LAPORAN PRAKTIKUM PROYEK AKHIR

BISNIS ANALITIK (A)

KMMI ITS

Pemodelan Variabel Penilaian Aplikasi Google Play Store Menggunakan Algoritma Naïve Bayes, Random Forest dan K-Means

Oleh:

Regina Suhadi (04311940000097)

Kevina Windy Arlianni (06211940000047)

Faril Ahmad (F1A218022)

DEPARTEMEN STATISTIKA

FAKULTAS SAINS DAN ANALITIKA DATA

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

2021

ABSTRAK

Google Play Store adalah pasar platform android yang lazim untuk pendistribusian aplikasi mobile. Google Play Store merupakan salah satu pasar aplikasi seluler yang saat ini marak digunakan oleh para *smartphone user* dengan sistem operasi Android. Tujuan dari paper ini adalah untuk mengetahui algoritma pemodelan yang sesuai untuk memprediksi variable Penilaian berdasarkan variabel Harga, Ulasan, Ukuran, dan Installs pada dataset 3, Google Play Store. Serta mengetahui komparasi hasil ketepatan prediksi paling baik dari beberapa metode pemodelan. Pada paper ini, kami menggunakan algoritma Naïve Bayes, Random Forest, dan K-Means untuk melihat rating dari aplikasi berdasarkan variabel yang ada. Dataset yang digunakan memiliki 10041 observasi dan 9 *attributes*. Untuk hasilnya, pada metode Naïve Bayes memiliki *accuracy* sebesar 55.58%, *recall* sebesar 53.32%, dan *precision* sebesar 95.27%. sementara untuk hasil dengan menggunakan metode *random forest* memiliki *accuracy* sebesar 99.98%, *recall* sebesar 99.98%, dan *precision* sebesar 99.98%. Serta K-Means memiliki nilai SC sebesar 93%

Kata Kunci: Google Play Store, K-Means, Naïve Bayes, Prediksi, Random Forest

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENADAHULUAN

1.1 Latar Belakang

Di era digital ini, aplikasi seluler adalah hal yang sangat marak digunakan oleh khalayak umum. Aplikasi seluler ini bisa digunakan jika diunduh dari pasar aplikasi. Salah satu pasar aplikasi yang marak digunakan adalah Google Play Store. Di dalam Google Play Store sendiri terdapat banyak aplikasi yang dapat diunduh untuk menunjang kehidupan sehari – hari sesuai kaegori yang kita butuhkan. Aplikasi yang tersedia juga sangat beragam, dari mulai yang dikhususkan untuk anak – anak hingga orang dewasa. Aplikasi yang tersedia pun ada yang gratis namun ada juga yang berbayar. Hal tersebut menjadikan data yang diperoleh pun sangat beragam, sehingga untuk memahaminya diperlukan pemodelan yang sesuai.

Pada kelas Bisnis Analitik yang diselenggarakan oleh Departemen Statistika Institut Teknologi Sepuluh Nopember, sebagai salah satu program unggulan dari KMMI ini, mahasiswa yang tergabung di dalam kelas Bisnis Anlatik ini diajarkan bagaimana untuk membuat pemodelan data yang baik dan mudah dipahami oleh banyak orang.

Dalam laporan ini, kami akan memberikan studi terkait pendataan aplikasi Google Play Store dengan informasi unik. Analisis kami dibagi menjadi beberapa fase, yaitu: ekstraksi data, pembersihan data, visualisasi data, dan prediksi model yang berbeda. Agar mendapatkan perbandingan keunggulan di antara pemodelan yang kami gunakan. Untuk itu penulisan laporan ini didasari oleh pengolahan data yang telah disediakan, yaitu dataset 3 yang berisikan data dari Google Play Store. Dengan harapan, penulisan laporan ini dapat membantu para mahasiswa dan khalayak umum untuk memahami pemodelan data statistika yang baik dan benar.

1.2 Rumusan Masalah

Rumusan masalah yang diambil untuk mendasari pelaksanaan praktikum dan pembuatan laporan ini, antara lain:

1. Bagaimana metode pemodelan yang sesuai untuk memprediksi variable dataset 3, Google Play Store?
2. Bagaimana komparasi hasil ketepatan prediksi variabel Penilaian dari beberapa metode pemodelan, mana yang paling baik?

1.3 Tujuan

Adapun tujuan dari pelaksanaan praktikum dan penulisan laporan ini, yaitu:

1. Untuk mengetahui metode pemodelan yang sesuai untuk memprediksi variable pada dataset 3, Google Play Store.
2. Untuk merngetahui komparasi hasil ketepatan prediksi variabel Penilaian dari beberapa metode pemodelan.

1.4 Manfaat

Adapun manfaat yang dapat diambil dari pelaksanaan praktikum dan penulisan laporan ini, yaitu:

1. Agar dapat mengetahui metode pemodelan yang sesuai untuk memprediksi variabel pada dataset 3, Google Play Store.
3. Agar dapat merngetahui komparasi hasil ketepatan prediksi variabel Penilaian dari beberapa metode pemodelan.

BAB II

TINJAUAN PUSTAKA

2.1 Google Play Store

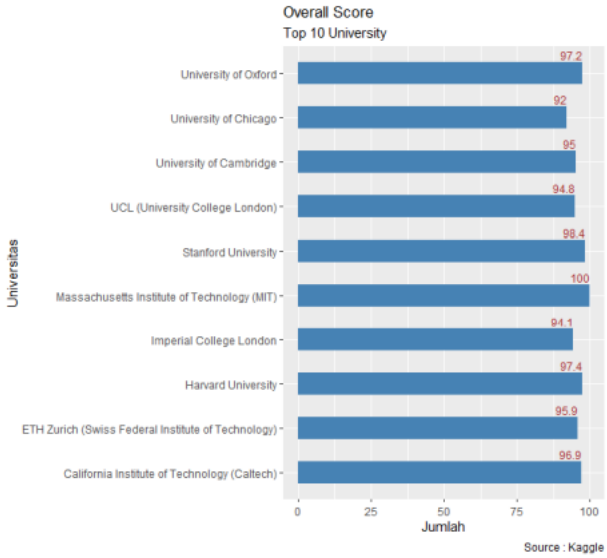
Google play store adalah pasar platform android yang penting untuk pendistribusian aplikasi mobile. Pada tahun 2011, total pengunduhan aplikasi android pada google play sudah mencapai 10 Milliar unduhan. Google play store memungkinkan pengguna untuk mengunduh dan menggunakan aplikasi-aplikasi pihak ketiga secara bebas (*Computer Science and ICT, 2019).* Google Play Store merupakan salah satu pasar aplikasi seluler yang saat ini marak digunakna oleh para *smartphone user* dengan sistem operasi Android.

2.2 Analisis Eksplorasi Data

Eksplorasi data merupakan proses pencarian informasi oleh konsumen data untuk membentuk analisis yang benar dari data yang di yang dikumpulkan. Untuk analisis yang benar, data yang jumlahnya besar terkadang tidak terorganisir dengan baik yang tentunya perlu di eksplorasi terlebih dahulu. Di sinilah diperlukan eksplorasi data digunakan untuk menganalisis data dan memperoleh informasi dari data tersebut dan selanjutnya di analisis lebih lanjut *(Oliveira and Levkowitz,2003).*

2.2.1 *Bar Chart*

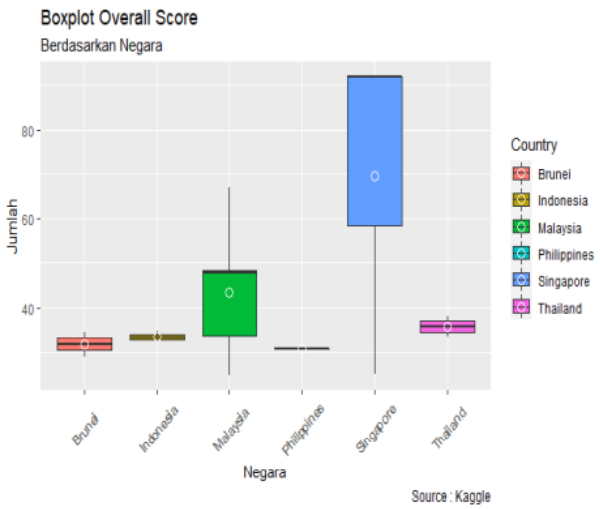
Bar Chart adalah jenis grafik yang direpresentasikan dengan bar atau batang, dimana panjang bar adalah representasi dari ukuran sebuah variable. *Bar chart* juga berfungsi untuk menunjukan frekuensi atau fraksi dari data kategori. Selain frekuensi atau fraksi, *bar chart* juga dapat digunakan untuk menjelaskan nilai kuantitatif lain untuk menjelaskan kategori pada data seperti mean dan persen perubahan.



Gambar 2. 1 Contoh Bar Chart

2.2.2 *Box Plot*

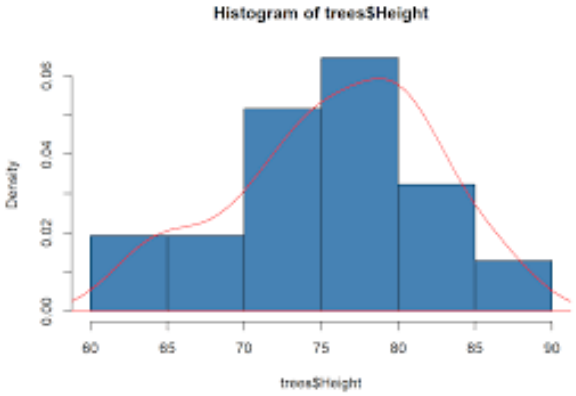
Boxplot merupakan suatu plot grafis yang terdiri dari suatu *box* dan *whisker plot*. *Box* menyatakan nilai kuartil satu hingga kuartil tiga. Sedangkan *whisker plot* berupa garis vertikal yang menunjukkan nilai minimum hingga maksimum. Garis horizontal di dalam *box* menyatakan nilai median. *Boxplot* juga dapat digunakan untuk mengidentifikasi nilai *outlier*, yang biasanya digambarkan dalam bentuk titik di luar *whisker plot.*



Gambar 2. 2 Contoh BoxPlot Chart

2.2.3 Histogram dan *Density Plot*

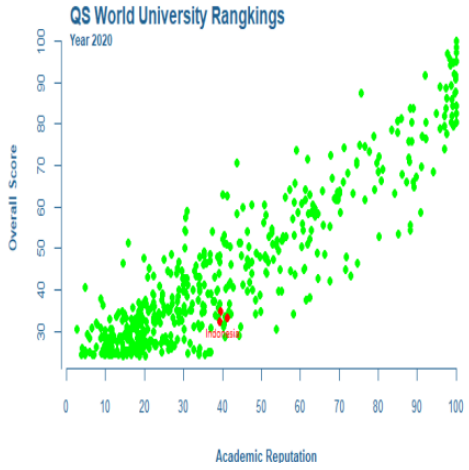
Hisatogram dan *Density Plot* adalah cara visualisasi data variable numerik untuk melihat bentuk distribusi datanya. Dengan menggunakan histogram atau *density plot* maka dapat dilihat apakah distribusi dari suatu data bersifat simetris atau tidak. Hal ini terkait dengan asumsi yang digunakan untuk menganalisa statistic yang akan digunakna pada dataset yang didapatkan.



