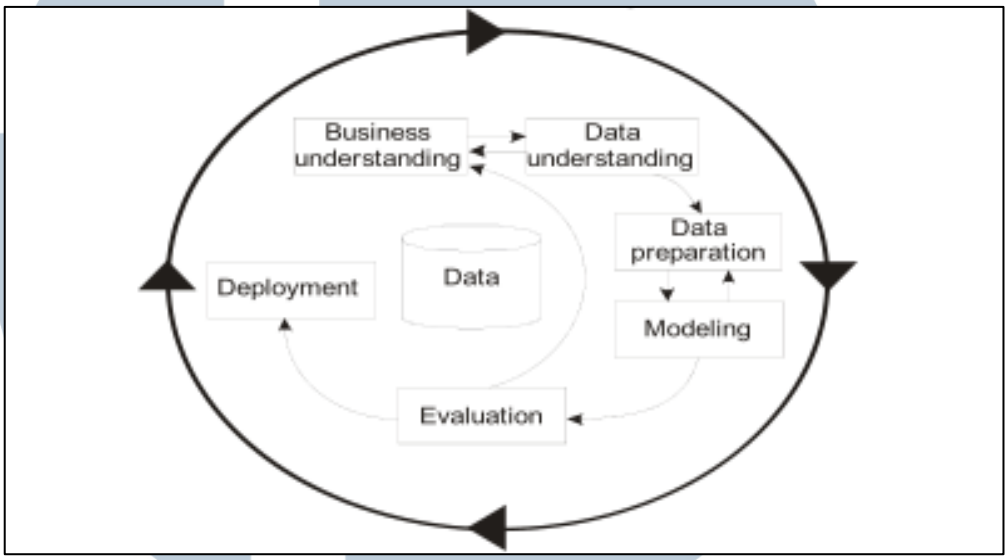
BAB II

TINJAUAN PUSTAKA

2.1. CRISP-DM (*Cross Industry Standard Process for Data mining*)



Gambar 2.1. Tahap Kerja CRISP-DM

Sumber : (Óscar Marbán, 2009)

CRISP-DM merupakan metode yang paling umum digunakan untuk *data mining*, menurut *poll* yang dilaksanakan oleh KdNuggets.com pada tahun 2002, 2004, dan 2007. CRISP-DM dikembangkan oleh Daimler Chrysler (kemudian Daimler-Benz), SPSS (lalu ISL) dan NCR pada tahun 1999, CRISP-DM 1.0 telah dirilis untuk umum (Shafique & Qaiser, 2014).

CRISP-DM bersifat *vendor-independent* sehingga bisa digunakan dengan *data mining tools* apa saja dan dapat diterapkan untuk mengatasi masalah dalam *data mining*. Gambar 2.1. menggambarkan alur proses dalam CRISP-DM. CRISP-DM dibagi menjadi enam tahapan (Óscar Marbán, 2009), yaitu sebagai berikut :

Tabel 2.1. Penjelasan Tahapan CRISP-DM

Business understanding	Data understanding	Data preparation	Modeling	Evaluation	Deployment
Determine business objectives	Collect initial data	Select data	Select modeling techniques	Evaluate results	Plan deployment
Assess situation	Describe data	Clean data	Generate test design	Review process	Plan monitoring & maintenance
Determine DM objectives	Explore data	Construct data	Build model	Determine next steps	Produce final report
Produce project plan	Verify data quality	Integrate data	Assess model		Review project
		Format data			

Sumber: (Óscar Marbán, 2009)

a. *Business Understanding*

Tahap ini berfokus pada pemahaman tujuan proyek tersebut dilaksanakan dari sudut pandang bisnis, kemudian merubah tujuan proyek tersebut menjadi *data mining* (DM) *problem domain*, juga menyusun perencanaan tahap awal untuk penyelesaian proyek.

