

**MENENTUKAN RANGE HARGA GAWAI BERDASARKAN  
SPESIFIKASI DENGAN METODE KLASIFIKASI  
RANDOM FOREST DAN SVM**

**TUGAS AKHIR**

Digunakan Sebagai Syarat Pengerjaan  
Tugas Akhir Mata Kuliah *Machine Learning*

**Oleh Kelompok 7 :**

**ARIEF NAUVAN RAMADHA  
BINTANG ADIYATMA A.P.  
MUCHAMAD RIZAL G.A.**

**NIM.2041720134  
NIM.2041720130  
NIM.2041720015**

**TI-3G**



**POLITEKNIK NEGERI MALANG  
JURUSAN TEKNOLOGI INFORMASI  
PROGRAM STUDI D4 TEKNIK INFORMATIKA  
2022**

## **DAFTAR ISI**

DAFTAR ISI	ii
DAFTAR GAMBAR	iii
DAFTAR TABEL	iv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Manfaat	2
BAB II METODE PELAKSANAAN	3
2.1 Landasan Teori	3
2.1.1. Relasi Harga dan Spesifikasi Handphone	3
2.1.2 Metode SVM(Support Vector Machine)	3
2.1.3 Metode Random Forest	5
2.2 Diagram Alur	7
2.2.1. Tahapan Pelaksanaan	7
2.2.2. Diagram Alur SVM	7
2.2.3 Diagram Alur Random Forest	9
BAB III HASIL DAN PEMBAHASAN	10
3.1 Dataset	10
3.2 Perhitungan Manual	10
3.2.1 Perhitungan Manual SVM	10
3.2.2 Perhitungan Manual Random Forest	17
3.3 Implementasi Kode Program	20
3.3.1 Kode Program Klasifikasi	20
3.3.2 Kode Program Modeling	25
BAB IV KESIMPULAN DAN SARAN	28
4.1 Kesimpulan	28
4.2 Saran	28
DAFTAR PUSTAKA	30

## DAFTAR GAMBAR

Gambar 2.1 Diagram Pelaksanaan	7
Gambar 2.2 Diagram Alur SVM	8
Gambar 2.3 Diagram Alur Random Forest	9
Gambar 3.1 Persiapan Data	21
Gambar 3.2 Check Dataset Info	21
Gambar 3.3 Check Dataset Describe	22
Gambar 3.4 Visualisasi Data Heatmap Correlation Matrix	23
Gambar 3.5 Drop Column	24
Gambar 3.6 Pembuatan Variabel x dan y	24
Gambar 3.7 Pemisahan Data Test dan Data Train	24
Gambar 3.8 Pembagian Panjang Data	25
Gambar 3.9. Standarisasi	25
Gambar 3.10 Mencari hypermeter Random Forest Grid Search	26
Gambar 3.11 Mencari Akurasi RandomForest	26
Gambar 3.12 Mencari hypermeter svm gridsearch	27
Gambar 3.13 Mencari Akurasi SVM	27

## DAFTAR TABEL

Tabel 3.1 Dataset Mobile Price	10
Tabel 3.2 Menentukan Nilai Bobot	11
Tabel 3.3 Menentukan Jumlah dari Range Harga	11
Tabel 3.4 Encode Price Range	12
Tabel 3.5 Menentukan Nilai Bobot(w)	13
Tabel 3.6 Menentukan Nilai Bias	14
Tabel 3.7 Menentukan Hasil Testing	15
Tabel 3.8 Membandingkan hasil testing dengan encode price range	16
Tabel 3.9 Menghitung Akurasi dengan Confusion Matrix	16
Tabel 3.10 Data spesifikasi gawai menurut price range 0	17
Tabel 3.11 Data spesifikasi gawai menurut price range 1	17
Tabel 3.12 Data spesifikasi gawai menurut price range 2	18
Tabel 3.13 Data spesifikasi gawai menurut price range 3	18
Tabel 3.14 Penentuan Root dan Node.	19
Tabel 3.15 Price range berdasarkan ram > 1500.	19
Tabel 3.16 Price range berdasarkan ram <= 1500	19
Tabel 3.17 Gini index ram > 1500	20
Tabel 3.18 Gini index ram > 1500	20

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi hingga terciptanya gawai tentu saja membawa dampak positif bagi masyarakat. Dengan adanya gawai, siapapun sangat mudah mengakses informasi terkini tanpa perlu membeli koran atau pergi ke perpustakaan. Selain itu komunikasi dengan orang-orang di luar kota maupun luar negeri sekalipun jadi lebih mudah dan mudah. Melepas rindu pada orang tercinta tak butuh waktu lama lewat penggunaan gawai.

Harga adalah suatu nilai tukar, dimana digunakan dalam pemasaran dan bisnis yang paling efektif. Harga menjadikan patokan pertama dalam pembelian dan penjualan. Setiap pelanggan pada saat ingin membeli sebuah ponsel pasti memikirkan spesifikasi ponsel tersebut yang akan dibelinya sesuai dengan estimasi harga yang disiapkan.

*Artificial Intelligence* (kecerdasan buatan) yang membuat mesin mampu menjawab pertanyaan secara cerdas sekarang ini dalam bidang rekayasa yang sangat luas. Machine Learning memberi kita teknik terbaik untuk kecerdasan buatan seperti klasifikasi, regresi, pembelajaran terawasi dan pembelajaran tanpa pengawasan dan banyak lagi. Berbagai jenis algoritma pemilihan fitur tersedia untuk memilih fitur yang terbaik dan meminimalkan kumpulan data. Karena ini adalah masalah pengoptimalan maka banyak teknik yang digunakan untuk mengoptimalkan atau mengurangi dimensi dari dataset. Pada hal ini dibahas banyak fitur yang sangat penting dipertimbangkan untuk memperkirakan harga ponsel. Misalnya prosesor dari ponsel, besaran baterai, ukuran dan ketebalan ponsel juga merupakan faktor penting. Memori internal, kualitas kamera, dan kualitas video harus dipertimbangkan. Sehingga menjadi latar belakang kami yaitu dengan menggunakan dataset untuk memprediksi harga produk yang tersedia dan meluncurkan baru.

Dengan adanya permasalahan diatas, maka diperlukan sebuah sistem cerdas yang mampu mengklasifikasikan gawai menurut range harga dan spesifikasi dari gawai tersebut. Klasifikasi merupakan sebuah proses pengolahan data yang dilakukan untuk menemukan suatu model atau menjelaskan dan membedakan konsep dari kelas data, bertujuan untuk memperkirakan suatu kelas dari objek tertentu yang tidak diketahui kelasnya. Penelitian kali ini metode Support Vector Machine (SVM), yaitu merupakan sistem pembelajaran dengan menggunakan ruang hipotesis yang berupa fungsi-fungsi linear didalam sebuah fitur yang memiliki dimensi tinggi dan dilatih menggunakan algoritma pembelajaran yang berdasarkan teori optimasi.

Terdapat beberapa metode dalam melakukan pengklasifikasian salah satunya adalah metode Random Forest. Dimana Random Forest dapat meningkatkan akurasi karena adanya pemilihan secara acak dalam membangkitkan

simpul anak untuk setiap node (simpul di atasnya) dan diakumulasikan hasil klasifikasi dari setiap pohon (tree), kemudian dipilih hasil klasifikasi yang paling banyak muncul.

## **1.2 Rumusan Masalah**

Rumusan masalah pada penelitian ini dapat disusun sebagai berikut: Dari latar belakang diatas, maka dapat dirumuskan dengan membuat beberapa pertanyaan, adapun pertanyaan itu adalah sebagai berikut :

1. Bagaimana menentukan metode Random Forest sesuai untuk permasalahan “Menentukan Range Harga Gawai Berdasarkan Spesifikasi” ?
2. Bagaimana menentukan metode SVM (Support Vector Machine) sesuai untuk permasalahan “Menentukan Range Harga Gawai Berdasarkan Spesifikasi” ?
3. Bagaimana pengguna/*user* mengetahui informasi khusus mengenai produk gawai dengan spesifikasi yang bagus?
4. Bagaimana cara mengklasifikasikan range harga gawai menurut spesifikasi dari gawai tersebut?

