UNIVERSITAS BHAYANGKARA JAKARTA RAYA
FAKULTAS ILMU KOMPUTER
Kampus I: Jl. Harsono RM No. 67, Ragunan, Pasar Minggu, Jakarta Selatan 12550
Telepon: (021) 27808121 – 27808882
Kampus II: Jl. Raya Perjuangan, Marga Mulya, Bekasi Utara, Jawa Barat, 17142
Telepon: (021) 88955882, Fax.: (021) 88955871
Web: fasilkom.ubharajaya.ac.id, E-mail: fasilkom@ubharajaya.ac.id

SURAT TUGAS

Nomor: ST/1327/XII/2022/FASILKOM-UBJ

1. Dasar: Kalender Akademik Ubhara Jaya Tahun Akademik 2022/2023.

2. Dalam rangka mewujudkan Tri Dharma Perguruan Tinggi untuk Dosen di Universitas Bhayangkara Jakarta Raya maka dihimbau untuk melakukan Penelitian.

3. Sehubungan dengan hal tersebut di atas, maka Dekan Fakultas Ilmu Komputer Universitas Bhayangkara Jakarta Raya menugaskan:

NO.	NAMA	NIDN	JABATAN
1.	Dr. Tb. Ai Munandar, S.Kom., M.T.	0413098403	Dosen Tetap Prodi Informatika

Membuat Buku dengan judul **"Data Mining Menggunakan R Teori dan Praktik"** yang diterbitkan oleh PT. Bale Damar Publishing, Cetakan Pertama: Januari 2023, ISBN: 978-623-09-1542-0.

4. Demikian penugasan ini agar dapat dilaksanakan dengan penuh rasa tanggung jawab.

Jakarta, 29 Desember 2022
DEKAN FAKULTAS ILMU KOMPUTER

Dr. Dra. Tyastuti Sri Lestari, M.M.
NIP. 1408206

TB AI MUNANDAR

Data Mining Menggunakan R

Teori dan Praktik

Pengantar Data Mining - Proses Penemuan Pengetahuan - Pre-Processing Data - Visualisasi Data - Pengenalan Paltform Kaggle - Analisis Regresi - Pohon Keputusan - Naive Bayes - Partitioning Arround Medoid - K-means - Hierarchical Agglomerative Clustering - Evaluasi Model - Materi Praktik Menggunakan pada Platform Kaggle

PT. BALE DAMAR PUBLISHING

Data Mining Menggunakan R
Teori dan Praktik

Data Mining Menggunakan R
Teori dan Praktik

Tb Ai Munandar

PT. BALE DAMAR PUBLISHING

DATA MINING MENGGUNAKAN R
Teori dan Praktik

Tb Ai Munandar

Desain Cover :
Achmad Fazi Alam

Tata Letak :
Achmad Fazi Alam

Ukuran :
6, 228, Uk: 15.5x23 cm

ISBN :
978-623-09-1542-0

Cetakan Pertama :
Januari 2023

PT BALE DAMAR PUBLISHING
Jl. KH. Syuhada Perum Griya Lestari Blok B.i No. 17 Kab. Serang, Banten 42184
Website: https://www.bd-publishing.store
E-mail: info@ bd-publishing.store

KATA PENGANTAR

Bismillahirrahmaanirrahiim

Segala puji Penulis panjatkan kehadirat Allah SWT atas rahmat dan hidayah Nya lah impian menulis buku tentang data mining menggunakan bahasa R dapat terwujud. Sholawat serta salam penulis haturkan pula kepada Nabi Muhammad SAW beserta keluarga, sahabat serta umatnya hingga akhir zaman.

Buku ini merupakan kumpulan keberanian penulis untuk berbagi sedikit pengetahuan terkait konsep data mining dan implementasinya menggunakan Bahasa R. Buku ini merupakan akumulasi berbagai sumber pengetahuan yang penulis dapatkan baik selama belajar dan mengajar, mengikuti pelatihan, workshop, seminar serta berbagai kegiatan lainnya terkait data mining, *machine learning* dan implementasinya.

Buku ini tidak akan pernah ada jika tidak disertai dengan dukungan, semangat dari berbagai pihak. Penulis mengucapkan terimakasih kepada teman-teman sejawat mengajar di Universitas Bhayangkara Jakarta Raya, keluarga, pihak penerbit serta pihak yang tidak bisa disebutkan satu-persatu namun telah memberikan dukungan dan bantuan sampai terwujudnya buku ini.

Semoga kehadiran buku ini bermanfaat bagi pembaca dan menjadi ladang amal bagi penulis sebagai upaya untuk menyebarkan ilmu pengetahuan. Tidak ada gading yang tak retak, saran dan masukan yang bersifat membangun untuk perbaikan buku sangat penulis harapkan.

Salam hormat,
Tb Ai Munandar
Email : tbaimunandar@gmail.com

DAFTAR ISI

BAB I
PENGANTAR DATA MINING

1.1. Definisi Data Mining

Data mining merupakan kegiatan analisis yang dilakukan terhadap data dengan jumlah sangat besar yang tersimpan pada komputer. Analisis yang dilakukan bisa terjadi melalui berbagai media, misalnya bar code yang digunakan untuk proses transaksi perdagangan pada sebuah toko atau supermarket. Banyak informasi dari suatu transaksi perdagangan tersimpan di dalam bar code. Dimana setiap bar code ini menyimpan banyak informasi terkait dengan harga produk, kuantitas item suatu produk dan sebagainya. Informasi yang dikumpulkan melalui bar code inilah yang kemudian dapat digunakan untuk analisis data mining (Olson dan Delen, 2008).

Gartner Group dalam Larose (2005) mendefinisikan data mining sebagai suatu proses yang dilakukan untuk menemukan korelasi, pola dan tren baru dengan memilah dan memilih sejumlah data yang besar dan tersimpan di dalam sebuah repositori. Proses tersebut dilakukan menggunakan teknologi pengenalan pola baik dengan teknik statistik maupun model matematika.

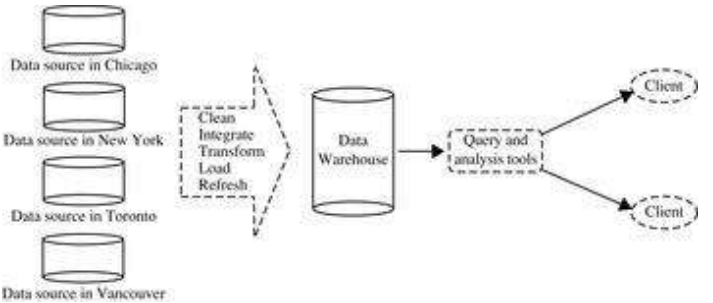
Definisi lain menyatakan bahwa data mining bertujuan untuk memahami unsupervised data dalam jumlah yang besar pada beberapa domain tertentu (Cios dkk, 2007).Proses memahami suatu data dipengaruhi oleh pengalaman user dalam me-mining data. Agar suatu data menghasilkan pengetahuan baru yang mudah dipahami, atribut seperti understandable, valid, novel dan useful harus menjadi hal utama pada saat melakukan mining data. Unsupervised data lebih mudah dan murah untuk dikumpulkan dibandingkan dengan

supervised data. Data yang bersifat unsuvervised tidak membutuhkan input data yang harus dicocokan dengan outputnya. Oleh karena itu, analisis data dengan jenis unsupervised biasanya dilakukan menggunakan algoritma yang mampu menemukan kelompok, hubungan dan asosiasi data secara natural, misalnya dengan teknik klaster dan association-rule.Meskipun demikian, sangat sulit untuk menemukan relasi dan kelompok natural dari suat data karena teknik klaster membutuhkan keahlian user untuk menentukan jumlah klaster data nantinya.Selain itu, asosiasi aturan (association-rule) dalam algoritma data mining membutuhkan penentuan parameter dari user untuk memperoleh jumlah asosiasi dengan kualitas yang sangat baik.

Istilah data mining berkaitan dengan proses yang dilakukan untuk menemukan pengetahuan yang berharga dari sejumlah besar data mentah. Istilah populer lainnya untuk data mining adalah Knowledge Discovery from Data (KDD), knowledge mining from data, knowledge extraction, data/pattern analysis, data archeologu dan data dredging (Han dkk, 2012).

1.2. Jenis dan Pola Data Yang Dapat Di Mining

Secara umum, semua jenis data dapat digunakan pada proses data mining sepanjang data tersebut memiliki arti sangat penting untuk aplikasi yang akan ditargetkan. Contoh data tersebut antara lain, data yang bersumber dari database, data warehousedan database transaksional.



Gambar 1.1 Skema proses pembentukan data warehouse

Database system merupakan sekumpulan data yang terelasi satu dengan lainnya, dimana sebuah relasi biasanya terdiri atas sekumpulan tabel-tabel data dengan nama yang unik. Setiap tabel berisi sekumpulan atribut (yang biasa disebut kolom atau field) dan berisi sejumlah data record atau rows.Data warehouse merupakan tempat penyimpanan informasi yang terkumpul dari banyak sumber data dan tersimpan pada satu tempat. Proses yang berlangsung untuk membentuk data warehouse adalah data cleaning, data integration, data transformation, data loading dan periodic data refreshing. Gambar 1.1 menunjukkan skema proses pembentukan data warehouse.

Database transaksional merupakan data yang diperoleh dari sebuah transaksi, misalnya dari proses pembelian yang dilakukan pelanggan disebuah super market, pemesanan tiket pesawat, atau pemilihan menu tertentu oleh user web di internet. Sebuah transaksi biasanya terdiri atas nomor identitas transaksi yang bersifat unik (misal ID_transaksi) dan sejumlah item untuk kemudian membentuk sebuah tranksasi tertentu. Database transaksional kemungkinan memiliki tabel-tabel tambahan yang berisis informasi terkait dengan

transaksi tertentu, seperti deskripsi item, informasi pelayan toko atau cabang, dan sebagainya.

Jenis data yang lainnya adalah data streaming (seperti data sensor dan data video pengamatan), data sekuen/order (seperti, data bursa saham, data time series dan data sekuen biologis), data jaringan (seperti data yang berasal dari media sosial), data spasial (seperti, peta), hypertext dan multimedia (seperti data teks, gambar, video dan suara) dan internet.

Secara umum fungsi data mining dapat dikelompokkan ke dalam dua kategori, pertama deksriptif yakni berkaitan dengan tugas data mining untuk mengkarakterisasi sejumlah data pada sekumpulan data target. Kedua adalah prediktif dimana data mining digunakan untuk membuat prediksi berdasarkan induksi dari data saat ini. Sedangkan berdasarkan jenis data yang digunakan pada proses data miningfungsionalitas data mining dapat dibagi ke dalam beberapa jenis, antara lain :

1. Karakterisasi dan Perbedaan (*characterization and discrimination*)

Karakterisasi merupakan proses sumarisasi dari karakteristik atau fitur yang bersifat umum dari sebuah class target. Biasanya dilakukan menggunakan query terhadap sejumlah data pada database. Contohnya adalah, untuk mengetahui karakteristik dari penjualan software yang mengalami peningkatan sebesar 10% pada tahun lalu, maka data produk yang terkait dapat dikumpulkan dengan mengeksekusi perintah SQL pada database penjualan.Hasil query dapat ditampilkan dalam bentuk grafik pie, bar, kurva, kubus data multidimensi dan tabel multidimensi.

Sedangkan *data discrimination* merupakan proses yang dilakukan untuk membandingkan fitur-fitur yang dimiliki data target class terhadap fitur suatu objek data dari satu atau banyak class yang

berbeda. Target data dan class yang berbeda ditentukan oleh user, sedangkan objek data yang cocok/sesuai diperoleh melalui query database. Contohnya adalah proses membandingkan fitur-fitur general sebuah penjualan software yang mengalami peningkatan 10% terhadap penjualan yang mengalami penurunan sebesar 30% pada periode yang sama.

2. Pola-pola berulang, asosiasi dan korelasi (*frequent patterns, associations, and correlations*)

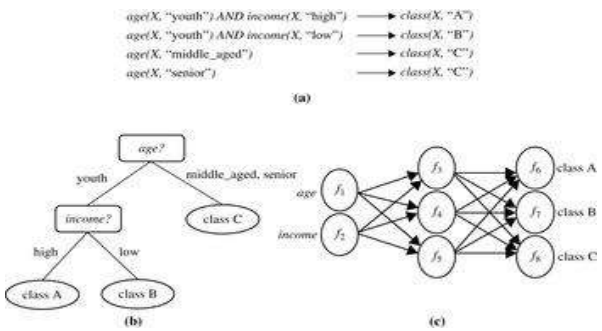
Frequent patterns merupakan pola yang sering muncul/terjadi secara berulang kali pada suatu data. Frequent patterns bisa berupa frekuen item, subsquence (sequential pattern) dan substruktur. Frekuen item biasanya terjadi ketika suatu item sering muncul secara bersamaan pada suatu data transaksi. Contohnya susu dan roti yang sering dibeli oleh banyak pelanggan disebuah toko makanan. Frekuen sequence merupakan kemunculan suatu item yang diikuti dengan item lainnya secara berurutan, atau kemunculan suatu item didahului dengan item-item lain sebelumnya. Contohnya adalah pola pembelian yang dilakukan pelanggan, ketika memutuskan untuk membeli peralatan elektronik, maka yang pertama kali dibeli adalah laptop, kemudian kamera digital dan selanjutnya kartu memori.Sedangkan frekuen substruktur berkaitan dengan perbedaan bentuk struktur (misalnya grafik, pohon keputusan dll) yang mungkin dikombinasikan dengan item atau subsequence. Frequent patterns dalam hal ini akan menghasilkan sebuah infomasi yang menarik berupa asosiasi dan korelasi suatu data. Berikut diberikan ilustrasi analisis asosiasi data yang diperoleh dari suatu data transaksi penjualan dalam bentuk aturan :

membeli (X,"Roti") => membeli (X, "mentega:) [support = 2%, confidence = 60%]

dimana X merupakan variabel yang merepresentasikan pelanggan, confidence sebesar 60% berarti bahwa jika pelanggan membeli roti, ada kemungkinan sebesar 60% bahwa dia akan membeli mentega pada waktu bersamaan. Sedangkan support sebesar 2% menunjukkan bahwa besarnya tingkat akurasi analisis bahwa roti dan mentega akan dibeli secara bersamaan.

3. Klasifikasi dan regresi (*classification and regression*)

Klasifikasi merupakan proses yang dilakukan untuk menemukan sebuah model (fungsi) yang menggambarkan dan membedakan klass data atau konsep berdasarkan hasil analisis sekumpulan training data (objek data yang memiliki label klass yang diketahui). Model yang dihasilkan kemudian digunakan untuk memprediksi label klass dari objek baru yang belum diketahui.Model tersebut dapat direpresentasikan dalam bentuk aturan klasifikasi, pohon keputusan, formula matematika atau jaringan syaraf tiruan seperti yang diperlihatkan pada Gambar 1.2.

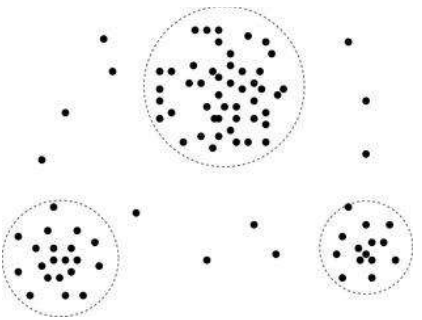


Gambar 1.2 Bentuk representasi model klasifikasi

Regresi digunakan untuk memprediksi nilai data numerik yang hilang atau tidak tersedia. Analisis regresi adalah metode statistik yang digunakan untuk memprediksi data numerik beserta identifikasi distribusi kecenderungan berdasarkan data yang ada.

4. Analisis Klaster (*Cluster analysis*)

Analisis klaster digunakan untuk membentuk label klass dari suatu group data dimana data tidak memiliki label klass sebelumnya. Suatu data dikelompokkan menurut prinsip similarity dan disimilarity suatu objek data. Data yang memiliki kesamaan yang kuat akan dikelompokan ke dalam suatu klaster sedangkan data dengan similarity yang lemah (disimilarity) akan dikelompokkan ke dalam klaster lainnya. Gambar 1.3 memperlihatkan proses klaster data ked lam tiga kelompok.



Gambar 1.3 Klaster data ke dalam tiga kelompok

5. Analisis *outlier*

Analisis *outlier* berkaitan dengan analisis yang dilakukan terhadap objek data yang tidak memiliki model, perilaku dan pola

yang secara umum ditemukan pada sebuah data, misalnya deteksi penipuan penggunaan kartu kredit dll. Analisis ini biasanya menggunakan pengujian statistik yang mengasumsikan distribusi atau kemungkinan dari suatu model data, atau menggunakan pengukuran jarak dimana objek dengan jarak yang jauh dari klaster lainnya dipandang sebagai outlier.

1.3. Perbedaan Data Mining Dengan Pendekatan Lain

Data mining bukan sekedar "payung" yang diciptakan dengan tujuan supaya data dapat dipertimbangkan untuk menghasilkan pengetahuan baru sehingga dapat dimanfaatkan sesuai kebutuhannya. Perbedaan utama data mining dengan model lain adalah, bahwa data mining bersifat data driven. Di dunia statistik para peneliti biasanya berurusan dengan proses pencarian estimasi yang cukup dipercaya berdasarkan ukuran data yang kecil. Sedangkan data mining adalah kebalikannya, data yang digunakan berukuran besar dan bertujuan membangun sebuah model data dengan ukuran kecil (namun tidak terlalu kompleks) tetapi masih dapat dideskripsikan dengan baik.Menemukan model data yang terbaik dan mudah dipahami merupakan jantung dari data mining.

1.4. Teknologi Yang Digunakan

Data mining menyatukan banyak teknik dari domain lainnya seperti statistik, machine learning, pattern recognition, sistem database dan data warehouse, information retrieval, visualisasi, algoritma, komputasi tingkat tinggi dan banyak domain aplikasi lainnya. Gambar 1.4 memperlihatkan teknik-teknik yang diadopsi oleh data mining.

Gambar 1.4 Teknik-teknik yang diadopsi oleh data mining

Statistik mempelajari proses pengumpulan, analisis, interpretasi atau penjelasan dan penyajian data. Sebuah model statistik merupakan sekumpulan fungsi matematis yang menjelaskan perilaku suatu objek pada sebuah klass target menurut distribusi variabel secara random dan kemungkinan asosiasi yang dimilikinya.Machine learning menyelidiki bagaimana komputer dapat belajar (atau meningkatkan kinerjanya) berdasarkan data tertentu yang akan dianalisis. Area penelitian utamanya adalah mengembangkan program komputer yang dapat belajar secara otomatis untuk mengenali pola-pola data yang kompleks dan membuat keputusan cerdas berdasarkan data yang tersedia. Contoh, program komputer yang dapat mengenali kode pos yang ditulis tangan pada surat yang akan dikirim.Machine learning secara umum terbagi ke dalam empat teknik utama, yakni supervised learning, unsupervised learning, semi-supervised learning dan active learning.

Fokus penelitian sistem database adalah membuat, memelihara dan menggunakan database untuk kebutuhan organisasi dan pengguna akhir. Secara khusus, penelitian dibidang ini berkaitan dengan model data, bahasa query, pemrosesan query, optimasi metode query, media penyimpanan data serta metode pengindekan dan pengaksesan data. Kaitannya dengan data mining, penelitian terkait sistem database adalah bagaimana menangani data dalam jumlah yang sangat besar, bersifat real time dan data streaming yang bergerak sangat cepat. Sedangkan data warehouse merupakan bidang ilmu yang mengintegrasikan data dari berbagai sumber dan jangka waktu yang beragam. *Information retrieval* (IR) merupakan bidang ilmu yang digunakan untuk mencari dokumen atau informasi yang tersimpan pada suatu dokumen. Dokumen yang dimaksud dapat berupa text atau multimedia yang mungkin tersedia pada situs Web. Ada dua perbedaan antara information retrieval yang bersifat tradisional dengan sistem database, yakni (1) information retrieval mampu menangani data yang tidak terstruktur, (2) query dibentuk berdasarkan kata kunci dengan struktur yang tidak terlalu kompleks.

1.5. Hal-Hal Yang Dapat Dilakukan Dengan Data Mining

Data mining merupakan disiplin ilmu yang telah berkembang dan menghasilkan banyak aplikasi sebagai alat bantu menyelesaikan permasalahan yang berkaitan dengan prediksi, klasifikasi, pengelompokkan dan sebagainya. Beberapa aplikasi penting telah banyak dikembangkan sebagai dimensi penting dari penelitian dan pengembangan data mining, dua dari aplikasi data mining yang paling populer adalah business intelligence dan search engines.

Tanpa data mining, banyak bisnis yang mungkin tidak mampu melakukan analisis pasar dengan efektif, membandingkan feedbak pelanggan terhadap produk yang mirip, menemukan kekuatan dan

kelemahan yang dimiliki para kompetitor, memelihara loyalitas pelanggan yang sangat bernilai bagi kelangsungan bisnis dan pembuatan keputusan bisnis yang cerdas. Melalui konsep *Business Intelligence* (BI) semua kegiatan tersebut sangat mungkin dilakukan. BI menyediakan operasi bisnis yang tersusun secara historis, yang sedang beredar dan dari sudut pandang prediktif. Contoh BI misalnya, pelaporan, pemrosesan analisis online (*online analytical processing*), tata kelola kinerja bisnis, proses persaingan cerdas, benchmarking dan analisis prediksi.Teknik klasifikasi dan prediksi merupakan analisis prediksi utama yang paling banyak digunakan dalam business intelligence, misalnya untuk analisis pasar, persediaan dan penjualan. Selain itu, teknik clustering juga banyak digunakan untuk pengeloaan customer relationship dengan mengelompokkan pelanggan berdasarkan kesamaan yang dimilikinya.

Web search engine merupakan sever komputer yang secara khusus digunakan untuk mencari informasi yang ada pada sebuah Web. Hasil pencarian biasanya ditunjukkan dalam bentuk daftar pencarian (list atau hits) yang berisi halaman web, gambar dan jenis file lainnya.

1.6. Data Science vs Data Mining vs Machine Learning

Data mining merupakan subset dari Data Science. Data science merupakan bidang yang lebih luas yang memadukan kemampuan analisis dan tools untuk menghasilkan pengetahuan data (data insight). Tujuannya adalah untuk menghasilkan rekomendasi sebagai alat bantu pendukung keputusan. Sementara data mining adalah proses penemuan pengetahuan melalui serangkaian prosedur kerja sehingga dapat menemukan pola data yang sebelumnya tidak diketahui. Data mining merupakan proses atau alat yang digunakan oleh data scientist

untuk memproduksi pengetahuan. Adapun machine learning merupakan cabang kecerdasan buatan yang dapat digunakan pada proses data mining maupun data science. Hubungan ketiganya saling menguatkan dan mendukung proses satu dengan lainnya.

1.7. Contoh-contoh Kasus Data Mining

Data mining banyak diimplementasikan untuk berbagai kebutuhan. Beberapa diantaranya adalah digunakan untuk kebutuhan segmentasi pelanggan, penentuan kelayakan penerima kredit di dunia perbankan. Di dunia asuransi, data mining dapat digunakan untuk memprediksi pelanggan yang memiliki tingkat resiko klaim tertentu atas kejadian kecelakaan yang dialami kendaraan, pribadi dan sebagainya. Bidang-bidang lainnya yang menggunakan data mining adalah marketing, telekomunikasi, jasa layanan online, IT dan teknologi komputer, kedokteran dan farmasi, biro perjalanan dan tour travel, akademisi dan sebagainya (www.easydatamining.com).

Dibidang IT dan teknologi komputer, data mining banyak digunakan untuk menganalisis hubungan antara waktu tanggap suatu website perusahaan terhadap penjualan produk secara online. Analisis yang dilakukan akan mengelompokkan berbagai permasalahan terkait waktu respon yang dimiliki website perusahaan sehingga membentuk pola-pola tertentu sehingga dapat digunakan untuk kebutuhan peningkatan kinerja operasional perusahaan.Dibidang marketing data mining digunakan untuk memprediksi kemungkinan pembelian yang dilakukan pelanggan terhadap produk-produk tertentu. Dengan demikian kegiatan penjualan produk dari perusahaan difokuskan pada produk spesifik tersebut sehingga biaya yang dikeluarkan untuk kegiatan penjualan dapat lebih diminimalisir.

BAB II
PROSES PENEMUAN PENGETAHUAN

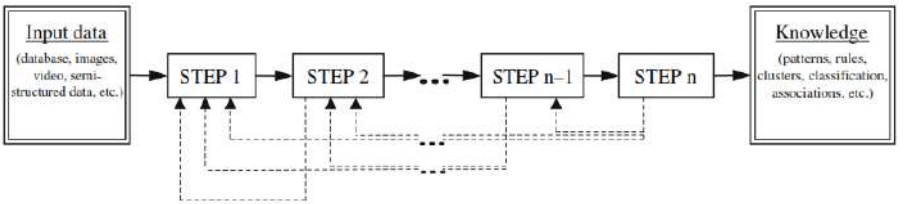
Bab ini membahas bagaimana proses ekstraksi pengetahuan dari sekumpulan data. Sangat penting untuk memahami berbagai pendekatan yang bisa digunakan untuk keberhasilan dalam menggali pengetahuan dari sejumlah data. Pada bab ini juga dibahas mengenai gambaran proses apa saja yang harus dilakukan untuk dalam rangka menemukan pengetahuan baru, gambaran proses tersebut kemudian disebut dengan model proceses.Beberapa alasan mengapa proses penemuan pengetahuan harus disusun secara terstruktur :

1. Hasil akhir proses penemuan pengetahuan harus memberikan manfaat bagi pengguna/pemilik data.
2. Model penemuan pengetahuan harus disusun secara logis, kohesif dan dapat didefinisikan dengan jelas sehingga pengambil keputusan mampu memahami tujuan dari proses ekstraksi pengetahuan.
3. Kegiatan penemuan pengetahuan membutuhkan pengelolaan proyek dengan kerangka kerja yang jelas.
4. Kegiatan penemuan pengetahuang harus mampu mengadopsi model-model dari disiplin keilmuan lain yang sudah ada sebelumnya.
5. Adanya kebutuhan untuk menstandarisasi proses penemuan pengetahuan.

2.1 Definisi Proses Penemuan Pengetahuan

Proses penemuan pengetahuan atau *knowledge discovery process* (KDP) merupakan proses yang dilakukan untuk mengidentifikasi

pola-pola yang tersembunyi dari sekumpulan data sehingga mampu menghasilkan pengetahuan baru yang mudah dipahami dan bermanfaat. KDP difokuskan pada proses ekstraksi pengetahuan, termasuk menyajikan dan mengakses data, memanfaatkan algoritma yang efisien dan terukur untuk menganalisis sejumlah data yang sangat besar. KDP juga berkaitan dengan bagaimana menginterpretasikan dan memvisualisasikan pengetahuan yang ditemukan, serta bagaimana membuat model yang mampu mendukung interaksi timbal balik antara manusia dengan mesin.Proses penemuan pengetahuan (KDP) bersifat sekuensial artinya dilakukan secara berurutan menurut prosedur tertentu sesuai dengan metode yang digunakan. Secara umum, tahapan sekuensial model KDP seperti diperlihatkan pada Gambar 2.1.



Gambar 2.1 Model Knowledge Discovery Procesess
(Sumber : Cios dkk, 2007)

Gambar 2.1 memperlihatkan bagaimana model KDP bekerja. Model KDP memuat banyak tahapan yang kesemuanya dieksekusi secara berurutan. Setiap sub sekuen proses yang ada pada model dipengaruhi oleh hasil dari tahapan sebelumnya baik pada proses sekuen maupun sub sekuen nya. Oleh karenanya, output dari setiap sekuen akan menjadi input bagi sekuen lainnya pada saat tahapan selanjutnya dieksekusi. Input data biasanya terdiri atas berbagai format baik yang bersifat numerik maupun nominal, gambar, video,

data semi struktur seperti XML atau HTML dan sebagainya. Sedangkan luaran yang dihasilkan berupa pengetahuan baru dalam bentuk aturan (rules), pola-pola data tertentu (patterns), model klasifikasi, asosiasi, kecenderungan data (trends), analisis statistik dan sebagainya.