***Gambar 2. 3 Contoh Histogram dan* Density Plot**

2.2.4 Scatter Plot

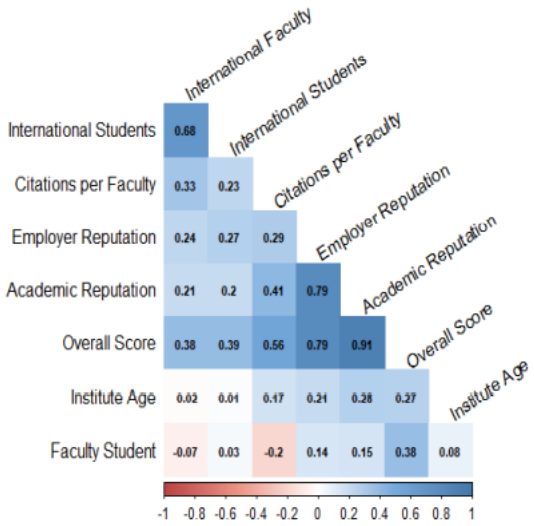
Scatter plot atau sering disebut dengan diagram pencar merupakan suatu grafik yang berisi titik-titik yang menyatakan pasangan nilai observasi (x, y). Plot ini sering digunakan untuk melihat hubungan dari dua variable.



***Gambar 2. 4 Contoh* Scatter Plot**

2.2.5 Correlation Plot

Correlation plot adalah grafik yang digunakan untuk lebih dari dua variable. *Correlation plot* dapat membantu kita untuk memvisualisasikan korelasi antara variabel kontinu. *Correlation plot* merupakan grafik matriks korelasi. Dalam plot ini, koefisien korelasi diberi warna sesuai dengan nilainya. Matriks korelasi juga dapat disusun ulang menurut derajat hubungan antar variable.



Gambar 2. 5 Contoh Correlation Plot

2.3 Pemodelan

Pemodelan merupakan proses membangun / membentuk sebuah model dari suatu sistem nyata dalam bahasa formal. Pemodelan juga dapat dikatakan sebagai sebuah alat menarik yang dapat memberikan sebuah metode untuk mengeksplorasi sistem yang kompleks, untuk bereksperimen dengan sistem tersebut tanpa menghancurkan pada saat yang sama. Bertujuan untuk memperkenalkan beberapa pendekatan pemodelan yang dapat membantu kita untuk memahami bagaimana dunia ini bekerja

Ada beberapa hal yang memengaruhi pemodelan, yaitu: sistem nilai yang dainut,pengetahuan, dan pengalaman. Adapun beberapa hal yang menjadi prinsip pemodelan, yaitu:

1. Perubahan *image* menjadi model: dengan menggunakan Bahasa yang formal

2. Elaborasi: memulai dengan yang sederhana secara bertahap dielaborasi hingga diperoleh model yang representatif.
3. Sinektik: metode yang dibuat utnuk mengembangkan pengenalan masalah seecara analogis.
4. Iteratif: pengulangan dan peninjauan kembali yang diperlukan.

2.3.1 Naïve Bayes

Naive Bayes adalah pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu class. Naive Bayes didasarkan pada teorema Bayes yang memiliki kemampuan klasifikasi serupa dengan decision tree dan neural network. Naive Bayes terbukti memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan ke dalam database dengan data yang besar. Algoritma ini menggunaan prinsip teorema Bayes. Sehingga pada proses pengolahan data, masing-masing feature dianggap independen atau tidak terkait satu sama lain.

2.3.2 *Random Forest*

Random forest adalah kombinasi dari masing – masing tree yang baik kemudian dikombinasikan ke dalam satu model. Random Forest bergantung pada sebuah nilai vektor random dengan distribusi yang sama pada semua pohon yang masing masing *decision tree* memiliki kedalaman yang maksimal. Random forest adalah classifier yang terdiri dari classifier yang berbentuk pohon {h (x, θ k), k = 1, . . .} dimana θk adalah *random vector* yang diditribusikan secara independen dan masing masing tree pada sebuah unit kan memilih class yang paling popular pada input x. Berikut ini karakteristik akurasi pada *random forest.* Terdapat *Classifier* h1(x), h2(x),…, hk(x) dan dengan *training set* dari distribusi *random vector* Y,X, berikut adalah funfsi yang terbentuk :

$$mg(\mathbf{X}, Y) = av_k I(h_k(\mathbf{X}) = Y) - \max_{j \neq Y} av_k I(h_k(\mathbf{X}) = j).$$

Fungsi error yang digunakan:

$$PE^* = P_{\mathbf{X},Y}(mg(\mathbf{X}, Y) < 0)$$

Hasil dari penggabungan fungsi:

$$P_{\mathbf{X},Y}(P_\Theta(h(\mathbf{X}, \Theta) = Y) - \max_{j \neq Y} P_\Theta(h(\mathbf{X}, \Theta) = j) < 0).$$

Pada hasil tersebut menjelaskan mengapa *random forest* tidak *overfit* saat *tree* ditambahkan, tetapi menghasilkan nilai yang terbatas pada error.

2.3.3 K-Means

Jumlah *cluster* ditentukan dengan perhitungan jarak data dengan *centroid*. Perhitungan ini dilakukan dengan menghitung jarak dari setiap nilai pada data dengan ke *centroid*-nya. Jika data berdekatan dengan *centroid*, maka data akan dituliskan sesuai sama dnegan niali *centeroid*-nya. Metode yang digunakan adalah jarak euclidean, seperti pada persamaan seperti di bawah ini :

$$\sum_{i=1}^{n} \sum_{j=1}^{k} (d(Xi, Mj))^2$$

Penentuan jumlah *cluster* yaitu K *cluster* merupakan permulaan dari pemodelan K-Means. Lalu tentukan *centeroid* secara acak, dan setelah itu, masukkan setiap data ke *centroid* yang terdekat. Maka *cluster* akan terbentuk sesuai dengan jarak data dengan *centroid*. *Cluster* yang terbentuk akan dihitung kembali.

2.3.3 Confusion Matrix

Confusion matrix merupakan metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining. Dimana evaluasi *confusion matrix* adalah sebuah matrik dari prediksi yang akan melakukan pengujian untuk memperkirakan obyek yang benar dan salah agar menghasilkan nilai akurasi, presisi dan *recall*. Presisi atau confidence adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya. *Recall* atau *sensitivity* adalah proporsi kasus positif yang sebenarnya yang diprediksi positif secara benar

Rumus *Accuracy*:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN},$$

Rumus *Precision*:

$$Precision = \frac{TP}{TP+FP},$$

Rumus *RecalL*:

$$Recall = \frac{TP}{TP+FN},$$

Rumus *RMSE*:

$$RMSE = = \frac{\overline{\sum_{i}^{n}{}_{=1}(6_i - \hat{Y}_i)^2}}{n}$$

Keterangan :

Y_i = data awal (data sebenarnya)

$\hat{Y}_i$ = data akhir (data hasil estimasi)

n = jumlah data

BAB III
METODOLOGI

3.1 Sumber Data

Sumber data yang digunakan dalam praktikum ini adalah data sekunder yang diperoleh dari *Project* Akhir Kelas Bisnis Analitik KMMI ITS dengan file dataset 3 Data Google Play Store. Dengan Jumlah variabel ada 13 *attributes* dan 10041 observasi. Pengambilan data dilakukan pada.

Hari/tanggal : Kamis, 16 September 2021

Pukul : 12.00 WIB

3.2 Variabel Penelitian

Variabel yang digunakan dalam praktikum ini terdiri dari 9 variabel. Dan dilakukan pemodelan kepada 5 variabel yaitu Penilaian, Harga, Ulasan, Ukuran, dan Installs. Berikut merupakan variabel data dari dataset 3 Data Google Play Store yang akan diamati pada *project* akhir ini adalah sebagai berikut :

Table 3. 1 Variabel Penelitian

No	Variabel	Keterangan
1	Kategori	Variable pembeda aplikasi berdasarkan tujuannya (education, entertaintment, events, dll).
2	Penilaian	Variabel yang berisikan penilaian dari pengguna aplikasi terhadap kepuasannya dengan rentang nilai 1 – 5.
3	Ulasan	Variabel penujuk jumlah *user* yang memberikan penilaian terhadap aplikasi yang diunduh.
4	Ukuran	Variabel penujuk besar atau kecilnya file dari suatu aplikasi yang terdapat pada Google Play Store.
5	Install	Variabel penunjuk berapa banyak *user* yang mengunduh aplikasi pada Google Play Store.
6	Tipe	Variabel penunjuk aplikasi tersebut gratis atau berbayar.

7	Harga	Variabel penujuk jumlah harga yang harus dibayar jika *user* ingin mengunduh aplikasi yang terdapat pada Google Play Store.
8	Penilaian Konten	Variabel penmunjuk rentang umur pengguna dari aplikasi yang terdapat pada Google Play Store.
9	Genre	Variabel penunjuk jenis aplikasi berdasarkan tujuan dari *user* yang mengunduh di Google Play Store.