b. *Data Understanding*

Tahap ini dimulai dengan pengumpulan data dan dilanjutkan dengan identifikasi masalah kualitas data yang kemungkinan muncul, bertujuan untuk mendapatkan gambaran dari *insight* yang akan didapat atau membentuk hipotesis awal.

c. *Data Preparation*

Tahap ini mencakup semua aktivitas yang dibutuhkan untuk membuat & menyusun *dataset* terakhir dari data mentah awal. Tahap ini cenderung dilakukan berulang kali.

d. *Modelling*

Tahap ini mencakup pemilihan berbagai teknik pemodelan yang akan dipakai dan diterapkan. Biasanya, ada beberapa teknik untuk menyelesaikan jenis masalah DM yang sama. Beberapa teknik memiliki persyaratan khusus dalam bentuk data yang dipakai, jadi merupakan hal yang wajar jika harus kembali ke tahap *Data Preparation*.

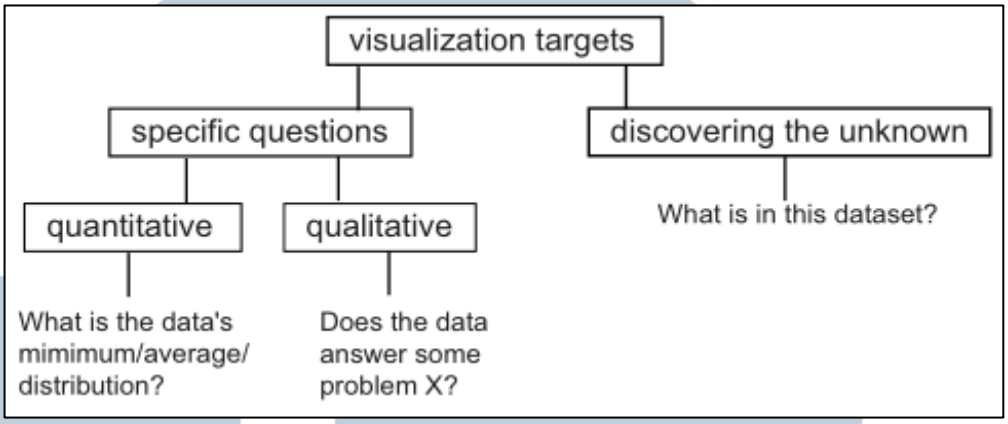
e. *Evaluation*

Sebelum melanjutkan ke tahap *Deployment*, sangatlah penting untuk mengevaluasi model secara teliti dan meninjau ulang langkah-langkah yang dilakukan dalam pengerjaannya agar tujuan awal proyek (kebutuhan bisnis) dapat dipenuhi dengan tepat.

f. *Deployment*

Dalam tahap ini, model dirilis & dipresentasikan kepada *end-user*. Laporan dari hasil proyek juga dibuat dan dilakukan peninjauan terhadap *deployment* yang dilakukan, apakah sudah memuaskan kebutuhan awal atau belum.

2.2. *Data Visualization*



Gambar 2.2. Rumusan Masalah dari Visualisasi Data

Sumber: (Telea, 2015)

Data visualization merupakan metode untuk mempresentasikan data dalam bentuk visual demi tujuan informasi, wawasan, dan pola dari data tersebut dapat lebih mudah dipahami. *Tools* yang umum digunakan dalam *data visualization* yaitu Cognos Insight oleh IBM, Tableau, QlikView, dan Microsoft Power BI. *Visualization Tools* dapat memberikan representasi dan analisis terhadap data secara menyeluruh. Pengguna dapat dengan mudah untuk meneliti lebih jauh (*drill down*, *up*, atau *across*) data yang digunakan dan bereksperimen dengan berbagai macam visual yang ditawarkan seperti *heatmap*, grafik, tabel, *spatial maps*, *timeline*, dan sebagainya (Kimball, Ross, Becker, Mundy, & Thornwaite, 2016). *Data visualization* merupakan metode untuk mempresentasikan data dalam bentuk visual demi tujuan informasi, wawasan, dan pola dari data tersebut dapat lebih mudah dipahami. *Tools* yang umum digunakan dalam *data visualization* yaitu Cognos Insight oleh IBM, Tableau, QlikView, dan Microsoft Power BI. *Visualization Tools*