## **1.3 Tujuan**

Tujuan yang diharapkan dari penelitian adanya potensi banjir tersebut yaitu :

1. Memastikan bahwa Random Forest dapat meningkatkan akurasi karena adanya pemilihan secara acak dalam membangkitkan simpul anak untuk setiap node (simpul di atasnya) dan diakumulasikan hasil klasifikasi dari setiap pohon (tree), kemudian dipilih hasil klasifikasi yang paling banyak muncul.
2. Menentukan Metode SVM dapat menawarkan akurasi yang lebih tinggi dibandingkan dengan metode klasifikasi lainnya seperti regresi linier berganda ketika satu set data yang sangat besar tersedia, sehingga SVM membuat menangani data dimensi tinggi lebih baik dan menghindari masalah yang kurang pas.
3. Membuat range antara 0-3 dengan melakukan klasifikasi berdasarkan harga gawai dan spesifikasi yang ditawarkan sehingga pengguna/*user* dapat dengan mudah menerima informasi tersebut.
4. Dapat mengklasifikasikan range harga gawai menurut spesifikasi dari gawai.

## **1.4 Manfaat**

Manfaat yang diharapkan dari penelitian menentukan range harga gawai tersebut adalah :

1. Penelitian ini dapat digunakan sebagai cara menentukan jangkauan harga berdasarkan spesifikasi yang dimiliki.
2. Sebagai pemenuhan tugas akhir dari mata kuliah machine learning.

## **BAB II**

### **METODE PELAKSANAAN**

#### **2.1 Landasan Teori**

##### **2.1.1. Relasi Harga dan Spesifikasi Handphone**

Handphone merupakan alat telekomunikasi elektronik bersifat dua arah yang mudah untuk dibawa kemana saja dan mempunyai kemampuan untuk bisa mengirim pesan baik berupa suara, gambar dan informasi. Dalam kehidupan di era sekarang, manusia tentu saja hampir tidak pernah lepas dari yang namanya handphone. Apalagi sekarang dengan semakin berkembangnya teknologi sehingga membuat handphone memiliki berbagai manfaat dan kegunaan sekaligus. Tidak hanya digunakan untuk alat komunikasi.

Dengan banyaknya manfaat yang dimiliki, membuat banyak orang ingin membeli handphone. Karena kebutuhan ini, membuat pasar handphone sangat luas. Jika kita lihat pada zaman sekarang, dengan harga murah kita sudah bisa memiliki handphone. Namun jika kita membutuhkan handphone dengan performa atau untuk mencukupi kebutuhan yang kita inginkan, kita harus mengeluarkan uang lebih banyak. Dengan harga yang lebih mahal, kita juga dapat handphone dengan spesifikasi yang tinggi dan sebaliknya. Oleh karena itu perlu mengetahui jangkauan harga untuk menentukan spesifikasi yang dibutuhkan.

##### **2.1.2 Metode SVM (Support Vector Machine)**

###### **a) Pengertian**

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 bersama rekannya Bernhard Boser dan Isabelle Guyon. SVM (Support Vector Machine) merupakan salah satu metode dalam supervised learning yang biasanya digunakan untuk klasifikasi (seperti Support Vector Classification) dan regresi (Support Vector Regression). Dalam pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya. SVM juga dapat mengatasi masalah klasifikasi dan regresi dengan linier maupun non linear.

###### **b) Tipe SVM**

###### **1. SVM Linier**

SVM linear digunakan untuk data yang dapat dipisahkan secara linear, yang berarti jika sebuah dataset dapat diklasifikasi menjadi dua kelas dengan menggunakan sebuah garis lurus tunggal, maka data tersebut disebut sebagai data yang dapat dipisahkan secara linear, dan classifier yang digunakan disebut sebagai Linear SVM classifier.

###### **2. SVM Non Linear**

SVM non-linear digunakan untuk data yang dapat dipisahkan secara non-linear, yang berarti jika sebuah dataset tidak dapat diklasifikasi menggunakan garis lurus, maka data tersebut disebut data non-linear dan classifier yang digunakan disebut sebagai Non-linear SVM classifier.

### c) Model Persamaan

Model Persamaan SVM Sederhana adalah seperti berikut ini :

Data pada suatu dataset diberikan variabel  $x_i$  , sedangkan untuk kelas pada dataset diberikan variabel  $y_i$  . Metode SVM membagi dataset menjadi 2 kelas. Kelas pertama yang dipisah oleh hyperplane bernilai 1, sedangkan kelas lainnya bernilai -1.

$$X_i.W + b \geq 1 \text{ untuk } Y_i = 1 \quad (1)$$

$$X_i.W + b \leq -1 \text{ untuk } Y_i = -1 \quad (2)$$

Dimana :

- $X_i$  = data ke -i
- $W$  = nilai bobot support vector yang tegak lurus dengan hyperplane
- $b$  = nilai bias
- $Y_i$  = kelas data ke - i

Bobot vector ( $w$ ) adalah garis vektor yang tegak lurus antara titik pusat koordinat dengan garis hyperplane. Bias ( $b$ ) merupakan koordinat garis relative terhadap titik koordinat. Persamaan (3) merupakan persamaan untuk menghitung nilai  $b$ , sedangkan persamaan (4) merupakan persamaan untuk mencari nilai  $w$ .

$$b = -\frac{1}{2}(w \cdot x^+ + w \cdot x^-) \quad (3)$$

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad (4)$$

Keterangan :

$b$  = nilai bias

$w \cdot x^+$  = nilai bobot untuk kelas data positif

$w \cdot x^-$  = nilai bobot untuk kelas data negatif

$w$  = bobot vektor

$\alpha_i$  = nilai bobot data ke-i

$y_i$  = kelas data ke-i

$x_i$  = data ke-i

Nilai-nilai  $a$  dan  $b$  dapat dihitung dengan menggunakan Rumus dibawah ini :

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{(\sum x^2) - (\sum x)^2}$$



$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

#### d) Urutan langkah - langkah SVM

Berikut ini adalah Langkah-langkah dalam melakukan Analisis Regresi Linear Sederhana :

- 1) Tentukan Tujuan dari melakukan Analisis metode SVM.
- 2) Lakukan Pengumpulan Data
- 3) Mencari persamaan hyperline
- 4) Identifikasi nilai bobot( $w$ ) dan bias( $b$ )
- 5) Menggunakan eliminasi dan substitusi untuk mencari nilai bobot( $w$ ) dan bias
- 6) Melakukan pengujian data dengan memasukan nilai bobot dan bias ke dalam persamaan hyperline
- 7) mengecek akurasi dengan confusion matrix

### 2.1.3 Metode Random Forest

#### a) Pengertian

Random Forest adalah teknik pembelajaran mesin yang digunakan untuk memecahkan masalah regresi dan klasifikasi. Ini menggunakan pembelajaran ensemble, yang merupakan teknik yang menggabungkan banyak pengklasifikasi untuk memberikan solusi untuk masalah yang kompleks. Metode random forest (RF) merupakan metode yang dapat meningkatkan hasil akurasi, karena dalam membangkitkan simpul anak untuk setiap node dilakukan secara acak. Metode ini digunakan untuk membangun pohon keputusan yang terdiri dari root node, internal node, dan leaf node dengan mengambil atribut dan data secara acak sesuai ketentuan yang diberlakukan. Root node merupakan simpul yang terletak paling atas, atau biasa disebut sebagai akar dari pohon keputusan. Internal node adalah simpul percabangan, dimana node ini mempunyai output minimal dua dan hanya ada satu input. Sedangkan leaf node atau terminal node merupakan simpul terakhir yang hanya memiliki satu input dan tidak mempunyai output.

#### b) Tipe Random Forest berdasarkan input

Metode Random Forest digolongkan menjadi beberapa tipe berdasarkan jenis inputan. Berikut ini penjelasannya.

##### 1. Random Forest Classifier

Random Forest bagus untuk klasifikasi. Dapat digunakan untuk membuat prediksi kategori dengan beberapa nilai yang mungkin dan dapat dikalibrasi untuk probabilitas output. Satu hal yang perlu diwaspadai adalah overfitting. Random Forest rawan terjadi overfitting, terutama ketika bekerja

dengan dataset yang relatif kecil. Perlu di curigai jika model data dapat membuat prediksi yang "terlalu bagus" pada set uji menggunakan Random Forest. Salah satu cara overfitting adalah menggunakan fitur yang benar-benar relevan dalam model data yang digunakan.