2.2 Model Proses Penemuan Pengetahuan

Secara umum, proses model penemuan pengetahuan terbagi ke dalam tiga pembahasan utama. Pertama model yang didasarkan atas hasil-hasil penelitian para akademisi (*Academic Research Models*), kedua model yang didasarkan atas standar proses industri (*Industrial Models*) dan ketiga adalah *hybrid models*.

1. Academic Research Models

Kemunculan bidang data mining dikalangan akademisi dimulai pada pertengahan tahun 1990an. Para peneliti mulai mendefinisikan berbagai prosedur data mining untuk permasalahan yang lebih kompleks. Tujuan utamanya adalah untuk menyediakan panduan yang tepat dalam upaya menemukan pengetahuan terhadap domain data yang kompleks.Tahun 1996 dan 1998 dikembangkan dua buah model KDP oleh Fayyad dkk serta Anand dan Buchner. Fayyad dalam penelitiannya mengemukakan sembilan tahapan KDP sebagai berikut :

1) *Developing and understanding the application domain*. Tahap ini mengharuskan seseorang yang akan melakukan kegiatan data mining harus mempelajari pengetahuan dasar yang relevan serta tujuan yang ingin dicapai dari proses penemuan pengetahuan yang akan dilakukan.
2) *Creating a target data set*. Pada tahap ini dilakukan pemilihan sejumlah variabel (atribut) dan data point (sample) yang akan digunakan untuk tujuan ekstrak pengetahuan. Kemampuan query

data pada tahap ini sangat dibutuhkan, karena akan digunakan untuk memilih data yang akan digunakan pada tahap selanjutnya.

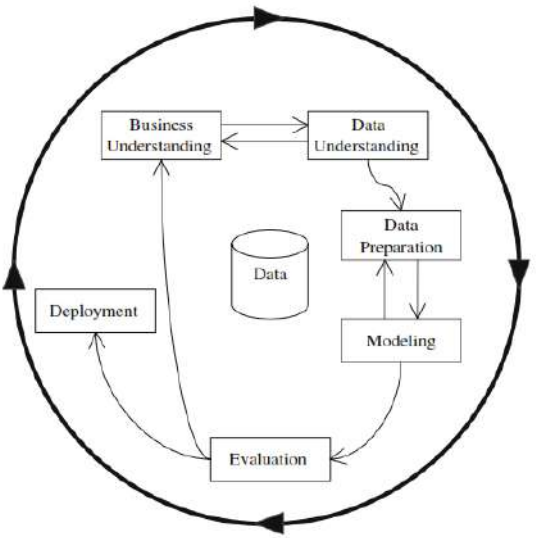
3) *Data cleaning and pre-processing*. Pada tahap ini dilakukan proses penghapusan data yang tidak diharapkan keberadaannya (outliers), penghilangan noise serta melengkapi atau menghilangkan value data yang akan digunakan.
4) *Data reduction and projection*. Tahap ini memuat proses pencarian atribut-atribut yang akan digunakan pada KD. Proses penentuan atribut untuk kebutuhan KD dilakukan dengan menerapkan teknik reduksi dan transformasi data serta menemukan representasi data yang berpola tetap (invariant).
5) *Choosing the data mining task*. Untuk memastikan tujuan dari data mining, maka pemilihan metode yang tepat untuk penemuan pengetahuan sangat dibutuhkan pada tahap ini. Meetode seperti klasifikasi, klastering, regresi dan sebagainya dipilih pada tahap ini sesuai dengan tujuan yang ingin dicapai.
6) *Choosing the data mining algorithm*. Penentuan algoritma yang tepat untuk mengekstrak pengetahuan dilakukan pada tahap ini. Pemilihan algoritma harus disesuaikan dengan tujuan sebagaimana pada tahap 1 dengan mempertimbangkan pemilihan metode yang dipilih pada tahap 5.
7) *Data mining*. Pada tahap ini, pola-pola yang tersembunyi dari sekumpulan data dibangkitkan ke dalam bentuk representasi tertentu, seperti bentuk aturan klasifikasi, pohon keputusan, model regresi, model kecenderungan data (trends) dan sebagainya.
8) *Interpreting mined patterns*. Pada tahap ini, hasil dari kegiatan di tahap 7 kemudian dianalisis untuk menghasilkan visualisasi yang lebih jelas dari model dan pola-pola yang terekstraksi pada tahap 7. Selain itu, data kemudian divisualisasikan dan dianalisis berdasarkan model yang dihasilkan.

9) *Consolidating discovered knowledge*. Tahap ini bisa disebut juga sebagai tahap evaluasi hasil penemuan pengetahuan. Pada tahap ini dilakukan pemeriksaan adanya hasil data mining yang bertentangan dengan pengetahuan / informasi yang sudah ada sebelumnya. Pada tahap ini juga dilakukan proses dokumentasi laporan hasil ekstrak pengetahuan yang dilakukan untuk kebutuhan lain dimasa mendatang.

Proses di atas berjalan secara iterative sampai tujuan dari kegiatan data mining tercapai.

2. Industrial Models

Model ekstraksi pengetahuan yang dikembangkan oleh dunia industri pada dasarnya mengikuti tahapan yang dikembangkan oleh para akedmisi. Yang membedakannya adalah bahwa pada industrial models, pendekatan yang digunakan serta model yang dikembangka didasarkan atas pengalaman panjang kegiatan industri melalui pembentukan konsorsium perusahaan sehingga menghasilkan model-model yang disesuaikan dengan kebutuhan namun tetap bisa digunakan oleh pihak lain sepanjang memenuhi kriteria model yang digunakan. CRISP-DM model yang memuat enam tahapan penemuan pengetahuan adalah salah satu contoh industrial model yang dapat digunakan untuk mengekstraksi pengetahuan. Gambar 2.2 memperlihatkan tahapan yang terjadi pada model CRISP-DM.



Gambar 2.2 Model CRISP-DM untuk penemuan pengetahuan

Model ini pertama kali diluncurkan pada akhir tahun 1990an oleh empat perusahaan besar di Eropa yaitu, Integral Solution Ltd. (provider solusi data mining komersil), NCR (perusahaan database provider), DaimlerChrysler (pabrik pembuat mobil) dan OHRA (perusahaan asuransi). Berikut adalah keenam tahapan model CRISP-DM :

1) *Business understanding*. Tahap ini fokus pada bagaimana memahami tujuan serta kebutuhan kegiatan data mining berdasarkan sudut pandang perusahaan. Pada tahap ini perlu juga dijelaskan hal-hal berupa permasalahan yang dihadapi serta bagaimana merancang persiapan rencana proyek sehingga kegiatan data mining mampu menjawab tujuan yang diharapkan. Berikut adalah sub tahapan yang dilakukan pada tahap pertama :

a. Menetapkan business objective. Menjelaskan sedetil mungkin target yang ditetapkan oleh manajemen atau pemilik perusahaan.
b. Melakukan penilaian atau evaluasi situasi perusahaan.
c. Menentukan tujuan data mining (DM)
d. Menjalankan project plan

2) *Data Understanding*. Tahap ini diawali dengan pengumpulan data dan pengenalan lebih mendalam terhadap data yang akan digunakan. Tujuannya adalah untuk mengidentifikasi permasalahan terkait dengan kualitas data yang digunakan, pemahaman awal terhadap data yang digunakan. Beberapa kegiatan yang dilakukan pada tahap ini adalah Pengumpulan data awal, Deskripsi data yang akan digunakan, eksplorasi data dan verifikasi kualitas data.
3) *Data preparation*. Tahap ini meliputi semua kegiatan yang dibutuhkan untuk membuat dataset akhir. Dataset ini nantinya yang akan digunakan pada model DM pada tahap selanjutnya. Tahapan ini termasuk juga di dalamnya pemilihan tabel, record dan atribut; pembersihan data (data cleaning); membuat atribut-atribut baru (construction of new attributes); dan perubahan bentuk dataset (transformation of data). Secara umum tahap ini terbagi ke dalam lima kegiatan utama, yaitu :
   a. Pemilihan data (*selection of data*)
   b. Pembersihan data (*cleansing of data*)
   c. Pembuatan Data (*construction of data*)
   d. Integrasi data (*integration of data*)
   e. Penyusunan bentuk data (*formating of data subset*)
4) *Modeling*. Pada tahap ini, beberapa metode untuk mengekstraksi pengetahuan dipilih dan digunakan. Pemodelan biasanya

melibatkan penggunaan beberapa metode untuk jenis kasus DM yang sama kemudian mencocokannya dengan parameter yang digunakan untuk mendapatkan hasil yang optimal. Tahap ini dibagi ke dalam beberapa langkah, yakni pemilihan model yang akan digunakan, merancang teknik pengujian hasil DM, pembuatan model dan evaluasi model yang dibuat.

5) *Evaluation*. Tahap ini dilakukan untuk mengevaluasi model-model yang telah digunakan pada tahap sebelumnya sehingga menyesuaikan dengan tujuan yang diharapkan sesuai dengan perspektif perusahaan. Fase terakhir dari tahap ini adalah keputusan terkait dengan digunakan atau tidaknya hasil DM. Beberapa tahapan yang dilakukan pada tahap ini yaitu, evaluasi hasil, peninjauan ulang process data mining dan menentukan langkah selanjutnya berdasarkan hasil data mining.
6) *Deployment*. Tahap ini merupakan kegiatan terakhir dari model CRISP-DM. Pengetahuan baru yang sudah diekstrak perlu disebarluaskan, diorganisir dan dipresentasikan sehingga dapat digunakan oleh pihak yang berkepentingan. Tahapan ini meliputi perencanaan penyebarluasan pengetahuan baru, perencanaan pengawasan dan perawatan proses deployment, menyusun laporan akhir dan melakukan peninjauan ulang terhadap setiap proses yang telah dilakukan sebelumnya.

3. Hybrid Models

Model ini merupakan gabungan dari model akademik dan industrial. Hybrid model mengadopsi CRISP-DM yang dikombinasikan dengan model academic research. Model ini menyediakan tahapan yang lebih general dan berbasis academic research, lebih mengedepankan tahapan-tahapan yang berlaku pada proses data mining pada umumnya, memuat beberapa mekanisme

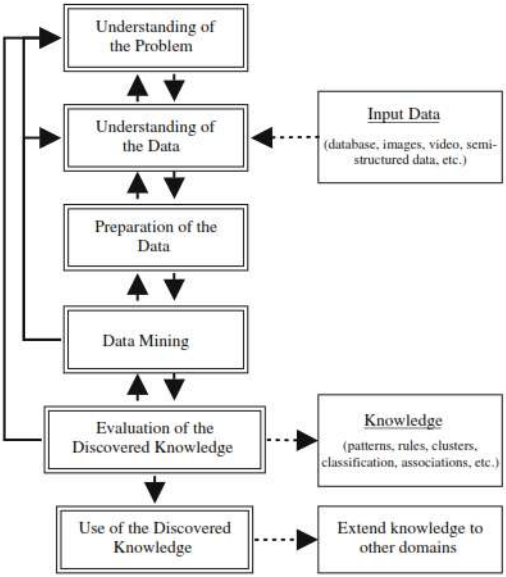
umpan balik antara satu tahap dengan tahap lainnya, baik untuk tahap sebelum maupun setelahnya. Pada model hybrid juga memuat tahap akhir yang tidak dimiliki model lainnya, yaitu pengetahuan yang dihasilkan untuk domain tertentu bisa jadi diimplementasikan dan disebarluarkan pada bidang atau domain lain.Model hybrid memiliki enam tahapan sebagai berikut :

1) *Understanding of the problem domain*. Tahap awal ini secara langsung melibatkan domain pakar untuk mendefinisikan permasalahan dan menetapkan tujuan, mengidentifikasi orang-orang yang menjadi kunci dari kegiatan DM yang dilakukan, dan mempelajari solusi yang sudah dilakukan untuk menghadapi permasalahan yang sedang dihadapi. Kegiatan akhir tahap ini berupa pemilihan dini alat bantu DM yang akan digunakan pada proses selanjutnya.
2) *Understanding of the data*. Tahap ini memuat kegiatan pengumpulan sampel data dan memutuskan data apa saja yang dibutuhkan, termasuk menentukan format dan ukuran data. Pada tahap ini pula dilakukan pengecekan ketidaklengkapan data, redudansi, missing value data dan sebagainya.
3) *Preparation of the data*. Pada tahap ini diputuskan data yang akan digunakan sebagai masukan (input) bagi metode DM yang sudah ditentukan. Tahap ini juga memuat proses penentuan sampel data, pengujian korelasi dan signifikansi data. Pembersihan data juga masuk ke dalam tahapan ini. Pembersihan data dimaksudkan untuk menghilangkan data-data yang tidak lengkap, menghapus atau mengkoreksi nilai-nilai data yang hilang ataupun noise, dan lain sebagainya. Tahap lanjutan dari pembersihan data adalah kegiatan feature selection dan feature extraction dengan menggali kembali atribut-atribut yang akan digunakan pada proses

selanjutnya (diskritisasi) dan data summarization. Hasil akhir dari tahap ini adalah yang memenuhi syarat sebagai masukan (input) untuk model DM yang sudah dipilih pada tahap 1.

4) *Data mining*. Tahap ini merupakan kegiatan dimana sesorang yang akan menggali pengetahuan menggunakan beragam metode DM untuk mendapatkan pengetahuan dari sejumlah data yang sedang diproses.
5) *Evaluation of the discovered knowledge*. Tahap ini memuat kegiatan memahami pengetahuan yang dihasilkan, memeriksa apakah pengetahuan yang dihasilkan benar-benar pengetahuan baru dan menarik atau tidak, interpretasi hasil oleh pakar serta memeriksa dampak yang ditimbulkan akibat hasil penemuan pengetahuan baru tersebut.
6) *Use of the discovered knowledge*. Tahap akhir dari model hybrid ini memuat perencanaan dimana dan bagaimana pengetahuan yang dihasilkan akan digunakan. Implementasi pengetahuan yang dihasilkan bisa diperluas terhadap domain permasalahan lainnya jika memungkinkan. Selain itu, perencanaan monitoring implementasi pengetahuan yang dihasilkan pun harus disusun dengan baik dan hasil dari kegiatan harus didokumentasikan.

Model hybrid banyak digunakan dibidang kesehatan dan pengembangan perangkat lunak. Selain itu, model ini telah diterapkan untuk menganalisis data yang berhubungan dengan kegiatan medis seperti perawatan intensif, *cystic fibrosis* dan klasifikasi citra medis. Gambar 2.3 memperlihatkan tahapan dan model hybrid.



Gambar 2.3 Enam tahapan KDP model hybrid

BAB III
PRE-PROCESSING DATA

Pre-processing data merupakan kegiatan yang dilakukan untuk memastikan bahwa data sudah memenuhi syarat untuk kebutuhan data mining. Banyak data mentah dari database yang masih bersifat tidak valid, tidak lengkap dan bahkan mengandung noisy. Redudansi atribut data, data value yang hilang atau kosong, data yang bersifat outlier, format data yang tidak sesuai untuk dianalisis dengan model data mining serta data value yang tidak konsisten adalah bentuk dari data mentah yang tidak mungkin langsung diproses ke dalam model data mining.

3.1 Pembersihan Data

Tahap pembersihan data biasanya dilakukan untuk memeriksa konsistensi value data dari sebuah dataset yang sudah kita tentukan untuk kebutuhan data mining. Pemeriksaan aturan pengisian value data setiap atribut menjadi fokus utama pada tahap ini. Misalnya, jika dalam suatu tabel data ada atribut Kode_Pos, dimana aturan baku Kode_Pos adalah data numerik yang terdiri atas lima karakter. Jika kemudian ditemukan ada value data yang kosong pada record tertentu, sangat penting bagi kita untuk mempertimbangkan apakah record tersebut akan digunakan atau dihilangkan. Begitu juga jika ada value data yang isinya tidak sesuai dengan aturan, misalnya gabungan antara angka dan huruf pada atribut Kode_Pos, value data yang kurang atau lebih dari lima karakter dan sebagainya. Kondisi seperti ini tentu saja harus benar-benar diperhatikan secara serius, sebab tujuan dari ekstrak pengetahuan adalah bagaimana kita membaca pola-pola tersembunyi dari setiap atribut data yang digunakan.

Kondisi lainnya yang mungkin terjadi adalah, ada atribut pada dataset namun value data nya tidak ada sama sekali, atau ada namun persebarannya tidak merata serta kondisi lainnya. Hal inilah yang menjadikan alasan kenapa harus dilakukan pembersihan data, sehingga data yang akan digunakan sudah benar-benar siap dianalisis.

3.2 Penanganan Missing Data

Pada tahap pembersihan data, adakalanya kita menemukan value data yang kosong (missing). Penyebabnya bisa beragam, terjadi kerusakan data pada saat proses pengumpulan data atau karena memang data itu tidak tersedia sama sekali. Data yang kosong merupakan masalah yang dapat mempengaruhi analisis data nantinya. Meskipun kekosongan data value tidak terlalu berpengaruh besar pada hasil akhir, namun tentu saja data yang lengkap jauh lebih mampu menghasilkan pengetahuan lebih baik. Oleh karenanya, harus dipikirkan dengan hati-hati masalah missing data value ini. Ada beberapa cara yang bisa dilakukan untuk menangani missing data yaitu (Larose, 2005) :

1. Ganti *missing data value* dengan nilai-nilai konstan yang ditetapkan oleh data analis. Misal menggantinya dengan keterangan “Kosong”, “0.000” dan seterusnya.
2. Ganti *missing data value* dengan nilai rata-rata yang diambil dari rerata nilai field bersangkutan
3. Ganti *missing data value* dengan nilai-nilai yang digenerate secara random berdasarkan distribusi nilai dari variabel yang diobservasi

3.3 Mengidentifikasi Kesalahan Klasifikasi Data

Kesalahan klasifikasi biasanya terjadi karena ketidak konsistenan pengisian data value dari suatu atribut, sehingga pada saat

dikelompokan berdasarkan jenis kategori tertentu, terdapat sebaran data yang tidak semestinya. Perhatikan contoh data pada Tabel 3.1. Pada Tabel 3.1 menunjukan jumlah mahasiswa Fakultas Teknologi Informasi berdasarkan klass/atribut **Asal_Daerah**.

Tabel 3.1 Klasifikasi Jumlah Mahasiswa Berdasarkan Asal Daerah

Asal Daerah	**Jumlah**
Pandeglang	20
Serang	30
Pandegelang	2
Cilegon	45
Cilegn	1

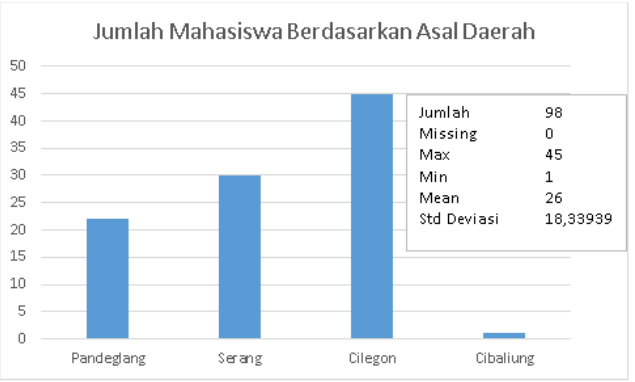
Tabel 3.1 menunjukan bahwa jumlah mahasiswa yang berasal dari **Pandegelang** hanya dua orang. Begitu juga dengan jumlah mahasiswa yang berasal dari **Cilegn**, hanya terdiri atas satu orang. Yang terjadi disini adalah adanya ketidak konsistenan dalam pengisian data value dari atribut **Asal_Daerah**. Pada saat pengisian data value untuk record tertentu, terjadi kesalahan input sehingga menghasilkan klass yang berbeda saat dilakukan query klasifikasi berdasarkan atribut **Asal_Daerah**.Untuk mengatasi hal ini, maka data value **Pandegelang** harus diganti sesuai dengan nama yang semestinya yakni **Pandeglang** dan data value **Cilegn** harus diganti dengan nama **Cilegon**. Dengan demikian, kesalahan klasifikasi data bisa diatasi.

3.4 Pendekatan Grafis untuk Identifikasi Outlier Data

Outliers adalah data value yang posisinya berdekatan dengan nilai limit interval suatu data atau bertentangan dengan trend data yang ada. Sangat penting mengidentifikasi outlier data sebelum melakukan

analisis, sebab bisa jadi outlier merepresentasikan kesalahan yang terjadi saat proses input data.Meskipun pada kasus tertentu, outlier data tidak selalu merepresentasikan kesalahan, namun seringkali mempengaruhi proses analisis sebab dapat menyebabkan hasil yang tidak stabil. Perhatikan Gambar 3.1.

Visualisasi dalam bentuk grafik histogram dapat membantu kita untuk mengidentifikasi outlier dari data yang akan kita gunakan. Pada Gambar 3.1 terlihat bahwa daerah **Cibaliung** memiliki jumlah data yang jauh bahkan di bawah nilai rata-rata (mean) dan standar deviasinya. Walaupun mungkin saja ini asli data yang sesungguhnya, pada tahap analisis akan membawa pengaruh terhadap hasil akhir. Selain dalam bentuk histogram, visualisasi grafis dalam bentuk scatter dapat membantu menunjukan data yang bersifat outlier.



Gambar 3.1 Histogram Klasifikasi Mahasiswa Berdasarkan Asal Daerah

3.5 Transformasi Data

Setiap variabel cenderung memiliki interval nilai yang sangat bervariasi antara satu dengan lainnya. Sebagai contoh, ada dua variabel data pembangunan yakni Indeks Pembangunan Manusia

(IPM) dan Pendapatan Domestik Regional Bruto (PDRB). Kedua variabel memiliki interval nilai yang berbeda. Variabel IPM memiliki interval antara 0 - 100, sementara untuk PDRB memiliki rentang antara 500.000 sampai tak terhingga (perhatikan Tabel 3.2). Pada variabel IPM-2021 misalnya, value atribut yang terisi tidak lebih dari 100, sementara pada variabel PDRB-2021 rentang nilai untuk value atribut tersebut antara 20 juta sampai dengan 100 jutaan. Terlihat perbedaan interval nilai yang terlalu jauh diantara kedua variabel.

Oleh karena itu, sebelum melalukan analisis data melalui tahapan data mining, disarankan untuk menormalisasi data value dari variabel dengan interval nilai yang sangat jauh. Meskipun demikian, beberapa peneliti menganjurkan untuk tidak melakukan normalisasi data agar terlihat pola yang sebenarnya sesuai dengan data aslinya. Ada beberapa teknik yang bisa digunakan untuk proses normalisasi, dua diantaranya adalah metode Min-Max dan Z-Scored.

1) Normalisasi dengan metode Min-Max

Jika diketahui data kecepatan kendaraan sebagai berikut (lihat Tabel 3.2), normalisasikan data value pada atribut kecepatan menggunakan pendekatan Min-Max. Jika diasumsikan simbol Y adalah data asli nilai kecepatan dan Y* adalah nilai kecepatan hasil normalisasi. Diketahui persamaan normalisasi dengan Min-Max adalah seperti pada persamaan (1).

$$Y^* = \frac{Y - \min(Y)}{range\ (Y)} = \frac{Y - \min(Y)}{\max(Y) - \min\ (Y)} \qquad (1)$$

Tabel 3.2 Data IPM dan PDRB Provinsi Banten Tahun 2021

Kabupaten/Kota	IPM-2021	PDRB-2021
Kab Pandeglang	65.17	20.127.757
Kab Lebak	64.03	21.245.040
Kab Tangerang	72.29	97.809.902
Kab Serang	66.82	54.992.522
Kota Tangerang	78.50	106.705.227
Kota Cilegon	73.35	77.071.368
Kota Serang	72.44	23.374.085
Kota Tangerang Selatan	81.60	62.364.100

Dengan menggunakan persamaan (1), normalisasi untuk data IPM-2021 Kab Pandeglang dapat dihitung sebagai berikut :

Nilai terkecil (min) IPM-2021 adalah 64.03 sedangkan nilai tertinggi (max) adalah 81.60. Sehingga IPM untuk Kab Pandeglang diperoleh normalisasi sebagai berikut :

$$Y_{Kab\ Pandeglang}{}^{*} = \frac{Y - \min(Y)}{range\ (Y)} = \frac{65.17 - 64.03}{81.60 - 64.03} = 0.0649$$

Dengan cara yang sama, data IPM-2021 Kab Lebak dapat dihitung sebagai berikut :

$$Y_{Kab\ Lebak}{}^{*} = \frac{Y - \min(Y)}{range\ (Y)} = \frac{64.03 - 64.03}{81.60 - 64.03} = 0$$

Untuk perhitungan normalisasi variable PDRB-2021 juga dilakukan hal yang sama. Perhitungan normalisasi data secara keseluruhan menghasilkan data hasil normalisasi seperti diperlihatkan pada Tabel 3.3.

Tabel 3.3 Data IPM dan PDRB Hasil Normalisasi Min-Max

Kabupaten/Kota	IPM-2021	PDRB-2021
Kab Pandeglang	0.0649	0.0000
Kab Lebak	0.0000	0.0129
Kab Tangerang	0.4701	0.8973
Kab Serang	0.1588	0.4027
Kota Tangerang	0.8236	1.0000
Kota Cilegon	0.5304	0.6577
Kota Serang	0.4787	0.0375
Kota Tangerang Selatan	1.0000	0.4878

2) Normalisasi dengan metode Z-scored

Z-scored standardization merupakan perbedaan antara value atribut dan nilai mean atribut tersebut dibandingkan dengan nilai standar deviasi atributnya. Formulasi perhitungan Z-scored seperti diperlihatkan pada persamaan (2).

$$Y^* = \frac{Y - mean\ (Y)}{STD\ Dev\ (Y)} \qquad (2)$$

Sebelum melakukan normalisasi data, nilai mean dan standar deviasi untuk setiap variable dihitung. Dengan bantuan analisis statistik deskriptif menggunakan Microsoft Excel, diperoleh mean untuk IPM2021 dan PDRB-2021 berturut-turut adalah 71.77 dan 57.961.250,135. Sedangkan untuk standar deviasi variable IPM-2021 dan PDRB-2021 diperoleh berturut-turut adalah 6.25 dan 34.502.609,87. Untuk data yang sama pada Tabel 3.2 data IPM-2021 Kab Pandeglang dapat dihitung sebagai berikut :

$$Y_{Kab\ Pandeglang}{}^* = \frac{65.17 - 71.77}{6.25} = -1.056$$

Dengan cara yang sama, data kecepatan untuk Kab Lebak dapat dihitung sebagai berikut :

$$Y_{Kab\ Pandeglang}{}^{*} = \frac{64.03 - 71.77}{6.25} = -1.238$$

Perhitungan normalisasi data secara keseluruhan baik variable IPM-2021 maupun PDRB-2021 semua wilayah menghasilkan data hasil normalisasi seperti diperlihatkan pada Tabel 3.4.

Tabel 3.4 Data IPM dan PDRB Hasil Normalisasi Z-Score

Kabupaten/Kota	**IPM-2021**	**PDRB-2021**
Kab Pandeglang	-1.056	-1.097
Kab Lebak	-1.238	-1.064
Kab Tangerang	0.083	1.155
Kab Serang	-0.792	-0.086
Kota Tangerang	1.077	1.413
Kota Cilegon	0.253	0.554
Kota Serang	0.107	-1.002
Kota Tangerang Selatan	1.573	0.128

BAB IV
VISUALISASI DATA

Mengutip laman resmi tableau.com, visualisasi data merupakan representasi grafis dari sekumpulan informasi dan data yang menggunakan berbagai macam elemen visual seperti bagan, grafik dan peta. Untuk dapat memvisualisaikan data, biasanya dibutuhkan alat bantu visualisasi sehingga dapat melihat dan memahami trend, outliers dan juga pola yang terdapat dalam data tersebut (tableau.com). Visualisasi data dapat membantu pembuat keputusan melihat hasil analisis yang disajikan secara visual, memahami pola data yang sulit bahkan untuk mengidentifikasi pola baru yang ada dalam data.