dapat memberikan representasi dan analisis terhadap data secara menyeluruh. Pengguna dapat dengan mudah untuk meneliti lebih jauh (*drill down*, *up*, atau *across*) data yang digunakan dan bereksperimen dengan berbagai macam visual yang ditawarkan seperti *heatmap*, grafik, tabel, *spatial maps*, *timeline*, dan sebagainya (Kimball, Ross, Becker, Mundy, & Thornwaite, 2016).

Visualisasi data dapat menjawab beberapa jenis pertanyaan seperti yang digambarkan dalam Gambar 2.2. Dalam visualisasi data hasil temuan atau pengetahuan baru yang didapat disebut dengan *insight* yang berarti "wawasan". Wawasan dalam konteks visualisasi data digunakan untuk menggambarkan dua jenis informasi yang diperoleh dari visualisasi aplikasi yaitu sebagai berikut:

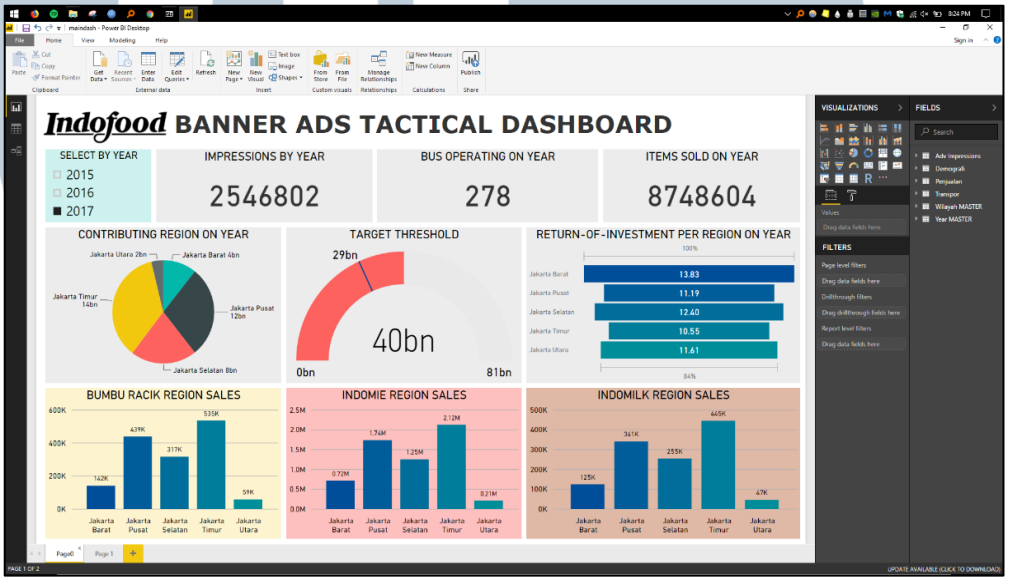
1. Jawaban atas pertanyaan tentang rumusan masalah yang diberikan.

2. Fakta baru tentang rumusan masalah yang tidak diketahui sebelumnya.

Pada umumnya visualisasi data dapat dibagi menjadi 2 jenis, yaitu *Scientific visualization* dan *Information visualization*. *Scientific visualization* merupakan visualisasi yang berkaitan dengan fenomena atau objek masalah yang berbentuk tiga dimensi seperti contoh dalam bidang arsitektur, meteorologi, medis, biologi, dan sebagainya) di mana hasil temuan ditekankan pada volume, permukaan, komponen waktu, dan sebagainya. Jenis ini juga dikenal sebagai *Spatial Data visualization*. Contoh dari jenis visualisasi ini adalah peta iklim berbentuk 3D. Sedangkan *Information*

visualization merupakan visualisasi yang sumber datanya tidak terikat dengan bentuk spasial yang berarti data tersebut dapat diperoleh dari berbagai bentuk seperti grafik, *tree*, tabel, dokumen, rangkaian waktu, dan sebagainya (Telea, 2015).