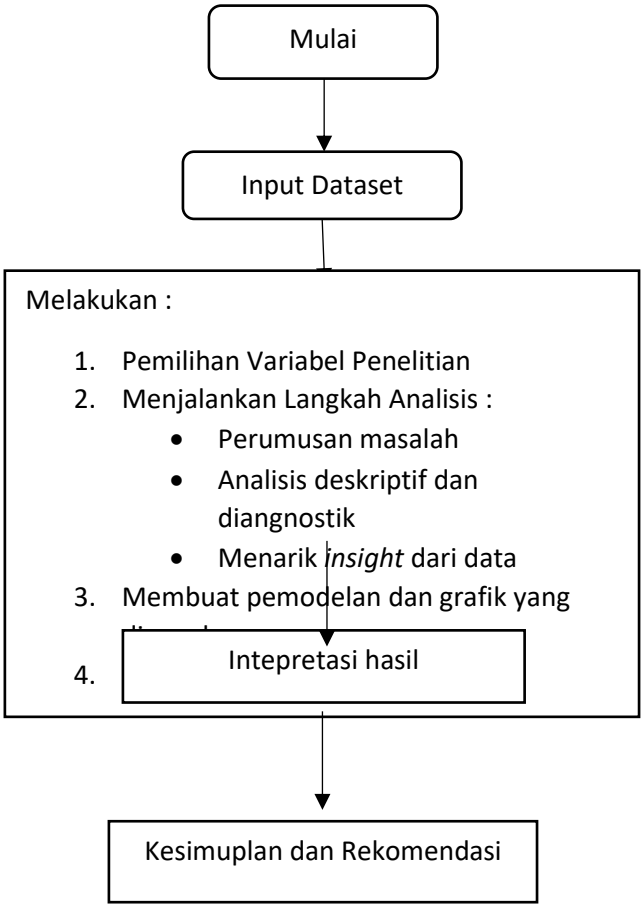
3.3 Langkah Analisis

Langkah analisis yang digunakan dalam praktikum ini yakni sebagai berikut :

1. Merumuskan masalah dan studi literatur terkait dataset 3 Google Play Store
2. Melakukan analisis deskriptif dan diagnostik dengan eksplorasi data menggunakan *software* R
3. Menarik insight dari data
4. Melakukan pemodelan dengan menggunakan metode Naïve Bayes, Random Forest dan K-Means
5. Menghitung ketepatan prediksi
6. Menarik kesimpulan dan rekomendasi.

3.4 Diagram Alir

Diagarm alir yang digunakan dalam penelitian ini adalah sebagai berikut.



BAB IV
HASIL ANALISIS DAN PEMBAHASAN

4.1 Hasil Analisis

4.1.1 Analisis deskriptif dan diagnostik

Sebelum melakukan analisis deskriptif, data yang akan digunakan harus disesuaikan terlebih dahulu. Seperti mengatasi apabila ada *missing value*, menghilangkan simbol-simbol pada data seperti '$', '+', 'M','k' serta penyesuaian tipe data. Dataset 3 Google Play Store memiliki 4434 missing value sehingga perlu diatasi. Data yang memiliki missing value merupakan data numeric maka cara mengatasi missing value adalah dengan nilai media dari masing-masing data tersebut.

```
> sum(is.na(datap))
[1] 4434
```

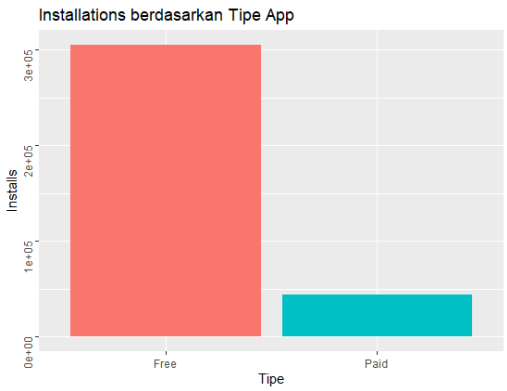
```
> sum(is.na(datap))
[1] 0
```

```
> str(datap)
'data.frame':   10041 obs. of  13 variables:
 $ App                : chr  "Learn 50 languages" "Rosetta Stone: Learn to Speak & Read New La
nguages" "Babbel â\200“ Learn Spanish" "Mango Languages: Lovable Language Courses" ...
 $ Kategori           : chr  "EDUCATION" "EDUCATION" "EDUCATION" "EDUCATION" ...
 $ Penilaian          : num  4.4 4.5 4.4 4 4.6 4.4 4.7 4.7 4.3 4.1 ...
 $ Ulasan             : num  55256 172508 54798 4815 75112 ...
 $ Ukuran             : num  14.01 76.08 11.01 19.02 6.51 ...
 $ Installs           : num  14 76 11 19 6.5 7 15 3.3 21 15 ...
 $ Tipe               : chr  "Free" "Free" "Free" "Free" ...
 $ Harga              : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Penilaian_Konten   : chr  "Everyone" "Everyone" "Everyone" "Everyone" "Everyone" ...
 $ Genres             : chr  "Education" "Education;Education" "Education" "Education" ...
 $ Terakhir_Diperbarui: chr  "19-Jun-18" "27-Jun-18" "30-Jul-18" "17-Jul-18" ...
 $ Versi_Sekarang     : chr  "10.9.1" "5.2.1" "20.7.2" "4.2.3" ...
 $ Versi_Android      : chr  "4.0 and up" "5.0 and up" "4.4 and up" "4.2 and up" ...
```

***Gambar 4. 1 Hasil setelah Mengatasi* Missing Value**

Data sudah bersih dan diberi nama "datap". Berikut adalah Analisis Deskriptif dan visualisasi dari dataset 3.

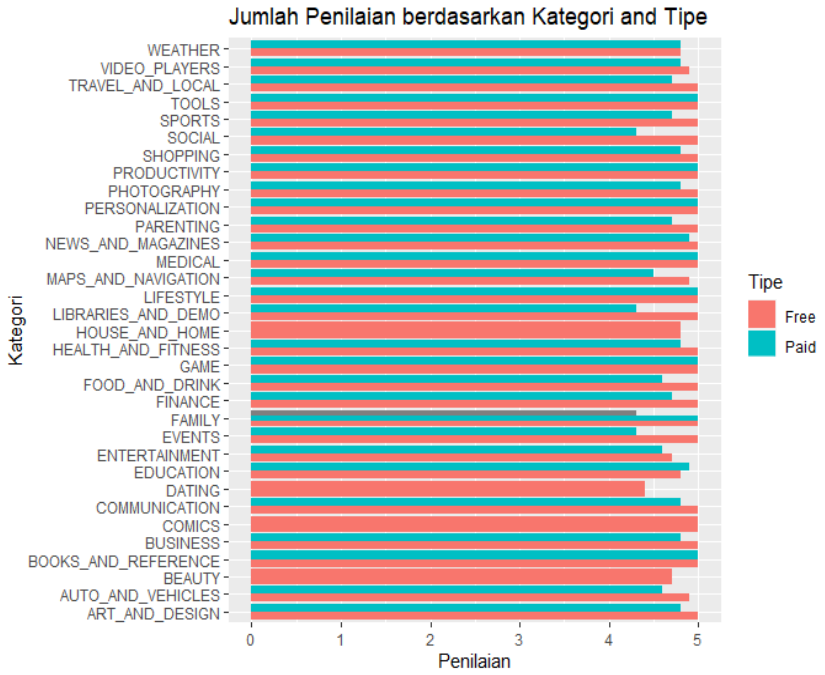
4.1.1.1 Bar chart Installs vs Tipe



Gambar 4. 2* Bar chart Installations *berdasarkan Tipe

Dari grafik tersebut, dapat dilihat bahwa aplikasi dengan tipe Free(gratis) jauh lebih banyak di install dari pada aplikasi bertipe Paid (berbayar).

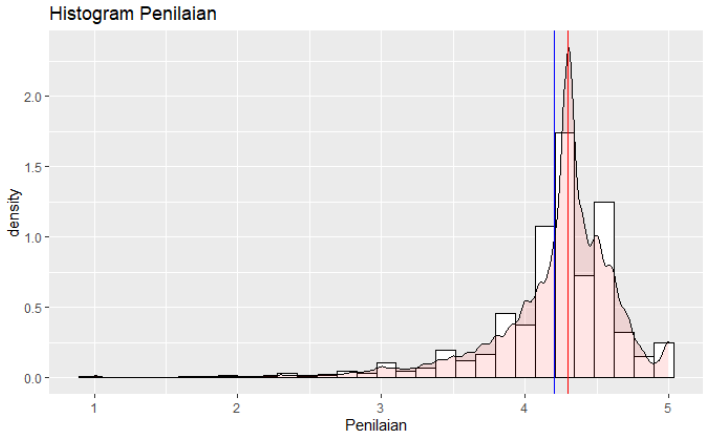
4.1.1.2 Bar chart Jumlah Rating (Penilaian) vs Kategori vs Tipe



Gambar 4. 3 *Bar Chart* Jumlah Rating Vs Kategori Vs Tipe

Berdasarkan grafik diatas, terdapat aplikasi yang hanya bertipe free yaitu dengan aplikasi berkategori House_and_home, Dating, Comics dan Beauty. Mayoritas aplikasi bertipe free memiliki penilaian maksimum yaitu pada angka 5. Sedangkan aplikasi bertipe paid sedikit memiliki rating (penilaian) 5.

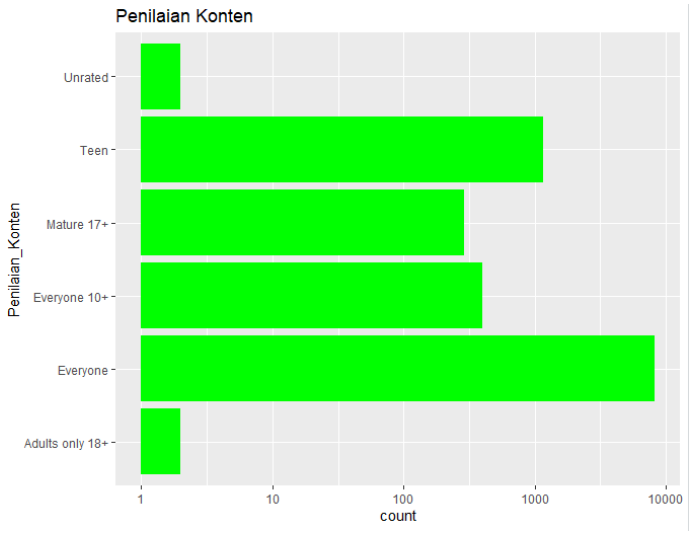
4.1.1.3 Histogram Penilaian



Gambar 4. 4 Histogram Penilaian

Berdasarkan histogram tersebut, dapat dilihat bahwa variabel Penilaian memiliki mean (garis biru) dan median (garis merah) yang berdekatan hal ini dapat disimpulkan bahwa data tersebut simetris. Terlihat pula, data ini memiliki outlier di Penilaian 1.