## 2. Random Forest Regressor

Random Forest dapat digunakan sebagai regresi dengan memperluas 'tree' sepenuhnya sehingga setiap daun memiliki tepat satu nilai. Breiman menyarankan untuk membuat regresi random forest dengan cara memperluas pohon secara acak. Kemudian sebuah prediksi secara sederhana mengembalikan variabel respon individu dari distribusi dapat dibangun jika 'forest' cukup besar. Satu peringatan bahwa perkembangan 'tree' sepenuhnya dapat menutupi atau melebihi kapasitas: jika itu terjadi, intervalnya akan sia-sia, seperti prediksi. Hal yang diharapkan adalah sama seperti akurasi dan presisi.

### c) Model Persamaan

Model Persamaan

$$Entropy(Y) = - \sum p(c|Y) \log_2 p(c|Y) \quad (1)$$

Dimana Y adalah himpunan kasus dan  $p(c|Y)$  merupakan proporsi nilai Y terhadap kelas c.

$$Information\ Gain(Y, a) = Entropy(Y) - \sum |Y_v| |Y_a| veValues Entropy(Y_v) \quad (2)$$

Dimana Values (a) merupakan semua nilai yang mungkin dalam himpunan kasus a.  $Y_v$  adalah subkelas dari Y dengan kelas v yang berhubungan dengan kelas a.  $Y_a$  adalah semua nilai yang sesuai dengan a.

### d) Langkah - langkah

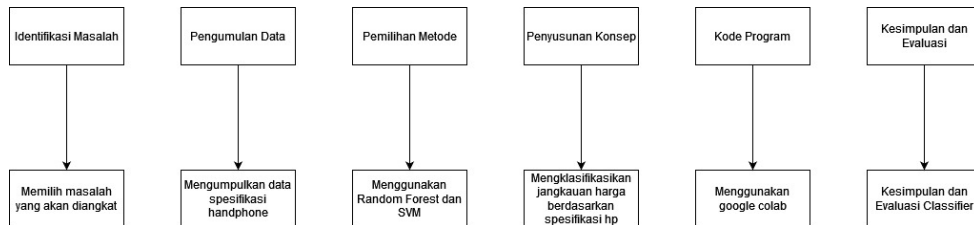
Random Forest bekerja dalam dua fase pertama adalah membuat hutan acak dengan menggabungkan N pohon keputusan, dan kedua membuat prediksi untuk setiap pohon yang dibuat pada fase pertama. Proses Kerja dapat dijelaskan dalam langkah-langkah dan diagram di bawah ini:

- Langkah-1: Pilih titik data K acak dari set pelatihan.
- Langkah-2: Bangun pohon keputusan yang terkait dengan titik data yang dipilih (Subset).
- Langkah-3: Pilih angka N untuk pohon keputusan yang ingin Anda bangun.
- Langkah-4: Ulangi Langkah 1 & 2.
- Langkah-5: Untuk poin data baru, temukan prediksi dari setiap pohon keputusan, dan ketetapan poin data baru ke kategori yang memenangkan suara mayoritas.

## 2.2 Diagram Alur

### 2.2.1. Tahapan Pelaksanaan

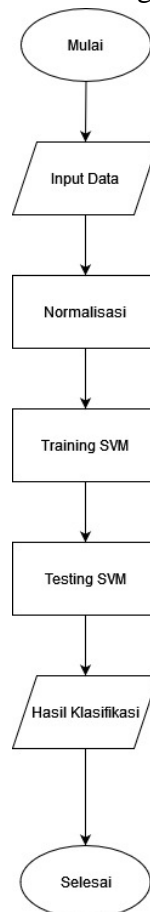
Tahapan pelaksanaan dari penelitian ini memuat enam tahapan dimulai dari identifikasi masalah hingga tahap pengujian dan evaluasi yang digambarkan pada diagram berikut ini :



Gambar 2.1. Diagram Pelaksanaan

### 2.2.2. Diagram Alur SVM

Berikut merupakan alur dari metode regresi linier sederhana.

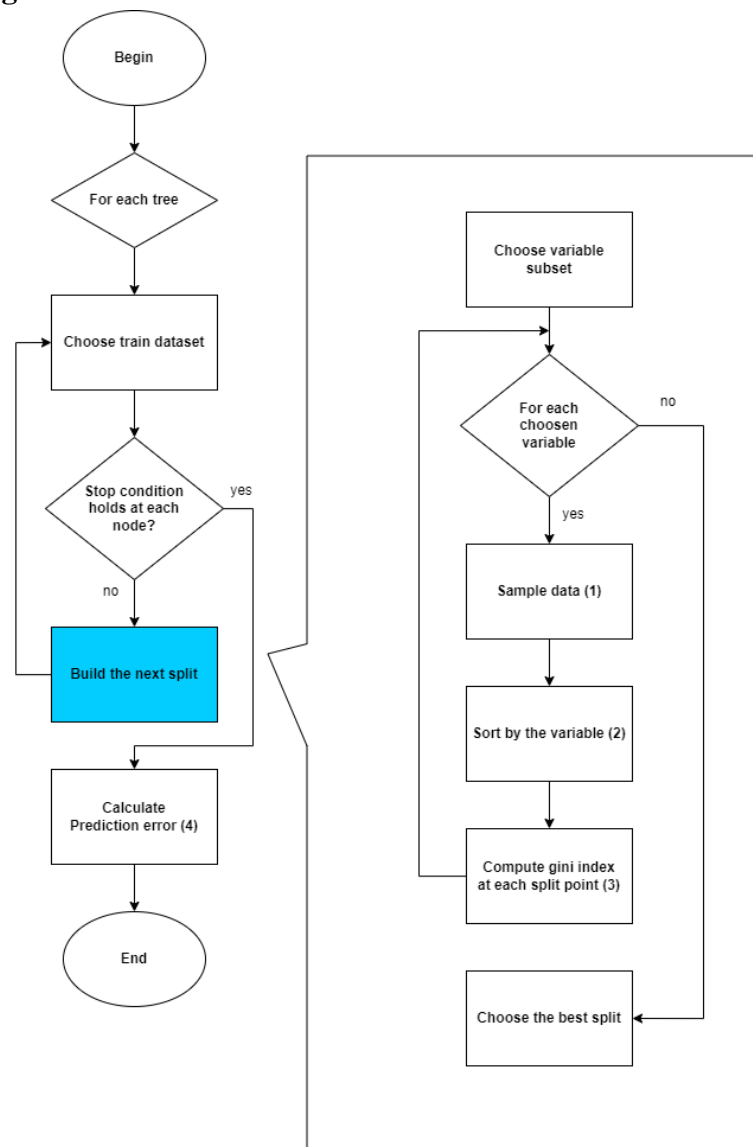


Gambar 2.2. Diagram Alur SVM

Berdasarkan Gambar 2.2, tahap pertama yaitu memasukkan dataset dengan format .xls yang mana data dari penelitian ini berupa data mobile price classification yang berjumlah 2000 data. Data tersebut berasal dari kaggle. Tahap

kedua yaitu proses training SVM yang terdiri dari matriks Hessian yang hasilnya akan digunakan untuk perhitungan berikutnya yaitu menghitung nilai  $\delta\alpha_i$  dan nilai  $\alpha_i$ . Selanjutnya menghitung sequential training yaitu melakukan perhitungan terhadap testing SVM, langkah pertama dalam proses testing adalah menghitung nilai bias, kemudian menghitung nilai  $K(x_i, x_{test})$  yaitu dengan melihat nilai terbesar pada kelas positif dan negatif dengan menggunakan fungsi kernel yang digunakan. Setelah nilai  $K(x_i, x_{test})$  didapatkan langkah selanjutnya adalah menghitung nilai  $f(x)_{test}$ , kemudian jika sudah didapat nilai  $f(x)_{test}$  maka sudah dapat diklasifikasikan berdasarkan hasil dari nilai  $f(x)_{test}$  tersebut. Jika nilai 0 maka data tersebut masuk pada jangkauan harga murah, nilai 1 maka data tersebut masuk pada jangkauan harga normal, nilai 2 maka data tersebut masuk pada jangkauan harga mahal sedangkan jika bernilai 3 maka data tersebut masuk pada jangkauan harga sangat mahal.