2.3. Microsoft Power BI



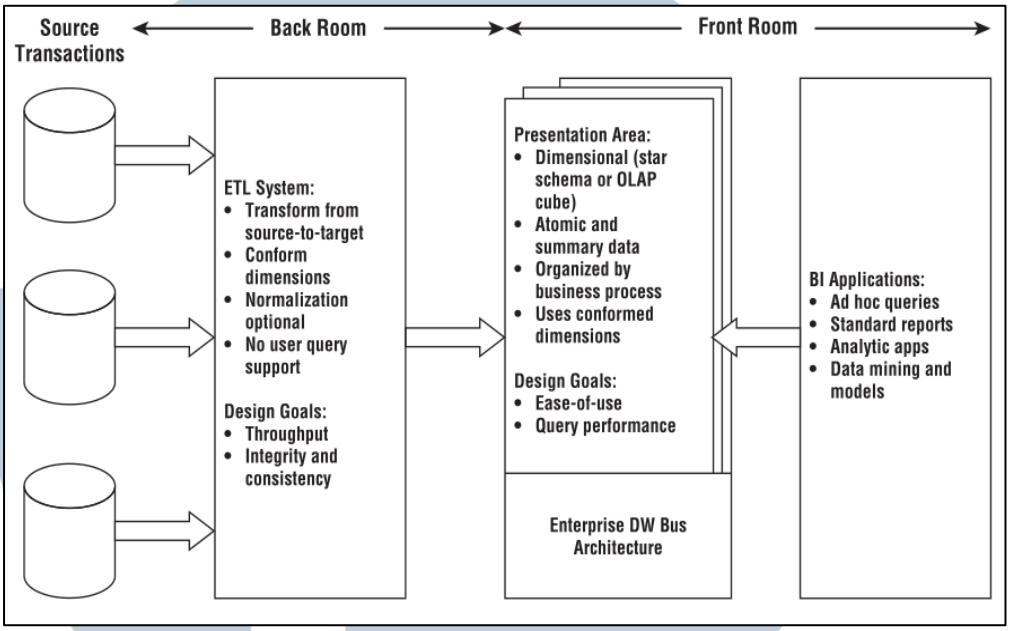
Gambar 2.3. *Dashboard* Visualisasi Microsoft Power BI

Sumber : Dokumentasi Pribadi

Microsoft Power BI merupakan program *Business Analytics* yang dikembangkan oleh Microsoft. Power BI dapat digunakan untuk keperluan *Business Intelligence*, mulai dari *data warehousing*, *data mining* hingga *visualization* (visualisasi grafik dan *dashboards*). Power BI menggunakan DAX (*Data Analysis Expressions*) sebagai dasarnya. DAX juga digunakan dalam produk Microsoft lainnya seperti SQL Server Analysis Services dan Microsoft Excel. Microsoft Power BI dapat memperoleh dan menggunakan data dari beragam sumber – sumber seperti Excel, Text/CSV, XML, JSON, Folder, SharePoint Folder, SQL Server Database, Access Database, SQL