Visualisasi data sendiri bukanlah hal yang baru dalam dunia statistika. Konsep visualisasi diawali dengan penggunaan gambar untuk memahami data sejak berabad-abad yang lalu. Pada abad ke-17 dimulai dengan penggunana peta dan grafik sampai dengan ditemukannya diagram lingkaran (pie) pada awal tahun 1800-an. Baru kemudian setelah beberapa dekade mulai muncul dan berkembang konsep visualisasi data yang lebih baik dengan dibuatnya grafik statistik pada era Charles Minard. Grafik statistik yang dibuat ini memberikan gambaran peta invasi Navoleon ke Rusia. Sejak saat itu sampai sekarang visualisasi data semakin banyak berkembang dan digunakan untuk berbagai kebutuhan. Perkembangan teknologi telah menghasilkan banyak alat bantu visualisasi data yang lebih artistik, bagus dan mudah dipahami (sas.com).

Pada bab ini pembahasan visualisasi data (penyajian data) dibahas berdasarkan jenis datanya, yakni data kuantitatif dan data kualitatif. Penanganan setiap jenis data ketika akan dibuat visualisasi tentu saja berbeda, termasuk pemilihan tipe grafik yang akan digunakan.

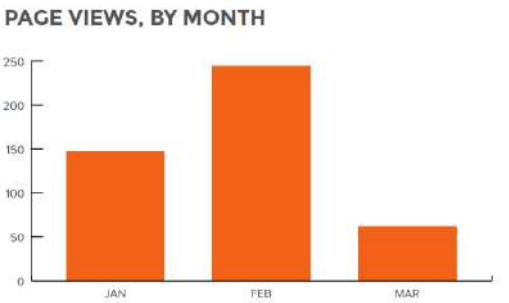
6.1. Visualisasi Data Kuantitatif

Kita tahu bahwa data kuantitatif berhubungan dengan kuantitas, dapat diukur dan berupa angka. Oleh karenanya, data bertipe ini dapat disajikan dalam bentuk grafik, diagram, tabel bahkan peta. Pada kondisi data berupa urutan waktu (time series) juga dapat disajikan dalam bentuk diagram (paling umum diargam line). Beberapa tipe grafik yang dapat digunakan untuk visualisasi data bertipe kuantitatif antara lain :

1. Grafik Bar (*Bar Chart*)

Grafik batang biasanya digunakan untuk menunjukan perubahan dari waktu ke waktu. Dapat juga digunakan untuk membandingkan kategori yang berbeda atau membandingkan bagian data terhadap keseluruhan data yang ada. Banyak yang mengatakan bahwa grafik ini sangat serbaguna karena mudah dan paling sering digunakan. Ada empat jenis grafik batang yang biasa digunakan, yakni vertical bar chart, horizontal bar chart, stacked bar chart dan 100% stacked bar chart.. Perhatikan ilustrasi pada Gambar 4.1 untuk vertical bar chart dan Gambar 4.2 untuk horizontal bar chart, Gambar 4.3 untuk stacked bar chart dan Gambar 4.4. untuk 100% stacked bar chart.

1) *Vertical bar chart* biasanya digunakan untuk menunjukan data secara kronologis (berupa deret waktu yang harus selalu bergerak dari kiri ke kanan), atau ketika akan memvisualisasikan nilai negatif di bawah sumbu x. Bisa juga digunakan untuk merepresentasikan data bertipe ordinal.



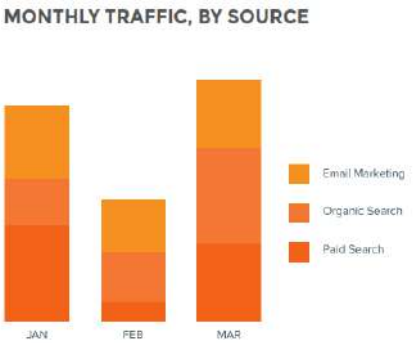
Gambar 4.1 Contoh Vertical Bar Chart
(Sumber : www.hubspot.net)

2) *Horizontal bar chart* biasanya digunakan untuk memvisualisasikan data dengan label kategori yang cukup panjang (bertipe ordinal).

Gambar 4.2 Contoh Horizontal Bar Chart
(Sumber : www.hubspot.net)

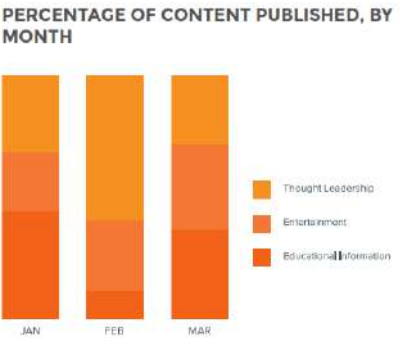
3) *Stacked bar chart* **(**grafik batang tumpuk**)** biasanya digunakan untuk memvisualisasikan data diskrit atau kontinyu. Visualisasi berorientasi pada bentuk vertikal atau horizontal. Grafik ini paling banyak digunakan untuk kebutuhan membandingkan beberapa

hubungan bagian tertentu ke data keseluruhan. Perhatikan Gambar 4.3.



Gambar 4.3 Contoh Stacked Bar Chart
(Sumber : www.hubspot.net)

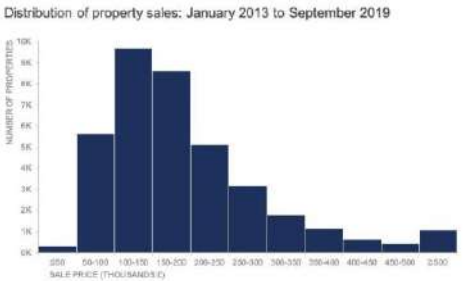
4) *100% Stacked bar chart* biasanya digunakan untuk merepresentasikan persentase relatif dari beberapa seri data. Pada grafik ini total (kumulatif) dari setiap batang grafik selalu sama dengan 100%, berbeda dengan jenis grafik batang tumpuk lainnya. Perhatikan contoh pada Gambar 4.4.



Gambar 4.4 Contoh 100% Stacked Bar Chart
(Sumber : www.hubspot.net)

2. Histogram

Histogram secara visual mirip dengan grafik batang (*Bar Chart*), namun histogram tentu saja bukanlah bar chart, atau bukan bagian dari bar chart. Histogram biasanya digunakan untuk menunjukan distribusi dari variabel. Sedangkan bar chart digunakan untuk membandingkan variabel. Histogram memplot data kuantitatif berdasarkan rentang data yang dikelompokan berdasarkan interval tertentu, sedangkan bar chart memplot data bertipe kategorikal (Robbins, 2012). Gambar 4.5 memperlihatkan contoh dari grafik histogram.



Gambar 4.5 Contoh Histogram
(Sumber : www.storytellingwithdata.com)

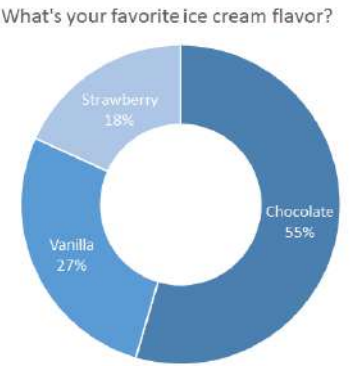
3. Pie Chart (Diagram Lingkaran)

Pie Chart merupakan jenis grafik yang digunakan untuk merepresentasikan data dalam bentuk grafik melingkar. Disebut pie, karena bentuknya mirip kue pie yang dapat dipotong ke dalam beberapa bagian. Potongan-potongan grafik tersebut biasanya sebanding dengan pecahan asli dari keseluruhan data pada setiap kategori. Dengan kata lain, potongan tersebut mewakili bagian dari keseluruhan, sedangkan keseluruhan potongan mewakili 100% keseluruhan data. Ada beberapa varian diagram lingkaran yang bisa digunakan, diantaranya 3D pie chart, Doughnut chart, Exploded pie chart, Ring chart, Spie chart dan Square chart / Waffle chart.

1. 3D pie chart atau perspective pie chart merupakan grafik yang biasa digunakan untuk menampilkan data dalam bentuk 3D. Alasan pokok penggunaan lebih pada nilai estetika karena tampilannya yang lebih menarik. Namun penggunaan Grafik Pie 3D ini pada dasarnya tidak memberikan pengaruh yang signifikan terhadap proses pembacaan data. Grafik dalam bentuk Pie 3D juga sebetulnya sulit untuk diinterpretasikan karena efek persepektif yang terdistorsi karena bentuk tiga dimensinya. Penggunaan grafik pie 3 dimensi ini tidak disarankan jika memang betul-betul tidak diperlukan. Gambar 4.6 memperlihatkan contoh dari garfik pie 3 dimensi.

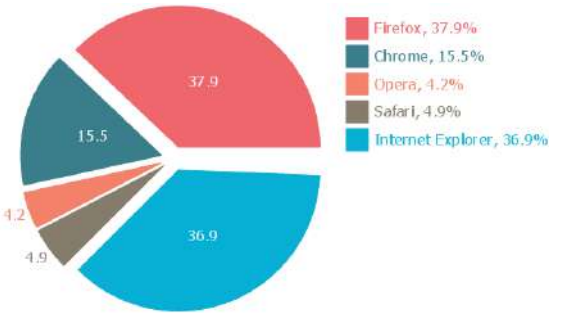
Gambar 4.6 Contoh Grafik Pie 3 Dimensi
(Sumber : www.advsofteng.com)

2. Doughnut chart disebut juga sebagai bagan donat karena bentuknya yang mirip donat. Grafik pie ini memiliki ruang kosong ditengahnya. Tujuan dari penggunaan diagram ini untuk merepresentasikan proporsi dan rasio intensitas suatu data. Bagian tengah diagram boleh berisi informasi boleh juga dikosongkan. Gambar 4.7 memperlihatkan contoh diagram donat.



Gambar 4.7 Contoh Grafik Donat
(Sumber : www. exceljet.net)

3. Exploded pie chart merupakan grafik dengan visualisasi pemisah potongannya lebih dari satu. Pemisah atau sektor potongan ini seolah memberikan kesan ada bagian potongan/segmen yang lebih kecil dari bagian lainnya. Gambar 4.8 memperlihatkan contoh Exploded pie chart.



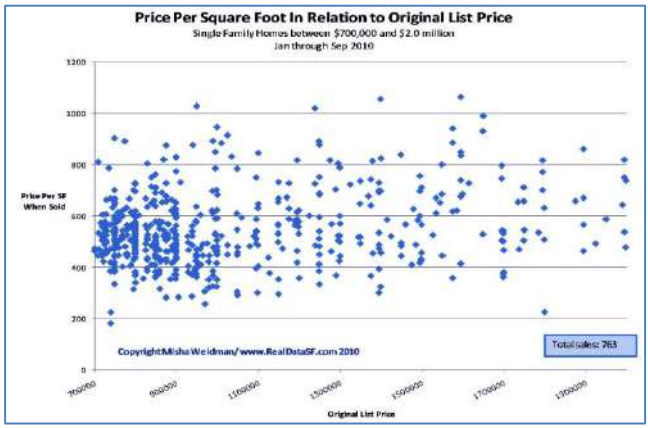
Gambar 4.8 Contoh Grafik Exploded Pie
(Sumber : https://conceptdraw.com/)

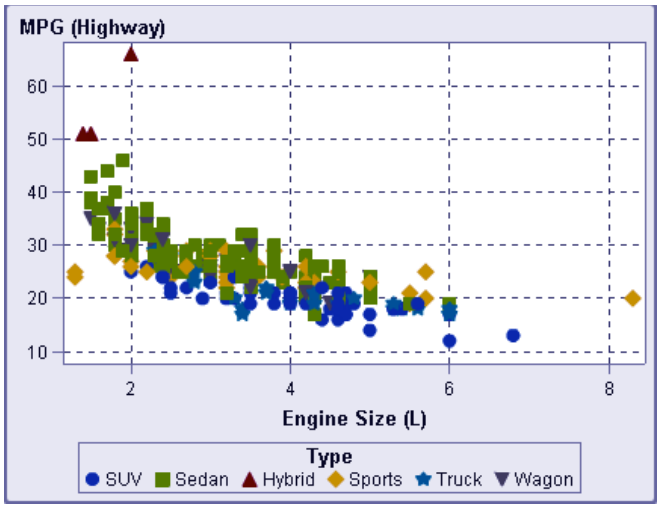
4. Ring chart disebut juga grafik cincin atau grafik semburan sinar matahari atau grafik pie bertingkat. Biasanya digunakan untuk memvisualisasikan data berbentuk hirarki. Grafik akan digambarkan dalam bentuk lingkaran konsentris. Gambar 4.9 merupakan contoh grafik ring.

Gambar 4.9 Contoh Grafik Ring
(Sumber : www.depositphotos.com)

4. Scatter Plot

Scatter plot pada umumnya memvisualisasikan data dalam bentuk sebaran titik pada bidang koordinat Cartesian. Sebaran titik tersebut memperlihatkan sebuah hubungan antara dua variabel. Scatter plot biasanya digunakan untuk mengetahui apakah suatu kelompok data yang berbeda memiliki korelasi atau tidak (Matias, 2021). Scatter plot dapat juga digunakan untuk melakukan analisis data ooutliers. Sebaran titik pada grafik scatter plot tidak hanya menampilkan titik data secara individu akan tetapi juga pola saat data diambil atau divisualisasikan secara keseluruhan (Yi, 2019). Gambar 4.10 memperlihatkan contoh dari grafik Scatter plot.





Gambar 4.10 Grafik Scatter Plot
(Sumber : https://data-flair.training/)

6.2. Visualisasi Data Kualitatif

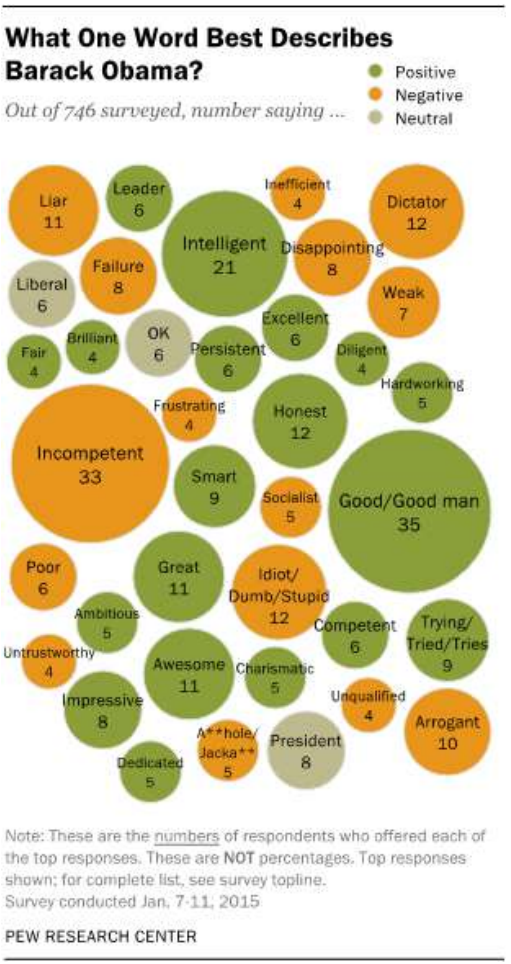
Berbeda dengan data kuantitatif yang dapat diukur, data kualitatif justru sebaliknya. Jenis data ini tidak dapat diukur namun berkaitan dengan kualitas suatu informasi. Ada beberapa cara yang bisa

digunakan untuk memvisualisasikan data bertipe kualitatif, antara lain dalam bentuk rangkuman tabel (table summarizes), grafik batang, pictograph, pie chart (diagram lingkaran). Pada sub pembahasan ini, penjelasan mengenai visualissasi data dalam bentuk tabel, grafik batang dan pie chart tidak akan dibahas ulang. Anda hanya perlu melihat pembahasan sebelumnya pada teknik visualisasi data kuantitatif.

Sumber lainnya menyebutkan bahwa teknik visualiasi data kualitatif dapat dilakukan dengan beberapa pendekatan yang representatif. Misalnya menggunakan teknik word clouds, graphic timelines dan infographic.

1. Word Clouds adalah representasi data dalam bentuk gambar yang terdiri atas sejumlah kata yang digunakan dalam teks atau subyek tertentu. Ukuran setiap kata yang terdapat dalam grafik word clouds memperlihatkan frekuensi atau tingkat kepentingan dari kata tersebut. Gambar 4.11 dan Gambar 4.12 memperlihatkan contoh word clouds yang saat ini banyak digunakan dalam penelitian kualitatif.

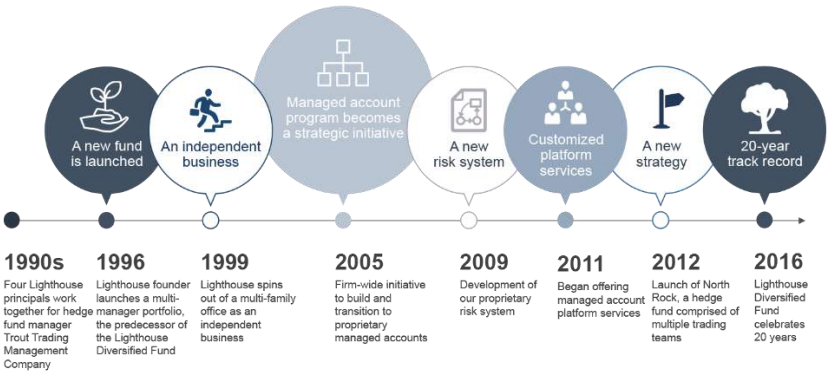
Gambar 4.11 Grafik Word Clouds [25]
(Sumber : Tong dkk, 2012)



Gambar 4.12 Grafik Word Clouds
(Sumber : https://cdn.kicksdigital.com/)

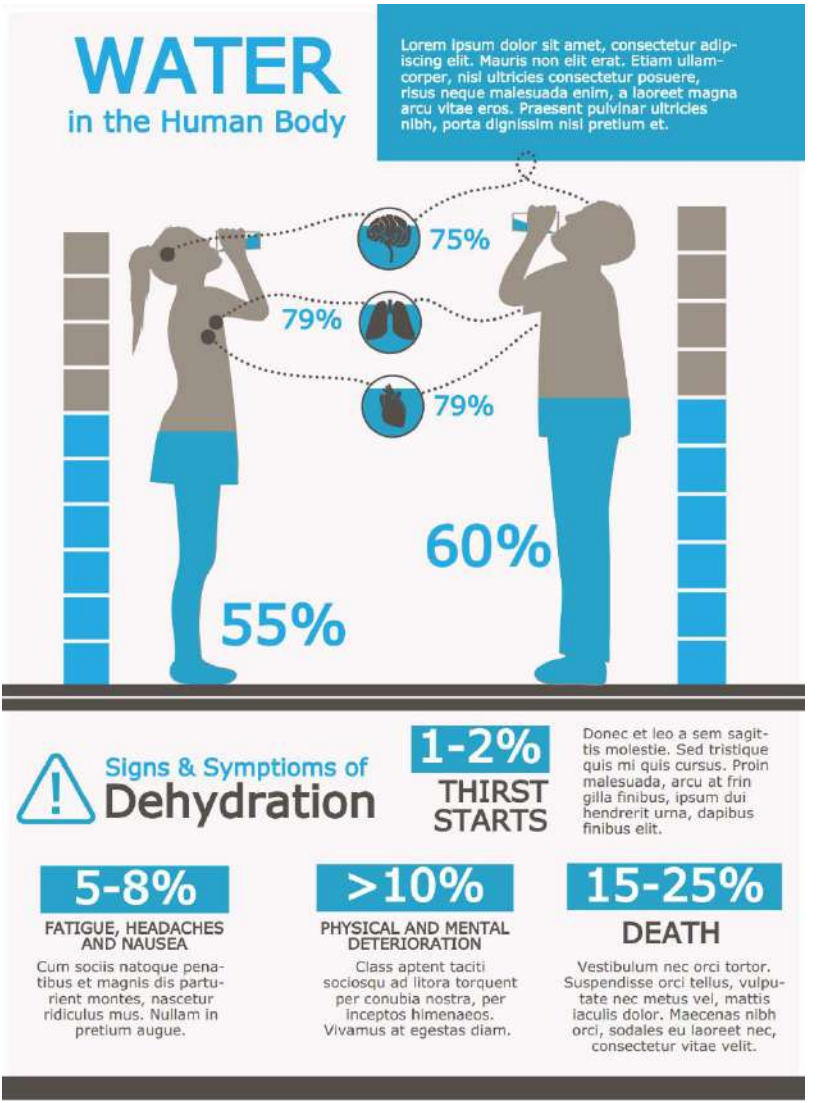
2. Grapich Timelines adalah jenis grafik yang merepresentasikan serangkaian peristiwa dalam urutan kronologis dalam skala waktu yang bersifat linier. Grafik ini mudah untuk dibaca dan dipahami. Banyak digunakan untuk membuat milestone suatu kegiatan, misal

untuk menggambarkan progress dari jadwal satu proyek. Gambar 4.13 merupakan contoh untuk Graphic timelines.



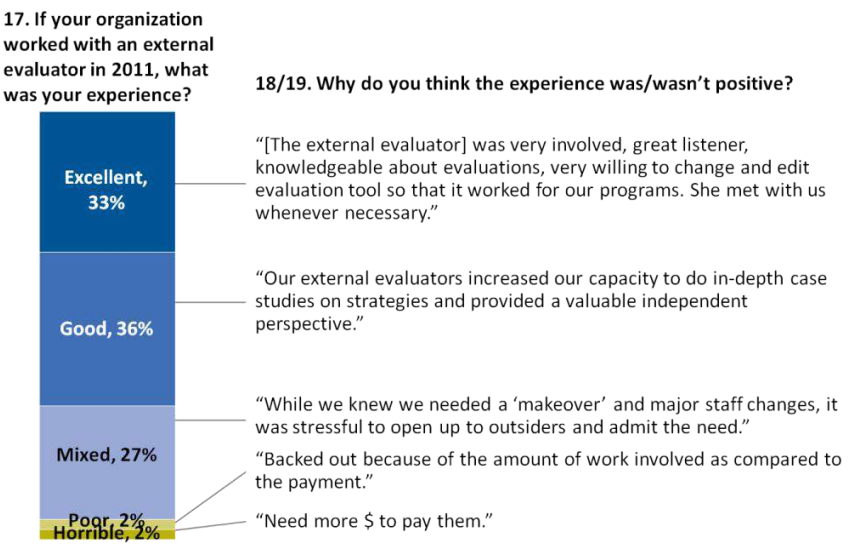
Gambar 4.13 Grapich Timelines
(Sumber : https://www.lighthousepartners.com/)

3. Infographic. Meskipun infografis banyak yang mengatakan bukan bagian dari visualisasi data, namun sebagian lain menganggap bahwa infografis juga merupakan bagian dari grafik. Pada dasarnya infografis merupakan representasi visual dari fakta, peristiwa atau angka. Biasanya tampilan infografis menggabungkan statistik dengan narasi atau sebuah deskripsi tertentu. Yang perlu dipahami adalah tidak semua infografis adalah visualisasi tetapi semua visualisasi adalah infografis. Gambar 4.14 memperlihatkan contoh grafik untuk infografis.



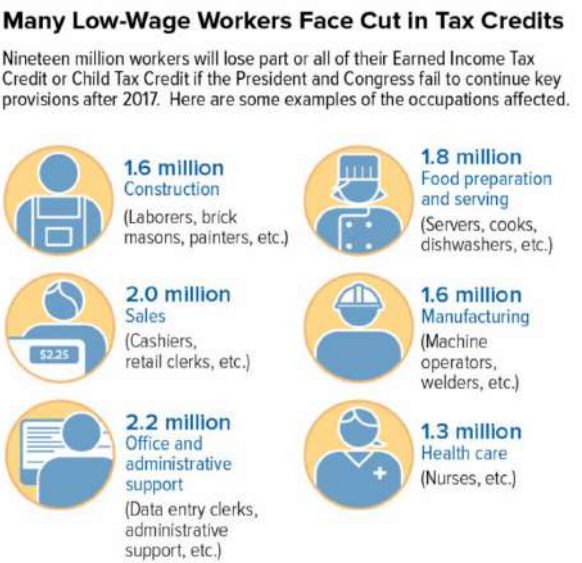
Gambar 4.14 Contoh Infografis
(Sumber : https://www.easel.ly/blog/)

Pendekatan lainnya untuk memvisualisasikan data kualitatif adalah menyajikan data dalam format showcase yang menggabungkan data dan narasi. Tekni ini biasanya digunakan untuk menyajikan data hasil survey dan ingin memperlihatkan tanggapan dalam satu grafik sebagai lampiran (lihat Gambar 4.15). Data hasil kegiatan inteview pun dapat disajikan dengan format memasukan foto narasumber beserta tanggapan yang diberikannya (lihat Gambar 4.16). Cara lain adalah dengan menambahkan icon pada respon atau deskripsi yang ingin disajikan (lihat Gambar 4.17) atau membuat diagram grafis untuk menjelaskan konsep dan proses yang sangat kompleks serta membutuhkan penjelasan yang komprehensif (lihat Gambar 4.18) (Emery, 2014)

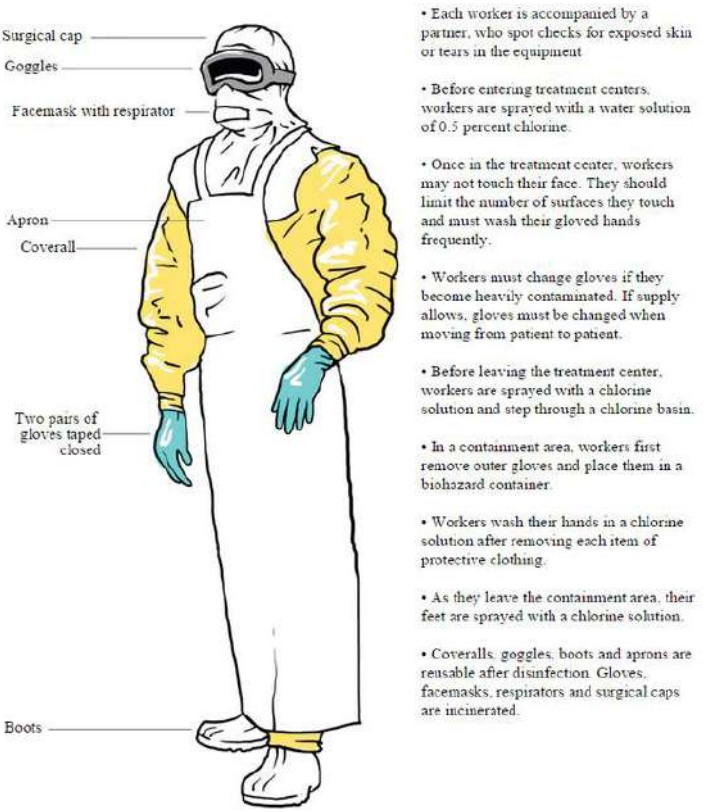


Gambar 4.15 Penyajian Data Dengan Showcase
(Sumber : Emery, 2014)

Gambar 4.16 Visualisasi Data Dengan Menambahkan Foto Narasumber
(Sumber : Emery, 2014)



Gambar 4.17 Visualisasi Data Dengan Menambahkan Icon
(Sumber : Emery, 2014)



Gambar 4.18 Visualisasi Data Untuk Representasi Proses atau Prosedur yang Kompleks
(Sumber : Emery, 2014)

BAB V
PLATFORM KAGGLE UNTUK AKSES R

5.1.Pengenalan Platform Kaggle

Kaggle merupakan platform komunitas online yang digunankan oleh ilmuwan data dan penggemar pembelaran mesin diseluruh dunia. Kaggle pertama kali ditemukan pada 2010 oleh Anthony Goldbloom dan Ben Hammer. Kemudian pada 2017 diakuisi oleh Google. Sampai tahun 2022 platform ini memiliki sebanyak 50.000 dataset yang bersifat publik dan dapat digunakan untuk kebutuhan seperti latihan mesin pembelajaran atau riset akademis. Menyediakan source code yang dibutuhkan ilmuwan data untuk bekerja dengan datanya.

