LAPORAN FINAL PROJECT

BISNIS ANALITIK

ANALISIS PEMODELAN DATA APLIKASI PADA GOOGLE PLAY STORE DENGAN METODE NAÏVE BAYES DAN DECISION TREE

Oleh :

Intan Citra Phonskaningtyas (0621940000007)

Jessica Zerlina Sarwono (5003201071)

Asisten Dosen:

Tiza Ayu Virania

Hasri Wiji Aqsari

Muhammad Adlansyah

Dosen:

Dr. Kartika Fithriasari, M.Si

Dr.rer.pol. Dedy Dwi Prastyo, M.Si

Adatul Mukarromah, S.Si, M.Si

Auliya Aziza

Win Heber Goklas Sianipar

DEPARTEMEN STATISTIKA

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

KREDENSIAL MIKRO MAHASISWA INDONESIA

2021

ABSTRAK

Dalam kehidupan sehari-hari di sekitar kita, terdapat banyak data yang dapat diolah sedemikian rupa untuk memberikan informasi yang jelas dan menarik. Salah satunya data pada aplikasi yang mungkin kita semua familiar terhadapnya, Google Play Store. Aplikasi-aplikasi yang diluncurkan pada Google Play Store terus bertambah setiap waktunya. Namun, bagaimana aplikasi tersebut dapat meningkatkan popularitasnya dan menambah penggunanya ketika sudah terlalu banyak aplikasi yang tersedia pada Google Play Store. Pada penelitian ini, dibahas mengenai pemodelan dari variabel-variabel yang menunjang suatu aplikasi dalam peningkatan popularitas serta kepuasan pelanggannya menggunakan metode klasifikasi data *Naïve Bayes Classifier* dan *Decision Tree Classifier*.

Kata kunci: Google Play Store, populer, klasifikasi, Naïve Bayes, Decision Tree.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan zaman yang pesat telah mendorong perkembangan dalam berbagai aspek, salah satunya adalah aspek teknologi. Manusia yang dulunya hanya dapat berkomunikasi secara langsung mulai beralih menggunakan telepon yang memperbolehkan mereka untuk dapat berkomunikasi meskipun terbatas oleh jarak. Telepon yang awalnya juga hanya digunakan sebagai media untuk berbicara secara berjauhan ini, juga telah dikembangkan lebih lanjut menjadi telepon genggam yang multifungsi. Dewasa ini, telepon genggam yang kita kenal sudah memiliki fitur-fitur yang memadai sehingga para penggunanya dapat melakukan pekerjaan lain tanpa harus memanfaatkan maupun membeli perangkat lain seperti kamera, TV, maupun mp3.

Adapun telepon genggam yang dimiliki bisa bertambah nilai gunanya apabila pengguna tersebut mengunduh aplikasi tambahan yang kompatibel. Misalnya, seseorang pengguna ingin memainkan *game* tetris atau pacman yang biasa tersedia pada *gameboy* tradisional. Berkat telepon genggam, pengguna tersebut tidak lagi perlu menggunakan *gameboy* untuk bermain *game* tersebut, melainkan pengguna dapat langsung mencari aplikasi untuk *game* yang dikehendaki dan mengunduhnya. Dalam waktu kurang dari 5 menit, aplikasi *game* yang diinginkan akan langsung dapat digunakan.

Salah satu media yang paling terkenal untuk mencari dan mengunduh aplikasi pada telepon genggam adalah Google Play Store. Pada umumnya, Google Play Store hadir sebagai aplikasi bawaan pada perangkat berbasis OS Android. Aplikasi yang ada pada Google Play Store pada awal mulanya diterbitkan oleh perusahaan-perusahaan besar. Namun, seiring perkembangan akses internet dan ilmu pengetahuan, perusahaan-perusahaan dan instansi-instansi kecil pun sudah mulai dapat meluncurkan aplikasi mereka ke Google Play Store untuk dipakai oleh umum. Adapun sebagian aplikasi yang diterbitkan oleh perusahaan maupun instansi kecil tersebut memang hanya digunakan untuk keperluan internal, salah satu contohnya adalah aplikasi myITS Classroom dan Sinadine Kementerian BUMN. Namun, sebagian perusahaan dan instansi lainnya memang

merencanakan dan membuat suatu aplikasi untuk tujuan komersil, contohnya Instagram dan juga Mobile Legends.

Apabila suatu perusahaan ingin membuat aplikasi untuk keperluan komersil, tentunya perusahaan tersebut akan membuat strategi dan perencanaan sedemikian rupa agar aplikasi tersebut dapat menarik para pengguna dengan meningkatkan kepuasan pengguna. Adapun pada umumnya, kepuasan pengguna yang ada direpresentasikan melalui penilaian dan ulasan yang tersedia pada Google Play Store. Pada penelitian kali ini, penulis bertujuan untuk menunjukkan faktor- faktor yang memengaruhi nilai kepuasan pelanggan pada aplikasi Google Play Store.

1.2 Permasalahan

Dalam praktikum ini, permasalahan yang muncul sebagai acuan untuk analisis adalah sebagai berikut.

1. Apakah Naïve Bayes merupakan metode yang cukup baik untuk pemodelan tingkat kepopuleran aplikasi pada Google Play Store?
2. Apabila seseorang *software developer* memiliki tawaran untuk membuat aplikasi dari tiga kategori aplikasi, kategori manakah yang harus dipilih dengan cara mempertimbangkan variabel lain pada data aplikasi di Google Play Store?

1.3 Tujuan

Perumusan masalah di atas menghasilkan tujuan yang akan dicapai dalam kegiatan praktikum ini, yaitu sebagai berikut.

1. Untuk mengetahui kebaikan Naïve Bayes sebagai metode untuk pemodelan data Google Play Store.
2. Untuk memberikan gambaran pada *software developer* untuk mengambil penawaran yang paling menarik dari penawaran pembuatan tiga kategori aplikasi yang berbeda.

1.4 Manfaat

Dari kegiatan praktikum ini, manfaat yang dapat diambil adalah sebagai berikut.

1. Mampu memahami faktor-faktor yang memengaruhi penilaian suatu aplikasi di Google Play Store
2. Mampu mememberikan gambaran pada *software developer* untuk mengambil penawaran yang paling menarik dari penawaran pembuatan tiga kategori aplikasi yang berbeda.

BAB II

TINJAUAN PUSTAKA

2.1 Preprocessing Data

Preprocessing data merupakan suatu proses transformasi suatu data mentah menjadi bentuk yang lebih mudah untuk dipahami dan diolah. Proses ini sangat penting untuk melihat kualitas dari data yang digunakan. Kualitas data dapat dilihat dan didasarkan pada akurasi, keutuhan, konsistensi, aktualitas, kepercayaan, dan tingkat penafsiran. Untuk menghasilkan data yang memenuhi dasar dari data yang berkualitas tersebut, dilakukan empat langkah utama dalam preprocessing data.

2.1.1 Pembersihan Data (*Data Cleaning*)

Data cleaning merupakan suatu proses membersihkan suatu data yang kotor menjadi data yang bersih. Pada dasarnya, data yang didapat oleh instansi-instansi besar cenderung "kotor". Diagnosa pada data yang memerlukan pembersihan atau data cleaning ini bisa diklasifikasikan menjadi dua, yaitu *single-source problems* dan *multi-source problems*. Pada *single-source problems*, permasalahan dalam data berakar dari data yang berasal dari sumber tertentu yang memiliki kesalahan dalam inputnya, contohnya seperti kesalahan dalam *unique values* seperti kesamaan data nomor KTP dua orang atau kesalahan kecil lain seperti kesalahan ejaan. Di sisi yang lain, *multi-source problems* adalah kekotoran data yang terjadi dalam menggabungkan data dari berbagai sumber yang berbeda. Hal ini bisa dicontohkan dengan perbedaan struktur antar data yang sama, kesalahan dalam penamaan, dan data waktu yang tidak konsisten.