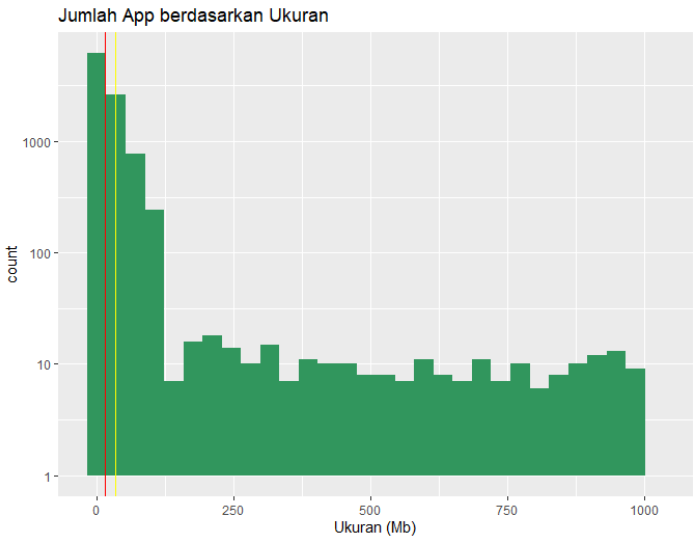
4.1.1.4 Count Plot Penilaian Konten



Gambar 4. 5* Count Plot *Penilaian Konten

Berdasarkan grafik diatas, aplikasi yang terdapat pada pasar Google Play Store kebanyakaan mendapatkan penilaian konten *'Everyone'* yang berarti aplikasi – aplikasi ini bisa digunakan oleh semua orang tanpa ada batasan umur.

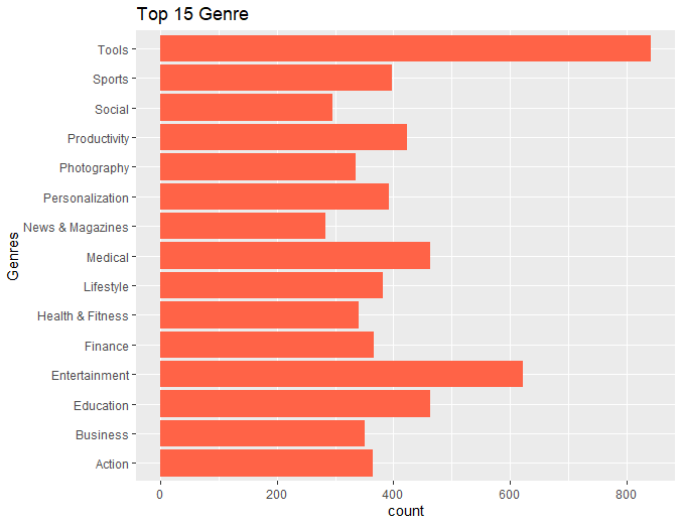
4.1.1.5 Histogram Jumlah Aplikasi Berdasarkan Ukuran



Gambar 4. 6 Jumlah Aplikasi Berdasarkan Ukuran

Grafik di atas menggambarkan bahwa mayoritas aplikasi berukuran dibawah 250 MegaByte. Dimana rata-rata ukuran aplikasi google play store adalah 34.769 Mb yang ditandai dengan garis kuning. Dan mediannya bernilai 15.015 Mb.

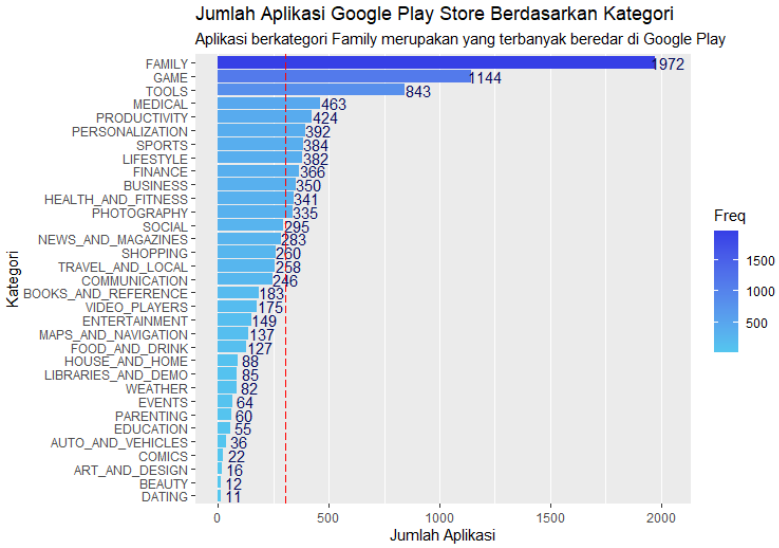
4.1.1.6 Bar Chart Genre Top 15



***Gambar 4. 7* Bar Chart Genre Top 15**

Dari grafik di atas, dapat dilihat bahwa Genre tertinggi dari daftar top 15 Genre yang diunduh oleh Android *users* adalah genre tools. Dimana pengunduh dari genre *tools* sendiri berada diangka lebih dari 800 pengunduh.

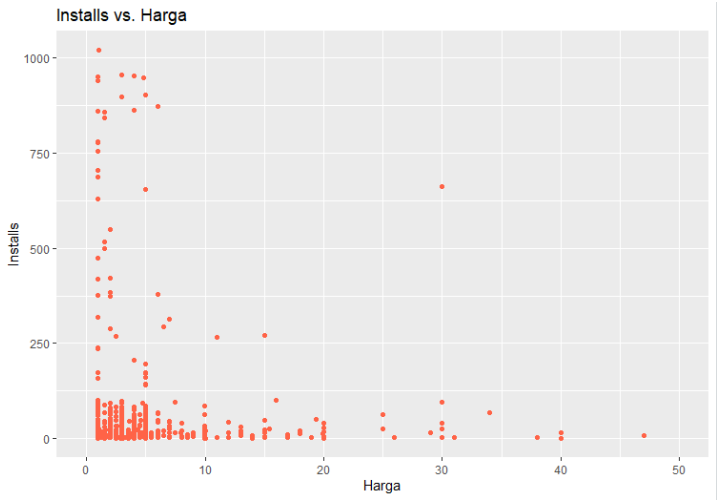
4.1.1.7 *Bar Chart* jumlah Aplikasi berdasarkan Kategori



Gambar 4. 8* Bar Chart *Jumlah Aplikasi Berdasarkan Kategori

Pada grafik di atas, dapat dilihat bahwa kategori teratas dari aplikasi yang ada di Google Play Store berdasarkan jumlah aplikasinya adalah kategori *family* paling banyak, dengan total angka 1972 aplikasi yang ada dapat diunduh melalui pasar Google Play Store.

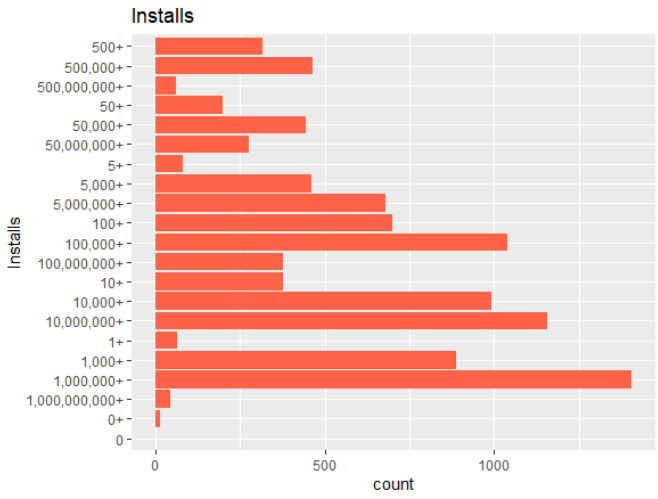
4.1.1.8 *Scatter plot Install* vs Harga



Gambar 4. 9* Scatter Plot Install *Vs Harga

Berdasarkan *scatter plot* pada gambar di atas, dapat dilihat bahwa Android *users* lebih suka mengunduh aplikasi yang tidak berbayar atau *free*. Hal tersebut dibuktikan dengan titik pada *scatter plot* yang menunjukan bahwa lebih banyak titik yang berada di kolom harga 0 Rupiah. Untuk *installs* nya sendiri ada yang mencapai lebih dari 1000 *installs* di aplikasi dengan harga 0 Rupiah.

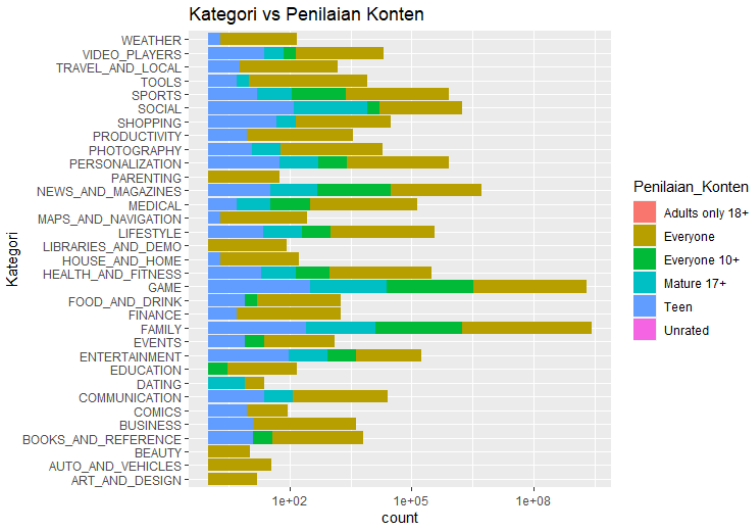
4.1.1.9 Count Plot Installs



Gambar 4. 10 Count Plot Installs

Dari grafik *count plot* di atas dapat dilihat bahwa aplikasi yang diunduh sebnayak 1.000.000 unduhan ada lebih dari 1000 aplikasi. Hal tersebut menyatakan bahwa kebanyakan aplikasi yang ada di Google Play Store diunduh sebanyak 1.000.000 kali.