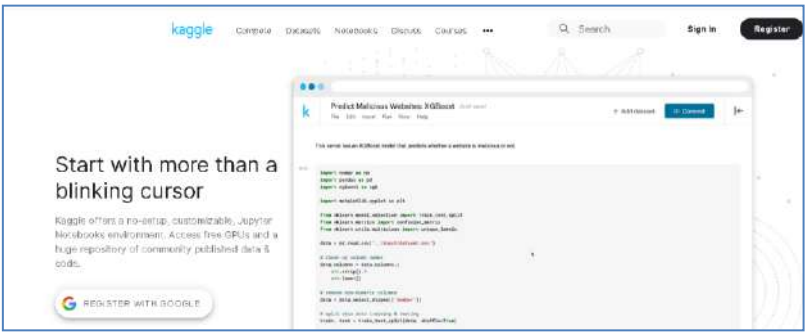
Tujuan platform ini dibuat adalah untuk menyediakan ruang bagi pembelajar dan professional untuk mencapai tujuan mereka dalam memperdalam ilmu data serta pembelajaran mesin. Kaggle dapat digunakan baik oleh pemula maupun pakar karena kemudahan penggunaannya. Platform ini mendukung dua buah tools data analis dan pembelajaran mesin, yakni Python dan Bahasa R. Powerfull untuk melakukan visualisasi data, pengolahan citra, pemrosesan bahasa alami, cluster analisis, analisis klasifikasi, reduksi dimensi data, design pembelajaran mesin dan masih banyak lagi.

Platform Kaggle bekerja dengan mode online saat digunakan. Untuk menggunakan setiap library (misal Bahasa R), tidak lagi dibutuhkan installasi package seperti saat menggunakan aplikasi R Studio secara stand alone. R pada Kaggle sudah mendukung hampir semua library yang dibutuhkan sehingga saat akan digunakan tinggal dipanggil. Selain untuk data science, data mining dan kecerdasan artifisial, Kaggle juga digunakan sebagai platform penyelenggaraan

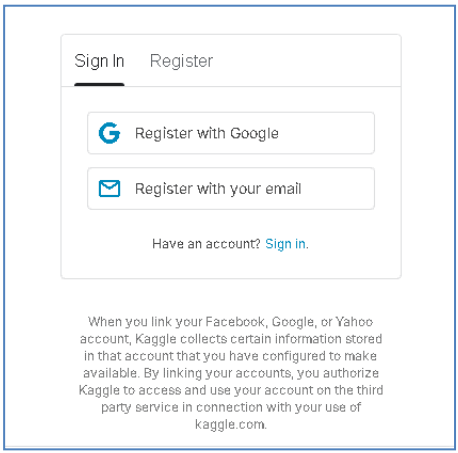
kompetisi, berbagi dan uji dataset, diskusi dengan pengguna lain diseluruh dunia bahkan dapat mengikuti kursus gratis yang disediakan oleh Kaggle dan mendapatkan sertifikat gratis.

5.2.Memulai Kaggle

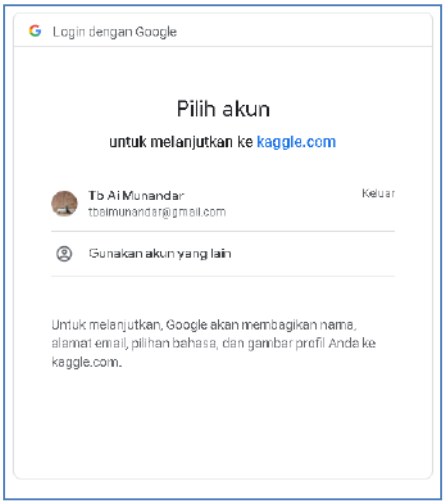
1. Pendaftaran Akun Kaggle
    1) Buka browser (disarankan Chrome), kemudian ketikan alamat URL → https://www.kaggle.com/ sehingga muncul halaman utama sebagai berikut :



    2) Jika belum memiliki akun, maka silahkan melakukan pendaftaran dengan memilih tombol **Register**. Disarankankan sudah memiliki akun e-mail google, dan akun email sedang aktif (mode login email), sehingga pendaftaran cukup dengan **Register with Google** sebagai berikut (Pastikan berada pada Tab **Register**):

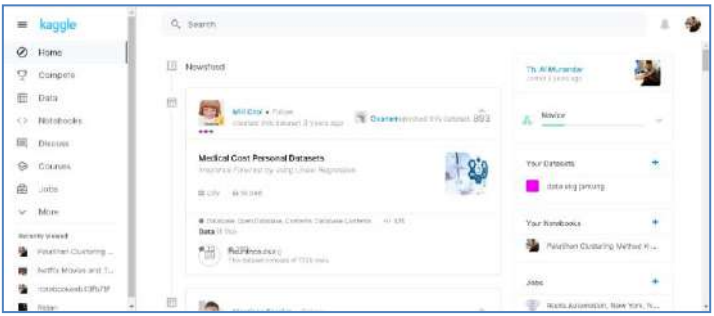


3) Setelah memilih pilihan **Register with Google** maka akan muncul halamn untuk memilih akun email Google sebagai berikut :



4) Pilih Akun email google yang dikehendaki, kemudian di klik lalu tunggu beberapa saat untuk sistem akan melengkapi pendaftaran secara otomatis dan jika berhasil akan diarahkan

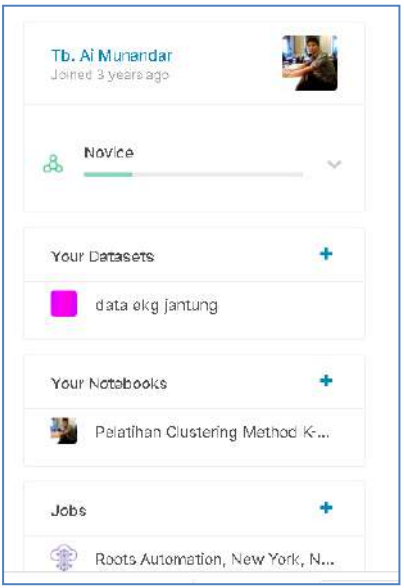
pada halaman dashboard Kaggle.com milik kita, seperti pada gambar di bawah ini :



5) Selamat!! Akun Kagle sudah siap digunakan. Anda bisa melakukan proses data mining baik menggunakan bahasa R ataupun Phyton melalui menu **< > Notebook.**

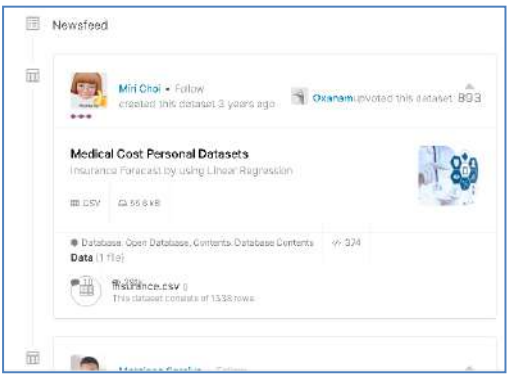
2. Pengenalan Ruang Lingkup Dashboard Kaggle
   1) Layer Profil

   Memuat informasi Profil Pengguna, informasi **Novice** atau Berita yang memberikan informasi terkait aktifitas apa saja yang sudah dilakukan. **Your Datasets** memuat informasi data yang pernah disimpan dalam kaggle. **Your Notebooks** memuat informasi file-file atau script yang pernah dibuat dan masih tersedia. **Jobs** memuat informasi lowongan pekerjaan terkait bidang data mining, data science dan sejenisnya dari berbagai negara



2) Layer News feed

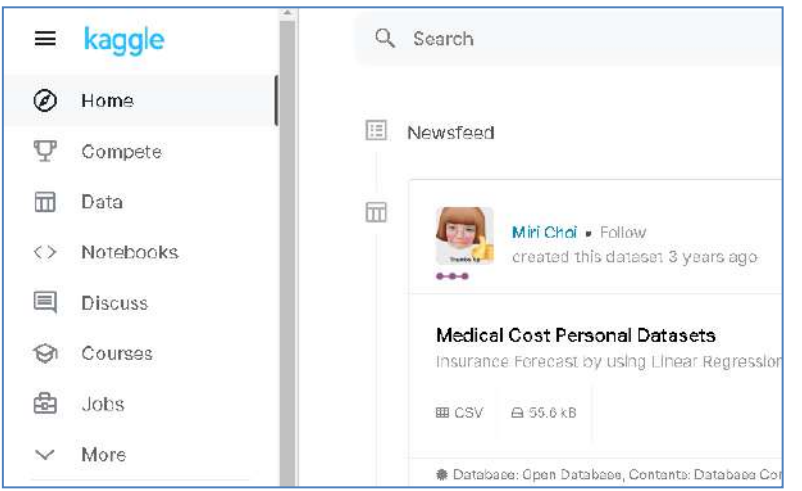
Memuat informasi berita-berita tentang perkembangan dunia data mining serta dataset terbaru yang dibagikan untuk publik serta informasi lainnya.



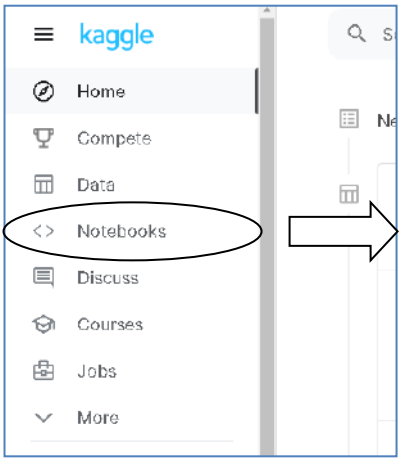
3) Layer Menu Dashboard

Home memuat informasi umum pada halaman dashboard. **Compete** memuat informasi kompetisi terkait data mining. **Data**

memuat informasi terkait data set yang dapat digunakan untuk latihan atau analisis. **Notebooks** memuat halaman kerja analisis data mining baik menggunakan Phyton maupun R. **Discuss** memuat informasi dan threat diskusi yang dilakukan sesama pengguna Kaggle dari berbagai negara dan berbagai kategori diskusi. **Courses** memuat informasi pelatihan atau kursus yang tersedia di Kaggle dan **Jobs** memuat informasi tawaran atau lowongan pekerjaan dari berbagai belahan dunia

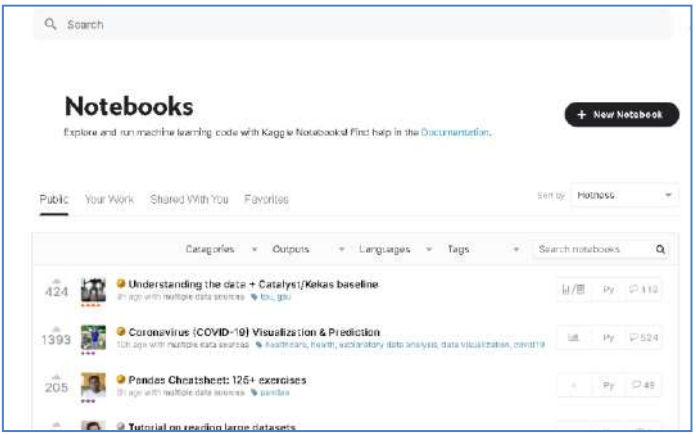


3. Akses R Programming Pada Kaggle
   1) Pada menu <> **Notebooks** klik menu tersebut (perhatikan gambar yang dilingkaran).



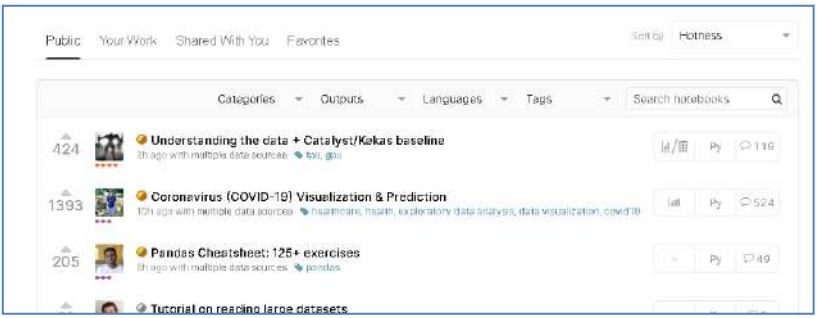
Klik pada menu **<> Notebooks**

2) Muncul halaman dashboard **Notebooks** sebagai berikut :

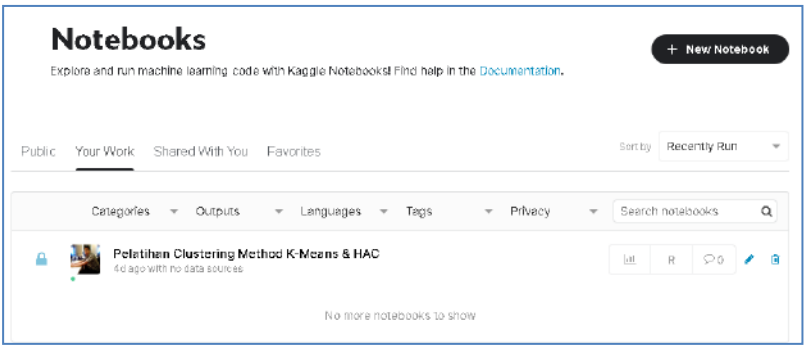


Terdapat empat tab opsi ruang kerja pada dashboard **Notebooks** yaitu **Public, Your Works, Shared With You** dan **Favorites**. Berikut deskripsi untuk masing-masing tab ruang kerja:

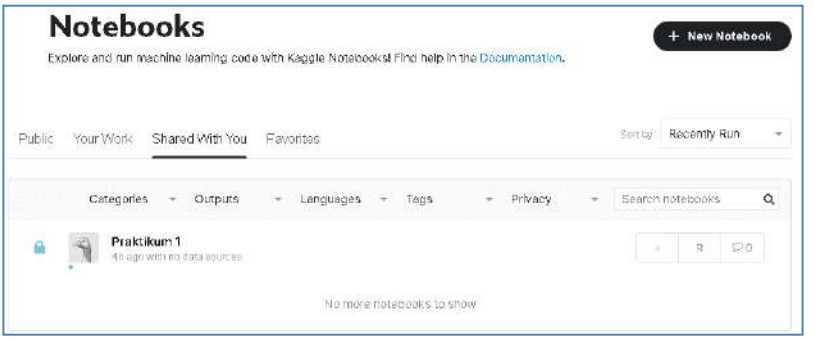
a. **Public** memuat informasi aktifitas atau kegiatan yang dibagikan oleh user yang tergabung pada Kaggle dan dapat diakses secara publik.



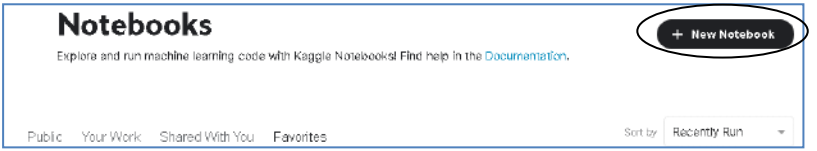
b. **Your Works** memuat informasi aktifitas analisis dan sebagainya yang sudah pernah dibuat atau dikerjakan. Jika sudah memiliki aktifitas maka daftar aktifitas akan muncul seperti pada gambar di bawah :



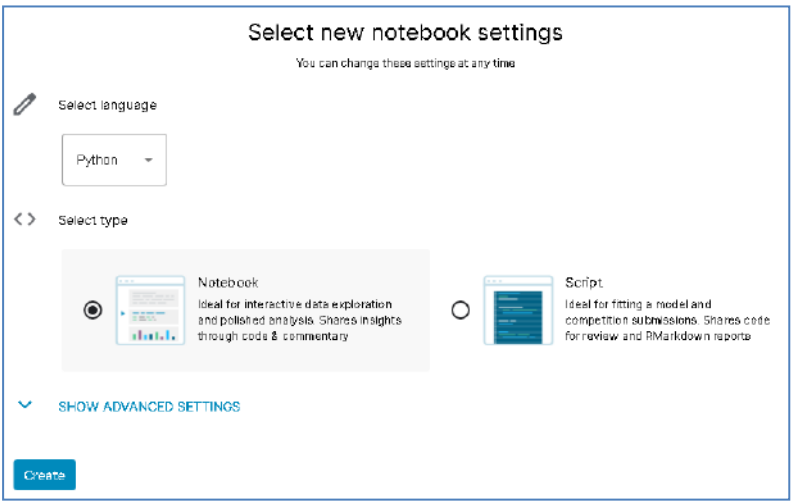
c. **Shared with You** memuat aktifitas atau latihan atau tugas yang di share oleh user lain. Tampilan halaman seperti pada gambar di bawah ini :



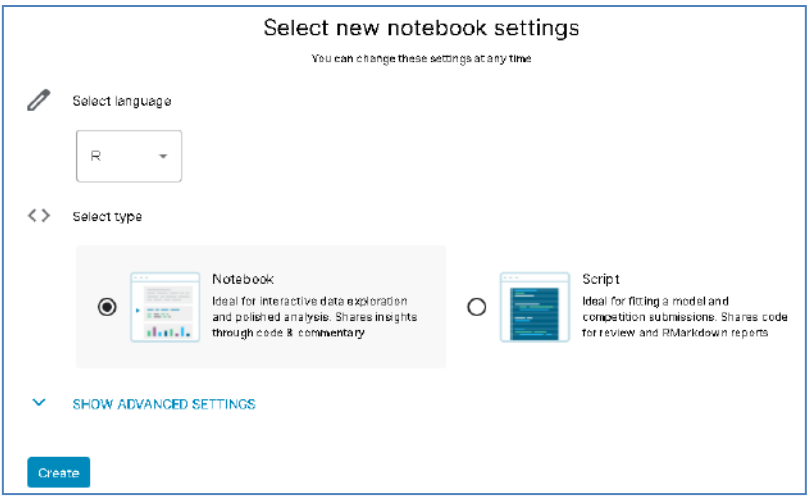
3) Untuk membuat file R yang baru, klik tombol **New Notebook** yang berada pada bagian atas tab dashboarb seperti pada gambar di bawah:



4) Sehingga terbuka halaman utama **Select New Notebook Settings** sebagai berikut :



5) Pada option **Select Language** klik drop down kemudian pilih **R,** pada option Select Type pilih option **Notebook** kemudian klik tombol **Create** yang berada di bawah halaman (perhatikan yang dilingkari).



Keterangan :

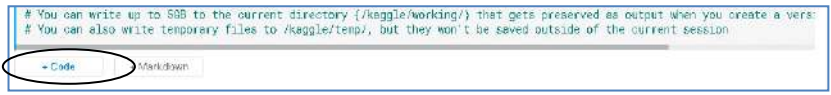
Jenis Type :

a. **Notebook** digunakan untuk eksplorasi data secara interaktif. Output, komentar dan source code bisa muncul dalam satu halaman console
b. **Scripts** digunakan untuk murni eksplorasi model dan pengiriman kompetisi data mining atau sejenis. Script bisa di review dan dibuatkan report RMarkdown nya.

6) Halaman console **R Programming** sebagai berikut

Informasi sesi, data yang disimpan, setingan lingkungan script R dan sebagainya

7) Untuk menambahkan console script, klik tombol **+ Code** yang terletak pada console script sebelumnya., perhatikan bagian yang dilingkari.

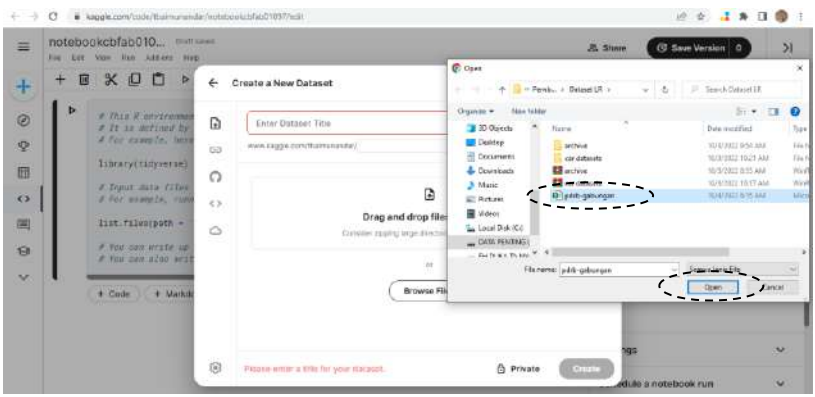


8) Sehingga menghasilkan line script baru seperti diperlihatkan pada gambar di bawah ini :

5.3.Import Dataset Ke Dalam Kaggle

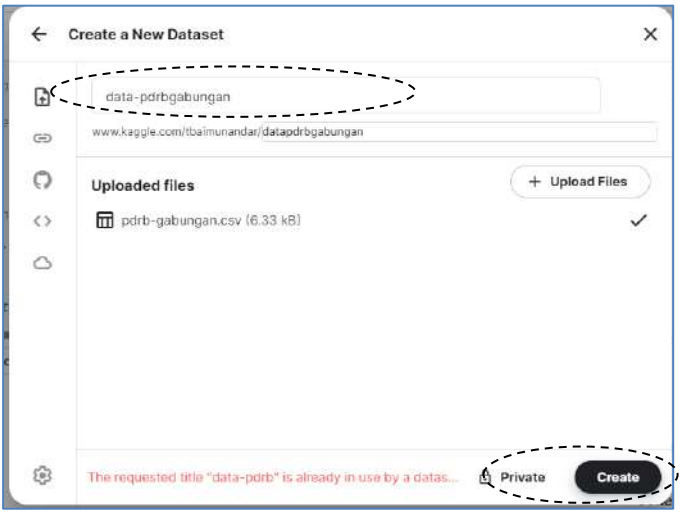
1. Buka notebook untuk bekerja dengan R. Kemudian pada jendela toolbox **Data** (pada bagian kanan layar) klik icon **submit** untuk menambahkan data dari berbagai sumber (termasuk dari folder kerja laptop / komputer anda). Perhatikan gambar yang diingkari di bawah :

2. Setelah diklik icon submit maka muncul kotak dialog Create a New Dataset seperti di bawah, kemudian klik tombol Browse untuk mengambil data dari folder kerja. Setelah diklik tombol Browse maka akan terbuka kotak dialgog Open. Arahkan ke folder kerja tempat dimana file dataset di simpan, klik atau pilih dataset kemudian klik tombol Open. Perhatikan gambar di bawah :

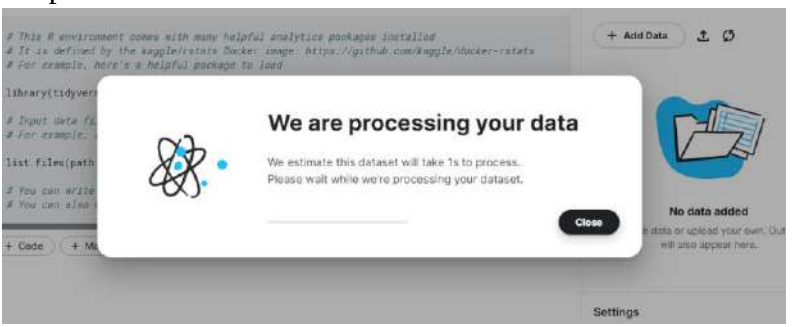


3. Setelah data dipilih, maka akan Kembali ke kotak dialog Create a New Dataset. Beri judul dataset dengan ketentuan Panjang judul antara 6 sampai 50 karkater. Sehingga akan muncul link / URL dataset pada bagian bawah judul dataset. Pada kotak dialog ini,

visibilitas data bisa diatur apakah akan dibuat Public atau Private dengan mengklik tombol dengan icon kunci di samping tombol Create. Perhatikan gambar di bawah :



4. Saat tombol Create diklik maka proses import dataset sedang berlangsung. Biarkan sampai proses selesai atau layar di bawah ini tertutup.



5. Jika proses import data sudah selesai, silahkan lihat pada jendela toolbox Data dibagian kanan, akan ada tambahan folde baru yakni input yang berisi dataset dengan judul yang sudah kita tentukan

pada langkah sebelumnya, serta output yang merupakan folder kerja URL yang diberikan Kaggle. Anda dapat mengklik nama dataset untuk menampilkan isinya pada jendela console. Perhatikan gambar di bawah :

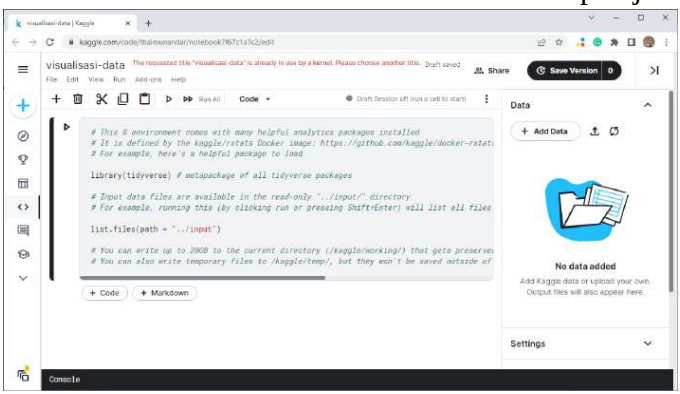
6. Setelah selesai mengimport dataset dan memeriksa isi dataset, silahkan berselancar melakukan analisis data menggunakan data tersebut dengan R.

BAB VI
VISUALISASI DATA MENGGUNAKAN R

Untuk memvisualisasikan data pada bahasa R, dibutuhkan library caret yang menyimpan fungsi untuk menyajikan data ke dalam grafik, khususnya untuk yang bersifat univariat. Sedangkan untuk menyajikan data yang bersifat multivariat, dapat menggunakan library corrplot. Pada pembahasan ini, untuk melakukan visualisasi digunakan data analis kredit yang terdiri atas tiga belas variabel / atribut. Data diberinama **dataset-pinjaman-nasabah.csv** yang tersimpan dalam folder kerja agar mudah untuk digunakan.

6.1. Menyiapkan Dataset

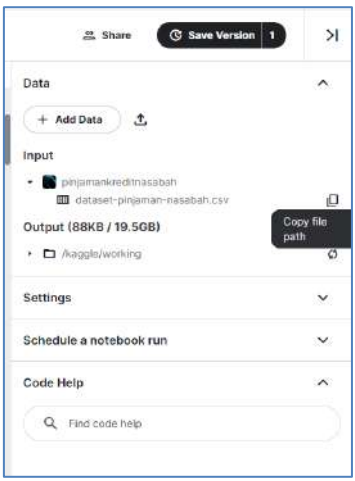
Untuk memulai visualisasi data menggunakan Bahasa R, langkah pertama adalah login ke dalam kaggle.com menggunakan akun masing-masing. Kemudian pada manu navigator bagian kiri pilih menu **< > Code** kemudian klik tombol **New Notebook** sampai terbuka halaman seperti pada Gambar 6.1. Untuk lebih jelas bagaimana membuat akun dan membuka notebook baru silahkan pelajari Bab 5.



Gambar 6.1 Tampilan Notebook R pada Kaggle.

Pada halaman notebook library **tidyverse** secara otomatis akan dipanggil setiap kali membuat file baru. Tidyverse sendiri merupakan library yang memuat kumpulan package R yang dirancang untuk kebutuhan data science. Ada delapan package utama tidyverse yang dapat digunakan, yaitu **ggplot2, dplyr, tidyr, readr, purr, tibble, stringr** dan **forcats**.