### 2.2.3 Diagram Random Forest



Gambar 2.3. Diagram Alur Random Forest

Pertama - tama menentukan data training yang diambil dari dataset dan menentukan apakah data tersebut berbentuk numerik atau tidak. Jika tidak, maka pada data training dilakukan penghitungan probabilitas dan dibuat tabel peluangnya. Jika ya, maka pada data training menghitung rata-rata dalam satu parameter dan menghitung standar deviasi dalam parameter. Setelah itu hasilnya dimasukkan ke dalam tabel peluang.

## BAB III HASIL DAN PEMBAHASAN

### 3.1 Dataset

Dataset yang digunakan pada penelitian ini terdiri dari 21 kolom, yaitu :

1. Battery Power (Total dalam mAh)
2. Bluetooth (dengan Ya = 1, Tidak = 0)
3. Clock Speed (kecepatan microprocessor)
4. Dual Sim (dengan Ya = 1, Tidak = 0)
5. Front Camera (megapixels)
6. 4G (dengan Ya = 1, Tidak = 0)
7. Internal Memory (Gigabytes)
8. Mobile Depth
9. Mobile Weight
10. Number of cores of processor
11. Primary Camera (megapixels)
12. Pixel Resolution Height
13. Pixel Resolution Width
14. RAM
15. Screen Height
16. Screen Width
17. Talk Time
18. 3G (dengan Ya = 1, Tidak = 0)
19. Touch Screen (dengan Ya = 1, Tidak = 0)
20. Wifi (dengan Ya = 1, Tidak = 0)
21. Price Range

Dalam dataset tersebut dapat dilihat pada tabel berikut :

Tabel 3.1 Dataset Mobile Price

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_ra
2	842	0	2.2	0	1	0	7	0.6	188	2	2	20	756	2549	9	7	19	0	0	1	
3	1021	1	0.5	1	0	1	53	0.7	136	3	6	905	1988	2631	17	3	7	1	1	0	
4	563	1	0.5	1	2	1	41	0.9	145	5	6	1263	1716	2603	11	2	9	1	1	0	
5	615	1	2.5	0	0	0	10	0.8	131	6	9	1216	1786	2789	16	8	11	1	0	0	
6	1821	1	1.2	0	13	1	44	0.6	141	2	14	1208	1212	1411	8	2	15	1	1	0	
7	1859	0	0.5	1	3	0	22	0.7	164	1	7	1004	1654	1067	17	1	10	1	0	0	
8	1821	0	1.7	0	4	1	10	0.8	139	8	10	381	1018	3220	13	8	18	1	0	1	
9	1954	0	0.5	1	0	0	24	0.8	187	4	0	512	1149	700	16	3	5	1	1	1	
10	1445	1	0.5	0	0	0	53	0.7	174	7	14	386	836	1099	17	1	20	1	0	0	
11	509	1	0.6	1	2	1	9	0.1	93	5	15	1137	1224	513	19	10	12	1	0	0	
12	769	1	2.9	1	0	0	9	0.1	182	5	1	248	874	3946	5	2	7	0	0	0	
13	1520	1	2.2	0	5	1	33	0.5	177	8	18	151	1005	3826	14	9	13	1	1	1	
14	1815	0	2.8	0	2	0	33	0.6	159	4	17	607	748	1482	18	0	2	1	0	0	
15	803	1	2.1	0	7	0	17	1	198	4	11	344	1440	2680	7	1	4	1	0	1	
16	1866	0	0.5	0	13	1	52	0.7	185	1	17	356	563	373	14	9	3	1	0	1	
17	775	0	1	0	3	0	46	0.7	159	2	16	862	1864	568	17	15	11	1	1	1	
18	838	0	0.5	0	1	1	13	0.1	196	8	4	984	1850	3554	10	9	19	1	0	1	
19	595	0	0.9	1	7	1	23	0.1	121	3	17	441	810	3752	10	2	18	1	1	0	
20	1131	1	0.5	1	11	0	49	0.6	101	5	18	658	878	1635	19	13	16	1	1	0	
21	682	1	0.5	0	4	0	19	1	121	4	11	902	1064	2337	11	1	18	0	1	1	
22	772	0	1.1	1	12	0	39	0.8	81	7	14	1314	1854	2819	17	15	3	1	1	0	
23	1709	1	2.1	0	1	0	13	1	156	2	2	974	1385	3283	17	1	15	1	0	0	
24	1949	0	2.6	1	4	0	47	0.3	199	4	7	407	822	1433	11	5	20	0	0	1	
25	1602	1	2.8	1	4	1	38	0.7	114	3	20	466	788	1037	8	7	20	1	0	0	

### 3.2 Perhitungan Manual

#### 3.2.1 Perhitungan Manual SVM

### a. Menganalisis Nilai Bobot ( $W$ )

Dalam penggunaan method SVM, harus menggunakan bobot( $w$ ) yang mana diambil dari fitur dalam dataset yang telah dibuat seperti berikut ini :

Tabel 3.2 Menentukan Nilai Bobot

AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU
w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	w14	w15	w16	w17	w18	w19	w20
battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	m_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wt

Karena dataset yang digunakan memiliki dua puluh kolom maka fitur yang dimiliki pun terdiri dari 20 nilai bobot. Nilai bobot ini tidak tergantung pada jumlah kolomnya saja melainkan relasi kolom satu dengan lainnya. Nilai bobot disini masih belum memiliki nilai, nantinya nilai bobot ini akan terisi dengan menggunakan substitusi dan eliminasi.

Berdasarkan hasil analisis di atas, terdapat dua puluh kolom yaitu berisi tentang spesifikasi handphone seperti battery power, front camera dan lain sebagainya. Berdasarkan hasil analisis dapat diketahui bahwa jumlah dari masing-masing range harga sama yaitu berjumlah 500.

Tabel 3.3 Menentukan Jumlah dari Range Harga

Murah	500
Normal	500
Mahal	500
Sangat Mahal	500

### b. Menganalisa Bias ( $b$ )

Untuk melakukan perhitungan SVM, selain memerlukan nilai bobot( $w$ ) perlu juga menentukan nilai dari bias. Karena pada saat nantinya dibuat, nilai bias dan nilai bobot akan dibuat sistem persamaan linear dimana  $y$  akan menjadi nilai dari klasifikasi dataset. jika melihat dari dataset, nilai  $y$  memiliki empat nilai yaitu 0,1,2 dan 3. Dimana nilai-nilai tersebut mengartikan klasifikasi jangkauan harga handphone. Keempat nilai ini akan dijadikan data sampel sebagai berikut :

- 0 akan menjadi -2
- 1 akan menjadi -1
- 2 akan menjadi 1
- 3 akan menjadi 2

Tabel 3.4 Encode Price Range

price_range	Data Sample
1	-1
2	1
2	1
2	1
1	-1
1	-1
3	2
0	-2
0	-2
0	-2
3	2
3	2
1	-1
2	1
0	-2
0	-2
3	2
3	2
1	-1
1	-1

Penggunaan data sampel, dimaksudkan agar memudahkan dalam melakukan perhitungan. Penentuan nilai bias disini masih hanya sekedar menentukan negatif atau positif nilai bias tersebut, contoh jika data sampel bernilai negatif satu maka pada sistem persamaan linear, nilai b akan bernilai negatif dan sebaliknya. Nantinya dalam menentukan nilai bias menggunakan substitusi dan eliminasi.