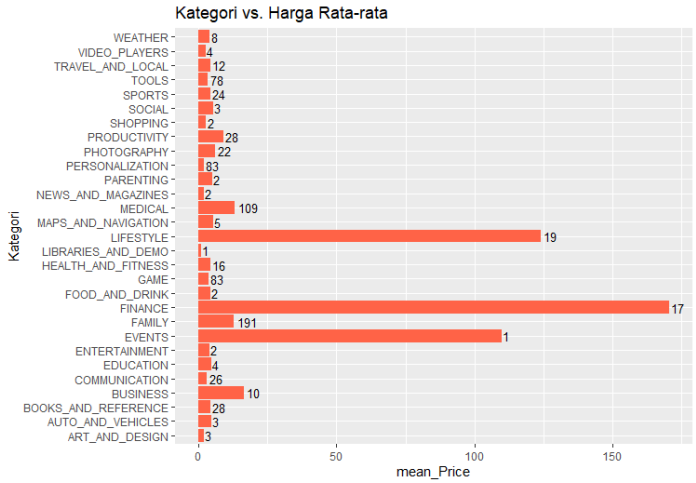
4.1.1.10 *Count Plot* Kategori Vs Penilaian Konten



Gambar 4. 11 *Count Plot* Kategori Vs Penilaian Konten

Satu-satunya aplikasi yang memiliki peringkat konten Dewasa saja ada dikategori comics. Setiap kategori memiliki lebih dari satu aplikasi yang memiliki penilaian konten Everyone.

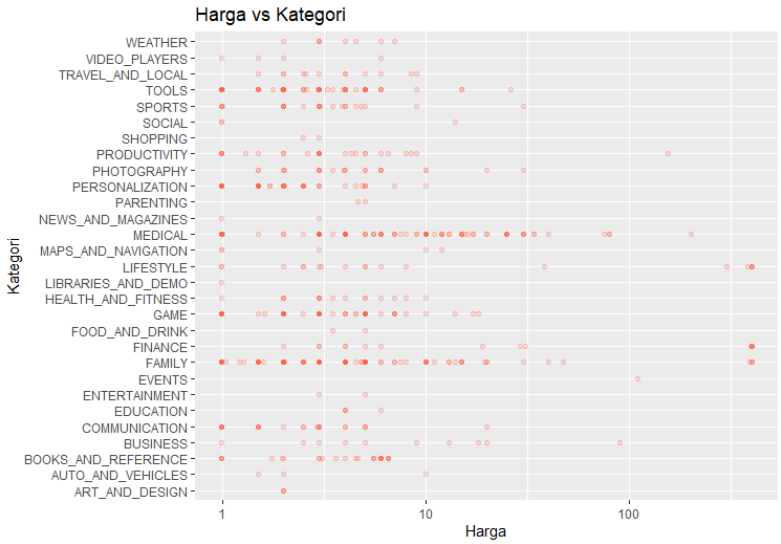
4.1.1.11 Bar chart jumlah Aplikasi berdasarkan Kategori Vs Harga Rata - rata



Gambar 4. 12* Bar Chart *Kategori Vs Harga Rata - Rata

Berdasarkan gambar grafik di atas, dapat diketahui bahwa kategori dengan harga rata – rata termahal adalah aplikasi dengan kategori *finance*. Kategori dengan harga rata – rata termahal yang kedua adalah aplikasi yang berkategori *lifestyle.* Sisanya, kebanyaka aplikasi yang ada di Google Play Store memiliki harga rata – rata 0 Rupiah (*free).*

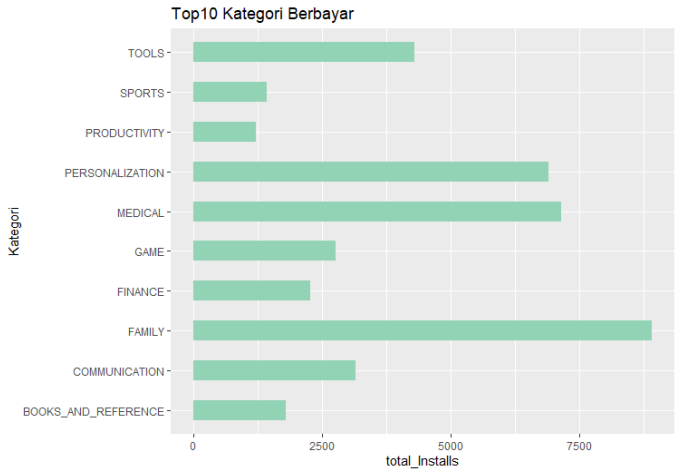
4.1.1.12 Scatter plot Harga Vs Kategori



Gambar 4. 13* Scatter Plot *Harga Vs Kategori

Pada gambar grafik di atas, dapat diketahui bahwa aplikasi dengan harga tertinggi yang terdapat pada Google Play Store adalah mayoritas aplikasi yang berkategori *finance* dan *family*, yaitu seharga lebih dari $100,00.

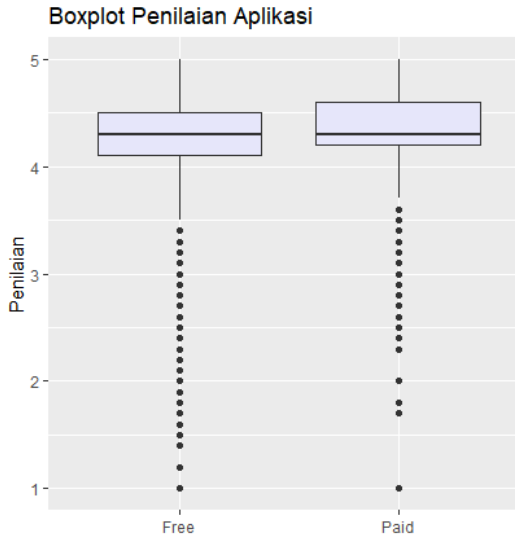
4.1.1.13 Count Plot Top 10 Kategori Vs Total Installs



Gambar 4. 14* Count Plot Top 10 *Kategori Berbayar

Dari hasil *plotting*an pada *count plot* di atas, aplikasi yang menduduki tingkatan teratas adalah aplikasi yang berkategorikan *family*. Dengan total jumah pengunduh lebih dari 7.500 pengunduh, dan hamper menyentuh angka *installs* 10.000 pengunduh.

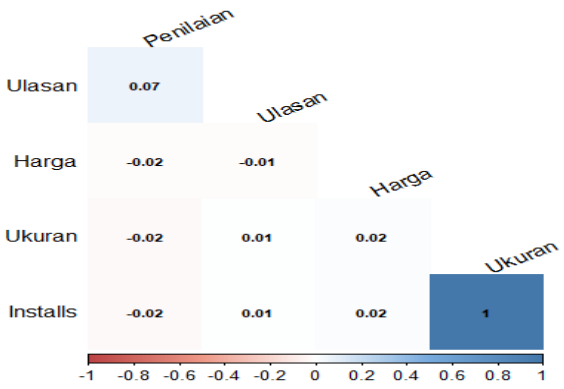
4.1.1.14 *Box Plot* Penilaian Aplikasi



Gambar 4. 15 Box Plot *Penilaian Aplikasi*

Pada hasil *box plot* di atas, dapat diketahui bahwa aplikasi bertope free lebih banyak dibandingkan aplikasi yang bertipe *paid.* Namun dapat dilihat juga dari grafik di atas bahwa untuk penilaian aplikasinya sendiri, aplikasi dengan tipe *paid* atau berbayar mendapatkan penilaian lebih tinggi yaitu berada di kisaran nilai 3.80 sampai dengan 5.00 dibandingkan dengan aplikasi gratis. Dan terlihat pula aplikasi berbayar memiliki outlier yaitu ada aplikasi yang memiliki penilaian pada nilai 1.

4.1.1.15 *Correlation Plot* Penilaian



```
> M
            Penilaian      Ulasan       Ukuran     Installs       Harga
Penilaian  1.00000000  0.069332773 -0.021471723 -0.021471723 -0.01952169
Ulasan     0.06933277  1.000000000  0.006174563  0.006174563 -0.01012649
Ukuran    -0.02147172  0.006174563  1.000000000  1.000000000  0.01581766
Installs  -0.02147172  0.006174563  1.000000000  1.000000000  0.01581766
Harga     -0.01952169 -0.010126495  0.015817659  0.015817659  1.00000000
```

Gambar 4. 16* Correlation Plot *Penilaian

Berdasarkan hasil *Correlation Plot* di atas, variabel ukuran dan variabel install bernilai 1 yang artinya kedua variabel tersebut berkorelasi positif atau berkorelasi kuat.

4.1.1.16 Korelasi Variabel Kategori

```
> chisq.test(datap$Kategori, datap$Penilaian_Konten, correct=FALSE)

        Pearson's Chi-squared test

data:  datap$Kategori and datap$Penilaian_Konten
X-squared = 3315.7, df = 160, p-value < 2.2e-16

> chisq.test(datap$Genres, datap$Penilaian_Konten, correct=FALSE)

        Pearson's Chi-squared test

data:  datap$Genres and datap$Penilaian_Konten
X-squared = 5138.9, df = 585, p-value < 2.2e-16
```

Berdasarkan hasil perhitungan Pearson tersebut, dapat dilihat bahwa variabel Kategori dengan Penilaian Konten saling independent begitu pula variabel Genre dan Penilaian Konten juga saling independent

4.2 Pemodelan dan Pembahasan

Pemodelan yang digunakan pada praktikum ini adalah menggunakan supervised learning method yaitu klasifikasi dan regresi. Metode supervised yang digunakan yaitu metode naïve bayes dan random forest pada variabel Penilaian. Serta menggunakan unsupervised learning yaitu clustering dengan K-Means. Variabel penilaian dibagi menjadi dua level yaitu high dan low dimana penulis mendefinisikan high bernilai antara 3.5-5 dan low bernilai antara 1-3.5.