Klik tombol + **Code** disamping tombol + **Markdown** untuk membuat line code baru, kemudian buat sebuah variable baru untuk menampung datasets, misal diberi nama **datanasabah**. Gunakan fungsi **read.csv()** untuk memanggil datasets yang sudah disiapkan. Pada fungsi **read.csv()** kita membutuhkan file path datasets. Untuk mendapatkan file path tersebut, arahkan kursos pada toolbox **Data**, kemudian pada opsi **Input** klik dataset sampai muncul level terakhir datasets dengan ekstensi file *.csv kemudian arahkan kursor pada bagian kanan datasets sampai muncul icon copy file path seperti diperlihatkan pada Gambar 6.2. Untuk menambahkan dataset ke dalam toolbox **Data** silahkan pelajari Bab 5.



Gambar 6.2 Menyalin File Path Datasets

Berikut adalah code lengkap untuk pemanggilan datasets menggunakan file path yang sudah disalin dari toolbox **Data.**

```
datanasabah=read.csv('../input/pinjamankreditnasabah/dataset-pinjaman-nasabah.csv', sep=',')
head(datanasabah[,6:12])
```

A data.frame: 6 × 7

	WIraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman	JangkaWaktuPinjaman	Credit_History	WilayahTempatTinggal
	<chr>	<int>	<int>	<int>	<int>	<int>	<chr>
1	No	5849	0	NA	360	1	Urban
2	No	4583	1508	128	360	1	Rural
3	Yes	3000	0	66	360	1	Urban
4	No	2583	2358	120	360	1	Urban
5	No	6000	0	141	360	1	Urban
6	Yes	5417	4196	267	360	1	Urban

+ Code + Markdown

Running code di atas dengan menekan icon **Play** pada bagian kiri source code, atau menekan tombol kombinasi **CTRL + ENTER.** Pada baris kode pertama setelah file path datasets, terdapat sebuah fungsi **sep=’,’** yang berfungsi untuk menentukan jenis pemisah setiap kolom data pada file *.csv. Untuk mengetahui separator (pemisah) antar kolom, sebelum dipanggil ada baiknya dicek terlebih dahulu, apakah menggunakan koma (,), titik koma (;), tab atau spasi atau bentuk lainnya. Pada buku ini, separator antar kolom menggunakan koma (,), oleh karenanya pada fungsi **sep** digunakan koma (,).

Fungsi **head(namadatasets)** dalam hal ini **head(datanasabah[,6:12])** digunakan untuk menampilkan enam data pertama dari datasets. Sebetulnya untuk menampilkan enam data pertama cukup menggunakan perintah **head()** saja tanpa kombinasi lain. Pada buku ini kombinasi **datanasabah[,6:12]** merupakan konfigurasi untuk menampilkan keseluruhan dari enam data pertama dimulai dari variable ke 6 sampai 12. Penyajian ini untuk lebih

mempermudah tampilan dataset karena jumlah variable sebanyak 13 dan terindikasi tidak tersajikan di layar untuk kebutuhan buku.

Untuk mengetahui struktur seperti jumlah data instance (observasi), jumlah variable, urutan variavel, tipe variable, serta sample data instance yang ada pada setiap variabel, kita dapat menggunakan fungsi **str()** atau kependekan dari **structure().** Pemanggilan fungsi **str()** diikuti dengan nama datasets yang sudah disiapkan. Pada buku ini kita menyiapkan datasets yang disimpan pada variable **datanasabah.** Dengan demikian penulisan fungsi secara lengkap menjadi **str(datanasabah).** Berikut baris kode yang bisa digunakan.

```
str(datanasabah)
```

```
'data.frame':   614 obs. of  13 variables:
 $ ID_Nasabah          : chr  "LP001002" "LP001003" "LP001005" "LP001006" ...
 $ JenisKelamin        : chr  "Male" "Male" "Male" "Male" ...
 $ StatusPernikahan    : chr  "No" "Yes" "Yes" "Yes" ...
 $ JumTanggungan       : int  0 1 0 0 0 2 0 3 2 1 ...
 $ Pendidikan          : chr  "Graduate" "Graduate" "Graduate" "Not Graduate" ...
 $ Wiraswasta          : chr  "No" "No" "Yes" "No" ...
 $ IncomeNasabah       : int  5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...
 $ IncomePasangan      : int  0 1508 0 2358 0 4196 1516 2504 1526 10968 ...
 $ JumlahPinjaman      : int  NA 128 66 120 141 267 95 158 168 349 ...
 $ JangkaWaktuPinjaman : int  360 360 360 360 360 360 360 360 360 360 ...
 $ Credit_History      : int  1 1 1 1 1 1 1 0 1 1 ...
 $ WilayahTempatTinggal: chr  "Urban" "Rural" "Urban" "Urban" ...
 $ StatusPinjaman      : chr  "Y" "N" "Y" "Y" ...
```

+ Code + Markdown

Selain mengetahui struktur datasets, kita juga dapat menyajikan statistik deskripsi sederhana menggunakan **summary(namadatasets)** menggunakan baris kode sebagai berikut :

```
summary(datanasabah)
```

```
 ID_Nasabah        JenisKelamin       StatusPernikahan   JumTanggungan
Length:614         Length:614         Length:614         Min.   :0.0000
Class :character   Class :character   Class :character   1st Qu.:0.0000
Mode  :character   Mode  :character   Mode  :character   Median :0.0000
                                                         Mean   :0.7629
                                                         3rd Qu.:2.0000
                                                         Max.   :3.0000
                                                         NA's   :15
 Pendidikan         Wiraswasta        IncomeNasabah    IncomePasangan
Length:614         Length:614         Min.   :  150   Min.   :    0
Class :character   Class :character   1st Qu.: 2878   1st Qu.:    0
Mode  :character   Mode  :character   Median : 3812   Median : 1188
                                      Mean   : 5403   Mean   : 1621
                                      3rd Qu.: 5795   3rd Qu.: 2297
                                      Max.   :81000   Max.   :41667

JumlahPinjaman  JangkaWaktuPinjaman Credit_History   WilayahTempatTinggal
Min.   :  9.0   Min.   : 12         Min.   :0.0000   Length:614
1st Qu.:100.0   1st Qu.:360         1st Qu.:1.0000   Class :character
Median :128.0   Median :360         Median :1.0000   Mode  :character
Mean   :146.4   Mean   :342         Mean   :0.8422
3rd Qu.:168.0   3rd Qu.:360         3rd Qu.:1.0000
Max.   :700.0   Max.   :480         Max.   :1.0000
NA's   :22      NA's   :14          NA's   :50
StatusPinjaman
Length:614
Class :character
Mode  :character
```

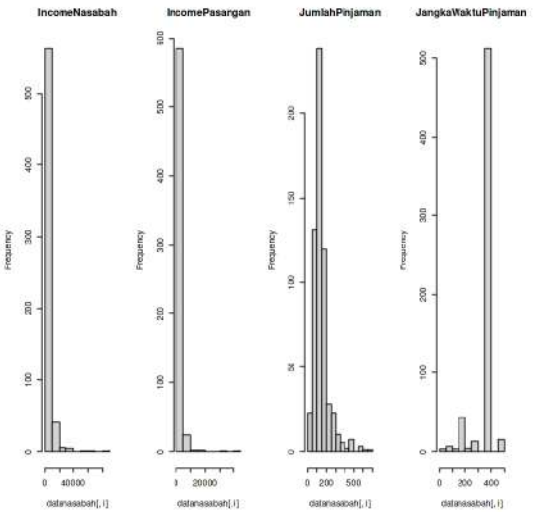
6.2. Visualisasi Univariat

Visualisasi plot univariat bertujuan untuk menampilkan distribusi data dari suatu atribut secara individual tanpa melibatkan interaksi apapun antar atribut. Biasanya visualisasi univariat lebih kepada memvisualisasikan kecenderungan utama suatu data serta penyebaran value attribute nya. Ada beberap jenis visualisasi data univariat antara lain :

1. **Visualisasi Histogram**

   Memvisualisasikan data bertipe numerik dengan menampilkan distribusi data dari suatu atribut yang dipilih. Berikut baris kode dan hasil visualisasi nya :

```
par(mfrow=c(1,4))
for (i in 7:10){hist(datanasabah[,i], main=names(datanasabah)[i])}
```

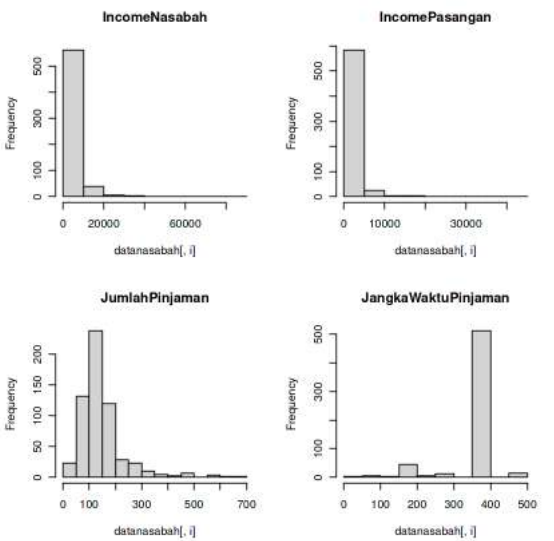


Fungsi **par(mfrow=c())** digunakan untuk mengatur layout dari grafik yang disajikan dengan membuat sebuah multi-panel. Sedang fungsi mfrow() digunakan untuk membentuk baris dan kolom tempat grafik disajikan. Argumen pertama merupakan jumlah baris, dan argumen kedua merupakan jumlah kolom. Pada baris kode di atas fungsi **par(mfrow=c(1,4))** berarti kita sedang membuat satu panel untuk grafik dengan ukuran 1 baris dan 4 kolom. Sebab kita akan menyajikan grafik sebanyak empat variable dalam satu baris grafik.

Untuk dapat menyajikan grafik histogram, fungsi yang digunakan adalah **hist()** yang diikuti dengan nama datasets kemudian diberi judul untuk setiap variable pada komponen **main.** Sedangkan untuk perintah **for i in 7 :10** merupakan perintah untuk memanggil variable ke 7 sampai kesepuluh yang

akan disajikan ke dalam plot. Adapun variable ke 7 sampai ke sepuluh adalah IncomeNasabah, IncomePasangan, JumlahPinjaman dan JangkaWaktuPinjaman sesuai dengan hasil grafik yang ditampilkan pada gambar di atas. Kita juga bisa mengatur penyajian grafik dengan tampilan dua baris dua kolom atau lainnya sesuai kebutuhan. Untuk kebutuhan tersebut kita dapat memodifikasi perintah **par(mfrow=c()).** Berikut contoh baris kode untuk menyajikan grafik histogram ke dalam dua baris dan dua kolom.

```
par(mfrow=c(2,2))
for (i in 7:10){hist(datanasabah[,i], main=names(datanasabah)[i])}
```

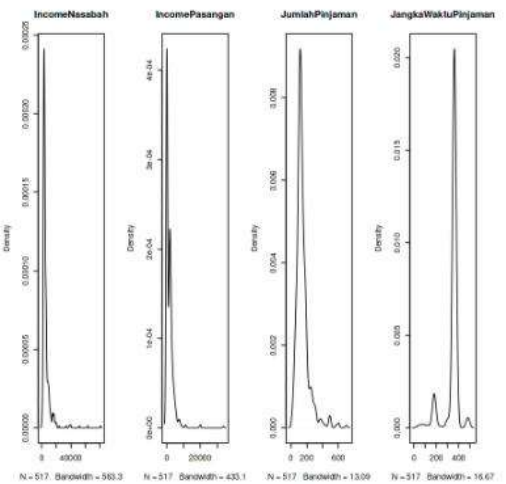


2. Density Plot

Bentuk lain dari grafik jenis histogram namun lebih rapat sehingga penggambaran distribusi dari masing-masing variabel lebih abstrak. Data yang diolah harus dalam bentuk numerik dan tidak diperkenankan memuat data yang bernilai N/A. Oleh

karenanya perlu dilakukan pre processing data terlebih dahulu menggunakan fungsi na.omit sebagai berikut :

```
visNasabah=na.omit(datanasabah)
par(mfrow=c(1,4))
for (i in 7:10){plot(density(visNasabah[,i]), main=names(visNasabah)[i])}
```



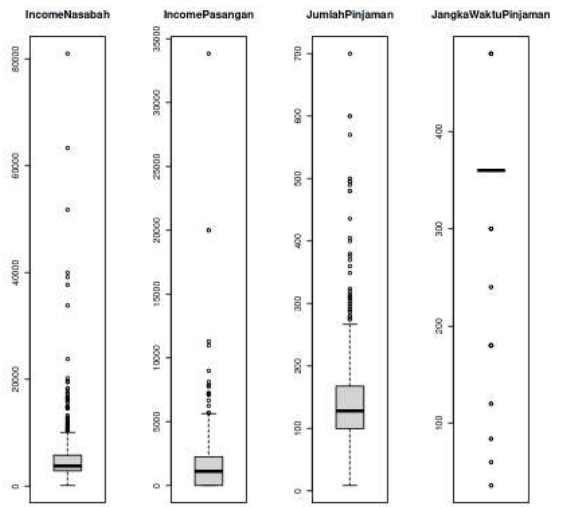
Fungsi na.omit(datasets) digunakan untuk menghapus semua data instance yang valuenya tidak lengkap. Fungsi ini dapat digunakan untuk objek data bertipe data frame, matrix atau vector. Pada buku ini, dataset awal **datanasabah** dilakukan pre-processing berupa penghapusan data instance dengan value Not Available (N/A) dan disimpan dalam variable baru dengan nama **visNasabah.** Untuk menyajikan plot ke dalam bentuk density, kita dapat menggunaka fungsi **plot(density()).**

3. Box And Whisker Plots

Grafik jenis ini menyajikan kumpulan data bertipe numerik berdasarkan informasi statistik seperti nilai minimu, maksimun, median, kuartil pertama, dan kuartil ketiga. Fungsi yang

digunakan adalah **boxplot()**. Berikut baris kode yang digunakan untuk menyajikan data dalam bentuk box and whisker plots.

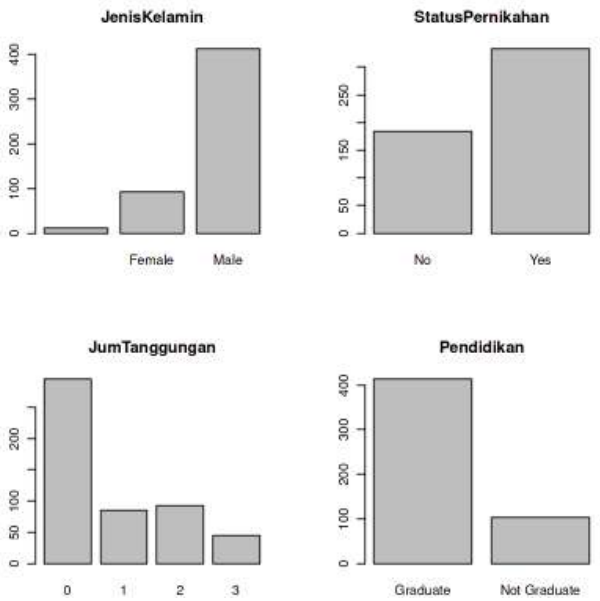
```
par(mfrow=c(1,4))
for (i in 7:10){boxplot(visNasabah[,i], main=names(visNasabah)[i])}
```



4. Barplots

Barplot mampu memvisualisasikan data baik yang bernilai numerik maupun non numerik sehingga memberikan gambaran tentang proporsi instance yang dimiliki oleh setiap kategori (atribut) data. Untuk membuat grafik Barplot dapat dilakukan menggunakan fungsi **barplot().** Beriku adalah baris kode untuk menyajikan data ke dalam bentuk bar plot.

```
par(mfrow=c(2,2))
for (i in 2:5){counts= table(visNasabah[,i])
               name = names(visNasabah)[i]
                barplot(counts, main=name)}
```



6.3. Visulasisasi Multivariat

Plot Multivariat merupakan visualisasi grafis yang memperlihatkan hubungan atau interaksi antar atribut sehingga dapat dipelajari pola distribusi data antar atribut, kecenderungan utama serta sebaran kelompok data yang terjadi antara satu atau lebih atribut. Visualisasi multivariat beberapa diantaranya adalah :

1. Correlation Plot

Plot korelasi biasanya digunakan untuk memvisualisasikan matriks korelasi antar atribut sehingga dapat diketahui atribut mana saja yang megalami perubahan bersama-sama. Sebelum membuat plot korelasi, terlebih dahulu setiap value variabel dihitung nilai korelasi antar variabel yang ada kemudian baru divisualisasikan menggunakan fungsi **corrplot().** Perhitungan nilai korelasi antar variabel dilakukan menggunakan fungsi **cor().**

Luaran dari fungsi **cor()** berupak matrik korelasi yang memperlihatkan hubungan antara satu variabel dengan variabel lainnya berdasarkan nilai antara -1 sampai +1. Nilai +1 merupakan representasi korelasi positif sempurna, sedangkan -1 merepresentasikan korelasi negatif sempurna, sedangkan 0 berarti tidak memiliki korelasi sama sekali. Yang perlu diingat adalah, fungsi **cor()** hanya akan dapat digunakan untuk tipe data numerik. Berikut adalah baris kode untuk menghitung korelasi antar variable.

```
korelasi=cor(visNasabah[,7:10])
korelasi
```

A matrix: 4 × 4 of type dbl

	IncomeNasabah	IncomePasangan	JumlahPinjaman	JangkaWaktuPinjaman
IncomeNasabah	1.00000000	-0.124310229	0.57044064	-0.063364126
IncomePasangan	-0.12431023	1.000000000	0.15747169	0.002262917
JumlahPinjaman	0.57044064	0.157471687	1.00000000	0.022322294
JangkaWaktuPinjaman	-0.06336413	0.002262917	0.02232229	1.000000000

+ Code + Markdown

Baris kode di atas menyajikan perhitungan korelasi empat buah variable **visNasbah[, 7:10]** yaitu variable ke 7 sampai 10 dari datasets **visNasabah.** Menghasilkan matrik korelasi yang memperlihatkan kekuatan hubungan antar variable.

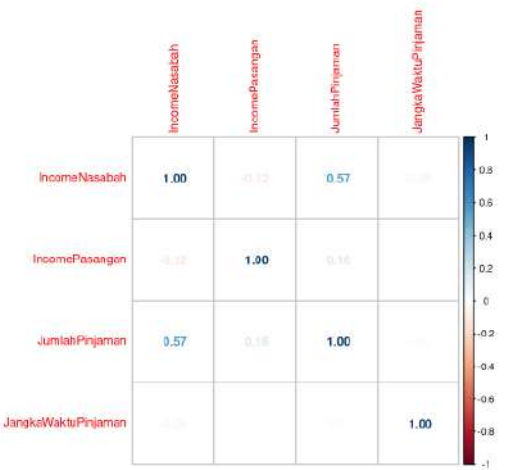
Untuk menyajikan nilai korelasi ke dalam bentuk plot, kita dapat menggunaka fungsi **corrplot().** Sebelum fungsi ini dipanggil maka terlebih dahulu panggil library nya dengan perintah **library(corrplot).** Pada fungsi **corrplot()** terdapat beberapa parameter penting yang harus dipahami, antara lain **method**, dan **type**. Pada parameter method terdiri atas tujuh teknik visualisasi nilai korelasi. Ketujuh teknik tersebut adalah circle, square, ellipse, number, shade, color dan pie. Berikut adalah penjelasan untuk masing-masing teknik pada parameter method.

Nama teknik	Penjelasan
circle dan square	Nilai absolut dari koefisien korelasi yang sesuai disajikan dalam area lingkaran atau bujur sangkar.
ellipse	Nilai korelasi diskalakan secara parametik secara eksentrisitas
number	Penyajian nilai korelasi dalam bentuk angka dengan warna berbeda
shade	Sama dengan color namun koefisien negatif glif diberi arsir
color	Panyajian nilai korelasi dalam bentuk bujur sangkar dengan warna berbeda tanpa informasi angka
pie	Dalam bentuk grafik pie

Sedangkan untuk parameter type, ada tiga teknik yang digunakan yaitu full, upper dan lower. Berikut adalah baris kode penggunaan corrplot dengan parameter berbeda.

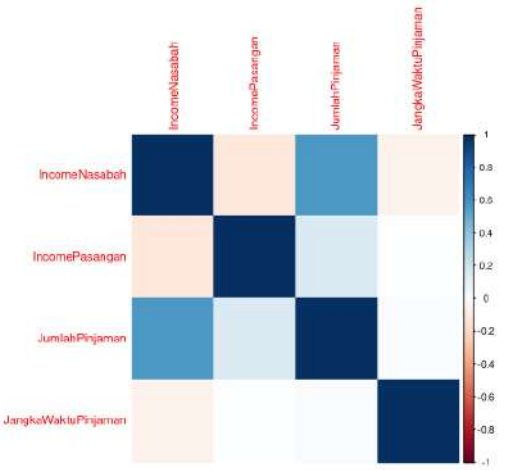
- Plot korelasi antar variable menggunakan '**number**'

```
library(corrplot)
corrplot(korelasi, method='number')
```



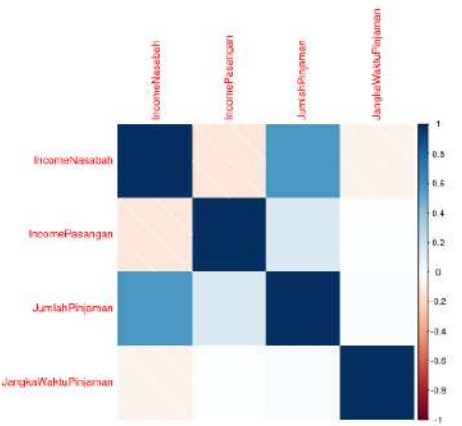
- Plot korelasi antar variable menggunakan '**color**'

```
library(corrplot)
corrplot(korelasi, method='color')
```



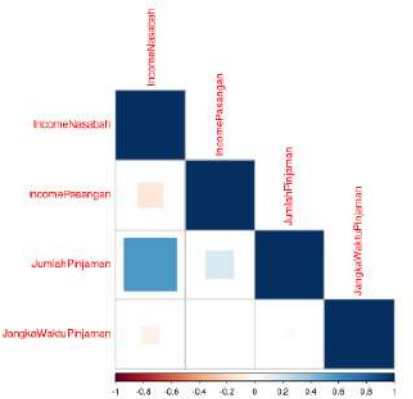
- Plot korelasi antar variable menggunakan '**shade**'

```
library(corrplot)
corrplot(korelasi, method='shade')
```



- Plot korelasi antar variable menggunakan '**square**' kombinasi parameter type **'lower'**

```
library(corrplot)
corrplot(korelasi, method='square', type='lower')
```

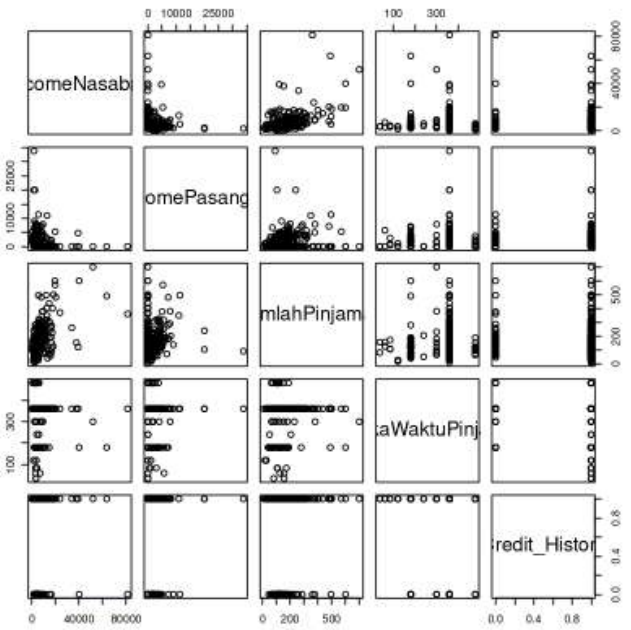


2. Scatterplot Matrix

Scatter plot biasanya memvisualisasikan dua variabel secara bersamaan, dimana variabel pertama pada sumbu x dan yang kedua pada sumbu y dengan titik-titik yang menunjukan interaksi antar kedua variabel. Penyebaran titik pada grafik memperlihatkan hubungan yang terjadi antar atribut. Pada scatter plot juga kita dapat membuat plot pencar untuk semua pasangan atribut dalam suatu dataset (matriks pencar). Untuk menyajikan data ke dalam scatterplot matix dapat dilakukan menggunakan fungsi **pairs().**

- Menyajikan data lima variable dari dataset **visNasabah** yakni variable 7 sampai 11 bertipe numerik.

```
pairs(visNasabah[,7:11])
```



3. Plot Menggunakan fungsi ggpairs()

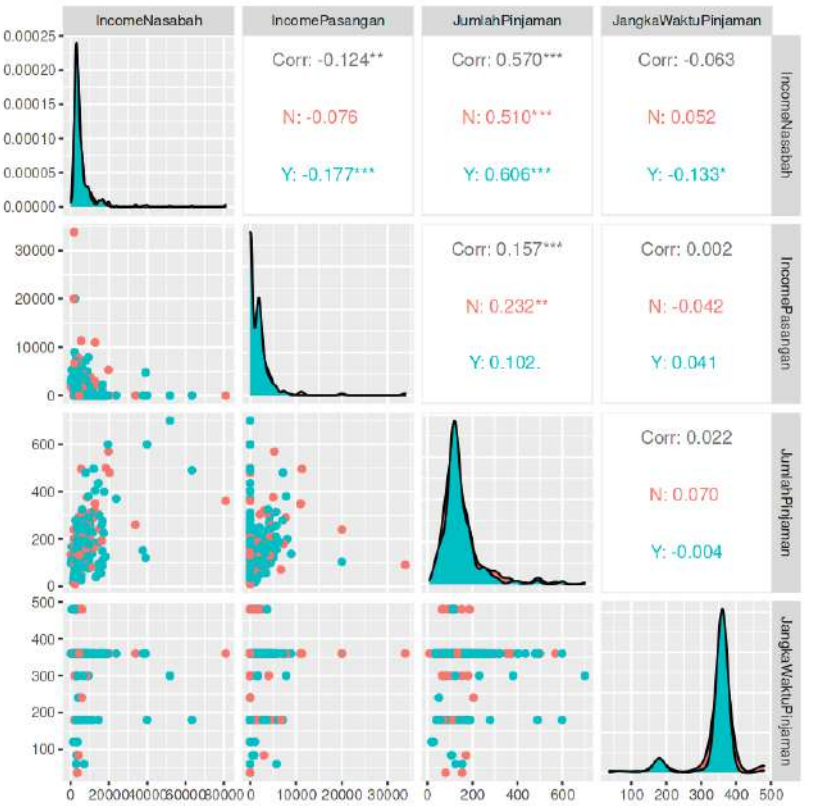
Fungsi **ggpairs()** memungkinkan kita menyajikan data dengan visualisasi yang lebih menarik. Untuk bisa menyajikan visualisasi data menggunakan fungsi **ggpairs()** terlebih dahulu harus membangun matrik scatterplot. Matriks ini kemudian memvisualisasikan pasangan variabel yang kita inginkan terhadap nilai korelasinya pada bagian kiri grafik, serta signifikansi hubungannya pada bagian kanan. Untuk dapat menggunakan fungsi **ggpairs()**, ada tiga library yang harus dipanggil, yaitu **library(ggplot2), library(ggthemes),** dan **library(GGally).**