Server Analysis Services Database, Oracle Database, IBM DB2 Database, PostgreSQL Database, SAP Business Warehouse Application Server, Amazon Redshift, Impala, Google BigQuery, Azure SQL Database, Azure SQL Data Warehouse, SharePoint Online List, Microsoft Exchange Online, Dynamics 365 (online), Twilio, tyGraph, Webtrends, Zendesk, TeamDesk. Visualisasi yang dihasilkan oleh Power BI dapat diluncurkan (*publish*) ke Power BI Service, dengan beragam akses dari *web browser* atau *mobile app* berbasis iOS, Android, atau Windows (UWP). Power BI bekerja dengan cara memilih data yang akan dimasukkan, mengolah data yang masuk, memuat ke dalam *dashboard*, dan memilih jenis visualisasi yang cocok sesuai dengan pertanyaan yang akan dijawab dari visualisasi data. Setiap visualisasi mempunyai beragam *Parameter* yang dapat diubah sesuai dengan kebutuhan. Gambar 2.3. merupakan salah satu contoh *dashboard* yang sedang dimuat, memiliki beragam visualisasi yang menjabarkan performa penjualan berdasarkan iklan produk yang memiliki beberapa *Parameter* yaitu penjualan terbesar, angka penjualan, angka *Return-of-Investment*, angka *Impressions*, dan lain sebagainya. (Ferrari & Russo, 2015)

2.4. *Data Mining*

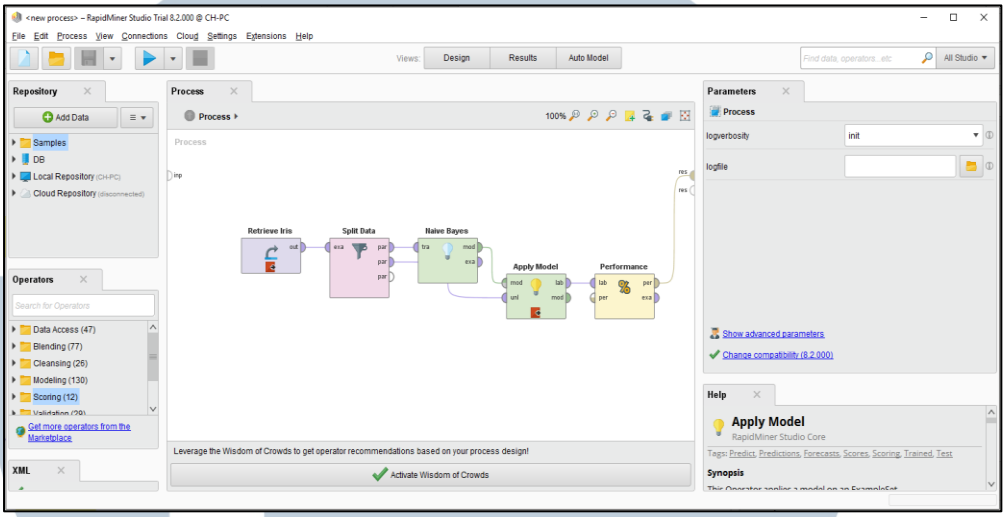


Gambar 2.4. Proses ETL dalam arsitektur DW/BI

Sumber : (Kimball & Ross, 2013)

Data mining merupakan metode untuk mencari wawasan dan pola dalam suatu kumpulan data yang terorganisir setelah melalui proses ETL (Extract, Transform, Load). Pola data yang terbentuk harus valid, mempunyai informasi yang berbobot dan bisa dimengerti. Proses ETL merupakan fase pemrosesan data dari sumber data masuk ke dalam *data warehouse* seperti yang digambarkan dalam Gambar 2.4. Tujuan dari ETL adalah mengumpulkan, menyaring, mengolah dan menggabungkan data-data yang relevan dari berbagai sumber untuk disimpan ke dalam *data warehouse*. (Dharayani, Laksitowening, & Yanuarfiani, 2015).