2.1.2 Partisi/Seleksi Data (*Data Selection*)

Data selection merupakan merupakan suatu proses pemilihan data dari rangkaian observasi dengan banyak variabel. Proses ini dilakukan untuk mengambil data yang sesuai untuk diteliti karena tidak semua variabel dari data yang diperoleh akan berguna untuk analisis yang dikehendaki peneiliti. Pada kasus tertentu, seleksi data ini juga diperlukan untuk mengekerucutkan variabel penelitian dan menghindari kesalahan penafsiran pada data.

2.1.3 Tranformasi Data (*Data Transformation*)

Data transformation merupakan suatu proses mengubah data dari rangkaian observasi yang memiliki berbagai variabel menjadi variabel-variabel lain yang lebih sesuai untuk dianalisis peneliti. Pada transformasi data dapat dilakukan hal seperti menambahkan kolom untuk mengkategorikan suatu variabel numerik untuk tujuan tertentu. Selain itu, dapat juga dilakukan dengan cara mengubah jenis data atau kegiatan lainnya yang dapat membantu peneliti untuk mendapatkan *insight* maksimal dari data.

2.2 Visualisasi Data

Visualisasi data merupakan suatu proses menyajikan data ke dalam suatu plot atau yang biasa dikenal sebagai grafik, baik itu merupakan barplot, boxplot, histogram, atau visualisasi lainnya. Memvisualisasikan data yang diperoleh dari observasi merupakan salah satu tahapan yang penting untuk mendapatkan *insight* dari data sekaligus merupakan sebuah awalan untuk mengemahami struktur atau tren data yang didapat. Misalnya, suatu data kotor yang didapat dari hasil observasi memiliki banyak nilai *null* atau kekosongan dalam beberapa baris. Untuk dapat mengetahui cara yang terbaik itu mengatasi kekosongan nilai tersebut, hal yang paling mudah untuk dilakukan adalah mengetahui sebaran datanya. Dalam kasus ini, contohnya apabila data tersebut memiliki distribusi normal, maka ukuran pemusatan yang digunakan untuk mengisi kekosongan tersebut adalah mean dari data. Di sisi yang lain, apabila distribusinya memiliki *skewness*, maka ukuran pemusatan yang tepat untuk digunakan adalah median. Distribusi akan lebih mudah dilihat apabila divisualisasikan terlebih dahulu menggunakan probability density plot atau histogram.

Pemahaman akan cara memvisualisasikan data sangat penting dilakukan. Hal ini disebabkan karena penyajian data yang salah akan menyebabkan pengguna data dan bahkan orang awam yang membaca data tersebut, salah paham akan makna dari data. Tidak hanya dari segi sains dan keakuratan penyajian data, visualisasi data juga memiliki segi artistik. Maka dari itu dua aspek, yaitu sains dan juga seni perlu diperhatikan dalam visualisasi data. Meskipun skala yang disajikan dalam data sudah benar, data yang divisualisasikan dengan pilihan warna yang buruk dan dalam beberapa keperluan memiliki tema atau warna yang tidak cocok untuk kalangan tertentu (misalnya

memiliki pilihan warna yang tidak bisa dibedakan oleh orang yang buta warna), akan mengakibatkan ketidaktepatan visualisasi.

2.2.1 Barchart dan Lollipop Plot

Barchart, atau yang biasa disebut dengan bar plot merupakan salah satu visualisasi yang berguna untuk data yang berjenis kategorik. Pada *barchart*, suatu variabel kategorik akan direpresentasikan dalam suatu batang, dengan jumlah kemunculan pada rangkaian observasi direpresentasikan dalam tinggi (vertikal) ataupun panjang (horizontal) batang tergantung bentuk dari bar chart tersebut. Barchart ini memiliki versi dimodifikasi yaitu *lollipop plo*t, yang pada dasarnya memiliki fungsi yang sama, hanya saja batang pada *barchart* direpresentasikan menjadi garis dan titik dengan guna memperindah visualisasi.

2.2.2 Density Plot dan Histogram

Histogram merupakan salah satu metode visualisasi data yang dapat membantu menjelaskan mengenai distribusi dari suatu data. Secara umum, histogram digunakan untuk mengetahui frekuensi kemunculan dari suatu variabel kontinu, berbeda dengan barchart yang digunakan untuk melihat kemunculan dari variabel-variabel kategorik. Apabila suatu batang dalam barchart merepresentasikan suatu kategori, suatu batang pada histogram merepresentasikan suatu kelas yang menandakan nilai dari data kontinu. Semakin tinggi atau panjang batang pada histogram, maka akan semakin banyak anggota dalam kelas tersebut dan begitu juga sebaliknya.

Pada visualisasi tertentu, histogram seringkali dikombinasi dengan sebuah *density plot.* Adapun *density plot* ini sendiri memiliki fungsi yang sama dengan histogram, yaitu untuk mengetahui sebaran data. Hanya saja, *density plot* memiliki bentuk halus yang mempermudah dalam pembacaan distribusi terutama pada ilmu mengenai teori probabilitas, *density plot* dikenal merepresentasikan probabilitas kejadian pada data sehingga membantu melengkapi histogram yang visualisasinya ditekankan pada mengetahui frekuensi setiap kelasnya.

2.2.3 Pie Chart

Pie chart, sama dengan bar chart, merupakan visualisasi untuk menghitung kemunculan dari suatu variabel kategorik pada sebuah rangkaian observasi. Namun, kemunculan variabel pada

suatu observasi dalam suatu pie chart akan direpresentasikan dalam suatu lingkaran yang menunjukkan komposisi dari kemunculan masing-masing kategori pada suatu rangkaian observasi. Luas dari satu bagian mendeskripsikan seberapa sering kemunculan suatu kategori dibandingkan kategori-kategori lain pada satu rangkaian percobaan yang sama. Semakin mendominasi luasan tersebut, maka kategori tersebut cenderung semakin banyak muncul dibandingkan kategori-kategori lainnya. Di sisi yang lain, semakin kecil luasan tersebut, maka kategori tersebut cenderung semakin jarang muncul dibandingkan kategori-kategori lainnya. Pada umumnya, pie chart disajikan dengan bantuan *label* yang menunjukkan persentase komposisi suatu kategori dalam rangkaian observasi.

2.2.4 Korelasi

Korelasi merupakan salah satu statistik deskriptif yang digunakan untuk mengetahui hubungan antara dua variabel numerik. Dalam statistika, korelasi atau yang biasa disebut sebagai koefisien korelasi pada umumnya dilambangkan dengan huruf *r*. Nilai pada koefisien korelasi ditunjukkan dalam interval $-1 \leq r \leq 1$ dan tidak memiliki satuan. Dua variabel dianggap memiliki korelasi atau hubungan yang kuat apabila memiliki nilai mendekati -1 atau 1. Di sisi yang lain, dua variabel yang memiliki korelasi cenderung mendekati 0 dikatakan memiliki korelasi yang semakin lemah, atau memiliki hubungan yang lemah.