Pada pembahasan buku ini kita akan menyajikan empat buah variable bertipe numerik dari datasets **visNasabah.** Keempat variable tersebut adalah IncomeNasabah, IncomePasangan, JumlahPinjaman, dan JangkaWaktuPinjaman. Adapun untuk variable StatusPinjaman digunakan sebagai klass untuk melakukan maping terhadap koreasi yang terbentuk dari keempat variable tersebut. Berikut baris kode yang dapat digunakan untuk menyajikan data instance ke dalam bentuk **ggpairs().**

- Memanggil library ggplot2, ggthemes, dan GGally

- Baris kode dan hasil visualisasi

```
ggpairs(data = visNasabah,
        columns = 7:10,
        mapping = aes(col = StatusPinjaman))
```



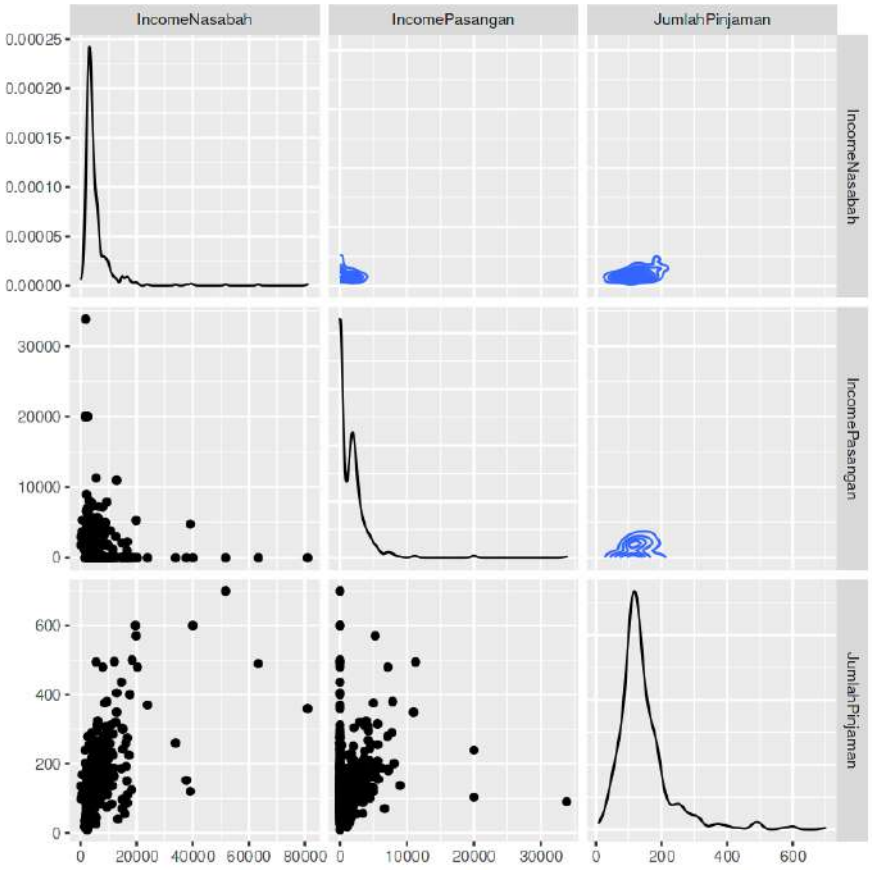
4. Visualisasi Data Plot Density dan Points Sekaligus

Dengan fungsi **ggpairs()** kita juga dapat menyajikan data ke dalam dua bentuk sekaligus. Bagian atas visualisasi dengan bentuk density, dan bagian bawah menggunakan bentuk points. Pada pembahasan buku ini, akan disajikan data variable IncomeNasabah, IncomePasangan dan JumlahPinjaman. Plot terbagi ke dalam segitiga

atas dan bawah. Segitiga atas memuat visualisasi dalam bentuk density, sedangkan bagian bawah dalam bentuk scatters (points). Berikut baris kode yang digunakan untuk menyajikan data ke dalam dua bentuk plot.

```
ggpairs(visNasabah[,7:9],
upper = list(continuous = 'density', combo = 'box_no_facet'),
lower = list(continuous = 'points', combo = 'dot_no_facet'))
```

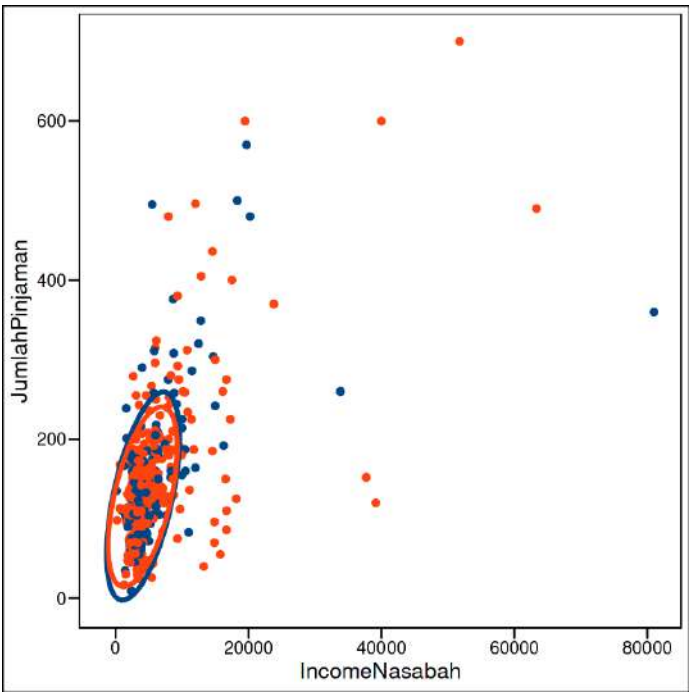
Hasil dari baris kode tersebut sebagai berikut.



5. **Plot Manual terhadap data menggunakan ggplot2() dan gridExtra()**

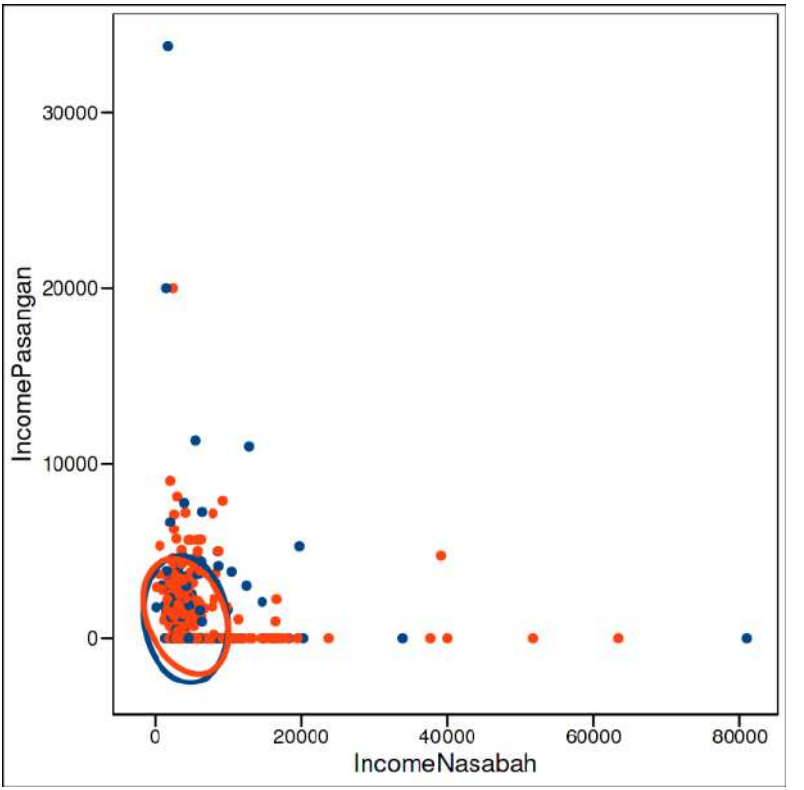
Dengan funsgsi ggplot() dan gridExtra() kita juga dapat memvisualisasikan data ke dalam beberapa tampilan sekaligus sesuai kebutuhan. Pada pembahasan buku ini, akan disajikan dua buah plot dalam layout grid menggunakan **gridExtra().** Plot yang pertama memvisualisasikan hubungan antara variable IncomeNasabah dan JumlahPinjaman terhadap Variabel StatusPinjaman dari datasets **visNasabah.** Baris kode yang bisa digunakan sebagai berikut :

```
ggplot(visNasabah, aes(IncomeNasabah,JumlahPinjaman, col= StatusPinjaman)) +
  geom_point(size= 2, show.legend = F) +
  stat_ellipse(size= 1.3, linetype=1, show.legend = F) +
  scale_color_calc()
```



Untuk plot yang kedua, menyajikan grafik hubungan antara variable IncomeNasabah dengan IncomePasangan terhadap variable StatusPinjaman. Dataset yang digunakan sama seperti pada plot pertama yaitu **visNasabah.** Berikut adalah baris kode untuk visualisasi grafis hubungan ketiga variabek yang dimaksud.

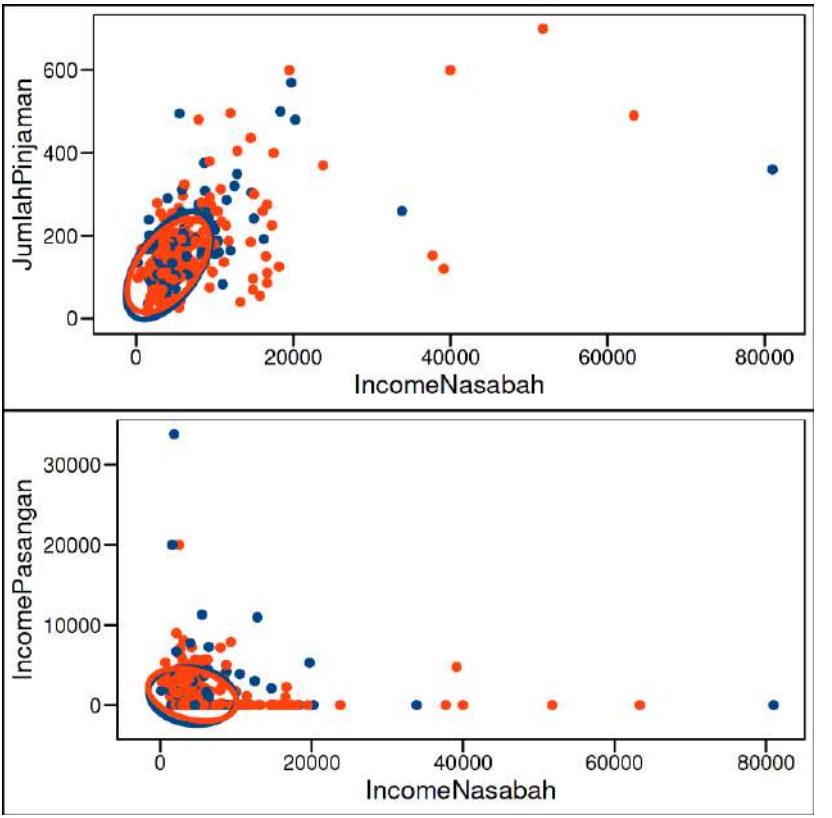
```
ggplot(visNasabah, aes(IncomeNasabah,IncomePasangan, col= StatusPinjaman)) +
  geom_point(size= 2, show.legend = F) +
  stat_ellipse(size= 1.3, linetype=1, show.legend = F) +
  scale_color_calc()
```



Kita juga bisa menggabungkan kedua plot menjadi satu tampilan menggunakan fungsi **grid.arrange().** Sebelum menggunakan fungsi ini, terlebih dahulu panggil library(gridExtra). Berikut baris kode yang dapat digunakan.

```
library(gridExtra)
theme_set(theme_base())

a=ggplot(visNasabah, aes(IncomeNasabah,JumlahPinjaman, col= StatusPinjaman)) +
  geom_point(size= 2, show.legend = F) +
  stat_ellipse(size= 1.3, linetype=1, show.legend = F) +
  scale_color_calc()
b=ggplot(visNasabah, aes(IncomeNasabah,IncomePasangan, col= StatusPinjaman)) +
  geom_point(size= 2, show.legend = F) +
  stat_ellipse(size= 1.3, linetype=1, show.legend = F) +
  scale_color_calc()

grid.arrange(a,b)
```

Untuk dapat menyajikan grafik ke dalam satu grid menggunakan library **gridExtra** kita dapat memanggil fungsi **grid.arrange().** Sebelum itu, baris kode plot pertama disimpan dalam variable **a**, dan plot kedua disimpan dalam variable **b**. dengan demikian lebih mudah untuk dimodifikasi tampilan plotnya dengan **grid.arrange().** Berikut hasil dari penggabungan dua buah plot.


JumlahPinjaman
600
400
200
0
0
20000
40000
60000
80000
IncomeNasabah
IncomePasangan
30000
20000
10000
0
0
20000
40000
60000
80000
IncomeNasabah

BAB VII
ANALISIS REGRESI

Regresi merupakan sebuah model analisis statistik yang digunakan untuk melihat hubungan dan pengaruh antara dua variabel atau lebih. Hubungan yang dimaksud adalah hubungan fungsional dalam bentuk model matematis. Pada analisis regresi ada dua jenis variabel yakni variabel terikat atau dependent variabel dan variabel bebas atau independent variabel atau predictor variabel. Pada analisis regresi jumlah variabel terikat harus sama dengan 1 karena hanya akan dicari satu nilai variabel berdasarkan nilai-nilai pada variabel bebas yang biasanya lebih dari satu.

Penggunaan analisis regressi dilakukan dengan melihat jenis datanya, jika nilai variabel terikat adalah data kontinyu, maka analisis dapat dilakukan dengan pendekatan regresi linear ataupun non-linear, sedangkan jika data untuk variabel terikat berupa data kategorikal maka pendekatan yang bisa digunakan adalah regresi logistik. Ada banyak jenis dan turunan dari metode regresi yang telah dikembangkan oleh para ilmuwan, namun pada buku ini hanya akan dibahas dua buah jenis metode regresi, yaitu regresi linear sederhana (*simple linear regression*) dan regresi liniear berganda (*multiple linear regression*).

7.1.Regresi Linear Sederhana (*Simple Linear Regression*)

Model regresi linear sederhana biasanya digunakan untuk mempelajari hubungan antara dua buah variabel. Dua variabel yang dimaksud adalah variabel bebas dan variabel terikat. Pada modul ini untuk mempermudah memahami pembahasan, istilah variabel bebas

disimbolkan dengan X sedangkan variabel terikat disimbolkan dengan Y. Hubungan dua buah variabel X dan Y pada regresi linear sederhana dapat dinyatakan dengan persamaan sebagai berikut :

$$Y = a + bX$$

Dimana Y merupakan variabel terikat, X merupakan variabel bebas, a merupakan konstanta dan b merupakan koefisien regresi yang ditimbulkan oleh variabel bebas. Untuk mencari nilai konstanta a dan koefisien b dapat dilakukan menggunakan persamaan sebagai berikut :

$$a = \frac{(\sum_{i=1}^{n} Y_i).(\sum_{i=1}^{n} X_i^2) - (\sum_{i=1}^{n} X_i).(\sum_{i=1}^{n} X_i.Y_i)}{n(\sum_{i=1}^{n} X_i^2) - (\sum_{i=1}^{n} X_i)^2}$$

$$b = \frac{n(\sum_{i=1}^{n} X_i.Y_i) - (\sum_{i=1}^{n} X_i).(\sum_{i=1}^{n} Y_i)}{n(\sum_{i=1}^{n} X_i^2) - (\sum_{i=1}^{n} X_i)^2}$$

Langkah-langkah yang dilakukan untuk analisis regresi linear sederhana terdiri atas beberapa tahapan sebagai berikut :

1) Tentukan tujuan yang akan dicapain pada saat melakukan analisis dengan pendekatan regresi linear
2) Kenali mana yang merupakan variabel bebas (X) dan mana yang merupakan variabel terikat (Y)
3) Melakukan pengumpulan data untuk kegiatan analisis
4) Menghitung nilai-nilai yang akan digunakan untuk mencari nilai konstanta a dan koefisien b. Nilai yang dihitung adalah X^2, Y^2 dan X.Y.
5) Hitung nilai konstanta a dan nilai koefisien b.

6) Tentukan model persamaan regresi linear sederhana berdasarkan persamaan umum dengan memasukan nilai konstanta a dan koefisien b ke dalamnya.
7) Lakukan prediksi terhadap variabel terikat (Y) untuk menentukan hasil akhir berdasarkan model persamaan regresi linear yang sudah terbentuk.

Contoh Kasus

Seorang mahasiswa yang sedang tugas akhir ingin mempelajari hubungan antara kemampuan mata kuliah Kecerdasan Artifisial dengan Pembelajaran Mesin yang dimiliki setiap mahasiswa. Dengan demikian dapat diprediksi nilai Pembelajaran Mesin mahasiswa jika nilai Kecerdasan Artifisial nya diketahui. Sebagai bahan analisis sederhana mahasiswa tugas akhir tersebut mengumpulkan 5 buah data mahasiswa yang memiliki nilai kemampuan Pembelajaran Mesin dan Kecerdasan Artifisial.

Penyelesaian :

Berdasarkan informasi studi kasus di atas, untuk melakukan analisis data dengan pendekatan regresi linear sederhana dapat dilakukan dengan mengikuti tahapan penyelesaian regresi linear sederhana seperti yang sudah dijelaskan di atas.

Langkah pertama : Tentukan tujuan analisis

Tujuan : Memprediksi nilai kemampuan Pembelajaran Mesin mahasiswa berdasarkan nilai Kecerdasan Artifisial yang dimilikinya.

Langkah kedua : Kenali mana variabel terikat dan mana variabel bebas

Berdasarkan ilustrasi kasus di atas dapat diketahui bahwa variabel terikat (Y) adalah nilai kemampuan Pembelajaran Mesin, sedangkan variabel bebas (X) nya adalah nilai Kecerdasan Artifisial.

Langkah ketiga : Kumpulkan data pendukung

Hasil pengumpulan data, misalnya diperoleh data nilai mahasiswa untuk kedua mata kuliah sebagai berikut :

No.	**Nama Mahasiswa**	**Kecerdasan Artifisial (X)**	**Pembelajaran Mesin (Y)**
1	Rahmat Wijaya	80	70
2	Indra Wijaya	85	90
3	Angga Prawira	70	70
4	Dina Septiani	90	70
5	Riky Faza	80	85

Langkah keempat : Hitung nilai-nila X^2, Y^2 dan X.Y

Berdasarkan data pada langkah ketiga, maka nilai-nilai total dapat dihitung sebagai berikut :

No.	**Nama Mahasiswa**	**X**	**Y**	**X^2**	**Y^2**	**X.Y**
1	Rahmat Wijaya	80	70	6400	4900	5600
2	Indra Wijaya	85	90	7225	8100	7650
3	Angga Prawira	70	70	4900	4900	4900
4	Dina Septiani	90	70	8100	4900	6300
5	Riky Faza	80	85	6400	7225	6800
$\sum$		**405**	**385**	**33025**	**30025**	**31250**

Langkah kelima : Hitung nilai konstanta a dan koefisien b

Untuk dapat menentukan model persamaan regresi linear sederhana, langkah selanjutnya adalah menghitung nilai konstanta a dan juga koefisien b. Berikut tahap perhitungannya :

1) Perhitungan nilai konstanta a

   Masukan nilai-nilai setiap persamaan sesuai dengan hasil perhitungan pada langkah keempat.

$$a = \frac{(\sum_{i=1}^{n} Y_i).(\sum_{i=1}^{n} X_i^2) - (\sum_{i=1}^{n} X_i).(\sum_{i=1}^{n} X_i.Y_i)}{n(\sum_{i=1}^{n} X_i^2) - (\sum_{i=1}^{n} X_i)^2}$$

$$a = \frac{(385).(33025) - (405).(31250)}{5(33025) - (405)^2}$$

$$= \frac{12714625 - 12656250}{165125 - 164025}$$

$$a = \frac{58375}{1100} = 53.06$$

2) Perhitungan nilai koefisien b

   Masukan nilai-nilai setiap persamaan sesuai dengan hasil perhitungan pada langkah keempat.

$$b = \frac{n(\sum_{i=1}^{n} X_i.Y_i) - (\sum_{i=1}^{n} X_i).(\sum_{i=1}^{n} Y_i)}{n(\sum_{i=1}^{n} X_i^2) - (\sum_{i=1}^{n} X_i)^2}$$

$$b = \frac{5(31250) - (405).(385)}{5(33025) - (405)^2} = \frac{156250 - 155925}{165125 - 164025}$$

$$b = \frac{325}{1100} = 0.29$$

Langkah keenam : Buatkan model persamaan regresi linear sederhana

Pada tahap ini masukan nilai-nilai konstanta a dan koefisien b yang sudah diperoleh pada tahap kelima ke dalam persamaan regresi linear sederhana yang baku. Sehingga, pada kasus ini, model persamaan regresi linear sederhana dapat ditentukan sebagai berikut :

Nilai konstanta a = 53.06; nilai koefisien b = 0.29 sehingga diperoleh model persamaan linear regresi sederhana :

$$Y = a + bX$$

$$Y = 53.06 + 0.29X$$

Atau

Pembelajaran Mesin (Y) = 53.06 + 0.29 (Kecerdasan Artifisial)

Langkah ketujuh : Lakukan prediksi terhadap data baru

Pada tahap ini lakukan prediksi terhadap data baru. Dalam melakukan prediksi ada dua tipe prediksi yang bisa dilakukan berdasarkan model persamaan regresi linear sederhana yang diperoleh. Pertama, memprediksi nilai variabel terikat (Y) terhadap data variabel bebas yang diketahui, kedua memprediksi besarnya nilai variabel bebas (X) jika batasan variabel terikat (Y) ditentukan. Berikut contoh implementasi kedua tipe prediksi tersebut :

1) Prediksi nilai Pembelajaran mesin (Y) jika nilai Kecerdasan Artifisial (X) mahasiswa diketahui X = 80, maka diperoleh :

   $Y = 53.06 + 0.29.X$

   $Y = 53.06 + 0.29\ (80)$

   $Y = 53.06 + 23.2$

   $Y = 76.8$

2) Prediksi besarnya nilai Kecerdasan Artifisial (X) yang harus dicapai mahasiswa jika ingin mendapatkan nilai Pembelajaran Mesin (Y) sebesar 90, dapat dihitung sebagai berikut :
   Y = 53.06 + 0.29.X
   90 = 53.06 + 0.29.X
   X = (90 – 53.06)/0.29
   X = 36.4/0.29
   X = 127.4

7.2.Regresi Linear Berganda (*Multiple Linear Regression*)

Regresi linear berganda merupakan pendekatan statistik yang digunakan untuk melihat hubungan antara variabel terikat (Y) dengan variabel bebas (X), dimana jumlah variabel bebas yang mempengaruhi variabel terikat tersebut lebih dari satu variabel. Pada dasarnya regresi linear berganda sama dengan linear regresi sederhana, hanya saja pada regresi linear berganda jumlah variabel bebasnya lebih dari satu sehingga perkiraan prediksi nilai dari variabel terikat (Y) dipengaruhi oleh variabel bebas X_1, X_2, X_3, ..., X_n. Secara umum model persamaan regresi linear berganda dapat dituliskan sebagai berikut :

$$Y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n + \varepsilon$$

Dimana b_0 merupakan konstanta nilai Y jika variabel bebas (X_1, X_2, X_3,...,X_n = 0), sedangkan b_1, b_2, $b_n$merupakan koefisien regresi untuk setiap variabel bebas (X_n), dimana nilainya bisa berupa nilai peningkatan ataupun penurunan. Untuk memperoleh nilai konstanta b_0, b_1, b_2, b_n dapat dilakukan dengan menggunakan persamaan sebagai berikut :

$$\sum_{i=1}^{N} Y_i = nb_0 + b_1 \sum_{i=1}^{N} x_1 + b_2 \sum_{i=1}^{N} x_2 + \ldots + b_k \sum_{i=1}^{N} x_n$$

$$\sum_{i=1}^{N} x_{i1} y_i = b_0 \sum_{i=1}^{N} x_{i1} + b_1 \sum_{i=1}^{N} {x_{i1}}^2 + b_2 \sum_{i=1}^{N} x_{i1} x_{i2} + \ldots + b_k \sum_{i=1}^{N} x_{i1} x_{ik}$$

$$\sum_{i=1}^{N} x_{ik} y_i = b_0 \sum_{i=1}^{n} x_{ik} + b_1 \sum_{i=1}^{n} x_{ik} x_{i1} + b_2 \sum_{i=1}^{n} x_{ik} x_{i2} + \cdots + b_k \sum_{i=1}^{n} {x_{ik}}^2$$

Dimana *k* merupakan urutan variabel bebas ke – *i.* Perhitungan nilai konstanta a dan koefisien b pada regresi linear berganda berbeda dengan regresi linear sederhana. Pada regresi linear berganda dapat dilakukan dengan proses substitusi dan eliminasi persamaan awal yang sudah terbentuk (persamaan normal).

Contoh Kasus

Seorang mahasiswa yang sedang tugas akhir ingin mempelajari hubungan antara kemampuan Pembelajaran Mesin dengan kemampuan Kecerdasan Artifisial, Data Mining dan Statistika yang dimiliki setiap mahasiswa. Dengan demikian dapat diprediksi nilai kemampuan **Pembelajaran Mesin** jika nilai Kecerdasan Artifisial, Data Mining dan Statistika diketahui. Sebagai bahan analisis sederhana mahasiswa tugas akhir tersebut mengumpulkan 5 buah data mahasiswa yang memiliki nilai kemampuan Pembelajaran Mesin, Kecerdasan Artifisial, Data Mining dan Statistika.

Penyelesaian :

Berdasarkan informasi studi kasus di atas, untuk melakukan analisis data dengan pendekatan regresi linear sederhana dapat dilakukan dengan mengikuti tahapan penyelesaian regresi linear sederhana seperti yang sudah dijelaskan di atas.

Langkah pertama : Tentukan tujuan analisis

Tujuan : Memprediksi nilai kemampuan **Pembelajaran Mesin** mahasiswa berdasarkan nilai Kecerdasan Artifisial, Data Mining dan Statistika yang dimilikinya.