### c. Menghitung Persamaan dan menentukan nilai bobot dan bias

Dengan menggunakan persamaan dalam melakukan perhitungan SVM ini didapatkan data sampel dua puluh persamaan yang mana didapat dari unsur sistem persamaan linear yang harus memiliki jumlah matriks yang sama dalam melakukan perhitungannya. Karena pada dataset memiliki dua puluh kolom maka harus berjumlah dua puluh juga pada barisnya. Adapun beberapa contoh persamaan yang didapat dari dataset yaitu :

- $w1.x1 + w2.x2 + w3.x3 + w4.x4 + w5.x5 + w6.x6 + w7.x7 + w8.x8 + w9.x9 + w10.x10 + w11.x11 + w12.x12 + w13.x13 + w14.x14 + w15.x15 + w16.x16 + w17.x17 + w18.x18 + w19.x19 + w20.x20 - b > -1$
- $w1.x1 + w2.x2 + w3.x3 + w4.x4 + w5.x5 + w6.x6 + w7.x7 + w8.x8 + w9.x9 + w10.x10 + w11.x11 + w12.x12 + w13.x13 + w14.x14 + w15.x15 + w16.x16 + w17.x17 + w18.x18 + w19.x19 + w20.x20 - b > -1$



Nilai x disini merupakan nilai dari dataset. Jadi jika nilai dari battery power adalah 2314 maka dapat dituliskan menjadi  $(2314)w_1$ . Dengan melakukan substitusi dan eliminasi didapatkan nilai  $w_1$ - $w_{20}$  adalah sebagai berikut :

Rumus untuk menentukan nilai  $w_1$ - $w_{20}$

=MMULT(MINVERSE(A2:T21);V2:V21), melakukan eliminasi matriks 20x20 agar mendapat nilai bobot( $w$ )

Tabel 3.5 Menentukan Nilai Bobot( $w$ )

	Mencari nilai $w_1$ - $w_{20}$
$w_1$	0,00111985
$w_2$	0,21880328
$w_3$	-0,549945022
$w_4$	0,094877072
$w_5$	-0,191185082
$w_6$	-0,155059356
$w_7$	-0,003973268
$w_8$	0,492959543
$w_9$	-0,011701902
$w_{10}$	-0,126730887
$w_{11}$	0,076790912
$w_{12}$	0,000228052
$w_{13}$	-4,44172E-05
$w_{14}$	0,00163308
$w_{15}$	-0,220152812
$w_{16}$	0,117449485
$w_{17}$	-0,071847973
$w_{18}$	1,368129012
$w_{19}$	-0,540261069
$w_{20}$	-0,119336934

Setelah mendapatkan nilai bobot ( $w$ ), selanjutnya mencari nilai bias dengan cara substitusi nilai bobot ke dua puluh persamaan sehingga mendapat bias sebagai berikut :

Rumus untuk menentukan nilai bias

=((A2\*Y2)+(B2\*Y3)+(C2\*Y4)+(D2\*Y5)+(E2\*Y6)+(F2\*Y7)+(G2\*Y8)+(H2\*Y9)+(I2\*Y10)+(J2\*Y11)+(K2\*Y12)+(L2\*Y13)+(M2\*Y14)+(N2\*Y15)+(O2\*Y16)+(P2\*Y17)+(Q2\*Y18)+(R2\*Y19)+(S2\*Y20)+(T2\*Y21))+1

, melakukan substitusi nilai bobot( $w$ ) ke dalam persamaan hyperline, nilai +1 didapat dari persamaan hyperline yang pertama dimana memiliki nilai = -1. Karena dipindah ruas maka nilai -1 menjadi +1. Konsep ini diterapkan disemua persamaan hyperline untuk mencari nilai bias.

Tabel 3.6 Menentukan Nilai Bias

Nilai B
-3,55271E-15
-3,44169E-15
-6,77236E-15
-1,77636E-15
-3,9968E-15
0
-4,66294E-15
-3,55271E-15
-229,66666
-6,66134E-15
-3,55271E-15
-5,9952E-15
-3,10862E-15
-4,44089E-15
-4,44089E-15
-5,32907E-15
-7,77156E-15
-4,44089E-15
-3,10862E-15
-3,10862E-15

Setelah mendapatkan nilai bobot serta bias, masukan semua nilai tersebut ke dalam persamaan sehingga menghasilkan hasil tes sebagai berikut :

Rumus untuk menentukan nilai w1-w20

$$= ((A2*Y2)+(B2*Y3)+(C2*Y4)+(D2*Y5)+(E2*Y6)+(F2*Y7)+(G2*Y8)+(H2*Y9)+(I2*Y10)+(J2*Y11)+(K2*Y12)+(L2*Y13)+(M2*Y14)+(N2*Y15)+(O2*Y16)+(P2*Y17)+(Q2*Y18)+(R2*Y19)+(S2*Y20)+(T2*Y21))+Z2,$$

Menjumlahkan semua nilai yang sudah disubstitusikan ke dalam persamaan hyperline.

Tabel 3.7 Menentukan Hasil Testing

Hasil Testing
-1
1
1
1
-1
-1
2
-2
-461,33332
-2
2
2
-1
1
-2
-2
2
2
-1
-1

**d. Menganalisa dan Menghitung Akurasi**

Dari perhitungan manual yang sudah dilakukan, ternyata terdapat perbedaan hasil dari data model dengan hasil testing sebagai berikut :

Tabel 3.8 Membandingkan hasil testing dengan encode price range

Hasil Testing	Data Sample
-1	-1
1	1
1	1
1	1
-1	-1
-1	-1
2	2
-2	-2
-461,33332	-2
-2	-2
2	2
2	2
-1	-1
1	1
-2	-2
-2	-2
2	2
2	2
-1	-1
-1	-1

Untuk menghitung akurasi yang didapatkan, pada perhitungan manual ini menggunakan confusion matrix akurasi sebagai berikut :

Rumus untuk menentukan akurasi confusion matrix

$$=(AB25+AC26)/(AB25+AC25+AB26+AC26)$$

Tabel 3.9 Menghitung Akurasi dengan Confusion Matrix

n=20	Aktual Positif	Aktual Negatif
Prediksi Positif	19	0
Prediksi Negative	1	0
	20	0
Accuary	95%	

Dengan menggunakan confusion matrix dapat diambil akurasi sebesar 95% berdasarkan 20 sampel yang digunakan.

### 3.2.2 Perhitungan Manual Random Forest

#### a. Menentukan filtering dataset

Data dilakukan filtering berdasarkan range harga antara 0-3, yang memiliki tujuan untuk mempermudah dalam membaca dataset dan melakukan perhitungan manual dari data yang berjumlah 2000 baris.

Tabel 3.10 Data spesifikasi gawai menurut price range 0

battery	po dual_sim	int memor ram	touch_scre	wifi	price_range	
1954	1	24	700	1	1	0
1445	0	53	1099	0	0	0
509	1	9	513	0	0	0
1866	0	52	373	0	1	0
775	0	46	568	1	1	0
1602	1	38	1037	0	0	0
851	0	21	478	0	1	0
1568	0	33	508	1	1	0
644	0	22	1262	0	0	0
1195	0	20	504	1	0	0
578	1	57	1270	0	1	0
823	1	60	905	1	1	0
799	0	63	431	1	1	0
1156	1	50	470	1	0	0
1242	1	10	1050	0	1	0
605	0	9	392	0	0	0
797	1	38	590	1	0	0
1114	0	9	907	1	0	0
1234	1	33	1449	0	1	0
1496	1	42	854	1	1	0
885	1	15	1410	1	1	0
879	0	14	349	0	1	0
1355	0	23	880	1	0	0
657	0	37	1412	1	0	0

Tabel 3.11 Data spesifikasi gawai menurut price range 1

battery	po_dual_sim	int_memor_ram	touch_scre	wifi	price_range
842	0	7	2549	0	1
1821	0	44	1411	1	0
1859	1	22	1067	0	0
1815	0	33	1482	0	0
1131	1	49	1835	1	0
682	0	19	2337	1	1
1949	1	47	1433	0	1
503	1	8	2583	0	0
1319	0	41	2227	1	1
725	1	60	1326	0	0
589	1	61	2113	0	1
1253	1	5	1878	1	0
1656	0	34	1629	1	0
1514	0	27	1152	1	1
1054	1	40	2296	0	1
1457	1	16	1440	0	1
1757	0	49	2056	0	0
1484	0	12	1457	1	0
819	1	42	1814	0	1
1103	1	29	1837	1	1
1008	0	61	2009	0	0
1127	1	57	2048	1	1
1412	0	25	837	0	1
1322	1	7	1418	1	0