Dilakukan partisi data menjadi data training dan data testing terlebih dahulu. Diambil 70% untuk data training dan 30% untuk data testing. Karena data imbalanced, maka dilakukan resampling data menggunakan over sampling.

```
> #select 70% of the data for training
> train<-dataapps[index1,]
> dim(train)
[1] 7030   13
> #use the remaining to testing the models
> test<-dataapps[-index1,]
> dim(test)
[1] 3011   13
```

```
> table(train$Penilaian)

 Low High
 581 5442

> library(ROSE)
> oversample<-sample(Lower,length(High),replace=TRUE)
> over<-training[c(oversample,High),]
> table(over$Penilaian)

 Low High
5442 5442
```

4.2.1 Hasil Pemodelan

4.2.1.1 Algoritma Naïve Bayes

Metode Naïve Bayes digunakan karena dapat menghasilkan akurasi yang maksimal dengan data latih (*training data)* yang sedikit untuk menentukan estimasi paremeter yang diperlukan dalam proses pengklasifikasian.

```
> classifier_o <- naiveBayes(as.factor(Penilaian) ~ ., data = over)
> y_predtest<-predict(classifier_o, over)
> cm<-table(y_predtest,over$Penilaian)
> confusionMatrix(cm)

          Confusion Matrix and Statistics
          y_predtest  Low High
               Low  5123 3994
               High  319 1448

                    Accuracy : 0.6037
                      95% CI : (0.5945, 0.6129)
         No Information Rate : 0.5
         P-Value [Acc > NIR] : < 2.2e-16

                     Kappa : 0.2075

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

                  Sensitivity : 0.9414
```

```
                Specificity : 0.2661
             Pos Pred Value : 0.5619
             Neg Pred Value : 0.8195
                 Prevalence : 0.5000
             Detection Rate : 0.4707
       Detection Prevalence : 0.8377
          Balanced Accuracy : 0.6037

           'Positive' Class : Low

> y_predtest2<-predict(classifier_o, test)
> cm2<-table(y_predtest2,test$Penilaian)
> confusionMatrix(cm2)
          Confusion Matrix and Statistics
          y_predtest2  Low High
               Low    186 1084
               High    62 1248

                Accuracy : 0.5558
                  95% CI : (0.5364, 0.5751)
     No Information Rate : 0.9039
     P-Value [Acc > NIR] : 1

                   Kappa : 0.1004

          Mcnemar's Test P-Value : <2e-16

             Sensitivity : 0.75000
             Specificity : 0.53516
          Pos Pred Value : 0.14646
          Neg Pred Value : 0.95267
              Prevalence : 0.09612
```

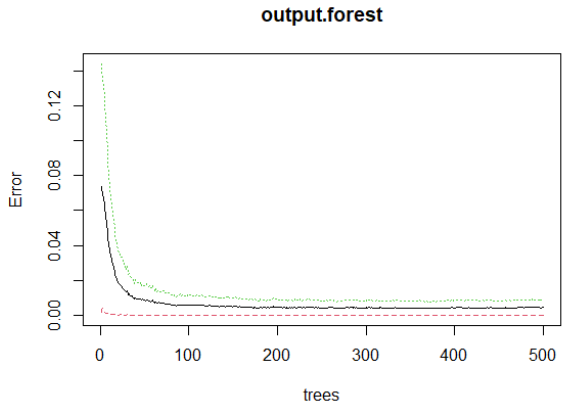
```
                Detection Rate : 0.07209
          Detection Prevalence : 0.49225
             Balanced Accuracy : 0.64258

              'Positive' Class : Low
```

4.2.1.2 Algoritma Random Forest

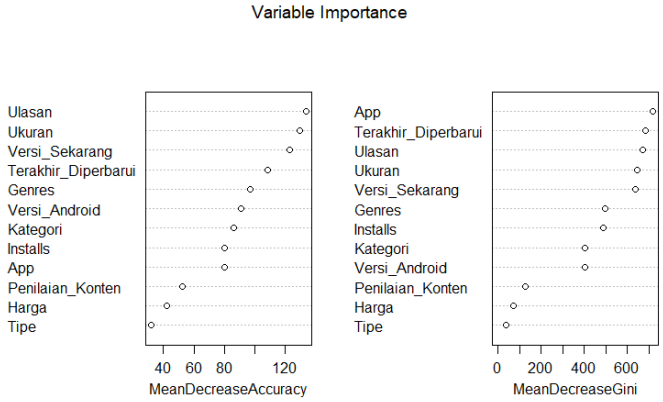
Menggunakan Random Forest karena data yang digunakan dalam penelitian ini tidak seimbang sehingga menggunakan seleksi input yang random. Serta lebih cocok untuk pengklasifikasian data serta dapat digunakan untuk menangani data sampel yang banyak.

```
> print(output.forest)
        Call:
         randomForest(formula = Penilaian ~ ., data = rating.over,
              importance = TRUE)
                    Type of random forest: classification
                          Number of trees: 500
        No. of variables tried at each split: 3

               OOB estimate of  error rate: 0.45%
        Confusion matrix:
             Low High class.error
        Low  5441    1 0.000183756
        High    1 5441 0.000183756

        attributes(output.forest)
        $names
         [1] "call"          "type"          "predicted"       "err.rate"
         [5] "confusion"     "votes"         "oob.times"       "classes"
         [9]  "importance"         "importanceSD"      "localImportance"
              "proximity"
        [13] "ntree"         "mtry"          "forest"         "y"
        [17] "test"          "inbag"         "terms"
```

```
$class
[1] "randomForest.formula" "randomForest"
```



Gambar 4. 17 Output Random Forest



Gambar 4. 18 Importance Variable

Dari grafik importance variabel tersebut dapat dilihat

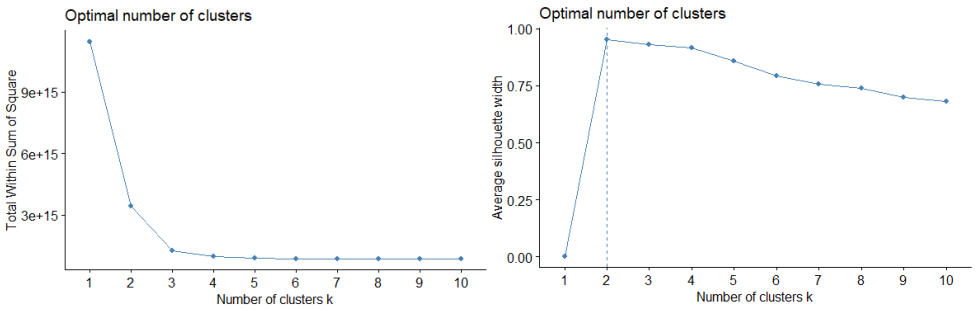
1. Menurut Mean Decrease Accuracy(MDA)

   Variabel tersebut penting apabila MDA semakin tinggi. Urutan variabel dari paling penting yang digunakan pada penelitian ini untuk memprediksi variabel Penilaian adalah Ulasan, ukuran, Installs, Harga.
2. Menurut Mean Decrease Gini (MDG)

   Variabel tersebut penting apabila MDG semakin tinggi. Urutan variabel dari paling penting yang digunakan pada penelitian ini untuk memprediksi variabel Penilaian adalah ulasan, ukuran, installs, dan harga.

4.2.1.3 Algoritma K-Means

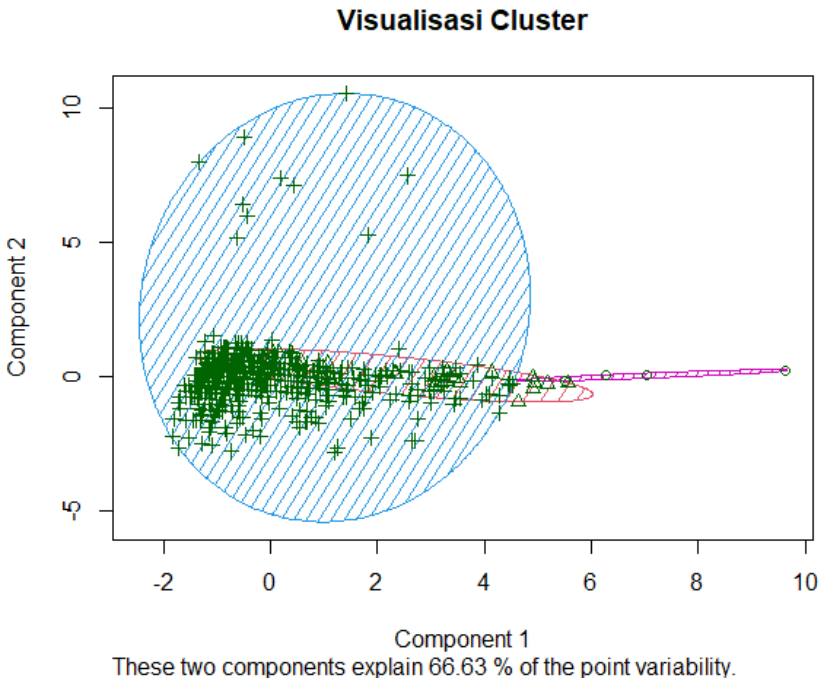
Algoritma ini digunakan karena memiliki kelebihan yaitu, mudah diimplementasikan dan mampu mengelompokkan data yang besar dengan waktu komputasi yang cepat dan efisien serta dapat mengelompokkan data berdasarkan kemiripannya. Dikarenakan data berdistribusi tidak normal, maka dilakukan reduksi baris menjadi 1000.



(a) Metode Elbow (b)Metode Shilouette

Gambar 4.19 Penentuan Nilai k Optimal

Sebelum menjalankan model K-Means, nilai k yang terbaik ditentukan dulu dengan metode elbow dan metode silhouette. Metode elbow menguji setiap nilai K sampai terlihat garis siku. Pengujian K dilakukan dari 1 sampai 10. Pada metode elbow gambar 4.19(a), nilai K ke-3 terdapat garis siku, itulah nilai k yang terbaik. Pada metode Silhouette gambar 4.19(b), nilai k ke-2 merupakan k optimal. Penulis menggunakan k=3.