Nilai negatif atau positif pada korelasi menunjukkan jenis hubungan kedua variabel. Apabila korelasi memiliki tanda negatif, maka kedua variabel saling berbanding terbalik. Dalam arti, kenaikan satu variabel akan menyebabkan penurunan pada variabel yang lain dan sebaliknya, penurunan satu variabel akan menyebabkan penurunan pada variabel yang lain. Di sisi lain, korelasi yang memiliki tanda positif menandakan bahwa kedua variabel berbanding lurus. Hal ini berarti bahwa kenaikan satu variabel akan menyebabkan kenaikan pada variabel yang lain dan sebaliknya, kenaikan satu variabel akan menyebabkan kenaikan pada variabel yang lain. Apabila didapat dari dua rangkaian observasi yang memiliki hasil $r = -0{,}83$ dan $r = 0{,}83$ maka bisa dikatakan bahwa hubungan antara kedua variabel dalam masing-masing rangkaian observasi tersebut memiliki nilai yang sama. Hanya saja, variabel dependen pada rangkaian observasi yang pertama menyebabkan penurunan pada variabel independennya, sedangkan variabel dependen

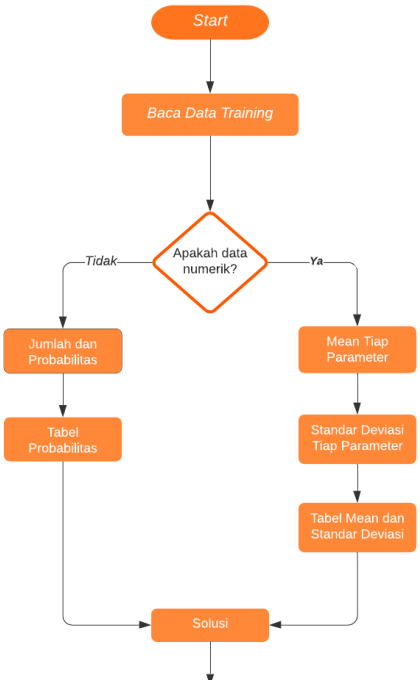
pada rangkaian observasi yang kedua menyebabkan kenaikan pada variabel independennya. Adapun untuk menentukan sebuah koefisien korelasi, dapat memanfaatkan rumus berikut.

2.3 Klasifikasi Data

Klasifikasi data merupakan suatu metode analisis yang digunakan dimana suatu model atau *classifier* dibangun untuk memprediksi nama-nama dari tiap kelas kategori (Morgan Kauffman). Berbeda dengan metode *clustering*, pada klasifikasi data, label atau nama-nama dari tiap kelas telat ditentukan sehingga kita hanya perlu melihat dari ciri-ciri yang tersedia untuk mengelompokkannya sesuai kelas. Maka dari itu, klasifikasi merupakan salah satu metode *supervised learning*. Langkah pertama yang harus dilakukan dari proses klasifikasi adalah mempelajari pemetaan dan fungsi yang memisahkan kelas-kelas data. Setelah dilakukan pemetaan, dilakukan pemodelan klasifikasinya dan mengestimasi akurasi untuk prediksi modelnya.

2.3.1 Naive Bayes Classifier

Metode klasifikasi Naive Bayes Classifier didasarkan pada teorema Bayes yang dikemukakan oleh Thomas Bayes dimana kita menghitung peluang terjadinya suatu kejadian dengan syarat kejadian lainnya. Jika $P(H|\boldsymbol{X})$ merupakan peluang akhir, dimana H dikondisikan terhadap $\boldsymbol{X}$, dan $P(H)$ adalah peluang awal. Maka teorema Bayes dapat berguna untuk mengkalkulasi peluang akhirnya.



***Gambar 2.1** Alur Metode Naïve Bayes*

Dengan berdasarkan pada teorema Bayes, maka metode Naive Bayes Classifier juga mengasumsikan bahwa tiap variabel penjelas bersifat independen atau tidak dipengaruhi atau memengaruhi variabel penjelas lainnya. Berdasarkan hal tersebut, Naive Bayes Classifier menjadi salah satu metode klasifikasi data yang sederhana. Kesederhanaan tersebut menjadi salah satu alasan kami

memilih metode Naive Bayes classifier. Selain itu, metode ini juga relatif mudah untuk dimengerti, memiliki perhitungan yang efisien dan dapat mengklasifikasi sesuai kategori dengan bentuk matematis yang sederhana (Yuliana). Naive Bayes classifier dalam proses pengklasifikasian juga sesuai digunakan karena tidak membutuhkan data latih (*Training Data*) yang besar dalam penentuan parameternya (Alfa Saleh). Berdasarkan metode Naive Bayes classifier, alur dalam proses pengklasifikasian datanya dapat digambarkan pada **Gambar 2.1** atau sebagai berikut:

1. Baca data training
2. Hitung jumlah dan probabilitas, namun apabila data numerik maka:
    a. Cari nilai mean dan standar deviasi dari masing-masing parameter yang merupakan data numerik.
    b. Cari nilai probabilistik dengan cara menghitung jumlah data yang sesuai dari kategori yang sama dibagi dengan jumlah data pada kategori tersebut.
3. Menghasilkan nilai dalam tabel mean, standar deviasi, dan probabilitas
4. Solusi dapat dihasilkan

2.3.2 Decision Tree Classifier

Decision tree atau pohon keputusan merupakan salah satu metode dalam *data mining* yang paling efektif untuk membantu dalam mengambil keputusan. Metode ini pertama kali dikenalkan pada tahun 1960-an dan merupakan metode yang dianggap fleksibel. Hal ini disebabkan karena data yang digunakan dalam metode ini bisa dalam bentuk apapun, baik dalam bentuk diskrit maupun kontinu atau numerik maupun kategorik. Hasil dari pemodelan dengan menggunakan pohon keputusan dapat disajikan dalam sebuah gambaran grafis yang menunjukkan hubungan dari suatu variabel yang diinputkan dengan variabel targetnya. Metode ini dianggap sebagai metode yang mudah diinterpretasikan karena grafik hasil yang ditampilkan dapat secara langsung dianalisis untuk membuat keputusan.

BAB III

METODOLOGI PENULISAN

3.1 Sumber Data

Data yang digunakan pada analisis kali ini adalah data primer berupa dataset Google Play Store yang telah disediakan oleh tim pengajar Bisnis Analitik KMMI.

3.2 Variabel Penelitian

Variabel yang digunakan dalam analisis kali ini adalah variabel Penilaian, Ulasan, Harga, Installs, serta Selisih Hari dengan penjelasan sebagai berikut.

Tabel 3.1 *Variabel Penelitian yang Digunakan*

Nama Variabel	***Jenis Variabel***	***Keterangan***
Kategori	*Kategorik*	*Kategori aplikasi yang tertera pada Google Play Store*
Penilaian	*Numerik*	*Rating pengguna aplikasi pada ulasan yang diberikan*
Ulasan	*Numerik*	*Jumlah ulasan yang diberikan dari pengguna aplikasi*
Tipe	*Kategorik*	*Tipe aplikasi apakah berbayar atau tidak*
Harga	*Numerik*	*Harga aplikasi*
Penilaian_Konten	*Numerik*	*Kategori usia target aplikasi*
Installs	*Numerik*	*Banyaknya pengguna yang mengunduh aplikasi*
Selisih_Hari	*Numerik*	*Selisih hari terakhir diperbarui dengan 25 September 2021 atau hari dilakukannya analisis*
Popular	*Kategorik*	*Level kepopuleran dari aplikasi berdasarkan penilaian*

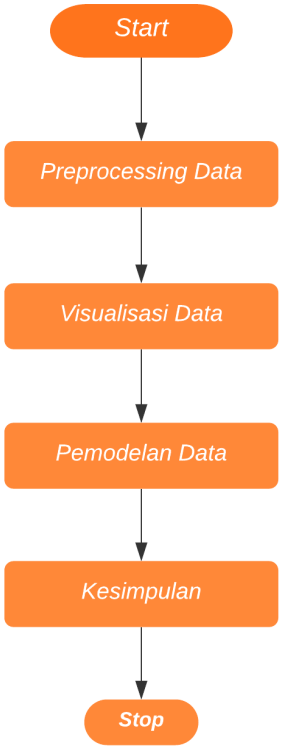
3.3 Langkah Analisis

Langkah yang dilakukan dalam analisis data kali ini adalah sebagai berikut:

1) Merumuskan masalah berdasarkan dataset yang disediakan
2) Melakukan studi literatur
3) Melakukan preprocessing data
4) Melakukan eksplorasi data
5) Melakukan klasterisasi data untuk menentukan banyak klaster
6) Melakukan pemodelan data dengan klasifikasi data
7) Melakukan penarikan kesimpulan

3.4 Diagram Alir

Diagram alir menggambarkan alur perjalanan pembuatan laopran ini, mulai dari proses perumusan masalah hingga pemberian kesimpulan dan saran. Diagram alir yang dipakai dalam laporan ini adalah:



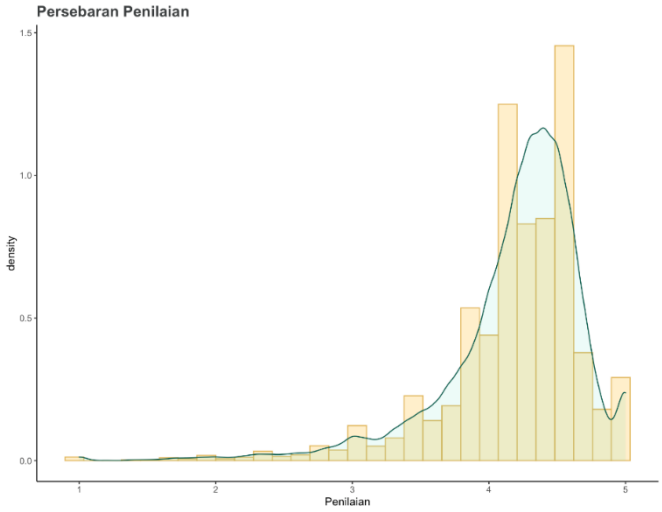
Gambar 3.1 *Diagram Alir Penelitian*

BAB IV

ANALISIS DAN PEMBAHASAN

4.1 Preprocessing Data

Pada penelitian kali ini digunakan beberapa metode untuk melakukan preprocessing data yaitu pembersihan data (*data cleaning*), partisi atau seleksi data (*data partition/data selection*), dan transformasi data (*data transformation*).



Gambar 4.1 *Density Plot pada Variabel Penilaian*

4.1.1 Pembersihan Data (*Data Cleaning*)

Pada metode pembersihan data, data null pada variabel penilaian diatasi dengan mensubstitusikan nilai median dari variabel penilaian ke data *null* tersebut. Pada penelitian kali ini, peneliti juga melakukan seleksi dalam pembersihan pada data dengan cara menghapus salah satu baris yang berisi *data corrupt* dengan kesalahan entri data dan kekosongan yang sekiranya menganggu rangkaian analisis. Adapun dalam ada data yang juga mengalami pemasukkan data yang error sehingga harus diganti dengan cara melihat kolom lainnya.

Pada dataset yang *dimiliki*, variabel “Penilaian” memiliki banyak nilai *null*. Maka dari itu, diperlukan suatu proses untuk menutupi hilangnya data tersebut. Untuk megetahui cara yang tepat agar mengisi kekosongan nilai tersebut dapat dilakukan dengan cara melihat terlebih dahulu sebaran dari datanya.

Dapat dilihat bahwa data variabel penilaian tersebut tidak memiliki distribusi normal, atau yang bisa disebut memiliki distribusi yang *skewed-left.* Maka dari itu, peneliti memutuskan untuk

mengubah nilai-nilai kosong tersebut dengan ukuran pemusatan berupa median dari variabel penilaian.

4.1.2 Partisi atau Seleksi Data (*Data Partition*)

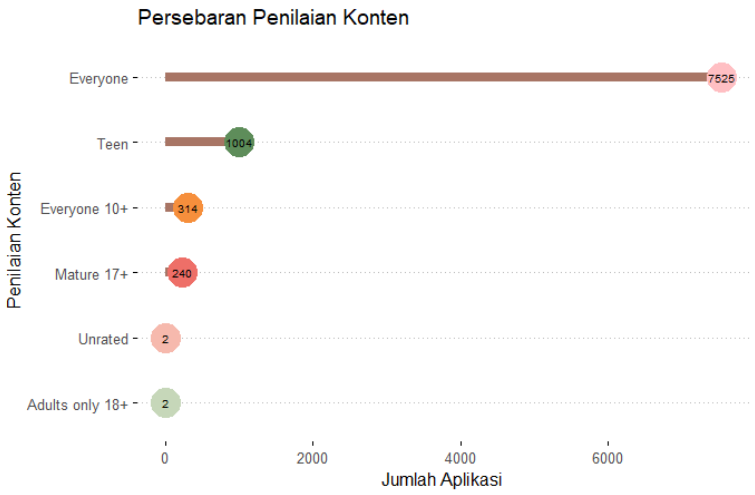
Pada metode ini, dilakukan seleksi data dan pengambilan bagian dari data untuk kemudian dilakukan pemodelan. Proses ini dilakukan untuk mempermudah proses pemodelan dan mencegah galat yang akan terjadi ketika proses pemodelan data. Pada penelitian ini, beberapa kolom memang tidak diikutkan karena dianggap tidak memengaruhi, contohnya seperti variabel App dan Versi_Sekarang yang sekiranya dianggap khas dan tidak bisa dibandingkan antar aplikasi yang berbeda. Selain itu, variabel Genre yang tidak dianggap karena sudah direpresentasikan oleh Kategori yang merupakan variabel serupa. Variabel Android_Version juga tidak terlalu berpengaruh dan cenderung bervariasi antar perangkat sehingga bukan merupakan tolok ukur yang terlalu penting dalam analisis.

4.1.4 Data Transformation

Data yang kurang sesuai untuk dimodelkan kemudian akan ditransformasi atau diubah agar dapat diproses. Pada penelitian kali ini, variabel baru ditambahkan pada dataset yaitu variabel Popular dimana variabel ini didasarkan pada penilaian suatu aplikasi yang akan dijelaskan pada klasifikasi data. Selain penambahan variabel tersebut, beberapa variabel juga dilakukan transformasi seperti variabel Installs dan Ukuran ke dalam variabel numerik. Sedangkan pada variabel Terakhir_Diperbarui dilakukan transformasi data menjadi Selisih_Hari yang merupakan selisih dari hari dibuatnya penelitian ini (25 September 2021) dengan tanggal terakhir aplikasi tersebut diperbarui.

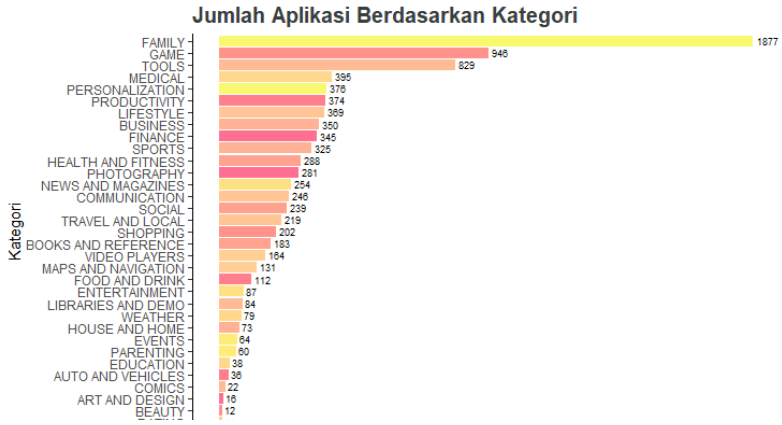
4.2 Visualisasi Data

Visualisasi data merupakan langkah yang penting untuk mengetahui elemen-elemen dalam suatu data. Maka dari itu, visualisai data merupakan langkah awal yang baik untuk memahami data hasil observasi. Pada kasus kali ini, jumlah persebaran penilaian konten dan juga jumlah aplikasi berdasarkan kategori.akan dieksplor terlebih dahulu.