2.5. RapidMiner



Gambar 2.5. *RapidMiner Canvas*

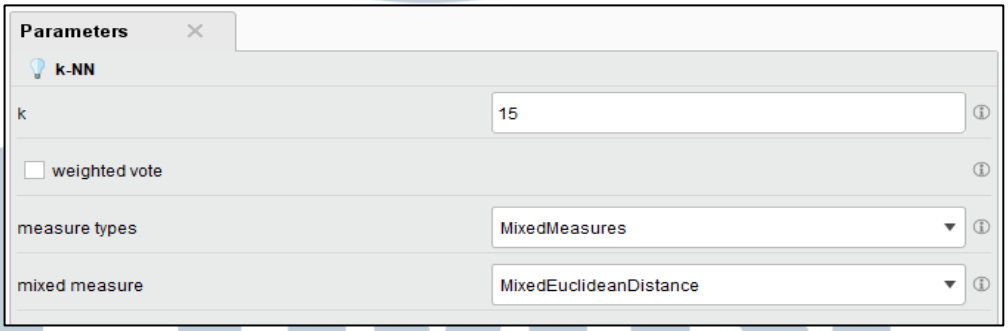
Sumber : Dokumentasi Pribadi

RapidMiner adalah platform yang digunakan untuk keperluan *data mining* (DM), dimana proses – proses dari DM diprogram per *block* yang memiliki fungsi spesifik yang disebut operator. Setiap operator melakukan fungsi terhadap data seperti contoh memuat data (*load data*), normalisasi data (*normalize*), mnegurutkan data (*sort*), dan sebagainya. Pengguna dapat membuat proses dari operator dengan menempatkan operator di kanvas dan menghubungkan garis *input* mereka ke port *output*, seperti ditunjukkan pada Gambar 2.5.

Operator dalam RapidMiner dibagi ke dalam beberapa jenis, yaitu sebagai berikut:

1. *Data Access* (berhubungan dengan operasi *read*, *write*, *update*, *delete*)

2. *Blending* (berhubungan dengan operasi untuk memanipulasi data mentah)
3. *Cleansing* (berhubungan dengan operasi untuk membersihkan atau normalisasi data)
4. *Modeling*, *Scoring*, *Validation* (berhubungan dengan algoritma dan metode yang dipakai untuk mengolah, menguji dan memvalidasi data)
5. *Utility* (berhubungan dengan operasi terkait program RapidMiner itu sendiri)
6. *Extensions* (berisi operator dari *extension* yang di install dalam RapidMiner) (RapidMiner GmbH, 2017)



Gambar 2.6. *Tab Parameter*

Sumber : Dokumentasi Pribadi

Salah satu keunggulan dari program RapidMiner adalah penggunaan operator yang bersifat *drag & drop*, khususnya dalam menggunakan operator jenis *Modelling* seperti operator Decision Tree, k-Means Clustering, Deep Learning, Neural Net, dan sebagainya. Pengaturan *Parameter* dalam tiap operator tersebut juga sangat mudah dengan menggunakan *tab Parameters* di

bagian pojok kanan layar, *user* dapat dengan cepat mengatur seluruh variabel yang digunakan dalam metode atau algoritma yang dipakai, seperti yang digambarkan dalam Gambar 2.6.

2.6. *Text mining*

Text mining dapat didefinisikan sebagai proses ekstraksi informasi atau *knowledge* dimana dokumen – dokumen dianalisa menggunakan *tools* untuk mencari informasi yang berguna diantara data yang belum terstruktur didalam dokumen – dokumen tersebut melalui identifikasi and ekplorasi pattern. Tidak seperti pada *preprocessing* dalam *data mining* yang dirangkum dalam 2 proses penting, yaitu *scrubbing* atau proses membersihkan teks dari variabel yang tidak diinginkan lalu *normalize* atau normalisasi teks, dan membuat *table join* untuk dilanjutkan dalam proses ETL, *text mining* berfokus pada identifikasi dan ekstraksi bahasa dalam dokumen secara alami atau natural. Proses *preprocessing* ini bertanggung jawab untuk mengubah data yang belum terstruktur dalam dokumen - dokumen menjadi format yang lebih terstruktur. Secara umum *preprocessing* terdiri dari beberapa proses, yaitu sebagai berikut:

1. *Tokenization*

   Tokenization atau tokenisasi merupakan proses dimana karakter dalam dokumen dipecah menjadi beberapa bab, bagian, paragraf, kalimat, kata-kata, dan bahkan suku kata. Pendekatan yang paling sering ditemukan dalam *text mining* adalah "pemutusan" teks menjadi kalimat dan kata-kata (Feldman & Sanger, 2017). Seperti

contoh, dalam kalimat "Kalimat berikut ini adalah sebuah kalimat yang ditulis menggunakan Bahasa Indonesia" jika melalui proses *tokenization* berdasarkan kata akan menghasilkan *tokens* sebagai berikut: 'Kalimat', 'berikut', 'ini', 'adalah', 'sebuah', 'kalimat', 'yang', 'ditulis', 'menggunakan', 'Bahasa', 'Indonesia'.

2. *Stemming*

Stemming merupakan proses penghilangan imbuhan, awalan, dan akhiran dari suatu kata menggunakan algoritma, hal ini bertujuan untuk merubah kata ke dalam bentuk asalnya (*root word*) (Sulman & Kurniawan, 2014). Seperti contoh dalam kata '*Fishing*', '*Cars*' dan '*Driving*' jika melalui proses *stemming* maka akan menghasilkan kata '*Fish*', '*Car*', dan '*Driv*'.

3. *Stop word Removal*

Stop word removal merupakan tahap penghapusan *stop word*, dimana *stop word* itu sendiri merupakan sekumpulan kata yang tidak berhubungan (*irrelevant*) dengan subyek utama yang dimaksud, meskipun kata tersebut sering muncul didalam data yang digunakan. Umumnya *stop word* merupakan kata sambung yang digunakan dalam sebuah kalimat atau teks, seperti contoh dalam bahasa Inggris terdapat kata *the, of, a,* dan *and* (Setiawan, Kurniawan, & Handiwidjojo, 2013). Seperti contoh dalam kutipan teks "*This is a sample of a sentence*" jika melalui proses *stop word removal* akan menghasilkan "*This sample sentence*".

4. *n-Grams Technique*

Teknik *n-gram* didasarkan pada pemisahan teks menjadi *string* dengan panjang *n* mulai dari posisi tertentu dalam suatu teks. Posisi *n-gram* berikutnya dihitung dari posisi yang sebenarnya bergeser sesuai dengan *offset* yang diberikan. Nilai offset bergantung pada pembagian yang igunakan dalam *n-gram* (sukukata, huruf, kata, dan sebagainya). Pembagian *n-gram* dapat bervariasi tergantung dari pendekatan dalam membagi teks menjadi bentuk *n-gram*. *N-gram* untuk setiap *string* dihitung dan kemudian dibandingkan satu per satu. *N-gram* dapat berupa *unigram* (n=1), *bigram* (n=2), *trigram* (n=3), dan seterusnya (Lisangan, 2013). Seperti contoh dalam kutipan teks berbahasa inggris "*The aeroplane and the radio have brought us closer together. The very nature of these inventions cries out for the goodness in men - cries out for universal brotherhood - for the unity of us all. Even now my voice is reaching millions throughout the world - millions of despairing men, women, and little children - victims of a system that makes men torture and imprison innocent people.*" Jika melalui proses analisis *n-gram* dengan pengaturan *bigram* maka kutipan teks akan dipecah ke dalam pasangan yang terdiri dari 2 kata, menghasilkan *token* '*The aeroplane*', '*aeroplane and*', '*and the*', '*the radio*', '*radio have*','*have brought*', '*brought us*', '*us closer*', '*closer together*', dan sebagainya.