Tabel 3.12 Data spesifikasi gawai menurut price range 2

battery_po	dual_sim	int_memor	ram	touch_scre	wifi	price_range
1021	1	53	2631	1	0	2
563	1	41	2603	1	0	2
615	0	10	2769	0	0	2
803	0	17	2680	0	1	2
961	1	57	2782	1	1	2
1453	1	52	2373	1	1	2
790	1	11	3169	0	1	2
560	1	50	2150	1	0	2
1347	0	44	2484	0	1	2
1547	1	14	2059	0	1	2
1760	1	63	2053	1	0	2
1073	1	51	2908	1	0	2
1936	1	46	2552	0	0	2
1063	1	48	2910	1	1	2
1720	0	55	2148	1	1	2
702	1	9	2955	0	0	2
1407	0	22	2192	0	1	2
999	1	64	2593	1	0	2
668	1	3	2746	1	1	2
1309	1	33	2334	0	1	2
1977	1	54	1971	1	0	2
1665	0	60	1601	0	0	2
1310	1	7	3182	1	0	2
909	0	53	3187	0	0	2

Tabel 3.13 Data spesifikasi gawai menurut price range 3

battery_po	dual_sim	int_memor	ram	touch_scre	wifi	price_range
1821	0	10	3220	0	1	3
769	1	9	3946	0	0	3
1520	0	33	3826	1	1	3
838	0	13	3554	0	1	3
595	1	23	3752	1	0	3
772	1	39	2819	1	0	3
1709	0	13	3283	0	0	3
519	1	51	3763	0	1	3
956	0	41	3286	1	0	3
1579	1	5	3532	0	1	3
1310	1	51	3845	1	0	3
1725	1	6	3429	1	0	3
1646	0	41	3339	1	0	3
1723	1	42	3587	0	0	3
596	1	64	3238	0	0	3
1654	1	43	3112	1	0	3
987	1	61	3963	0	1	3
616	1	44	3366	1	0	3
1358	0	36	3068	1	0	3
1866	0	30	3834	0	0	3
1166	1	43	3993	1	0	3
1448	1	45	3378	1	1	3
1038	0	43	3709	1	1	3
1199	1	16	3448	0	1	3

#### b. Menentukan nilai pembanding

Pada tahap ini adalah menentukan pembanding yang digunakan untuk menjadi acuan dari range harga gawai, yang dimana menggunakan kolom ram sebagai pembanding range harga. Kemudian dari kolom pembanding tersebut dilakukan sortir menjadi 2 yaitu ram > 1500 dan ram <= 1500.

Tabel 3.14 Penentuan Root dan Node.

Root	Ram	Total
Node	Ram > 1500	1319
	Ram <= 1500	681

**c. Menghitung jumlah data berdasarkan klasifikasi**

Setelah dilakukan perbandingan price range berdasarkan ram maka dilakukan klasifikasi berdasarkan dataset untuk ram dengan besaran kurang dari sama dengan 1500 dan ram dengan besaran lebih dari 1500.

Tabel 3.15 Price range berdasarkan ram > 1500.

Ram > 1500	Price range	Total
	0	17
	1	310
	2	492
	3	500

- Rumus mencari nilai probabilitas range 0 kondisi “ram > 1500” : =COUNTIF(Sheet1!L2:L501;">1500"), dimana menggunakan dataset yang sudah dilakukan filter pada langkah sebelumnya .
- Rumus mencari nilai probabilitas range 1 kondisi “ram > 1500” : =COUNTIF(Sheet1!T2:T502;">1500"), dimana menggunakan dataset yang sudah dilakukan filter pada langkah sebelumnya .
- Rumus mencari nilai probabilitas range 2 kondisi “ram > 1500” : =COUNTIF(Sheet1!AB2:AB502;">1500"), dimana menggunakan dataset yang sudah dilakukan filter pada langkah sebelumnya .
- Rumus mencari nilai probabilitas range 3 kondisi “ram > 1500” : =COUNTIF(Sheet1!D2:D502;">1500"), dimana menggunakan dataset yang sudah dilakukan filter pada langkah sebelumnya .

Tabel 3.16 Price range berdasarkan ram <= 1500.

Ram <= 1500	Price range	Total
	0	483
	1	190
	2	8
	3	0

- Rumus mencari nilai probabilitas range 0 kondisi “ram <= 1500” : =COUNTIF(Sheet1!L2:L501;">1500"), dimana menggunakan dataset yang sudah dilakukan filter pada langkah sebelumnya .
- Rumus mencari nilai probabilitas range 1 kondisi “ram <= 1500” : =COUNTIF(Sheet1!L2:L501;">1500"), dimana menggunakan dataset yang sudah dilakukan filter pada langkah sebelumnya .

- Rumus mencari nilai probabilitas range 2 kondisi “ram <= 1500” : =COUNTIF(Sheet1!AB2:AB502;"<=1500"), dimana menggunakan dataset yang sudah dilakukan filter pada langkah sebelumnya .
- Rumus mencari nilai probabilitas range 3 kondisi “ram <= 1500” : =COUNTIF(Sheet1!D2:D502;"<=1500"), dimana menggunakan dataset yang sudah dilakukan filter pada langkah sebelumnya .

#### d. Menghitung Gini Index

Pada tahap selanjutnya yaitu dengan cara perhitungan *gini index* yang dihitung dengan mengurangi jumlah probabilitas kuadrat dari setiap kelas dari satu. Ini lebih menyukai partisi yang lebih besar dan sangat mudah diterapkan. Sederhananya, ini menghitung probabilitas fitur tertentu yang dipilih secara acak yang diklasifikasikan secara tidak benar. Maka dari perhitungan yang telah dilakukan diperoleh *gini index* seperti berikut:

Tabel 3.17 Gini index ram > 1500.

Ram >1500	
Gini Index	0,85630210
	0,860863647
	0,944762528
	0,999833885

Tabel 3.18 Gini index ram > 1500.

Ram <= 1500	
Gini Index	0,49696288
	0,460929229
	0,999861998
	1

#### e. Hasil

Dari perhitungan manual yang sudah dilakukan dengan menggunakan metode random forest maka didapatkan bahwa hasil dari perhitungan Weighted Gini Index dari node kiri (ram > 1500) adalah 0,881 dan node kanan (ram <= 1500) adalah 0,493.

### 3.3 Implementasi Kode Program

#### 3.3.1 Kode Program Klasifikasi

1. Mempersiapkan Data



```

- Import Library dan Dataset

Pertama kita import pandas terlebih dahulu untuk membaca file csv.

[1] import pandas as pd

Membaca file mobile.csv yang berada pada folder dataset.

- Import Dataset

[2] mobile = pd.read_csv('mobile.csv')

Kemudian kita bisa melihat dataset mobile kita dengan menulis variable mobile.

[4] mobile.head()

```

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi
0	1	1043	1	1.8	1	14	0	5	0.1	193	...	16	226	1412	3476	12	7	2	0	1	0
1	2	841	1	0.5	1	4	1	61	0.8	191	...	12	746	857	3895	6	0	7	1	0	0
2	3	1907	1	2.8	0	1	0	27	0.9	186	...	4	1270	1366	2396	17	10	10	0	1	1
3	4	1546	0	0.5	1	18	1	25	0.5	96	...	20	295	1752	3893	10	0	7	1	1	0
4	5	1434	0	1.4	0	11	1	49	0.5	108	...	18	749	810	1773	15	8	7	1	0	1

5 rows x 21 columns

Gambar 3.1 Persiapan Data

Melakukan import library dan dataset “mobile.csv”. Lalu cek data dengan menampilkan 5 urutan teratas dataset dengan menggunakan `.head()`. import library pandas digunakan untuk membaca sebuah data seperti `.csv`.