Gambar 4.2 *Lollipop Plot Persebaran Penilaian Konten*

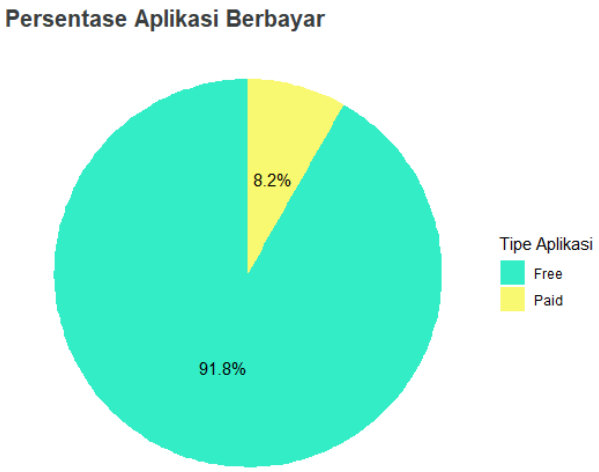
Berdasarkan lollipop plot pada **Gambar 4.2** yang menyajikan mengenai persebaran penilaian konten dalam data Google Play Store, dapat dilihat bahwa sebagian besar dari aplikasi yang ada memiliki penilaian konten "Everyone". Hal ini berarti bahwa sebagian aplikasi dari data Google Play Store yang diteliti bisa digunakan untuk semua kalangan. Setelah penilaian "Everyone", dapat dilihat bahwa penilaian konten kedua tertinggi jatuh pada penilaian konten "Teen" atau remaja. Hal ini berarti bahwa diantara ada sebagian aplikasi pada data Google Play Store yang memang ditujukan untuk remaja. Pada posisi ketiga terdapat penilaian konten "Everyone 10+" atau yang biasa diartikan untuk semua kalangan dengan usia di atas 10. Lalu, pada posisi keempat terdapat penilaian konten "Mature 17+" dan hal ini berarti bahwa ada juga sebagian aplikasi pada data Google Play Store yang berisi konten dewasa dan sekiranya dicocokkan untuk pengguna dengan usia 17 keatas. Adapun pada posisi terakhir, dapat dilihat bahwa diantara aplikasi-aplikasi di Google Playstore tersebut, ada 2 diantaranya yang tidak memiliki penilaian konten atau "Unrated" dan 2 aplikasi juga dinyatakan memiliki konten "Adults only 18+" atau konten dewasa yang dicocokkan untuk pengguna dengan usia 18 keatas.



Gambar 4.3 *Bar Plot Jumlah Aplikasi Berdasarkan Kategori*

Berdasarkan barplot pada **Gambar 4.3** yang menyajikan mengenai jumlah aplikasi berdasarkan 33 kategori pada data Google Play Store, dapat dilihat bahwa sebagian besar dari aplikasi yang ada berada pada kategori "FAMILY" atau

keluarga. Dapat dilihat bahwa nilai ini sangat besar dan hampir dua kali lipat dari kategori paling tinggi kedua, yaitu "GAME" (permainan). Adapun pada data Google Play Store ini, kategori aplikasi yang paling sedikit ditemukan berada pada kategori "DATING" atau aplikasi untuk mencari jodoh.



Gambar 4.4 *Pie Chart Aplikasi berdasarkan Tipe Aplikasi*

Berdasarkan pie chart pada **Gambar 4.4** mengenai persentase aplikasi yang berbayar, dapat dilihat bahwa sebagian besar dari aplikasi yang ada pada data Google Play Store memiliki merupakan aplikasi yang "Free" atau gratis. Adapun komposisi dari aplikasi gratis tersebut mencangkup sebesar 91,8% dari keseluruhan aplikasi pada data Google Play Store. Di sisi yang lain, hanya ada sebesar 8,2% aplikasi yang memiliki jenis "Paid" atau yang merupakan aplikasi berbayar.

Gambar 4.5 *Matriks Korelasi Variabel Numerik*

Berdasarkan correlation plot pada data Google Play Store yang telah diolah tersebut, dapat dilihat bahwa ada 6 variabel yang dibandingkan korelasinya. Keenam variabel tersebut adalah Harga, Selisih_Hari, Ukuran, Penilaian, Ulasan, dan Installs. Dapat dilihat dengan jelas bahwa diantara 6 variabel tersebut, sepasang variabel yang memiliki korelasi paling tinggi adalah variabel Installs dan Ulasan, dengan korelasi sebesar 0.63 yang berarti bahwa kedua variabel tersebut memiliki hubungan positif yang cukup kuat. Dalam

arti, kenaikan pada jumlah Installs akan menyebabkan kenaikan pada jumlah Ulasan dan begitu juga sebaliknya. Sepasang variabel lain yang dapat dilihat memiliki korelasi adalah variabel Selisih_Hari dengan Ukuran, dengan korelasi sebesar -0.2 yang berarti bahwa kedua variabel tersebut memiliki hubungan saling berbanding terbalik yang sangat rendah. Sehingga, dapat diartikan bahwa semakin besar selisih dari hari penelitian dengan update aplikasi terakhir akan menyebabkan sedikit tren penurunan terhadap ukuran aplikasi dan begitu pula sebaliknya. Adapun variabel-variabel lain dalam observasi memiliki korelasi yang sangat kecil atau nyaris tidak ada dan dapat dilihat dari angka korelasi yang berada dibawah angka 0.15 atau -0.15.

4.4 Klasifikasi Data

Dilakukan partisi data pada p=0.8 pada data set untuk penentuan data latih dan data tes. Klasifikasi data dilakukan dengan dua metode berbeda yakni Naïve Bayes Classifier dan Decision Tree Classifier. Pada metode Naïve Bayes akan diteliti mengenai pengaruh variabel lainnya terhadap kepopuleran suatu aplikasi yang didasarkan pada Penilaian. Sedangkan dengan metode Decision Tree, akan diteliti mengenai penilaian dan banyak instalasi aplikasi tertentu, yakni Finance, Business, dan Shopping.

```
> confusionMatrix(cm)
Confusion Matrix and Statistics

          y_pred
           Moderate Popular Unpopular
  Moderate      271       6       156
  Popular       883    4830      1024
  Unpopular       2       0       100

Overall Statistics

               Accuracy : 0.7152
                 95% CI : (0.7047, 0.7256)
    No Information Rate : 0.665
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2344

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: Moderate Class: Popular Class: Unpopular
Sensitivity                  0.23443         0.9988          0.07812
Specificity                  0.97351         0.2172          0.99967
Pos Pred Value               0.62587         0.7169          0.98039
Neg Pred Value               0.87060         0.9888          0.83543
Prevalence                   0.15897         0.6650          0.17602
Detection Rate               0.03727         0.6642          0.01375
Detection Prevalence         0.05954         0.9264          0.01403
Balanced Accuracy            0.60397         0.6080          0.53890
```

Gambar 4.6 *Confusion Matrix Data Latih Naïve Bayes*

4.4.1 Naïve Bayes Classifier

Pada klasifikasi data, objek-objek data baik data latih maupun data tes dikelompokkan ke dalam tiga kategori yang didasarkan pada penilaian yang didapatkan. Ketika rating yang dimiliki