Langkah kedua : Kenali mana variabel terikat dan mana variabel bebas

Berdasarkan ilustrasi kasus di atas dapat diketahui bahwa variabel terikat (**Y**) adalah nilai kemampuan **Pembelajaran Mesin**, sedangkan variabel bebas (X) nya adalah nilai kemampuan Kecerdasan Artifisial **(X_1),** Data Mining **(X_2)** dan Statistika **(X_3).**

Langkah ketiga : Kumpulkan data pendukung

Hasil pengumpulan data, misalnya diperoleh data nilai untuk keempat mata kuliah dari lima mahasiswa sebagai berikut :

No.	Nama Mahasiswa	(X_1)	(X_2)	(X_3)	(Y)
1	Rahmat Wijaya	80	70	75	70
2	Indra Wijaya	85	80	70	90
3	Angga Prawira	70	80	85	70
4	Dina Septiani	90	80	70	70
5	Rostya	80	80	90	85

Langkah keempat : Tentukan persamaan normal untuk model regresi linear berganda

Pada tahap ini ditentukan persamaan normal berdasarkan jumlah variabel bebas yang diketahui. Penentuan persamaan dilakukan berdasarkan persamaan umum regresi linear berganda seperti yang telah disebutkan di atas. Berikut adalah persamaan normal yang digunakan pada kasus ini :

$$Y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3$$

Sedangkan penentuan nilai konstanta b_0 dan koefisien b_1, b_2 dan b_3 dapat diperoleh dari persamaan sebagai berikut :

$$\sum_{i=1}^{N} Y_i = nb_0 + b_1 \sum_{i=1}^{N} x_1 + b_2 \sum_{i=1}^{N} x_2 + b_3 \sum_{i=1}^{N} x_3$$

$$\sum_{i=1}^{N} x_{i1} y_i = b_0 \sum_{i=1}^{N} x_{i1} + b_1 \sum_{i=1}^{N} {x_{i1}}^2 + b_2 \sum_{i=1}^{N} x_{i1} x_{i2} + b_3 \sum_{i=1}^{N} x_{i1} x_{i3}$$

$$\sum_{i=1}^{N} x_{i2} y_i = b_0 \sum_{i=1}^{N} x_{i2} + b_1 \sum_{i=1}^{N} {x_{i2}}^2 + b_2 \sum_{i=1}^{N} x_{i1} x_{i2} + b_3 \sum_{i=1}^{N} x_{i2} x_{i3}$$

$$\sum_{i=1}^{N} x_{i3} y_i = b_0 \sum_{i=1}^{N} x_{i3} + b_1 \sum_{i=1}^{N} {x_{i3}}^2 + b_2 \sum_{i=1}^{N} x_{i3} x_{i1} + b_3 \sum_{i=1}^{N} x_{i3} x_{i2}$$

Langkah kelima : Hitung nilai-nilai sebagai berikut

$${x_1}^2, {x_2}^2, {x_3}^2, x_1 . x_2, x_1 . x_3, x_2 . x_3, x_1 . y, x_2 . y, x_3 . y$$

Berdasarkan data pada langkah ketiga, maka nilai-nilai total dapat dihitung sebagai berikut :

No.	Nama Mahasiswa	X_1	X_2	X_3	Y	x_1^2	x_2^2	x_3^2
1	Rahmat Wijaya	80	70	75	70	6400	4900	5625
2	Indra Wijaya	85	80	70	90	7225	6400	4900
3	Angga Prawira	70	80	85	70	4900	6400	7225
4	Dina Septiani	90	80	70	70	8100	6400	4900
5	Riky Faza	80	80	90	85	6400	6400	8100
$\sum$		405	390	390	385	33025	24100	30750

No.	$x_1.x_2$	$x_1.x_3$	$x_2.x_3$	$x_1.y$	$x_2.y$	$x_3.y$
1	5600	6000	5250	5600	4900	5250
2	6800	5950	7200	7650	7200	6300
3	5600	5950	6800	4900	5600	5950
4	7200	6300	5600	6300	5600	4900
5	6400	7200	7200	6800	6800	7650
$\sum$	31600	31400	32050	31250	30100	30050

Langkah keenam : Hitung nilai konstanta b_0 dan koefisien b_1,b_2,b_3

Untuk dapat menentukan model persamaan regresi linear sederhana, langkah selanjutnya adalah menghitung nilai konstanta b_0 dan juga koefisien b_1,b_2,b_3. Berikut tahap perhitungannya :

Bentuk persamaan normal :

$5b_0 + 405b_1 + 390b_2 + 390b_3 = 385$ (1)

$405b_0 + 33025b_1 + 31600b_2 + 31400b_3 = 31250$ (2)

$390b_0 + 24100b_1 + 31600b_2 + 32050b_3 = 30100$ (3)

$390b_0 + 390b_1 + 31400b_2 + 32050b_3 = 30050$ (4)

Langkah selanjutnya adalah menggunakan teknik eliminasi untuk menyederhanakan persamaan sehingga terbentuk persamaan baru yang memudahkan untuk menentukan nilai konstanta dan juga koefisien. Eliminasi persamaan (2) ke dalam persamaan (3), sehingga diperoleh :

$405b_0 + 33025b_1 + 31600b_2 + 31400b_3 = 31250$
$390b_0 + 24100b_1 + 31600b_2 + 32050b_3 = 30100$ -

$15b_0 + 8925b_1 – 650b_3 = 1150$ (5)

Eliminasi persamaan (1) ke dalam persamaan (2), sehingga diperoleh :
$5b_0 + 405b_1 + 390b_2 + 390b_3 = 385$ *x 81.02*
$405b_0 + 33025b_1 + 31600b_2 + 31400b_3 = 31250$ *x 1*

$405,1b_0 + 32813,1b_1 + 31600b_2 + 31600b_3 = 31192,7$
$405b_0 + 33025b_1 + 31600b_2 + 31400b_3 = 31250$ –

$0.1b_0 – 211,9b_1 + 200b_3 = - 57,3$ (6)

Eliminasi persamaan (1) ke dalam persamaan (3), sehingga diperoleh :
$5b_0 + 405b_1 + 390b_2 + 390b_3 = 385$ *x 81.02*
$390b_0 + 24100b_1 + 31600b_2 + 32050b_3 = 30100$ *x 1*

$405b_0 + 32813b_1 + 31600b_2 + 31600b_3 = 31192,7$
$390b_0 + 24100b_1 + 31600b_2 + 32050b_3 = 30100$ -

$15b_0 + 8713b_1 – 450b_3 = 1092,7$ (7)

Eliminasi persamaan (5) ke dalam persamaan (6), sehingga diperoleh :
$15b_0 + 8925b_1 – 650b_3 = 1150$ *x1*
$0.1b_0 – 211,9b_1 + 200b_3 = - 57,3$ *x150*

$15b_0 + 8925b_1 – 650b_3 = 1150$
$15b_0 – 31785b_1 + 30000b_3 = -8595$ -

$40710b_1 - 30650b_3 = 9745$ (8)

Eliminasi persamaan (5) ke dalam persamaan (7), sehingga diperoleh :
$15b_0 + 8925b_1 – 650b_3 = 1150$
$15b_0 + 8713b_1 – 450b_3 = 1092,7$ –

$212b_1 – 200b_3 = 57,3$ (9)

Eliminasi persamaan (8) ke dalam persamaan (9), sehingga diperoleh :
$40710b_1 - 30650b_3 = 9745$ x 1
$212b_1 – 200b_3 = 57,3$ x 153,25

$40710b_1 - 30650b_3 = 9745$
$32489b_1 – 30650b_3 = 8781,23$ -

$8221b_1 = 963,77$

$$b_1 = \frac{963,77}{8221} = 0.117 \approx 0.12$$

Substitusi nilai b_1 ke dalam persamaan (9), sehingga diperoleh :
$212b_1 – 200b_3 = 57,3$
$212(0.12) – 200b_3 = 57,3$
$– 200b_3 = 57,3 – 25.44$
$-200b_3 = 31,86$

$$b_3 = \frac{31.86}{-200} = -0.1593 \approx -0.16$$

$b_3 = - 0.16$

substitusikan nilai b_1 dan b_3 ke dalam persamaan (7), sehingga diperoleh :
$15b_0 + 8713b_1 – 450b_3 = 1092,7$
$15b_0 + 8713(0.12) – 450(-0.16) = 1092,7$

$15b_0 + 1045.56 + 72 = 1092,7$
$15b_0 + 1117.56 = 1092,7$

$15b_0 = 1092,7 - 1117.56$
$15b_0 = 975,14$
$b_0 = 975,14/ 15 = 65.009 \approx 65.01$

substitusikan nilai b_0, b_1 dan b_3 ke dalam persamaan (1) untuk menentukan nilai b_2, sehingga diperoleh :

$5b_0 + 405b_1 + 390b_2 + 390b_3 = 385$
$5(65.01) + 405(0.12) + 390b_2 + 390(-0.16) = 385$
$325.05+ 48.6 + 390b_2 - 62,4 = 385$
$390b_2 + 311,25 = 385$
$390b_2 = 385 – 311.25$
$390b_2 = 73.75$
$b_2 = 73.75 / 390$
$b_2 = 0.189 \approx 0.19$

Langkah ketujuh : Buatkan model persamaan regresi linear sederhana

Pada tahap ini masukan nilai-nilai konstanta a dan koefisien b yang sudah diperoleh pada tahap kelima ke dalam persamaan regresi linear sederhana yang baku. Sehingga, pada kasus ini, model persamaan regresi linear sederhana dapat ditentukan sebagai berikut :

Nilai konstanta b0 = 65.01; nilai koefisien b1 = 0.12; koefisien b2 = 0.19; koefisien b3 = -0.16 sehingga diperoleh model persamaan linear regresi berganda :

$$Y = b_0 + b_1x_1 + b_2x_2 + b_3x_3$$

$$Y = 65.01 + 0.12x_1 + 0.19x_2 - 0.16x_3$$

Langkah kedelapan : Lakukan prediksi terhadap data baru
Pada tahap ini lakukan prediksi terhadap data baru. Dalam melakukan prediksi ada dua tipe prediksi yang bisa dilakukan berdasarkan model persamaan regresi linear sederhana yang diperoleh.Prediksi nilai kemampuan Pembelajaran Mesin (Y) seorang mahasiswa jika diketahui nilai kemampuan Kecerdasan Artifisial (X_1) = 80, nilai Data Mining (X_2) = 80 dan nilai Statistika (X_3) = 70.
Masukan nilai nilai variabel bebas X1, X2 dan X3 tersebut ke dalam persamaan :

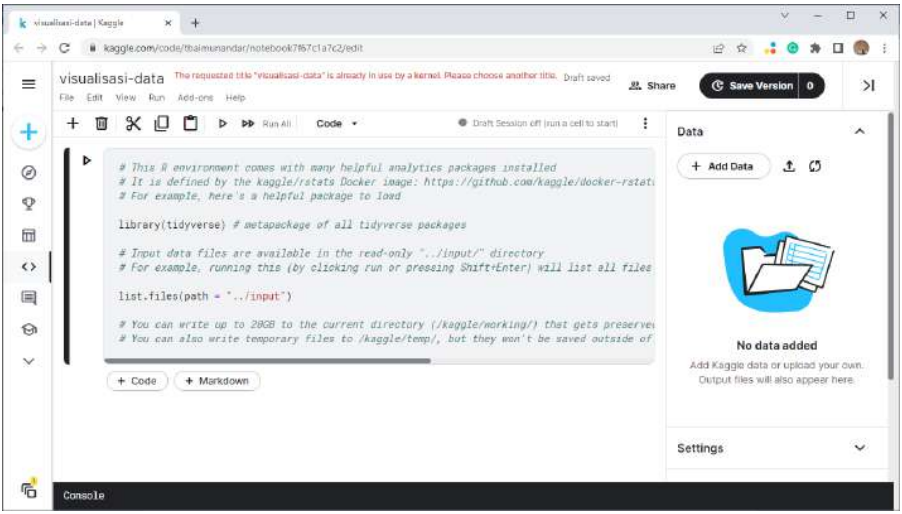
$$Y = 65.01 + 0.12x_1 + 0.19x_2 - 0.16x_3$$

Sehingga diperoleh :

$$Y = 65.01 + 0.12(80) + 0.19(80) - 0.16(70)$$
$$Y = 65.01 + 9.6 + 15.2 - 11.2$$
$$Y = 78.61$$

7.3.Analisis Regresi Menggunakan R

Untuk memulai visualisasi data menggunakan Bahasa R, langkah pertama adalah login ke dalam kaggle.com menggunakan akun masing-masing. Kemudian pada manu navigator bagian kiri pilih menu **< > Code** kemudian klik tombol **New Notebook** sampai terbuka halaman seperti pada gambar di bawah. Untuk lebih jelas bagaimana membuat akun dan membuka notebook baru silahkan pelajari Bab 5.



Langkah selanjutnya adalah menyiapkan dataset menggunakan perintah "read.csv" pastikan data sudah diupload ke dalam kaggle dengan format *.csv. Perintah head digunakan untuk menampilkan 6 buah data pertama. Berikut baris kode yang digunakan :

```
data=read.csv('../input/pinjamankreditnasabah/dataset-pinjaman-nasabah.csv', sep=',')
head(data)
```

A data.frame: 6 × 13

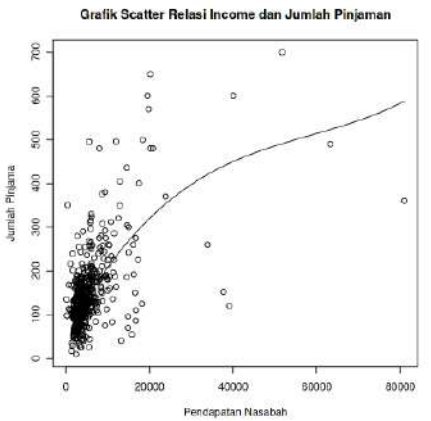
	ID_Nasabah	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman	JangkaWaktuPi
	<chr>	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>	
1	LP001002	Male	No	0	Graduate	No	5849	0	NA	
2	LP001003	Male	Yes	1	Graduate	No	4583	1508	128	
3	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	
4	LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	
5	LP001008	Male	No	0	Graduate	No	6000	0	141	
6	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	

+ Code + Markdown

Sebelum membangun model regresi, visualisasikan data untuk mengetahui hubungan antara variabel dependen dan independen. Pada

kasus ini, teknik pertama yang dibahas adalah SLR. Variabel dependen yang digunakan adalah JumlahPinjaman sedangkan variabel independen adalah IncomeNasabah. Pastikan bahwa bentuk grafik linear. Gunakan **library scatter.smooth** untuk memvisualisasikan data ke dalam grafik scatter.

```
scatter.smooth(x=data$IncomeNasabah, y=data$JumlahPinjaman, xlab='Pendapatan Nasabah',
               ylab='Jumlah Pinjama', main='Grafik Scatter Relasi Income dan Jumlah Pinjaman')
```



Hitung korelasi antara variabel **IncomeNasabah** dan **JumlahPinjaman**. Gunakan fungsi **cor()** untuk menghitung nilai korelasi sehingga diperoleh nilai hubungan keduanya. Namun sebelum menggunakan fungsi **cor(),** lakukan removal terhadap nilai dengan status N/A untuk setiap variable menggunakan perintah **na.omit()**, kemudian simpan ke dalam variable **data2**. Berikut baris kode untuk mengukur korelasi dua variabel.

Berdasarkan hasil pengukuran koefisien korelasi, dapat diketahui bahwa hubungan keduanya memiliki nilai sebesar 0.57. Selain mencari tahu hubungan antara variable IncomeNasabah terhadap JumlahPinjaman, kita juga bisa mengetahui hubungan antara **IncomePasangan** terhadap **JumlahPinjaman**. Dengan menganalisis hubungan antar variable kita dapat mengetahui apakah suatu variable mempengaruhi variable lainnya dengan kuat atau tidak sama sekali. Berikut hasil pengukuran koefisien korelasi antara variable **IncomePasangan** dengan **JumlahPinjaman**.

Hasil pengukuran koefisien korelasi terhadap variable IncomeNasabah, IncomePasangan dan JumlahPinjaman, dapat diketahui bahwa JumlahPinjaman lebih dipengaruhi oleh variable **IncomeNasabah** dibandingkan **IncomePasangan**. Hal ini terlihat dari nilai koefisien korelasi variable **IncomePasangan** yang menjauhi +1 terhadap **JumlahPinjaman**. Analisis korelasi variable lainnya bisa dilakukan dengan cara yang sama.

Untuk mengetahui semua koefisien korelasi antar variable bertipe numerik, seperti IncomeNasabah, Income Pasangan, Jumlah Pinjaman dan JangkaWaktuPinjaman, kita juga dapat menggunakan fungsi **cor().** Ambil data untuk keempat variable dan simpan ke dalam variable **KoefVar**, kemudian masukan ke dalam fungsi **cor()** untuk dihitung. Berikut baris kode yang bisa digunakan.

```
KoefVar=data2[,7:10]
cor(KoefVar)
```

A matrix: 4 × 4 of type dbl

	IncomeNasabah	IncomePasangan	JumlahPinjaman	JangkaWaktuPinjaman
IncomeNasabah	1.00000000	-0.124310229	0.57044064	-0.063364126
IncomePasangan	-0.12431023	1.000000000	0.15747169	0.002262917
JumlahPinjaman	0.57044064	0.157471687	1.00000000	0.022322294
JangkaWaktuPinjaman	-0.06336413	0.002262917	0.02232229	1.000000000

+ Code + Markdown

1. Simple Linear Regression

Untuk membangun model regresi sederhana, kita dapat memanggil fungsi **lm().** Kemudian tentukan formulasi dari model linear berdasarkan variabel dependen dan independen nya. Pada buku ini, variabel independen yang digunakan adalah IncomeNasabah (X) dan variabel dependen nya adalah JumlahPinjaman (Y). Buat sebuah variabel baru dengan nama **simpleRegresi** untuk menampung hasil pembelajaran dataset terhadap metode simple linear regresi. Untuk menampilkan model regresi yang terbentuk, gunakan fungsi **summary(namaVariabel)**. Berikut baris kode yang digunakan.

```
simpleRegresi=lm(formula=JumlahPinjaman ~ IncomeNasabah, data=data2)
summary(simpleRegresi)
```

```
Call:
lm(formula = JumlahPinjaman ~ IncomeNasabah, data = data2)

Residuals:
    Min      1Q  Median      3Q     Max
-350.90  -32.37   -9.58   24.17  349.51

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.047e+02  4.044e+00   25.88   <2e-16 ***
IncomeNasabah 7.484e-03  4.749e-04   15.76   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 69.82 on 515 degrees of freedom
Multiple R-squared:  0.3254,    Adjusted R-squared:  0.3241
F-statistic: 248.4 on 1 and 515 DF,  p-value: < 2.2e-16
```

+ Code + Markdown

Untuk mengetahui nilai konstanta (B_0) dan koefisien (B_1) variable IncomeNasabah, silahkan lihat pada opsi **Coefficients** pada kolom **Estimate.** Nilai **Intercept** merupakan konstanta persamaan linear (1.047e+02 atau 104.7), sedangkan nilai Estimate pada variable **IncomeNasabah** merupakan nilai koefisien untuk variable itu sendiri (7.48e-03 atau 0.00748). Dengan demikian, persamaan linear regresi sederhana yang terbentuk adalah :

$$JumlahPinjaman\ (Y) = 104.7 + 0.00748\ (IncomeNasabah)$$

Pisahkan variabel IncomeNasabah dan JumlahPinjaman dari datasets, kemudian simpan dalam variabel Pr. Gunakan dataset dari variable data2. Gunakan baris kode sebagai berikut untuk mengambil variable IncomeNasabah (7) dan JumlahPinjaman (9) :

```
Pr=data2[c(7,9)]
head(Pr)
```

A data.frame: 6 × 2

	IncomeNasabah	JumlahPinjaman
	<int>	<int>
2	4583	128
3	3000	66
4	2583	120
5	6000	141
6	5417	267
7	2333	95

+ Code + Markdown

Ambil 10 data pertama dari variable **Pr** untuk kebutuhan prediksi menggunakan model linear regresi yang sudah dibentuk. Simpan kesepuluh data ke dalam variable **dataPrediksi.** Gunakan baris kode berikut untuk mengambil 10 data pertama :

```
dataPrediksi=Pr[1:10,]
dataPrediksi
```

A data.frame: 10 × 2

	IncomeNasabah	JumlahPinjaman
	<int>	<int>
2	4583	128
3	3000	66
4	2583	120
5	6000	141
6	5417	267
7	2333	95
8	3036	158
9	4006	168
10	12841	349
11	3200	70

+ Code + Markdown

Gunakan fungsi **predict()** untuk melakukan prediksi terhadap data baru (dalam hal ini data pada variabel **dataPrediksi**). Gunakan variabel IncomeNasabah untuk memprediksi JumlahPinjaman. Simpan hasil prediksi dalam variabel **hasilPrediksi**. Berikut Perintahnya :

```
hasilPrediksi=predict(simpleRegresi,newdata=dataPrediksi,interval='confidence')
data.frame(hasilPrediksi)
```

A data.frame: 10 × 3

	fit	lwr	upr
	<dbl>	<dbl>	<dbl>
2	138.9704	132.6722	145.0686
3	127.1227	120.6412	133.6042
4	124.0018	117.3680	130.6355
5	149.5756	143.5282	155.6230
6	145.2123	139.1789	151.2457
7	122.1307	115.3966	128.8648
8	127.3921	120.9228	133.8615
9	134.6519	128.4519	140.8520
10	200.7758	191.6787	209.8729
11	128.6196	122.2040	135.0352

+ Code + Markdown

Syntax fungsi **predict ()** adalah : *predict (model linear regresi, databaru, interval). Model linear regresi* merupakan model yang sudah dibangun dengan fungsi **lm(),** *databaru* merupakan data yang akan digunakan untuk melakukan prediksi sedangkan *interval* digunakan untuk mengukur ketidakpastian dalam prediksi, biasanya menggunakan interval *'confidence'*. Hasil prediksi akan menampilkan tiga buah kolom, pertama merupakan hasil fit prediksi, kolom kedua merupakan batas terbawah hasil prediksi sedangkan kolom ketiga menunjukan batas teratas hasil prediksi.

Bandingkan antara hasil prediksi (pada variabel **hasilPrediksi**) dengan data aktual (pada variabel **dataPrediksi**). Gunakan **data.frame** dan **cbind** (untuk mengcombine matrik atau data frame menjadi kolom), simpan pada variable **bandingHasil**. Luaran dari perintah ini adalah tabel antara data actual dengan data hasil prediksi menggunakan model pada tahap sebelumnya. Baris kode berikut digunakan untuk menyajikan tabel perbandingan tersebut :

```
bandingHasil=data.frame(cbind(actuals=dataPrediksi$JumlahPinjaman,
                              predicted=hasilPrediksi))
bandingHasil
```

A data.frame: 10 × 4

	actuals	fit	lwr	upr
	<dbl>	<dbl>	<dbl>	<dbl>
2	128	138.9704	132.8722	145.0686
3	66	127.1227	120.6412	133.6042
4	120	124.0018	117.3680	130.6355
5	141	149.5756	143.5282	155.6230
6	267	145.2123	139.1789	151.2457
7	95	122.1307	115.3966	128.8648
8	158	127.3921	120.9228	133.8615
9	168	134.6519	128.4519	140.8520
10	349	200.7758	191.6787	209.8729
11	70	128.6196	122.2040	135.0352

+ Code + Markdown

Gunakan fungsi **cor()** untuk mengukur korelasi hasil prediksi dengan data aktual, jika nilai korelasi mendekati +1 maka model dapat dikatakan baik.

```
cor(bandingHasil)
```

A matrix: 4 × 4 of type dbl

	actuals	fit	lwr	upr
actuals	1.0000000	0.8539317	0.8554208	0.8520500
fit	0.8539317	1.0000000	0.9996918	0.9997271
lwr	0.8554208	0.9996918	1.0000000	0.9988391
upr	0.8520500	0.9997271	0.9988391	1.0000000

+ Code + Markdown

2. Multiple Linear Regression

Pada bagian ini dibahas tentang model *Multiple Linear Regression* (MLR). Pada dasarnya fungsi yang digunakan sama yakni **lm(),** hanya saja pada penulisan formula, untuk variabel prediktor tidak hanya satu, namun dua atau bahkan lebih. Pada kasus ini, variabel predictor yang digunakan adalah IncomeNasabah, IncomePasangan dan JangkaWaktuPinjaman, sedangkan variabel respon (dependen) adalah JumlahPinjaman. Simpan model yang dibangun ke dalam variable **multilinear.** Berikut perintahnya.

```
multilinear=lm(JumlahPinjaman~IncomeNasabah + IncomePasangan
               + JangkaWaktuPinjaman, data=data2)
summary(multilinear)
```

```
Call:
lm(formula = JumlahPinjaman ~ IncomeNasabah + IncomePasangan +
    JangkaWaktuPinjaman, data = data2)

Residuals:
    Min      1Q  Median      3Q     Max
-372.63  -28.26   -6.87   20.52  354.15

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         6.328e+01  1.641e+01   3.857 0.000129 ***
IncomeNasabah       7.913e-03  4.599e-04  17.207  < 2e-16 ***
IncomePasangan      7.748e-03  1.167e-03   6.641 7.96e-11 ***
JangkaWaktuPinjaman 7.888e-02  4.571e-02   1.726 0.085038 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 66.96 on 513 degrees of freedom
Multiple R-squared:  0.382,     Adjusted R-squared:  0.3784
F-statistic: 105.7 on 3 and 513 DF,  p-value: < 2.2e-16
```

+ Code + Markdown

Sama seperti pada model simple linear regression (SLR), untuk melihat nilai konstanta (B_0) dan koefisien (B_1, B_2, … B_n) dapat dilihat pada opsi **Coefficients.** Hasil model menunjukan bahwa nilai konstanta (intercept) sebesar 6.328e+01 atau 63.28, sementara untuk B_1, B_2 dan B_3 berturut-turut adalah 7.913e-03 (0.0079), 7.748e-03 (0.0077) dan 7.888e-02 (0.078). dengan demikian model multiple linear regression yang terbentuk adalah :

$$Y = 63.28 + 0.0079\,X_1 + 0.0077\,X_2 + 0.078\,X_3$$

Dimana X1 merupakan IncomeNasabah, X2 adalah IncomePasangan dan X3 adalah JangkaWaktuPinjaman. Sedangkan Y adalah JumlahPinjaman.