Gambar 4. 20 Visualisasi Cluster

Dengan penentuan nilai K adalah 3 maka akan ada 3 cluster yang terbagi. Hasil cluster dalam bentuk scatter plot bisa dilihat pada gambar 4.20. Berikut adalah centroid atau rata-rata dari masing-masing cluster yang terbentuk.

```
>k.means.fit$centers
  Penilaian     Ulasan   Ukuran  Installs     Harga
1  4.454717  7025687.3 46.17821 46.13208 0.00000000
2  4.510000 28650486.9 84.08400 84.00000 0.00000000
3  4.292850   170054.4 22.13774 22.11563 0.05056564
```

Pada cluster pertama, rata-rata dari semua atribut berada di antara cluster 2 dan cluster 3. Namun, nilai rata-rata Ulasan memiliki nilai yang lebih tinggi dibandingkan cluster 3.

Pada cluster kedua, semua atribut memilki nilai rata-rata yang terbesar kecuali pada atribut harga yang lebih rendah dari cluster ketiga. Namun , nilainya sama dengan nilai rata-rata cluster pertama. Hal ini menunjukan bahwa nilai dari cluster kedua dikelompokkan berdasarkan nilai yang terbesar.

Pada cluster ketiga, semua atribut memiliki nilai rata-rata yang terendah dari ketiga cluster. Hal ini berarti bahwa cluster ketiga dikelompokkan berdasarkan nilai yang terkecil.

Pada output centroid di atas, terdapat cluster kedua dengan ciri-ciri aplikasi yang tergolong ideal. Ciri-ciri tersebut adalah penilaian , Installs, ukuran dan ulasan yang tinggi dan juga harga aplikasi yang kecil.

```
>print(paste("silhouette score=",round(score,3)))
        [1] "silhouette score= 0.93"
```

Ketepatan prediksi dari metode K-Means ini ditunjukkan dengan silhouette score yang bernilai 93%

4.2.2 Perbandingan Ketepatan Prediksi Antar Metode

Setelah dilakukan pemodelan menggunakan metode-metode terebut, maka dilakukan perbandingan ketepatan prediksi berdasarkan nilai akurasinya. Dibawah ini tabel yang menunjukan confusion matriks dari

Tabel 4. 1 Confusion Matriks

Algoritma	**TP**	**FP**	**TN**	**FN**
Naïve Bayes	1248	62	186	1084
Random Forest	5441	1	5441	1

Tabel 4. 2 Perbandingan Hasil Ketepatan Prediksi

Algoritma	**Accuracy(%)**	**Recall(%)**	**Precision(%)**
Naïve Bayes	55.58	53.52	95.27
Random Forest	99.98	99.98	99.98

Sedangkan untuk K-Means memiliki silhouette score sebesar 93%. Kriteria untuk silhouette score ini adalah sebagai berikut. $0.7<SC\leq1$ adalah strong structure, $0.5<SC\leq0.7$ adalah mediun structure, $0.25<SC\leq0.5$ adalah weak structure, $SC\leq0.25$ adalah no structure. Maka berdasarkan Shilouette Score, ketepatan prediksi bersifat strong structure.

BAB V
KESIMPULAN DAN REKOMENDASI

5.1 Kesimpulan

Kesimpulan yang didapatkan dari hasil proses analisis deskriptif, diagnostik dan pemodelan menggunakan algoritma Naïve Bayes, Random Forest dan K-Means pada dataset 3 Google Play Store adalah sebagai berikut :

1. Ketepatan prediksi pada algoritma Naïve Bayes adalah *accuracy* sebesar 55.58%, *recall* sebesar 53.32%, dan *precision* sebesar 95.27%.
2. Ketepatan prediksi pada algoritma Random Forest memiliki *accuracy* sebesar 99.98%, *recall* sebesar 99.98%, dan *precision* sebesar 99.98%,
3. Ketepatan prediksi pada algoritma K-Means memiliki nilai shilouette sebesar 93%
4. Algoritma yang paling sesuai untuk memprediksi variabel Penilaian aplikasi Google Play Store dalam kriteria High dan Low berdasarkan variabel Harga, Ulasan, Ukuran, dan Installs adalah algoritma Random Forest yaitu dengan ketepatan prediksi sebesar 99.98%

5.2 Rekomendasi

Berdasarkan hasil analisis yang telah dilakukan, rekomendasi yang disarankan oleh penulis, antara lain:

1. Pada penelitian selanjutnya dapat menggunakan algoritma lain dan menggunakan metode validasi yang berbeda sehingga dapat membandingkan ketepatan prediksinya agar memaksimalkan *performance* dan mengurangi error dalam pemodelan.
2. Untuk *developer* dari aplikasi pada Google Play Store bisa lebih meningkatkan lagi terkait sistem pendataan agar kededpannya mendapatkan data penilaian yang lengkap sehingga ketepatan analisis menjadi lebih baik lagi. Dan agar dapat bersaing dengan aplikasi lainnya maka, Harga dan Ukuran aplikasi dapat diminimalisir sehingga semakin banyak *user* yang mengunduh aplikasi tersebut.

DAFTAR PUSTAKA

Breiman L (2001). "*Random Forests". Machine Learning. 45 (1): 5–32*. doi:1023/A:1010933404324.

Effendi, Jannes, and M. Jorgi Ramadhan. "Analisa Cluster Aplikasi pada Google Play Store dengan Menggunakan Metode K-Means." *Annual Research Seminar (ARS)*. Vol. 4. No. 1. 2019.

De Oliveira, MC Ferreira, dan Haim Levkowitz. "From visual data exploration to visual data mining: A survey." *IEEE transactions on visualization and computer graphics* 9.3 (2003): 378-394.

G. K. Bhattacharya dan A. R. Johnson. 2009. *Statistics; Principles and Methods, 6th ed.* United States: John Wiley & Sons, Inc.

Ho, Tin Kam. 1995. *Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal.

J. L. Hintze dan R. D. Nelson.1998. *Violin Plots: A Box Plot- Density Trace Synergis.* America: The American Statistician.

L. Breiman.2001. *Random Forests, Machine Learning.*

Nugroho, Yuda Septian. 2014. *Data Mining Menggunakan Algoritma Naive Bayes Untuk Klasifikasi Kelulusan Mahasiswa Universitas Dian Nuswantoro.* Dian Nuswantoro Fakultas Ilmu Komputer Skripsi.

R. E. Walpole, R. H. Myers, S. L. Myers dan K. Ye. 2016. *Probability & Statistics for Engineers & Scientists, 9th ed.* United States: Person Education.