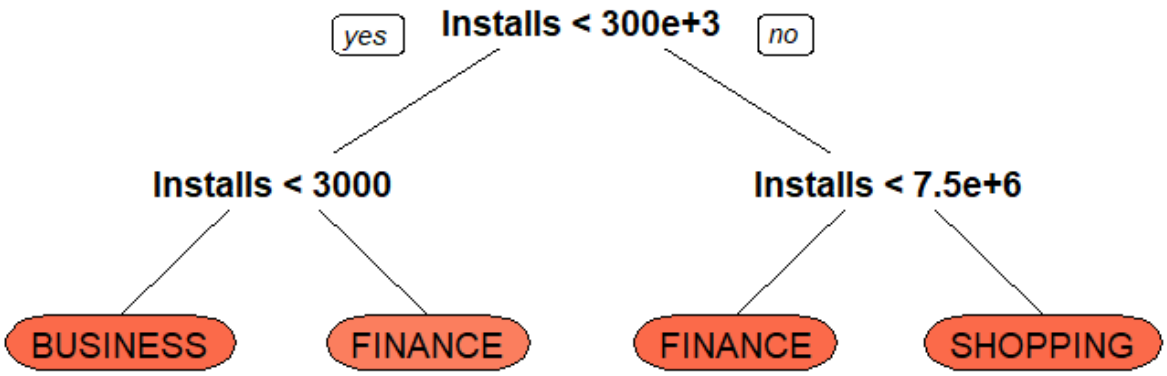
adalah lebih dari 3.5 maka digolongkan ke dalam “Popular”, di antara 2.5 hingga 3.5 akan digolongkan ke dalam “Moderate”, dan di bawah 2.5 akan digolongkan ke dalam “Unpopular”. Setelah digolongkan ke dalam beberapa kategori dilanjutkan dengan pemodelan data latih terlebih dahulu dan didapatkan hasil seperti pada **Gambar 4.6.** Pada data latih ini dapat dilihat bahwa akurasi dari pemodelannya adalah sebesar 0.7152 atau 71.52% yang menunjukkan bahwa model dari data latih ini merupakan model yang cukup baik.

Setelah terbentuk pemodelan dengan data latih, digunakan data tes untuk melihat keakurasian model dengan data sebenarnya. Pada langkah kali ini akan digunakan fungsi confusion matrix kembali dan dihasilkan output seperti tertera pada Gambar 4.7. Didapatkan akurasi dan kebaikan dari model ini sebesar 0.7972 atau 79.72% yang dapat diinterpretasikan bahwa model prediksi dari data Google Play Store sudah cukup baik.

```
> confusionMatrix(cm2)
Confusion Matrix and Statistics

           y_pred2
            Moderate Popular Unpopular
  Moderate        50       1        82
  Popular        127    1374       158
  Unpopular        0       0        23

Overall Statistics

               Accuracy : 0.7972
                 95% CI : (0.778, 0.8155)
    No Information Rate : 0.7576
    P-Value [Acc > NIR] : 3.304e-05

                  Kappa : 0.3209

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: Moderate Class: Popular Class: Unpopular
Sensitivity                  0.28249         0.9993          0.08745
Specificity                  0.94933         0.3523          1.00000
Pos Pred Value               0.37594         0.8282          1.00000
Neg Pred Value               0.92449         0.9936          0.86607
Prevalence                   0.09752         0.7576          0.14490
Detection Rate               0.02755         0.7570          0.01267
Detection Prevalence         0.07328         0.9140          0.01267
Balanced Accuracy            0.61591         0.6758          0.54373
```

Gambar 4.7 *Confusion Matrix Data Latih Naïve Bayes*

4.4.2 Decision Tree Classifier



Gambar 4.8 Analisis *Decision Tree pada Kategori "Business", "Finance", dan "Shopping".*

Pada metode *decision tree*, diibaratkan sebuah kasus seorang *software engineer* yang ingin menerapkan sains data untuk menentukan keputusannya dalam memilih proyek pembuatan aplikasi pada sektor tertentu. Dalam kasus ini, dianggap bahwa orang tersebut menerima tawaran untuk membuat aplikasi dari kategori "FINANCE", "BUSINESS", dan "SHOPPING". Namun, ia hanya dapat memilih salah satu tawaran pekerjaan. Maka dari itu, *software engineer* tersebut akan meneliti pasar untuk melihat aplikasi pada sektor manakah yang cenderung dapat menghasilkan uang banyak (dengan instalasi terbanyak). Pengunaan metode *decision tree* dianggap sebagai metode yang paling mudah untuk mempertimbangkan pilihan dari *software engineer* ini karena metode tersebut dapat memberikan

```
> dtree_fit
CART

257 samples
  2 predictor
  3 classes: 'BUSINESS', 'FINANCE', 'SHOPPING'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 232, 232, 233, 230, 231, 231, ...
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.00000000  0.6190095  0.37951463
  0.03605592  0.6367123  0.40452471
  0.07211185  0.6029098  0.34237243
  0.10816777  0.5560817  0.25963148
  0.14422369  0.5094435  0.17910109
  0.18027962  0.5094435  0.17910109
  0.21633554  0.4514055  0.07123600
  0.25239146  0.4321159  0.03769713
  0.28844739  0.4321159  0.03702711
  0.32450331  0.4254492  0.02591600

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.03605592.
```

Gambar 4.9 *Output Hasil Pemodelan Decision Tree*

penjabaran dan keputusan untuk mengetahui pasar dari masing-masing sektor dengan baik.

Seperti pada kasus *software engineer*, *p*ada metode kali ini akan diteliti mengenai ciri dari setiap kategori yang telah kami tentukan, yaitu "FINANCE", "BUSINESS", dan "SHOPPING". Dengan menggunakan dua variabel prediktornya adalah Penilaian dan Installs (banyak instalasi) didapatkan pemodelan seperti tertera pada **Gambar 4.8** yang didasarkan pada output pada **Gambar 4.9.** Dari decision tree yang telah dibuat, dapat dilihat bahwa terdapat beberapa percabangan pada angka instalasinya. Pada studi kasus kali ini, *software engineer* mengharapkan nilai instalasi yang tinggi, sehingga ketika sampai pada percabangan pertama, yaitu apakah banyak instalasi kurang dari 300.000, dipilih arah cabang ke kanan yang menyatakan "tidak" atau "*no*" karena yang dikehendaki adalah banyak instalasi yang lebih tinggi. Lalu pada percabangan kedua, dihadapkan kembali pada keputusan apakah banyak instalasi kurang dari 7.500.000 dan dipilih arah cabang ke kanan karena dikehendaki banyak instalasi yang lebih tinggi. Maka dari itu, dari ketiga kategori tersebut didapatkan keputusan untuk memilih kategori "SHOPPING" apabila *software engineer* menghendaki proyek aplikasi yang memiliki angka instalasi tinggi.

```
> confusionMatrix(cm)
Confusion Matrix and Statistics

          y_pred_dt
           BUSINESS FINANCE SHOPPING
  BUSINESS       16       6        0
  FINANCE         7      15        1
  SHOPPING        6       9        2

Overall Statistics

               Accuracy : 0.5323
                 95% CI : (0.4012, 0.6602)
    No Information Rate : 0.4839
    P-Value [Acc > NIR] : 0.262510

                  Kappa : 0.2706

 Mcnemar's Test P-Value : 0.005916

Statistics by Class:

                     Class: BUSINESS Class: FINANCE Class: SHOPPING
Sensitivity                   0.5517         0.5000         0.66667
Specificity                   0.8182         0.7500         0.74576
Pos Pred Value                0.7273         0.6522         0.11765
Neg Pred Value                0.6750         0.6154         0.97778
Prevalence                    0.4677         0.4839         0.04839
Detection Rate                0.2581         0.2419         0.03226
Detection Prevalence          0.3548         0.3710         0.27419
Balanced Accuracy             0.6850         0.6250         0.70621
```

Gambar 4.10 *Confusion Matrix Analisis Decision Tree*

Setelah dilakukan pemodelan dengan menggunakan metode Decision Tree Classifier, perlu dihitung akurasi dan kebaikan dari model yang telah dibuat dengan menggunakan fungsi *confusion matrix* didapatkan hasil output seperti pada **Gambar 4.10**. Seperti yang dapat dilihat, ternyata nilai akurasi atau kebaikan model yang dihasilkan pada Decision Tree ketika diproses dengan data tes berada di angka 0.5323 atau 53.23% sehingga dapat dikatakan bahwa model ini kurang baik.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis di atas, dapat ditarik beberapa kesimpulan yaitu:

1. Naïve Bayes merupakan metode yang cukup baik untuk melakukan pemodelan data tingkat kepopuleran suatu aplikasi pada Google Play Store dengan kebaikan model sebesar 79.27%.
2. Berdasarkan hasil analisis, aplikasi dengan kategori "SHOPPING" lebih sesuai dengan preferensi *software engineer* karena dapat berpotensi menghasilkan angka instalasi yang lebih tinggi dibandingkan kedua kategori lainnya.