## 2. Memahami Data

```

- Info Dataset

Kita juga harus mengecek info dari dataset kita dengan fungsi info(), fungsi berikut memberi tau semua data kita dari tipe data, nilai null, kolom baris, dan nama atribut.

[6] mobile.info()

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype
---  -
 0   id                  1000 non-null   int64
 1   battery_power       1000 non-null   int64
 2   blue                1000 non-null   int64
 3   clock_speed         1000 non-null   float64
 4   dual_sim            1000 non-null   int64
 5   fc                  1000 non-null   int64
 6   four_g              1000 non-null   int64
 7   int_memory          1000 non-null   int64
 8   m_dep               1000 non-null   float64
 9   mobile_wt           1000 non-null   int64
10   n_cores             1000 non-null   int64
11   pc                  1000 non-null   int64
12   px_height           1000 non-null   int64
13   px_width            1000 non-null   int64
14   ram                 1000 non-null   int64
15   sc_h                1000 non-null   int64
16   sc_w                1000 non-null   int64
17   talk_time           1000 non-null   int64
18   three_g             1000 non-null   int64
19   touch_screen        1000 non-null   int64
20   wifi                1000 non-null   int64
dtypes: float64(2), int64(19)
memory usage: 164.2 KB

```

Gambar 3.2 Check Dataset Info

Describe Dataset

Kita juga akan melihat deskripsi dari dataset dengan menggunakan fungsi `describe()`

```
mobile.describe()
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	n_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	...
count	2000.000000	2000.0000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	...	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	1238.518500	0.4950	1.522250	0.509500	4.309500	0.521500	32.046500	0.501750	140.249000	4.520500	...	645.108000	1251.515500	2124.213000	12.306500	5.761
std	439.418206	0.5001	0.816004	0.500035	4.341444	0.499662	18.145715	0.288416	35.399655	2.287837	...	443.780611	432.199447	1084.732044	4.213245	4.356
min	501.000000	0.0000	0.500000	0.000000	0.000000	0.000000	2.000000	0.100000	80.000000	1.000000	...	0.000000	500.000000	256.000000	5.000000	0.000
25%	851.750000	0.0000	0.700000	0.000000	1.000000	0.000000	16.000000	0.200000	109.000000	3.000000	...	282.750000	874.750000	1207.500000	9.000000	2.000
50%	1226.000000	0.0000	1.500000	1.000000	3.000000	1.000000	32.000000	0.500000	141.000000	4.000000	...	564.000000	1247.000000	2146.500000	12.000000	5.000
75%	1615.250000	1.0000	2.200000	1.000000	7.000000	1.000000	48.000000	0.800000	170.000000	7.000000	...	947.250000	1633.000000	3064.500000	16.000000	9.000
max	1998.000000	1.0000	3.000000	1.000000	19.000000	1.000000	64.000000	1.000000	200.000000	8.000000	...	1960.000000	1998.000000	3998.000000	19.000000	18.000

8 rows x 21 columns

Gambar 3.3 Check Dataset Describe

Memahami dataset yang tersedia dengan mengecek info data dan deskripsi data.

### 3. Visualisasi Korelasi Data Antar Kolom Dengan Correlation Matrix

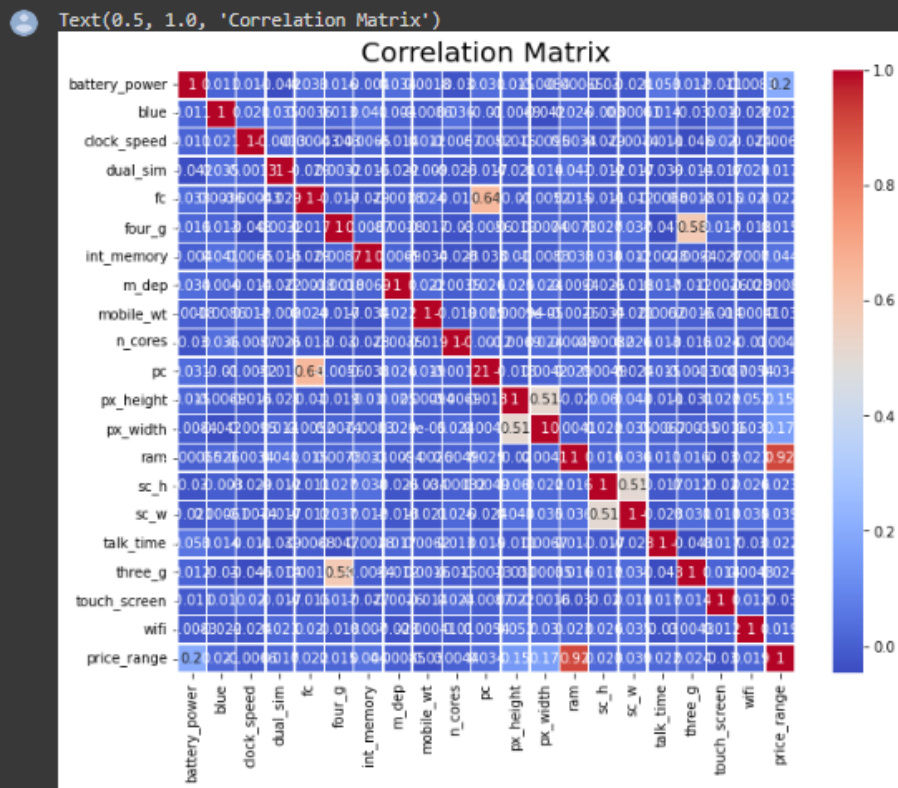
```
[ ] import matplotlib.pyplot as plt
    %matplotlib inline
    import seaborn as sns

[ ] mobile['price_range'].value_counts()

1    500
2    500
3    500
0    500
Name: price_range, dtype: int64

plt.figure(figsize=(10, 8))
correlation_matrix = mobile.corr().round(6)

sns.heatmap(data=correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5, )
plt.title("Correlation Matrix", size=20)
```



Gambar 3.4 Visualisasi Data Heatmap Correlation Matrix

Pada visualisasi heatmap tersebut, spesifikasi handphone dengan korelasi kuat yaitu RAM dengan nilai 0,92. sedangkan import library matplotlib digunakan untuk melakukan visualisasi data seperti membuat plot grafik untuk satu sumbu atau lebih. Setiap sumbu memiliki sumbu horizontal (x) dan sumbu vertikal (y).

#### 4. Melakukan drop column

```
mobile.drop(['touch_screen'], inplace=True, axis=1)
mobile.head()
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	wifi	price_range
0	842	0	2.2	0	1	0	7	0.6	188	2	2	20	756	2549	9	7	19	0	1	1
1	1021	1	0.5	1	0	1	53	0.7	136	3	6	905	1988	2631	17	3	7	1	0	2
2	563	1	0.5	1	2	1	41	0.9	145	5	6	1263	1716	2603	11	2	9	1	0	2
3	615	1	2.5	0	0	0	10	0.8	131	6	9	1216	1786	2769	16	8	11	1	0	2
4	1821	1	1.2	0	13	1	44	0.6	141	2	14	1208	1212	1411	8	2	15	1	0	1

```
mobile.drop(['mobile_wt'], inplace=True, axis=1)
mobile.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 19 columns):
#   Column             Non-Null Count  Dtype
---  -
0   battery_power      2000 non-null   int64
1   blue               2000 non-null   int64
2   clock_speed        2000 non-null   float64
3   dual_sim           2000 non-null   int64
4   fc                 2000 non-null   int64
5   four_g             2000 non-null   int64
6   int_memory         2000 non-null   int64
7   m_dep              2000 non-null   float64
8   n_cores            2000 non-null   int64
9   pc                 2000 non-null   int64
10  px_height           2000 non-null   int64
11  px_width           2000 non-null   int64
12  ram                 2000 non-null   int64
13  sc_h                2000 non-null   int64
14  sc_w                2000 non-null   int64
15  talk_time           2000 non-null   int64
16  three_g            2000 non-null   int64
17  wifi                2000 non-null   int64
18  price_range         2000 non-null   int64
dtypes: float64(2), int64(17)
memory usage: 297.0 KB
```

Gambar 3.5 Drop Column

Melakukan Drop column “touch\_screen” dan column “mobile\_wt”. Karena memiliki korelasi paling kecil di antara yang lain. Karena ingin memaksimalkan proses train dengan data yang powerfull, data yang memiliki sangat sedikit korelasi bisa mempengaruhi proses train.