2.7. *Naïve Bayes Classifier*

Naïve Bayes Classifier adalah metode klasifikasi probabilistik yang menghitung probabilitas dengan cara mencari dan menghitung frekuensi dan kombinasi atribut dalam data yang diberikan. Algoritma ini menggunakan teorema Bayesian dan mengasumsikan semua atribut diberikan secara independen yang diberikan oleh nilai dari variabel kelas. Penggunaan *Naïve* adalah dikarenakan asumsi independensi antar atribut dalam data yang dilakukan jarang berlaku di dunia nyata sehingga dikatakan naif namun dalam pengaplikasiannya algoritma ini cenderung berkinerja baik dan dapat belajar dengan cepat dalam beragam kasus penggunaan algoritma untuk klasifikasi (Patil & Sherekar, 2013). Salah satu keuntungan dari penggunaan algoritma *Naïve Bayes Classifier* adalah algoritma ini dapat bekerja dengan jumlah data latihan (*Training dataset*) yang berukuran kecil untuk menentukan estimasi paremeter yang diperlukan dalam proses klasifikasi. *Naïve Bayes Classifier* menggunakan rumus seperti yang digambarkan dalam Gambar 2.7.

$$P(C|F1 \dots Fn) = \frac{P(C)P(F1 \dots Fn|C)}{P(F1 \dots Fn)}$$

Gambar 2.7. Rumus *Naïve Bayes Classifier*

Sumber : Dokumentasi Pribadi

Dalam rumus ini, variabel C mewakili *class*, sementara variabel F1…Fn mewakili karakteristik petunjuk yang dibutuhkan untuk melakukan

klasifikasi. Rumus tersebut menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas C (*posterior*) adalah peluang munculnya kelas C (sebelum masuknya sampel tersebut di *input*, disebut *prior* yang berarti sebelum), dikali dengan peluang kemunculan karakteristik-karakteristik sampel pada kelas C (disebut juga *likelihood*), dibagi dengan peluang kemunculan karakteristik-karakteristik sampel secara global (disebut juga *evidence*). *Naïve Bayes Classifier* mampu menghitung hasil probabilitas yang paling mungkin yang tergantung oleh *input*, maka semakin banyak data mentah baru yang dimasukkan maka hasil klasifikasi juga menjadi lebih akurat. *Naïve Bayes Classifier* mengasumsikan bahwa keberadaan (atau ketiadaan) fitur tertentu dari suatu kelas tidak terkait dengan keberadaan (atau ketiadaan) fitur lainnya. Misalnya, buah dapat dianggap sebagai apel jika berwarna merah, bulat, dan berdiameter sekitar 4 inci. Bahkan jika fitur ini bergantung satu sama lain atau pada keberadaan fitur lainnya, *Naïve Bayes Classifier* menganggap semua ini properti untuk secara mandiri berkontribusi pada kemungkinan bahwa buah ini adalah apel (Parveen & Pattekari, 2015).

2.8. *Confusion matrix*

Confusion matrix merupakan metode yang digunakan dalam menguji suatu metode klasifikasi. Matrix dibentuk berisi kolom kolom yang membandingkan antara hasil klasifikasi yang ingin diuji dengan klasifikasi yang sebenarnya. Dalam klasifikasi 2 kelas, *confusion matrix* dapat dibuat seperti yang digambarkan dalam Gambar 2.8.

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
Positif	TP (True Positive)	FN (False Negative)
Negatif	FP (False Positive)	TN (True Negative)

Gambar 2.8. *Confusion matrix* 2 kelas

Sumber: (Solichin, 2017)

Perhitungan akurasi dinyatakan dalam rumus di Gambar 2.9. dengan penjelasan sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} * 100\,\%$$

Gambar 2.9. Rumus *Confusion matrix*

Sumber : Dokumentasi Pribadi

TP adalah *True Positive*, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem. TN adalah *True Negative*, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem. FN adalah *False Negative*, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem. FP adalah *False Positive*, yaitu jumlah data positif namun terklasifikasi salah oleh sistem (Indriani, 2014) .