5.2 Saran

Kegiatan penelitian ini hendaknya dilakukan analisis yang lebih mendalam terkait data dengan mempertimbangkan dan mengeksplorasi data-data terkait lainnya.

DAFTAR PUSTAKA

Han, Jiawei & Kamber, Micheline (2012). *Data Mining: Concept and Techniques*. Elsevier Inc.

Rahm, Erhard & Do, Hong (2000). Data Cleaning: Problems and Current Approaches. *IEEE Data Eng. Bull.*

Saleh, Alfa (2015). *Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga*. Citec Journal, Vol. 2, No. 3. Universitas Potensi Utama

Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, *27*(2), 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044

Syarli & Muin, Asrul A. (2016). *Metode Naive Bayes Untuk Prediksi Kelulusan (Studi Kasus: Data Mahasiswa Baru Perguruan Tinggi).* Jurnal Ilmiah Ilmu Komputer, Vol. 2, No. 1. Universitas Islam Negeri Makassar

Taruna, Shyara R. & Hiranwal, Saroj (2013), *Enhanced Naïve Bayes Algorithm for Intrusion Detection in Data Mining.* (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (6), 960-962. Department of Information Technology, STBC, Jaipur, India

Taylor, R. (1990). Interpretation of the Correlation Coefficient: A Basic Review. *Journal of Diagnostic Medical Sonography*, *6*(1), 35–39. https://doi.org/10.1177/875647939000600106

Wilke, C. O. (2019). Fundamentals of Data Visualization: A Primer on making informative and compelling figures. O'Reilly Media, Inc.

LAMPIRAN

Lampiran 1 Dataset Google Play Store

No.	App	Kategori	Penilaian	Ulasan	Ukuran
1	Learn 50 languages	EDUCATION	4.4	55256	14M
2	Rosetta Stone: Learn to Speak & Read New Languages	EDUCATION	4.5	172508	76M
3	Babbel â€“ Learn Spanish	EDUCATION	4.4	54798	11M
4	Mango Languages: Lovable Language Courses	EDUCATION	4	4815	19M
5	Learn English with Aco	EDUCATION	4.6	75112	6.5M
6	Learn to Speak English	EDUCATION	4.4	33646	7.0M
7	Learn languages, grammar & vocabulary with Memrise	EDUCATION	4.7	1107903	Varies with device
8	Learn English with Wlingua	EDUCATION	4.7	314300	3.3M
9	busuu: Learn Languages - Spanish, English & More	EDUCATION	4.3	206527	21M
10	My Class Schedule: Timetable	EDUCATION	4.1	9348	Varies with device
11	Study Checker	EDUCATION	4.2	3816	2.6M
12	My Study Life - School Planner	EDUCATION	4.3	47847	21M
13	HomeWork	EDUCATION	4.3	16195	5.2M
14	Next Gen Science Standards	EDUCATION	4.3	206	18M
15	myHomework Student Planner	EDUCATION	4	28392	Varies with device
…	…	….	…	…	…
10041	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M

No.	Installs	Tipe	Harga	Penilaian_Konten	Genres	Terakhir_Diperbarui
1	5,000,000+	Free	0	Everyone	Education	19-Jun-18
2	5,000,000+	Free	0	Everyone	Education;Education	27-Jun-18
3	1,000,000+	Free	0	Everyone	Education	30-Jul-18
4	500,000+	Free	0	Everyone	Education	17-Jul-18
5	1,000,000+	Free	0	Everyone	Education	11-Nov-17
6	1,000,000+	Free	0	Everyone	Education	15-Jul-18
7	10,000,000+	Free	0	Everyone	Education	2-Aug-18
8	10,000,000+	Free	0	Everyone	Education	2-May-18
9	10,000,000+	Free	0	Everyone 10+	Education	1-Aug-18
10	1,000,000+	Free	0	Everyone	Education	1-Jun-16
11	500,000+	Free	0	Everyone	Education	4-May-16
12	1,000,000+	Free	0	Everyone	Education	13-Jun-17
13	1,000,000+	Free	0	Everyone	Education	20-Sep-16
14	50,000+	Free	0	Everyone	Education	20-Dec-16
15	1,000,000+	Free	0	Everyone	Education	7-Mar-18
…	…	….	…	…	…	…
10041	10,000,000+	Free	0	Everyone	Lifestyle	25-Jul-18

No.	Versi_Sekarang	Versi_Android
1	10.9.1	4.0 and up
2	5.2.1	5.0 and up
3	20.7.2	4.4 and up
4	4.2.3	4.2 and up
5	2.09	4.1 and up
6	2.3.9	4.0 and up
7	Varies with device	Varies with device
8	1.94.9	4.0 and up
9	13.9.0.161	5.0 and up
10	Varies with device	Varies with device
11	3.6.0.115_FN	2.3 and up
12	6.1.3	4.0.3 and up
13	8.5.2	4.0 and up
14	1.12.1	2.3 and up
15	Varies with device	Varies with device
…	…	….
10041	Varies with device	Varies with device