## 5. Tahapan split data train dan test

### a) Membuat variabel x dan y

```
[53] X = mobile.drop(['price_range'], axis=1)
      y = mobile['price_range']
```

Gambar 3.6 Pembuatan Variabel x dan y

Membuat variabel x dan y berdasarkan analisis korelasi dari kegiatan sebelumnya yaitu :

x : berisi semua data kecuali data pada kolom price\_range

y : data pada kolom price\_range

### b) Pemisahan Data Testing dengan Data Training

```
[54] from sklearn.model_selection import train_test_split

      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=32)
```

Gambar 3.7 Pemisahan Data Test dan Data Train

Fungsi dari import library train\_test\_split diatas digunakan untuk memisahkan data train dan data test dengan proporsi data test : 0,2. dimana digunakan untuk mengevaluasi performa model data test dan data train.

c) Pembagian data x\_train dan x\_test

```
[55] print(f'Total # of sample in whole dataset: {len(X)}')  
      print(f'Total # of sample in train dataset: {len(X_train)}')  
      print(f'Total # of sample in test dataset: {len(X_test)}')  
  
Total # of sample in whole dataset: 2000  
Total # of sample in train dataset: 1600  
Total # of sample in test dataset: 400
```

Gambar 3.8 Pembagian Panjang Data

Berikut merupakan pembagian data dengan x\_train dan x\_test. Dengan hasil di atas kita bisa menyimpulkan, bahwa :

- a. data train = 1600
- b. data test = 400

d) Standarisasi

```
[56] from sklearn.preprocessing import StandardScaler  
      sc = StandardScaler()  
      X_train = sc.fit_transform(X_train)  
      X_test = sc.transform(X_test)
```

Gambar 3.9 Standarisasi

Kegunaan dari import library StandardScaler untuk melakukan Standarisasi agar data memiliki rentang yang sama antara 0 dan 1. Ini juga akan membantu membantu pada saat proses train nanti.

### 3.3.2 Kode Program Modeling

#### A. Random Forest

1. Mencari hypermeter random forest terbaik menggunakan grid search.

```
[ ] from sklearn.ensemble import RandomForestClassifier

rfc=RandomForestClassifier(random_state=32)
param_grid = {
    'n_estimators': [200, 500],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth' : [4,5,6,7,8],
    'criterion' :['gini', 'entropy']
}

[ ] from sklearn.model_selection import GridSearchCV
CV_rfc = GridSearchCV(estimator=rfc, param_grid=param_grid, cv= 5)
CV_rfc.fit(X_train, y_train)

GridSearchCV(cv=5, estimator=RandomForestClassifier(random_state=32),
             param_grid={'criterion': ['gini', 'entropy'],
                          'max_depth': [4, 5, 6, 7, 8],
                          'max_features': ['auto', 'sqrt', 'log2'],
                          'n_estimators': [200, 500]})

[ ] CV_rfc.best_params_

{'criterion': 'entropy',
 'max_depth': 8,
 'max_features': 'auto',
 'n_estimators': 500}
```

Gambar 3.10 Mencari hypermeter Random Forest Grid Search

Melakukan import library dan dataset “mobile.csv”. Setelah melakukan GridSearch mendapatkan hasil :

- criterion : ‘entropy’
- max\_depth : 8
- max\_feature : auto
- n\_estimator : 500

## 2. Mencari akurasi Random Forest

```
Parameter itulah yang akan kita pakai pada algoritma RandomForest kita

[ ] RF = RandomForestClassifier(criterion='entropy', max_depth=8, max_features='auto', n_estimators=500)
RF.fit(X_train, y_train)

RandomForestClassifier(criterion='entropy', max_depth=8, n_estimators=500)

[ ] from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = RF.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)

[[100  6  0  0]
 [ 3  79 11  0]
 [ 0 13 84  9]
 [ 0  0  7 88]]
0.8775
```

Gambar 3.11 Mencari Akurasi RandomForest

Dengan RandomForest kita mendapatkan akurasi sekitar 87% yang mana terbilang cukup baik.

## B. SVM

### 1. Mencari hypermeter svm terbaik menggunakan gridsearch

```
from sklearn.svm import SVC
svm = SVC(random_state=32)

param_grid_svm = {
    'kernel': ['linear', 'rbf', 'sigmoid'],
    'gamma': ['scale', 'auto'],
}

cv_rfc = GridSearchCV(estimator=svm, param_grid=param_grid_svm, cv= 5)
cv_rfc.fit(X_train, y_train)

GridSearchCV(cv=5, estimator=SVC(random_state=32),
             param_grid={'gamma': ['scale', 'auto'],
                         'kernel': ['linear', 'rbf', 'sigmoid']})

[ ] cv_rfc.best_params_

{'gamma': 'scale', 'kernel': 'linear'}
```

Gambar 3.12 Mencari hypermeter svm gridsearch

Melakukan import library dan dataset “mobile.csv”. Setelah melakukan GridSearch mendapatkan hasil :

- Kernel : Linear
- gamma : auto

### 2. Mencari Akurasi SVM

```
[ ] classifier = SVC(kernel = 'linear', gamma='scale', random_state = 32)
classifier.fit(X_train, y_train)

SVC(kernel='linear', random_state=32)

[ ] y_pred = classifier.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)

[[102  4  0  0]
 [ 1 87  5  0]
 [ 0  1 101  4]
 [ 0  0  3 92]]
0.955
```

Gambar 3.13 Mencari Akurasi SVM

Dengan SVM mendapatkan akurasi sekitar 95% cukup baik, dimana validasi untuk mencari akurasi yang lebih akurat.

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Berdasarkan penelitian tugas akhir ini telah dilakukan implementasi dengan dua metode machine learning pada dataset *mobile price classification*. Dataset yang telah dipilih berdasarkan topik yang akan dibahas, dan dataset tersebut diperoleh dari Kaggle. Dalam penelitian ini diambil topik Menentukan Jangkauan Harga Handphone berdasarkan Spesifikasi, berikut beberapa kesimpulan yang berhubungan dengan tujuan penelitian :

1. Penelitian ini dapat mengklasifikasikan jangkauan harga handphone dari spesifikasi handphone dengan tahapan pelaksanaan seperti berikut :
  - a) Identifikasi masalah
  - b) Pengumpulan data
  - c) Pemilihan metode
  - d) Penyusunan konsep
  - e) Pembuatan kode program
  - f) Evaluasi dan kesimpulan
2. Penelitian ini menggunakan dua metode yaitu SVM dan Random Forest Classifier.
3. Penelitian ini berhasil mengimplementasikan dua metode tersebut untuk menentukan jangkauan harga handphone berdasarkan spesifikasi yang dimiliki.
4. Pada metode pertama yaitu SVM, menggunakan persamaan hyperline yang nantinya akan mencari nilai bobot( $w$ ) serta nilai bias dengan menggunakan perhitungan sistem persamaan linear yaitu eliminasi dan substitusi. Setelah mendapatkan nilai yang diinginkan selanjutnya substitusi nilai tersebut ke dalam persamaan hyperline. Hasil dari persamaan hyperline akan dijadikan data dalam menghitung akurasi menggunakan confusion matrix.
5. Pada metode kedua yaitu Random Forest, yang mana mempunyai training data yang panjang, kemudian dilakukan pengambilan sedikit data training tersebut menggunakan subsample yang akan menghasilkan sebuah forest. Random Forest menggunakan dataset yang kemudian dilakukan pembagian menjadi data training dan data testing. Dari data training dilakukan pembagian lagi menjadi algoritma tree, kemudian dari tree tersebut didapatkan mayoritas keputusan yang akan menentukan hasil dari random forest.

#### **4.2 Saran**

Saran yang diperlukan untuk pengembangan program lebih lanjut antara lain :



1. Penambahan tampilan proses pelatihan sehingga pengguna dapat melihat proses yang terjadi.
2. Penggunaan metode klasifikasi data yang lain untuk perbandingan.
3. Penggunaan data yang lebih bervariasi.
4. Penambahan visualisasi data yang lebih bervariasi.

## DAFTAR PUSTAKA

- Arifiyani, N. (2020). *Price Mobile Classification*, 1-16.
- Khan, Z., & Asim, M. (2018). International Journal of Computer Applications. *Mobile Price Class prediction using Machine Learning Techniques*, 6-11.
- Parapat, I. M., Furqon, M. T., & Sutrisno. (2018). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. *Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi*, 6.
- Pipalia, K., & Bhadja, R. (2020). International Journal for Research in Applied Science & Engineering Technology. *Performance Evaluation of Different Supervised Learning Algorithms for Mobile Price Classification*, 8.
- Pudaruth, S. (2014). Computer Science and Engineering Department. *Predicting the Price of Used Cars using Machine Learning Techniques*, 12.
- Siburian, V. W., & Mulyana, I. E. (2018). Prosiding Annual Research Seminar. *Prediksi Harga Ponsel Menggunakan Metode*, 4.
- Srimuang, W., & Intrasonthongchun, S. (2015). 12th International Joint Conference on Computer Science and Software Engineering (JCSSE). *Classification Model of Network Intrusion using Weighted Extreme Learning Machine*, 190-194.