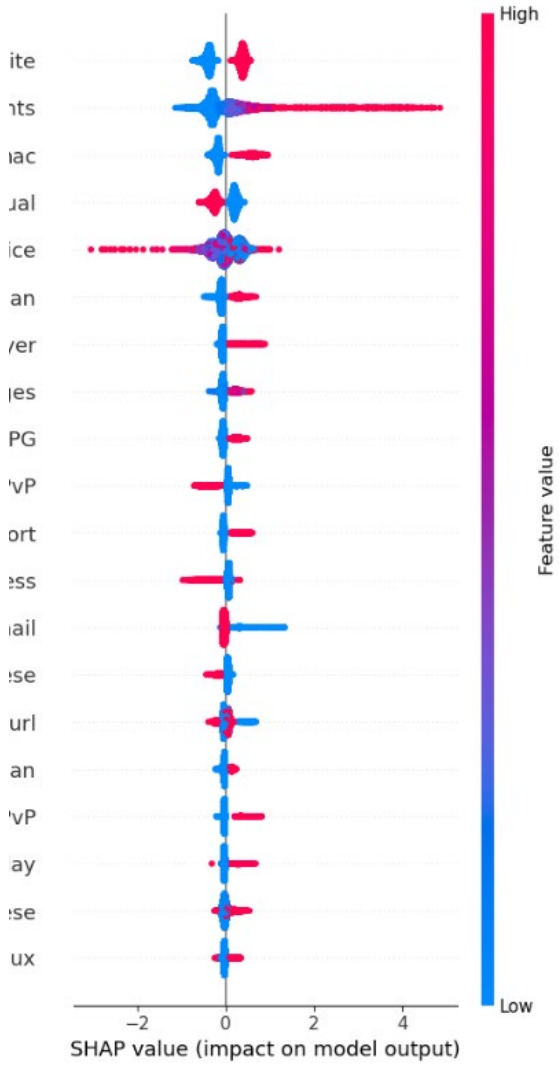
$$\phi_i = -0.2$$

   f) Instansi 1 (Satu) pada fitur x3

$$\phi_i = \frac{1}{3} \times 0.25 + \frac{1}{6} \times 0 + \frac{1}{6} \times 0.25 + \frac{1}{3} \times 0$$

$$\phi_i = 0.125$$

Setelah mendapatkan seluruh nilai *shapley* dari setiap instansi fitur, *Shapley Additive Explanations (SHAP)* akan menginterpretasikannya dengan visualisasi. Contoh hasil interpretasi menggunakan *SHAP* setelah mendapatkan nilai *shapley* menggunakan data yang berbeda dengan Tabel 3.11, ditunjukan pada Gambar 3.11



Gambar 3.11 Interpretasi *SHAP* terhadap model

Pada Gambar 3.11 menggunakan plot *beeswarm,* menunjukan daftar fitur dari paling atas hingga ke bawah yang merupakan urutan fitur paling berpengaruh pada model berdasarkan jumlah besaran nilai *shapley* pada semua sampel, dan konsistensi nilai *shapley* dari setiap instansi fitur terhadap hasil prediksi model.

Warna titik pada hasil interpretasi mengindikasikan nilai fitur tersebut. Warna titik yang semakin biru menunjukkan nilai yang rendah pada fitur tersebut, sedangkan warna titik yang semakin merah menunjukkan nilai yang semakin tinggi pada fitur tersebut. Letak titik terhadap sumbu x nilai *shapley* yang mengindikasikan pengaruh terhadap output model. Semakin kekanan letak titik dari sumbu x = 0, menunjukan semakin besar nilai positif *shapley* sehingga besarnya nilai atau adanya instansi tersebut memberikan kontribusi positif terhadap prediksi popular atau diatas 20000 *user owners*, sedangkan semakin kekiri letak titik dari sumbu x = 0, menunjukan semakin besar nilai negatif *shapley* sehingga kecilnya nilai atau ketiadaan instansi tersebut memberikan kontribusi positif pada prediksi popular atau diatas 20000 user owners.

3.3.10 Analisis Hasil

Pada tahap analisis hasil peneliti akan menyajikan ringkasan dari seluruh penelitian yang telah dilakukan. Peneliti akan melihat performa model serta hasil dari evaluasi *Shapley Additive Explanations*, dan mencatat hasil penelitian untuk kemudian diambil kesimpulan.