Lampiran 2 Syntax Analisis

```
setwd('E:/Bisnis Analitik')
df0           <-           read.csv('E:/Bisnis
Analitik/Dataset_3.csv',header=T,sep=',')
data <- df0[-9673,] #Remove Corrupt Data
data[8349,"Tipe"] <- "Free" #Harga Nol, Free

library(ggplot2)
library(stringr)
library(dplyr)
library(e1071)
library(catools)
library(ggcorrplot)
library(caret)
library(corrgram)

#Density Plot
ggplot(data,aes(x=Penilaian))+

geom_histogram(aes(y=..density..),colour="#E
2B659", fill="#FFEECA")+

geom_density(alpha=0.1,fill="#00D4B0",trim=F
, colour = "#005B4C") +
  ggtitle("Persebaran      Penilaian")      +
theme_classic() +
  theme(plot.title   =   element_text(face   =
"bold", size = 15, color = "#3a3e3f"))

data$Penilaian[is.na(data$Penilaian)]      <-
median(data$Penilaian, na.rm=T)
sum(is.na(data))

data$Kategori=as.factor(data$Kategori)
data$Tipe=as.factor(data$Tipe)
data$Penilaian_Konten=as.factor(data$Penilai
an_Konten)
summary(data)

data$Installs                              <-
str_replace_all(data$Installs, "[+]","")
data$Installs                              <-
str_replace_all(data$Installs, ",","")
data$Installs = as.numeric(data$Installs)

data$Harga   <-   str_replace_all(data$Harga,
"[$]","")
data$Harga = as.numeric(data$Harga)

data$Ukuran  <-  str_replace_all(data$Ukuran,
"0", "0k")
```

```
which(data$Ukuran == "0k")
measure                                    <-
substr(data$Ukuran,nchar(data$Ukuran),nchar(
data$Ukuran))
measure = as.data.frame(measure)
data$Ukuran  <-  str_replace_all(data$Ukuran,
"Varies with device", "0k")

data$Ukuran  <-  str_replace_all(data$Ukuran,
"[M]","")
data$Ukuran  <-  str_replace_all(data$Ukuran,
"[K]","")
data$Ukuran  <-  str_replace_all(data$Ukuran,
"[k]","")
data$Ukuran = as.numeric(data$Ukuran)

data$Ukuran[measure == 'k' | measure == 'K']
= data$Ukuran[measure == 'k' | measure == 'K']
* 1000
data$Ukuran[measure      ==      'M']      =
data$Ukuran[measure == 'M'] * 1000000

data$Ukuran[measure      ==      "e"]      =
median(data$Ukuran[measure == 'k' | measure ==
'K' | measure == 'M'])

data$Kategori = as.factor(data$Kategori)
levels(data$Kategori)
table(data$Kategori)

data$Kategori                              <-
str_replace_all(data$Kategori, "_"," ")

library(tidyverse)
data = data[order(-data$Ulasan),]
data = data[!duplicated(data$App),]


forlabel <- data %>%
  group_by(Kategori) %>%
  summarize(count = n()) %>%
  arrange(desc(count))

#JUMLAH APK PER KAT
#install.packages("ggthemes")
library(ggthemes)
ggplot(forlabel,  aes(x  =  reorder(Kategori,
count),y = count, fill = Kategori)) +
  geom_bar(stat = "identity") + coord_flip()
+
  ggtitle("Jumlah    Aplikasi    Berdasarkan
Kategori",)  +
  geom_text(aes(label = count), vjust = 0.4,
size = 2.5, hjust = -0.2) +
  labs(x = "Kategori", y = "Jumlah Aplikasi")
+
  theme_classic() +
  theme(plot.title   =   element_text(face   =
"bold", size = 15, color = "#3a3e3f"),
        legend.position = "none") +
  scale_fill_manual(values   =   c("#ff6f91",
"#ff808c", "#ff928b", "#ffa28e",
                                   "#ffb295",
"#ffbb94", "#ffc594", "#ffcf95",
                                   "#ffd88b",
"#ffe181", "#ffec78", "#f9f871",
                                   "#ff6f91",
"#ff808c", "#ff928b", "#ffa28e",
                                   "#ffb295",
"#ffbb94", "#ffc594", "#ffcf95",
                                   "#ffd88b",
"#ffe181", "#ffec78", "#f9f871",
                                   "#ff6f91",
"#ff808c", "#ff928b", "#ffa28e",
                                   "#ffb295",
"#ffbb94", "#ffc594", "#ffcf95",
                                   "#ffd88b",
"#ffe181", "#ffec78", "#f9f871"))


#PERSEBARAN
bayar <- data %>%
  group_by(Tipe) %>%
  summarise(count = n())

ggplot(bayar, aes(x = "", y = count, fill =
Tipe)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0) +
  geom_text(aes(x     =     1,     label     =
paste0(round(count/sum(count)*100,1),"%")),
            position = position_stack(vjust =
0.5)) +
  ggtitle("Persentase Aplikasi Berbayar") +
  labs(fill = "Tipe Aplikasi",
       x = NULL,
       y = NULL) +
  scale_fill_manual(values                   =
c("#33EDC6","#F9F871")) + theme_void() +
  theme(plot.title   =   element_text(face   =
"bold", size = 15, color = "#3a3e3f"),
        axis.line = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank())

#Konten

library(ggpubr)
kontenlabel <- data %>%
  group_by(Penilaian_Konten) %>%
  summarize(count = n()) %>%
  arrange(desc(count))

ggdotchart(kontenlabel,          x          =
"Penilaian_Konten", y = "count",
           rotate    =    TRUE,    ggtheme    =
theme_pubclean(), sorting = "descending",
           add   =   "segments",   label   =
kontenlabel$count,
           dot.size   =   9,   font.label   =
list(color = "black",size = 8,vjust = 0.5,

hjust = 0.5),
           add.params    =    list(color    =
"#A87565", size = 3),
           color = "Penilaian_Konten",
           palette                           =
c("#c6d7b9","#f6b9ad","#ee6f68","#f68f3c","#
5e8d5a","#FFBFC3"),
           xlab = "Penilaian Konten", ylab =
"Jumlah Aplikasi",
           main   =   "Persebaran   Penilaian
Konten",
           legend = "none")

class(data$Harga)
```

```
#PALING POPULER
pop <- data %>%
  filter(Installs == 1e+09)

#Menambahkan Selisih Hari
data$Terakhir_Diperbarui                    <-
as.Date(data$Terakhir_Diperbarui,
                                    format  =
"%d-%b-%y")

data = mutate(data, Selisih_Hari = Sys.Date()-
Terakhir_Diperbarui)

data_used <- data %>%
  select(Penilaian,     Ulasan,     Installs,
Selisih_Hari, Harga, Ukuran)

data_used$Selisih_Hari                      <-
as.numeric(data_used$Selisih_Hari)

data_used_naive <- data

data_used_naive$Selisih_Hari                <-
as.numeric(data_used_naive$Selisih_Hari)

#Matriks Korelasi
corrgram(data_used_naive                    %>%
select(where(is.numeric)), order=TRUE,
         upper.panel=panel.cor,  main="Google
Play Store App Classification")

##Naive Bayes##
#Partisi Data
set.seed(120)
index                                       <-
createDataPartition(data_used_naive$Penilaia
n, p=0.80, list=FALSE)
# select 80% of the data for Training
training <- data_used_naive[index,]
dim(training)
# use the remaining 80% of data to testing the
models
testing <- data_used_naive[-index,]
dim(testing)

set.seed(120) # Setting Seed
train = training %>%
  mutate(Popular = "Popularity")

train$Popular[which(train$Penilaian  >=  3.5)]
= "Popular"
train$Popular[which(train$Penilaian  <  3.5  &
train$Penilaian >= 2.5)] = "Moderate"
train$Popular[which(train$Penilaian < 2.5)] =
"Unpopular"

classifier_cl <- naiveBayes(Popular ~ ., data
= train)
classifier_cl

# Predicting on training data'
y_pred  <-  predict(classifier_cl,  newdata  =
train)
# Confusion Matrix
cm <- table(train$Popular, y_pred)
cm
# Model Evaluation
confusionMatrix(cm)

test = testing %>%
  mutate(Popular = "Popularity")

test$Popular[which(test$Penilaian >= 3.5)] =
"Popular"
test$Popular[which(test$Penilaian  <  3.5  &
test$Penilaian >= 2.5)] = "Moderate"
test$Popular[which(test$Penilaian  <  2.5)]  =
"Unpopular"

# Predicting on test data'
y_pred2  <-  predict(classifier_cl,  newdata  =
test)

# Confusion Matrix
cm2 <- table(test$Popular, y_pred2)
cm2

# Model Evaluation
confusionMatrix(cm2)

##Decision Tree##
data_dt = data %>%
  filter(Kategori == c("FINANCE", "BUSINESS",
"SHOPPING"))
#Partisi Data
set.seed(120)
index                                       <-
createDataPartition(data_dt$Installs, p=0.80,
list=FALSE)
# select 80% of the data for Training
training_dt <- data_dt[index,]
dim(training_dt)
# use the remaining 80% of data to testing the
models
testing_dt <- data_dt[-index,]
dim(testing_dt)

train_dt = training_dt %>%
  select(Installs, Penilaian, Kategori)

##Decision Tree##
library(rpart.plot)
trctrl <- trainControl(method = "repeatedcv",
number = 10, repeats = 3)
set.seed(120)
dtree_fit  <-  train(Kategori  ~.,  data  =
train_dt, method = "rpart",
                     parms   =   list(split   =
"information"),
                     trControl=trctrl,
                     tuneLength = 10)
dtree_fit

prp(dtree_fit$finalModel,    box.palette    =
"Reds", tweak = 1.2)

test_dt = testing_dt %>%
  select(Installs, Penilaian, Kategori)

y_pred_dt  <-  predict(dtree_fit,  newdata  =
test_dt)

cm <- table(test_dt$Kategori, y_pred_dt)

confusionMatrix(cm)
```

I