R. W. Pratiwi.2016. *Prediksi Rating Film Menggunakan Metode Naïve Bayes.*

LAMPIRAN

Awal dan Visualisasi data

```
setwd("D:/A KULIAH KVN/6. BA KMMI/project")

datap=read.csv("D:/A KULIAH KVN/6. BA KMMI/project/Dataset_3.csv",dec=".")


library(ggplot2)

library(dplyr)

datap$Penilaian=as.numeric(datap$Penilaian)

datap$Ulasan=as.numeric(datap$Ulasan)

datap$Installs=gsub("+","",as.character(datap$Ukuran))

datap$Installs=as.numeric(datap$Installs)

datap$Harga=gsub("$","",as.character(datap$Harga))

datap$Harga=as.numeric(gsub("\\$","",datap$Harga))

harga=datap$Harga

harga[3581]

library(tidyverse)

library(stringr)

mega_bytes <- as.numeric(str_remove_all(datap$Ukuran, "M"))

mega_bytes[is.na(mega_bytes)] <- 0

kilo_bytes <- as.numeric(str_remove_all(datap$Ukuran, "k")) / 1000

kilo_bytes[is.na(kilo_bytes)] <- 0

size_bytes <- kilo_bytes + mega_bytes

size_bytes[size_bytes==0] <- NA

summary(size_bytes)
```

```
datap$Ukuran <- size_bytes
datap$Ukuran=as.numeric(datap$Ukuran)

#statdes dan check missing value
summary(datap)
sum(is.na(datap))
#mengatasi missing value dgn median
datap$Penilaian[is.na(datap$Penilaian)]=median(datap$Penilaian,na.rm=TRUE)
datap$Ulasan[is.na(datap$Ulasan)]=median(datap$Ulasan,na.rm=TRUE)
datap$Harga[is.na(datap$Harga)]=median(datap$Harga,na.rm=TRUE)
datap$Ukuran[is.na(datap$Ukuran)]=median(datap$Ukuran,na.rm=TRUE)
datap$Installs[is.na(datap$Installs)]=median(datap$Installs,na.rm=TRUE)

sum(is.na(datap))
summary(datap)
str(datap)
datap <- subset(datap, Kategori != "1.9")

#EDA
datap$Tipe[datap$Tipe=="0"]<-NA
datap$Tipe[datap$Tipe=="NaN"] <- NA
a =datap %>% select(Tipe, Installs,Kategori) %>% filter(Tipe!=c("NA")) %>%
  group_by(Kategori) %>% arrange(Kategori)
```

```
ggplot(a, aes(x=Tipe, y=Installs, fill=Tipe))+
  geom_bar(stat="identity")+
  labs(x="Tipe",y="Installs",fill="Types",title="Installations berdasarkan Tipe App")+
  theme(legend.position = "None",axis.text.y = element_text(angle = 90))

b=datap%>%select(Penilaian, Kategori, Tipe)%>% filter(Kategori!="1.9")
ggplot(b, aes(x=Kategori, y=Penilaian, fill = Tipe)) +
  geom_bar(position='dodge',stat='identity') +
  coord_flip()+
  ggtitle("Jumlah Penilaian berdasarkan Kategori and Tipe")

#histogram Penilaian
med = median(subset(datap$Penilaian, datap$Penilaian >= 0.01))
mean = mean(subset(datap$Penilaian, datap$Penilaian >= 0.01))
ggplot(aes(x =Penilaian), data = datap )+
geom_histogram(aes(y=..density..),colour="black", fill="white")+
  geom_density(alpha=0.1, fill="red")+
  ggtitle('Histogram Penilaian')+
  geom_vline(xintercept = med, col = 'red')+
  geom_vline(xintercept = mean, col = 'blue')

#Jumlah Content Rating
c=datap%>%select(Penilaian_Konten)%>% filter(Penilaian_Konten!="")
```

```
ggplot(c, aes(x = Penilaian_Konten))+
  geom_bar(fill = 'green')+
  coord_flip()+
  scale_y_log10()+
  ggtitle('Penilaian Konten')

#Ukuran
d=datap
ggplot(d,aes(x = round(Ukuran)))+
  geom_histogram(fun.y = count, geom ='line', fill = '#31965D')+
  geom_vline(xintercept = median(subset(datap,!is.na(d$Ukuran))$Ukuran), col =
'red')+
  geom_vline(xintercept = mean(subset(datap,!is.na(d$Ukuran))$Ukuran), col =
'yellow')+
  ggtitle('Jumlah App berdasarkan Ukuran')+scale_y_log10()+
  xlab('Ukuran (Mb)')

#jumlah berdasarkan genre
topgenres = group_by(datap, Genres)%>%
  summarise(n = n())%>%
  arrange(desc(n))
topgenres = head(topgenres,15)
m= datap$Genres %in% topgenres$Genres
topgenres = datap[m,]
```

```
ggplot(aes(x = Genres), data = topgenres)+
  geom_bar(fill = 'tomato')+
  coord_flip()+
  ggtitle('Top 15 Genre')

#Jumlah Aplikasi terbanyak berdasar kategori
temp1 <- as.data.frame(table(datap$Kategori))
g1 <- ggplot(temp1, mapping = aes(x=reorder(Var1,Freq), y=Freq, fill=Freq))+
  geom_col()+
  scale_fill_gradient(high = "#363fe6",low = "#54C7EF" )+
  coord_flip()+
  geom_text(aes(label = temp1$Freq),nudge_y = 60,col = "#040b5b")+
  geom_hline(yintercept = mean(temp1$Freq), linetype = 5, col = "Red")+
  labs(title="Jumlah Aplikasi Google Play Store Berdasarkan Kategori",subtitle =
"Aplikasi berkategori Family merupakan yang terbanyak beredar di Google Play",
       x="Kategori", y="Jumlah Aplikasi")+
  theme(legend.position = "right",panel.grid.major.y = element_blank())
g1

#Instal vs Harga
ggplot(aes(x = Installs, y = Harga), data = subset(datap, Tipe == 'Paid'))+
  geom_jitter(alpha = 1, , color = 'tomato')+
  coord_flip(ylim = c(0,50))+
```

```
  ggtitle('Installs vs. Harga')


#kategory vs konten rating
e=subset(datap, Kategori != '1.9')
ggplot(e,aes(x = Kategori))+
  geom_bar(aes(fill = Penilaian_Konten))+
  coord_flip()+
  scale_y_log10()+
  ggtitle('Kategori vs Penilaian Konten')


#subsetting for Type
paidapp <- subset(datap, Tipe == "Paid")
#group the apps so we can get the mean, median and number of the price
paidappgroup <- paidapp%>%
  group_by(Kategori)%>%
  summarise(mean_Price = mean(Harga), n = n(), median_Harga = median(Harga))
#Kategori vs. Harga Rata-rata
ggplot(aes(x =Kategori, y =mean_Price ), data = paidappgroup)+
  geom_bar(stat = 'identity', position = 'dodge', fill = 'tomato')+
  coord_flip()+
  geom_text(aes(label = n), hjust=-0.17,size=3.5)+
  ggtitle('Kategori vs. Harga Rata-rata')
```

```
#Category vs price
ggplot(aes(y = Harga, x = Kategori), data = paidapp)+
  geom_point(alpha = 0.2, color = 'tomato')+
  coord_flip()+
  scale_y_log10()+
  ggtitle('Harga vs Kategori')

#top 10 paid
f=datap%>% filter(Tipe == "Paid") %>%
  group_by(Kategori) %>% summarize(total_Installs = sum(Installs)) %>%
  arrange(desc(total_Installs)) %>% head(10) %>%
  ggplot(aes(x = Kategori, y = total_Installs)) +
  geom_bar(stat="identity", width=.5, fill="#91D3B4") + labs(title= "Top10 Kategori
Berbayar" ) +
  coord_flip()
f

#boxplot
g <- datap[-which(datap$Kategori=='1.9'),]
ggplot(datap, aes(x=Tipe,y=Penilaian)) +
  geom_boxplot(fill="lavender") +
  ggtitle("Boxplot Penilaian Aplikasi") +
  ylab("Penilaian")+xlab("")
```

```
library(highcharter)
hcboxplot(x = datap$Ukuran, var = datap$Tipe, yAxis.bottom = 0, outliers = TRUE,
color = "#fb4901", fillColor = "lightblue") %>%
  hc_chart(type = "column") %>%
  hc_add_theme(hc_theme_ffx()) %>%
  hc_title(text = "Ukuran aplikasi (MB) berdasarkan Jenis Aplikasi")

z=read.csv("D:/A KULIAH KVN/6. BA KMMI/project/Dataset_3.csv",dec=".")
j = subset(z, z$Installs != 'Free')
#plotting a bar graph for level of installs.
ggplot(aes(x = Installs), data = j )+
  geom_bar(fill = 'tomato')+
  coord_flip()+
  ggtitle('Installs')



#coreleation plot
databaru=datap
databaru$Kategori=as.numeric(databaru$Kategori)
databaru$Tipe=as.numeric(databaru$Tipe)
databaru$Genres=as.numeric(databaru$Genres)
df<-datap %>% select (`Penilaian`,'Ulasan','Ukuran','Installs','Harga')
df<-na.omit(df)
```

```
M<-cor(df)
library(corrplot)
library(RColorBrewer)
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD",
"#4477AA"))
corrplot(M, method = "color", col = col(200),
         type = "lower", order = "hclust", number.cex = .7,
         addCoef.col = "black",
         tl.col = "black", tl.srt = 30,
         sig.level = 0.01, insig = "blank",
         diag = FALSE)
M
```

Pemodelan

1. Naïve bayes

```
dataapps=read.csv("D:/A KULIAH KVN/6. BA
KMMI/project/Dataset_3.csv",dec=".")
Penilaian<-
cut(dataapps$Penilaian,breaks=c(1,3.5,5),labels=c("Low","High"),right =
TRUE)
dataapps$Penilaian=Penilaian
#2. naive bayes
# Create Data Partition
library(caret)
library(ROSE)
library(dplyr)
library(randomForest)
```

```
library(e1071)
index1<-createDataPartition(dataapps$Penilaian,p=0.70,list=FALSE)

#select 70% of the data for training
train<-dataapps[index1,]
dim(train)
#use the remaining to testing the models
test<-dataapps[-index1,]
dim(test)
#cek proporsi data training
table(train$Penilaian)
High<-which(train$Penilaian=="High")
Lower<-which(train$Penilaian=="Low")
#Cek Proporsi Data
prop.table(table(Penilaian))

library(ROSE)
oversample<-sample(Lower,length(High),replace=TRUE)
over<-training[c(oversample,High),]
table(over$Penilaian)

#pemodelan
set.seed(123)
classifier_o <- naiveBayes(as.factor(Penilaian) ~ ., data = over)
y_predtest<-predict(classifier_o, over)
cm<-table(y_predtest,over$Penilaian)
confusionMatrix(cm)
y_predtest2<-predict(classifier_o, test)
cm2<-table(y_predtest2,test$Penilaian)
confusionMatrix(cm2)

y_predtest<-predict(classifier_o,newdata=test)
```

```
cm <- confusionMatrix(as.factor(y_predtest),as.factor(test$Penilaian), mode=
"prec_recall")
cm
```

2. Random Forest

```
dataapps=read.csv("D:/A KULIAH KVN/6. BA
KMMI/project/Dataset_3.csv",dec=".")

Penilaian<-
cut(dataapps$Penilaian,breaks=c(1,3.5,5),labels=c("Low","High"),right =
TRUE)
dataapps$Penilaian<-Penilaian
library(caret)
library(ROSE)
library(dplyr)
library(randomForest)
library(e1071)
index1<-createDataPartition(dataapps$Penilaian,p=0.70,list=FALSE)

#select 70% of the data for training
train<-dataapps[index1,]
dim(train)

#use the remaining to testing the models
test<-dataapps[-index1,]
dim(test)
High<-which(train$Penilaian=="High")
Lower<-which(train$Penilaian=="Low")
length(High)
length(Lower)
#Cek Proporsi Data
prop.table(table(Penilaian))
```

```
library(ROSE)
oversample<-sample(Lower,length(High),replace=TRUE)
over<-train[c(oversample,High),]
table(over$Penilaian)
##Random Forest
library(party)
library(randomForest)
library(caret)
library(e1071)

# Create the forest.
output.forest <- randomForest(Penilaian ~ .,
                        data = over,importance = TRUE)
print(output.forest)
attributes(output.forest)

# Predicting on train set
predTrain <- predict(output.forest, over, type = "class")
# Checking classification accuracy
table(predTrain, over$Penilaian)
confusionMatrix(predTrain,over$Penilaian)

# Predicting on Test set
predValid <- predict(output.forest, test, type = "class")
confusionMatrix(predValid,test$Penilaian)

plot(output.forest)

# Checking classification accuracy
mean(predValid == test$Penilaian)
table(predValid,test$Penilaian)
```

```
# To check important variables
importance(output.forest)
varImpPlot(output.forest,
        main="Variable Importance")
```

3. Kmeans

```
datap=datap[,c(3:6,8)]
datap=head(datap,1000)
#menentukan k optimum
library(factoextra)
fviz_nbclust(datap,kmeans,method="wss")
fviz_nbclust(datap,kmeans,method="silhouette")

#algoritma clustering
set.seed(123)
k.means.fit=kmeans(datap,iter.max=1000,3)
k.means.fit
#sentroid
k.means.fit$centers
#banyak iterasi mencapai optimum
k.means.fit$iter

#menghitung silhouette score
library(cluster)
jarak=as.matrix(dist(datap))
score=mean(silhouette(k.means.fit$cluster,dmatrix=jarak)[,3])
print(paste("silhouette score=",round(score,3)))
k.means.fit$cluster
silhouette(k.means.fit$cluster,dmatrix=jarak)

#visualisasi
clusplot(datap,k.means.fit$cluster,main="Visualisasi Cluster",
```

```
color=TRUE,shade=TRUE,lines=0)
```