Pisahkan variabel **IncomeNasabah, IncomePasangan, JangkaWaktuPinjaman** dan **JumlahPinjaman** dari datasets, kemudian simpan dalam variabel **Pr2**. Gunakan dataset dari variable **data2**. Gunakan baris kode sebagai berikut untuk mengambil variable **IncomeNasabah** (7) **IncomePasangan (8), JangkaWaktuPinjaman (10)** dan **JumlahPinjaman** (9) :

```
Pr2=data2[c(7,8,10,9)]
head(Pr2)
```

A data.frame: 6 × 4

	IncomeNasabah	IncomePasangan	JangkaWaktuPinjaman	JumlahPinjaman
	<int>	<int>	<int>	<int>
2	4583	1508	360	128
3	3000	0	360	66
4	2583	2358	360	120
5	6000	0	360	141
6	5417	4196	360	267
7	2333	1516	360	95

+ Code + Markdown

Ambil 12 data dari variable **Pr2** untuk kebutuhan prediksi menggunakan model linear regresi yang sudah dibentuk. Simpan keenam belas data ke dalam variable **dataPrediksiM**. Gunakan baris kode berikut untuk mengambil 12 data pertama :

```
dataPrediksiM=Pr2[20:31,]
dataPrediksiM
```

A data.frame: 12 × 4

	IncomeNasabah	IncomePasangan	JangkaWaktuPinjaman	JumlahPinjaman
	<int>	<int>	<int>	<int>
23	2600	1911	360	116
24	3365	1917	360	112
26	9560	0	360	191
27	2799	2253	360	122
28	4226	1040	360	110
29	1442	0	360	35
30	3750	2083	360	120
32	3167	0	360	74
33	4692	0	360	106
34	3500	1667	360	114
35	12500	3000	360	320
38	3667	1459	360	144

Gunakan fungsi **predict**() untuk melakukan prediksi terhadap data baru (dalam hal ini data pada variabel **dataPrediksiM**). Gunakan variabel **IncomeNasabah**, **IncomePasangan** dan **JangkaWaktuPinjaman** untuk memprediksi **JumlahPinjaman**. Simpan hasil prediksi dalam variabel **hasilPrediksiM**. Berikut Perintahnya :

```
hasilPrediksiM=predict(multilinear,newdata=dataPrediksiM, interval='confidence')
data.frame(hasilPrediksiM)
```

A data.frame: 12 × 3

	fit	lwr	upr
	<dbl>	<dbl>	<dbl>
23	127.0544	120.51348	133.5953
24	133.1544	126.83878	139.4700
26	167.3230	159.60981	175.0362
27	131.2789	124.69436	137.8635
28	133.1725	126.92572	139.4193
29	103.0845	95.02119	111.1478
30	137.4871	131.20916	143.7650
32	116.7346	109.32910	124.1401
33	128.8021	121.73870	135.8654
34	132.2856	126.03232	138.5390
35	213.8316	204.17962	223.4836
38	131.9955	125.76760	138.2235

Bandingkan antara hasil prediksi (pada variabel **hasilPrediksiM**) dengan data aktual (pada variabel **dataPrediksiM**). Gunakan **data**.**frame** dan **cbind** (untuk mengcombine matrik atau data frame menjadi kolom), simpan pada variable **bandingHasilM**. Luaran dari perintah ini adalah tabel antara data actual dengan data hasil prediksi menggunakan model pada tahap sebelumnya. Baris kode berikut digunakan untuk menyajikan tabel perbandingan tersebut :

```
bandingHasilM=data.frame(cbind(actuals=dataPrediksiM$JumlahPinjaman,
                               predicted=hasilPrediksiM))
bandingHasilM
```

A data.frame: 12 × 4

	actuals	fit	lwr	upr
	<dbl>	<dbl>	<dbl>	<dbl>
23	116	127.0544	120.51348	133.5953
24	112	133.1544	126.83878	139.4700
26	191	167.3230	159.60981	175.0362
27	122	131.2789	124.69436	137.8635
28	110	133.1725	126.92572	139.4193
29	35	103.0845	95.02119	111.1478
30	120	137.4871	131.20916	143.7650
32	74	116.7346	109.32910	124.1401
33	106	128.8021	121.73870	135.8654
34	114	132.2856	126.03232	138.5390
35	320	213.8316	204.17962	223.4836
38	144	131.9955	125.76760	138.2235

Gunakan fungsi **cor()** untuk mengukur korelasi hasil prediksi dengan data aktual, jika nilai korelasi mendekati +1 maka model dapat dikatakan baik.

```
cor(bandingHasilM)
```

A matrix: 2 × 2 of type dbl

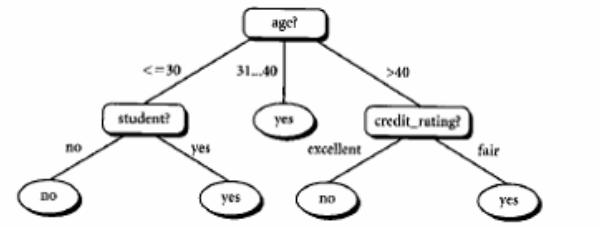
	actuals	predicted
actuals	1.0000000	0.4956762
predicted	0.4956762	1.0000000

+ Code + Markdown

BAB VIII
ALGORITMA POHON KEPUTUSAN

8.1. Konsep Pohon Keputusan

Pohon keputusan merupakan grafik diagram alir yang mewakili proses pengambilan keputusan, dimana grafik diagram alir tersebut menyerupai bentuk pohon. Pohon keputusan dapat digunakan seseorang untuk menentukan keputusan yang sulit dengan menyederhanakannya ke dalam pilihan yang lebih mudah. Setiap pohon keputusan memiliki simpul (*node*) dan cabang (*branch*) yang menghubungkan setiap simpul (*nodes*). Bagian simpul yang terletak di bagian bawah pohon keputusan disebut dengan Klass Keputusan (*leaves*) sedangkan simpul paling atas dari pohon keputusan disebut dengan root. Melalui root inilah dapat diketahui keseluruhan sample data training yang sudah terbagi ke dalam klas-klas keputusan tertentu. Setiap simpul pada pohon keputusan (kecuali *leaves*) disebut sebagai simpul keputusan. Simpul keputusan inilah yang memberikan informasi keputusan berdasarkan fitur tunggal berupa value yang merujuk pada salah satu leaves yang dimilikinya. Model pohon keputusan seperti diperlihatkan pada gambar di bawah



Berikut adalah algoritma pohon keputusan :

Input : Himpunan sample data training S, berisi sample training, label training dan atribut

1. Pilih atribut yang lebih dominan
2. Pisahkan himpunan S ke dalam beberapa subset dengan menggunakan feature selection, tempatkan pada root pohon keputusan. Jumlah anak simpul setiap root bergantung pada jumlah value dari masing-masing atributnya.
3. Lakukan secara berulang : Tentukan atribut yang paling dominan untuk setiap subset yang sudah digenerate pada langkah 2 kemudian bagi menjadi subset ke bawah dari setiap simpul cabang. Jika setiap subset berisi hanya satu klass (simpul leaf), maka BERHENTI; Jika tidak, ULANGI langkah 3.

Output : Pohon keputusan

Penentuan atribut yang dominan (significant) dapat dilakukan dengan menghitung nilai informasi Gain (Information Gain) sebagai berikut :

$$Information\ Gain\ (S, F_j) = Entropy\ (S) - \sum_{V_i \in V_{F_j}} \frac{|S_{V_i}|}{|S|} . Entropy\ (S_{V_i})$$

diamna V_{fj} merupakan himpunan semua nilai yang memungkinkan dari suatu atribut F_j dan S_{vi} merupakan subset dari S, dimana F_j memiliki nilai v_i. Perhitungan nilai Entropy dapat dilakukan menggunakan persamaan Shannon Entropy sebagai berikut :

$$Entropy\ (S) = \sum_{i=1}^{c} -p_i . log_2(p_i)$$

dimana p_i merupakan proporsi sample training terhadap klas ke – i

Studi kasus

Generate pohon keputusan pembelian tikel travel oleh calon konsumen agen travel jika diketahui data penjualan tiket agen travel sebagai berikut :

S	F1	F2	F3	F4	F5
Example	**Type of call**	**Lang. Fluency**	**Ticket type**	**Age**	**Decision attribute**
E1	Local (1)	Fluent(1)	Long(3)	Very young(1)	Buy(1)
E2	Local (1)	Fluent(1)	Local(1)	Old(4)	Buy(1)
E3	Long dist.(2)	Not fluent(3)	Short(3)	Very old(5)	Buy(1)
E4	Intern. (3)	Accent(2)	Long(3)	Very old(5)	Buy(1)
E5	Local (1)	Fluent(1)	Short(3	Middle(3)	Buy(1)
E6	Local (1)	Not fluent(3)	Short(3	Very young(1)	Not buy (2)
E7	Intern. (3)	Fluent(1)	Short(3	Middle(3)	Not buy (2)
E8	Intern. (3)	Foreign(4)	Long(3)	Young (2)	Not buy (2)
E9	Local (1)	Not fluent(3)	Long(3)	Middle(3)	Not buy (2)

Sumber : Cios dkk, 2007

Penyelesaian :

Langkah pertama, tentukan jumlah himpunan positif (keputusan membeli) dan negatif (keputusan tidak membeli)

Pada kasus ini, jumlah himpunan positif (konsumen membeli) yakni sebanyak 5 sedangkan himpunan negatif (konsumen tidak membeli) sebanyak 4.

Langkah kedua, hitung nilai entropy sample training S berdasarkan keputusan positif dan negatifnya.

$$Entropy\ (S) = \sum_{i=1}^{c} -p_i . log_2(p_i)$$

$$Entropy\ (S) = -\frac{5}{9} log_2\left(\frac{5}{9}\right) - \frac{4}{9} log_2\left(\frac{4}{9}\right) = 0.9911$$

Langkah ketiga, hitung nilai entropy setiap atribut terhadap entropy S. Perhitungan Entropy dilakukan terhadap klass atribut Decision.

1. Menghitung nilai **Entropy F1 (Type of Call)**

Atribut F1 memiliki tiga buah jenis Value yaitu **Local** sebanyak 5 buah, **Intern**. sebanyak 3 buah dan **Long dist**. sebanyak 1. Dengan demikian perhitungan nilai entropy untuk setiap value atribut diperoleh sebagai berikut :

$$Entropy\ (S, F1_{local})$$
$$= -\frac{F1Local\ ke\ Buy}{totalF1Local} log_2\left(\frac{F1Local\ ke\ Buy}{totalF1Local}\right)$$
$$-\frac{F1Local\ ke\ NotBuy}{totalF1Local} log_2\left(\frac{F1Local\ ke\ NotBuy}{totalF1Local}\right)$$

$$Entropy\ (S, F1_{local}) = -\frac{3}{5} log_2\left(\frac{3}{5}\right) - \frac{2}{5} log_2\left(\frac{2}{5}\right) = 0.971$$
$$Entropy\ (S, F1_{long\ dist.}) = -\frac{1}{1} log_2\left(\frac{1}{1}\right) - 0 = 0$$
$$Entropy\ (S, F1_{international}) = -\frac{1}{3} log_2\left(\frac{1}{3}\right) - \frac{2}{3} log_2\left(\frac{2}{3}\right)$$
$$= 0.918$$

2. Menghitung nilai **Entropy F2 (Lang. Fluency)**
Atribut F2 memiliki empat jenis value yaitu **Fluent** sebanyak 4 buah, **Accent** sebanyak 1 buah, **Not fluent** sebanyak 3 buah dan **Foreign** sebanyak 1 buah. Dengan demikian perhitungan nilai entropy untuk setiap value atribut diperoleh sebagai berikut :

$$Entropy\ (S, F2_{fluent}) = -\frac{3}{4} log_2\left(\frac{3}{4}\right) - \frac{1}{4} log_2\left(\frac{1}{4}\right) = 0.811$$
$$Entropy\ (S, F2_{Accent}) = -\frac{1}{1} log_2\left(\frac{1}{1}\right) - 0 = 0$$
$$Entropy\ (S, F2_{not\ fluent}) = -\frac{1}{3} log_2\left(\frac{1}{3}\right) - \frac{2}{3} log_2\left(\frac{2}{3}\right) = 0.918$$
$$Entropy\ (S, F2_{foreign}) = -0 - \frac{1}{1} log_2\left(\frac{1}{1}\right) = 0$$

3. Menghitung nilai **Entropy F3 (Ticket Type)**

Atribut F3 memiliki tiga jenis value yaitu **local** sebanyak 1 buah, **short** sebanyak 4 buah, **long** sebanyak 4 buah. Dengan demikian perhitungan nilai entropy untuk setiap value atribut diperoleh sebagai berikut :

$$Entropy\ (S, F3_{local}) = -\frac{1}{1} log_2 \left(\frac{1}{1}\right) - 0 = 0$$

$$Entropy\ (S, F3_{short}) = -\frac{2}{4} log_2 \left(\frac{2}{4}\right) - \frac{2}{4} log_2 \left(\frac{2}{4}\right) = 1$$

$$Entropy\ (S, F3_{long}) = -\frac{2}{4} log_2 \left(\frac{2}{4}\right) - \frac{2}{4} log_2 \left(\frac{2}{4}\right) = 1$$

4. Menghitung nilai **Entropy F4 (Age)**
   Atribut F4 memiliki lima jenis value yaitu **very young** sebanyak 2 buah, **young** sebanyak 1 buah, **middle** sebanyak 3 buah, **old** sebanyak 1 buah dan **very old** sebanyak 2 buah. Dengan demikian perhitungan nilai entropy untuk setiap value atribut diperoleh sebagai berikut :

$$\text{Entropy}\ \left(\text{S}, \text{F4}_{\text{very}_{\text{young}}}\right) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$$

$$\text{Entropy}\ \left(\text{S}, \text{F4}_{\text{young}}\right) = -0 - \frac{1}{1}\log_2\left(\frac{1}{1}\right) = 0$$

$$\text{Entropy}\ \left(\text{S}, \text{F4}_{\text{middle}}\right) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right) = 0.918$$

$$Entropy\ (S, F4_{old}) = -\frac{1}{1} log_2 \left(\frac{1}{1}\right) - 0 = 0$$

$$Entropy\ \left(S, F4_{very_{old}}\right) = -\frac{2}{2} log_2 \left(\frac{2}{2}\right) - 0 = 0$$

Langkah keempat, hitung nilai Information Gain setiap atribut untuk menentukan mana atribut yang lebih dominan dan akan dijadikan sebagai root tree.
Hasil perhitungan Informasi Gain untuk setiap atribut F1, F2, F3 dan F4 diperoleh sebagai berikut :

$$Information\ Gain\ (S, F_j)$$
$$= Entropy\ (S) - \sum_{V_i \in V_{F_j}} \frac{|S_{V_i}|}{|S|} . Entropy\ (S_{V_i})$$

Atribut F1 (Type of Call) :

$$Information\ Gain\ (S, F_1)$$
$$= 0.9911 - \left\{\frac{JumF1_{local}}{TotalS} . 0.971 + \frac{JumF1_{long_{dist}}}{TotalS} . 0 + \frac{JumF1_{inter}}{TotalS} . 0.918\right\} = 0.1456$$

$$Information\ Gain\ (S, F_1)$$
$$= 0.9911 - \left\{\frac{5}{9} . 0.971 + \frac{1}{9} . 0 + \frac{3}{9} . 0.918\right\}$$
$$= 0.1456$$

Atribut F2 (Lang. Fluency) :

$$Information\ Gain\ (S, F_2)$$
$$= 0.9911 - \left\{\frac{4}{9} . 0.811 + \frac{1}{9} . 0 + \frac{3}{9} . 0.918 + \frac{1}{9} . 0\right\}$$
$$= 0.324$$

Atribut F3 (Ticket Type) :

$$Information\ Gain\ (S, F_3) = 0.9911 - \left\{\frac{1}{9} . 0 + \frac{4}{9} . 1 + \frac{4}{9} . 1\right\}$$
$$= 0.102$$

Atribut F4 (Age) :

$$Information\ Gain\ (S, F_4)$$
$$= 0.9911 - \left\{\frac{2}{9} . 1 + \frac{1}{9} . 0 + \frac{3}{9} . 0.918 + \frac{1}{9} . 0 + \frac{2}{9} . 0\right\}$$
$$= 0.462$$

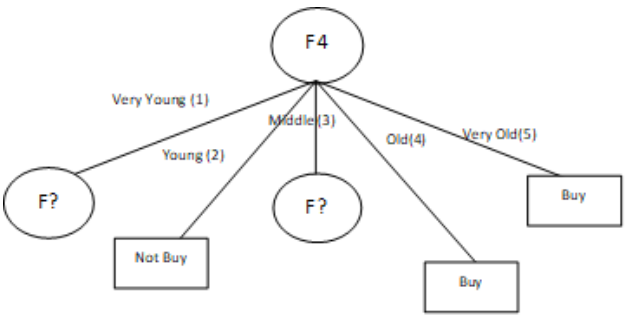
Langkah kelima, turunkan subset untuk root tree berdasarkan value atribut dari atribut yang terpilih sebagai root tree

Pada langkah ini dipilih atribut yang akan dijadikan root berdasarkan nilai informasi gain yang paling tinggi. Informasi gain masing-masing atribut hasil perhitungan pertama seperti di bawah ini :

Atribut	Informasi Gain
F1 (Type of Call)	0.1456
F2 (Lang. Fluency)	0.324
F3 (Ticket Type)	0.102
F4 (Age)	0.462

Berdasarkan perhitungan informasi gain, dapat dilihat bahwa nilai gain tertinggi dimiliki oleh atribut F4 (Age), sehingga dengan demikian F4 (Age) dijadikan sebagai root dari pohon keputusan yang akan dibentuk.

Sedangkan simpul keputusan dilihat berdasarkan value atribut dari F4, yakni dengan melihat apakah value tersebut memiliki class target berbeda atau hanya satu. Jika value atribut memiliki class target berbeda, maka dilakukan perhitungan nilai gain informasi kembali terhadap atribut tertentu yang merujuk pada value atribut yang menjadi root pohon keputusan. Gambar di bawah ini merupakan ilustrasi terbentuknya root pohon keputusan dengan simpul keputusan awal menurut value atribut yang dimilikinya.



Pada pohon keputusan awal yang terbentuk terlihat bahwa hanya value attribute very young (1) dan middle (3) saja yang memiliki simpul simpul lainnya yang harus dicari nilai informasi gainnya sesuai dengan value atribut F4 yang merujuk pada class targetnya. Ini dikarenakan, kedua value atribut tersebut memiliki dua jenis value class target yang berbeda, klass Buy dan klas Not Buy, sedangkan untuk tiga value attribut lainnya hanya memiliki satu klas saja, hanya buy saja atau not buy saja.

Langkah keenam, ulangi langkah ketiga sampai kelima hingga setiap subset hanya memiliki masing-masing satu klass.
Setelah pada langkah kelima dilakukan penentuan atribut root dan terpilih atribut F4 menjadi atribut root, maka data training awal kemudian menyusut setelah terjadi split data training sebagai berikut :

S	**F1**	**F2**	**F3**	**F5**
Example	**Type of call**	**Lang. Fluency**	**Ticket type**	**Decision attribute**
E1	Local (1)	Fluent(1)	Long(3)	Buy(1)
E5	Local (1)	Fluent(1)	Short(3	Buy(1)
E6	Local (1)	Not fluent(3)	Short(3	Not buy (2)
E7	Intern. (3)	Fluent(1)	Short(3	Not buy (2)
E9	Local (1)	Not fluent(3)	Long(3)	Not buy (2)

Menghitung Entropy berdasarkan keputusan positif dan negatif

$$Entropy\ (S) = \sum_{i=1}^{c} -p_i . log_2(p_i)$$

$$Entropy\ (S) = -\frac{2}{5} log_2\left(\frac{2}{5}\right) - \frac{3}{5} log_2\left(\frac{3}{5}\right) = 0.9709$$

Menentukan leaf node untuk **Very Young**

1. Menghitung nilai Entropy F1 (Type of Call)
   Atribut F1 memiliki dua buah jenis Value yaitu **Local** sebanyak 4 buah dan **Intern**. sebanyak 1 buah. Dengan demikian perhitungan nilai entropy untuk setiap value atribut diperoleh sebagai berikut :

$$Entropy\ (S, F1_{local})$$
$$= -\frac{F1Local\ ke\ Buy}{totalF1Local} log_2 \left(\frac{F1Local\ ke\ Buy}{totalF1Local}\right)$$
$$-\frac{F1Local\ ke\ NotBuy}{totalF1Local} log_2 \left(\frac{F1Local\ ke\ NotBuy}{totalF1Local}\right)$$

$$Entropy\ (S, F1_{local}) = -\frac{2}{4} log_2 \left(\frac{2}{4}\right) - \frac{2}{4} log_2 \left(\frac{2}{4}\right) = 1$$
$$Entropy\ (S, F1_{international}) = -0 - \frac{1}{1} log_2 \left(\frac{1}{1}\right) = 0$$

2. Menghitung nilai Entropy F2 (Lang. Fluency)
   Atribut F2 memiliki dua jenis value yaitu **Fluent** sebanyak 3 buah dan **Not fluent** sebanyak 2 buah. Dengan demikian perhitungan nilai entropy untuk setiap value atribut diperoleh sebagai berikut :
   $$Entropy\ (S, F2_{fluent}) = -\frac{2}{3} log_2 \left(\frac{2}{3}\right) - \frac{1}{3} log_2 \left(\frac{1}{3}\right) = 0.918$$
   $$Entropy\ (S, F2_{not\ fluent}) = -0 - \frac{2}{2} log_2 \left(\frac{2}{2}\right) = 0$$

3. Menghitung nilai Entropy F3 (Ticket Type)
   Atribut F3 memiliki dua jenis value yaitu long sebanyak 2 buah dan short sebanyak 3 buah. Dengan demikian perhitungan nilai entropy untuk setiap value atribut diperoleh sebagai berikut :
   $$Entropy\ (S, F3_{short}) = -\frac{1}{3} log_2 \left(\frac{1}{3}\right) - \frac{2}{3} log_2 \left(\frac{2}{3}\right) = 0.918$$
   $$Entropy\ (S, F3_{long}) = -\frac{1}{2} log_2 \left(\frac{1}{2}\right) - \frac{1}{2} log_2 \left(\frac{1}{2}\right) = 1$$

Menghitung Informasi Gain untuk setiap atribut :
Hasil perhitungan Informasi Gain untuk setiap atribut F1, F2 dan F3 diperoleh sebagai berikut :

$$Information\ Gain\ (S, F_j) = Entropy\ (S) - \sum_{V_i \in V_{F_j}} \frac{|S_{V_i}|}{|S|} . Entropy\ (S_{V_i})$$

Atribut F1 (Type of Call) :

$$Information\ Gain\ (S, F_1) = 0.9709 - \left\{\frac{4}{5} . 1 + \frac{1}{5} . 0\right\} = 0.1709$$

Atribut F2 (Lang. Fluency) :

$$Information\ Gain\ (S, F_2) = 0.9709 - \left\{\frac{3}{5} . 0.918 + \frac{2}{5} . 0\right\} = 0.4201$$

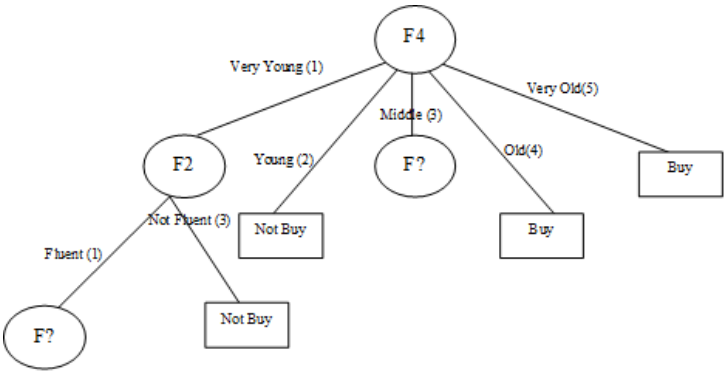
Atribut F3 (Ticket Type) :

$$Information\ Gain\ (S, F_3) = 0.9709 - \left\{\frac{3}{5} . 0.918 + \frac{2}{5} . 1\right\} = 0.0201$$

Pada langkah ini dipilih atribut yang akan dijadikan root berdasarkan nilai informasi gain yang paling tinggi. Informasi gain masing-masing atribut hasil perhitungan pertama seperti di bawah ini :

Atribut	Informasi Gain
F1 (Type of Call)	0.1709
F2 (Lang. Fluency)	0.4201
F3 (Ticket Type)	0.0201

Berdasarkan perhitungan informasi gain, dapat dilihat bahwa nilai gain tertinggi dimiliki oleh atribut **F2 (Lang. Fluency),** sehingga dengan demikian **F2 (Lang. Fluency)** dijadikan sebagai leaf node untuk Very Young. Berikut bentuk pohon keputusan setelah leaf node very young dipilih:



Menentukan leaf node untuk **Middle**

Setelah pada langkah sebelumnya dilakukan penentuan atribut root dan terpilih atribut F2 menjadi leaf node, maka data training awal kemudian menyusut setelah terjadi split data training sebagai berikut :

S	**F1**	**F3**	**F5**
Example	**Type of call**	**Ticket type**	**Decision attribute**
E1	Local (1)	Long(3)	Buy(1)
E5	Local (1)	Short(3	Buy(1)
E6	Local (1)	Short(3	Not buy (2)
E7	Intern. (3)	Short(3	Not buy (2)
E9	Local (1)	Long(3)	Not buy (2)

Menghitung Entropy berdasarkan keputusan positif dan negatif

$$Entropy\ (S) = \sum_{i=1}^{c} -p_i . log_2(p_i)$$

$$Entropy\ (S) = -\frac{2}{5} log_2\left(\frac{2}{5}\right) - \frac{3}{5} log_2\left(\frac{3}{5}\right) = 0.9709$$

1. Menghitung nilai Entropy F1 (Type of Call)

Atribut F1 memiliki dua buah jenis Value yaitu Local sebanyak 4 buah dan Intern. sebanyak 1 buah. Dengan demikian perhitungan nilai entropy untuk setiap value atribut diperoleh sebagai berikut :

$$Entropy\ (S, F1_{local})$$
$$= -\frac{F1Local\ ke\ Buy}{totalF1Local} log_2 \left(\frac{F1Local\ ke\ Buy}{totalF1Local}\right)$$
$$-\frac{F1Local\ ke\ NotBuy}{totalF1Local} log_2 \left(\frac{F1Local\ ke\ NotBuy}{totalF1Local}\right)$$

$$Entropy\ (S, F1_{local}) = -\frac{2}{4} log_2 \left(\frac{2}{4}\right) - \frac{2}{4} log_2 \left(\frac{2}{4}\right) = 1$$
$$Entropy\ (S, F1_{international}) = -0 - \frac{1}{1} log_2 \left(\frac{1}{1}\right) = 0$$

2. Menghitung nilai Entropy F3 (Ticket Type)
   Atribut F3 memiliki dua jenis value yaitu long sebanyak 2 buah dan short sebanyak 3 buah. Dengan demikian perhitungan nilai entropy untuk setiap value atribut diperoleh sebagai berikut :
   $Entropy\ (S, F3_{short}) = -\frac{1}{3} log_2 \left(\frac{1}{3}\right) - \frac{2}{3} log_2 \left(\frac{2}{3}\right) = 0.918$
   $Entropy\ \left(S, F3_{long}\right) = -\frac{1}{2} log_2 \left(\frac{1}{2}\right) - \frac{1}{2} log_2 \left(\frac{1}{2}\right) = 1$

Menghitung Informasi Gain untuk setiap atribut :
Hasil perhitungan Informasi Gain untuk setiap atribut F1 dan F2 diperoleh sebagai berikut :

$$Information\ Gain\ \left(S, F_j\right)$$
$$= Entropy\ (S) - \sum_{V_i \in V_{F_j}} \frac{\left|S_{V_i}\right|}{|S|} . Entropy\ \left(S_{V_i}\right)$$

Atribut F1 (Type of Call) :

$$Information\ Gain\ (S, F_1) = 0.9709 - \left\{\frac{4}{5} . 1 + \frac{1}{5} . 0\right\} = 0.1709$$

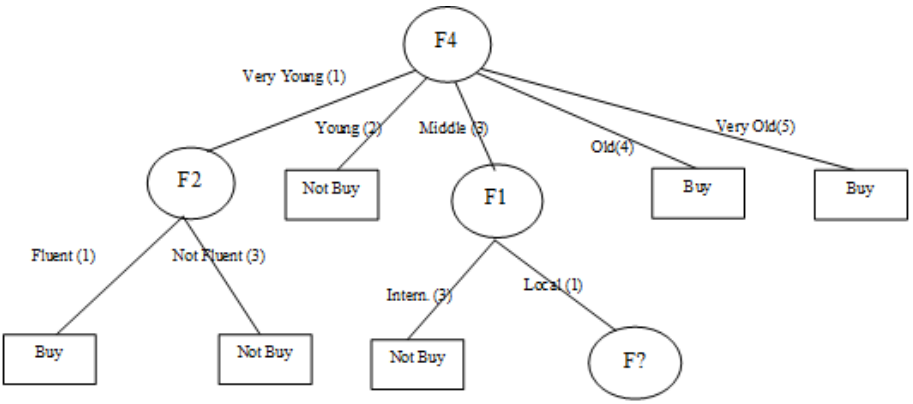
Atribut F3 (Ticket Type) :

$$Information\ Gain\ (S, F_3) = 0.9709 - \left\{\frac{3}{5} . 0.918 + \frac{2}{5} . 1\right\}$$
$$= 0.0201$$

Pada langkah ini dipilih atribut yang akan dijadikan root berdasarkan nilai informasi gain yang paling tinggi. Informasi gain masing-masing atribut hasil perhitungan pertama seperti di bawah ini :

Atribut	Informasi Gain
F1 (Type of Call)	0.1709
F3 (Ticket Type)	0.0201

Berdasarkan perhitungan informasi gain, dapat dilihat bahwa nilai gain tertinggi dimiliki oleh atribut F1 (Type of Call), sehingga dengan demikian F1 (Type of Call) dijadikan sebagai leaf node untuk Middle. Berikut bentuk pohon keputusan setelah leaf node very young dipilih :



Setelah pada langkah sebelumnya dilakukan penentuan atribut root dan terpilih atribut F1 menjadi leaf node, maka data training awal kemudian menyusut setelah terjadi split data training sebagai berikut :

S	F3	F5
Example	**Ticket type**	**Decision attribute**
E1	Long(3)	Buy(1)

E5	Short(3	Buy(1)
E6	Short(3	Not buy (2)
E9	Long(3)	Not buy (2)

Menghitung Entropy berdasarkan keputusan positif dan negatif

$$Entropy\ (S) = \sum_{i=1}^{c} -p_i . log_2(p_i)$$

$$Entropy\ (S) = -\frac{2}{4} log_2\left(\frac{2}{4}\right) - \frac{2}{4} log_2\left(\frac{2}{4}\right) = 1$$

Menghitung nilai Entropy F3 (Ticket Type)

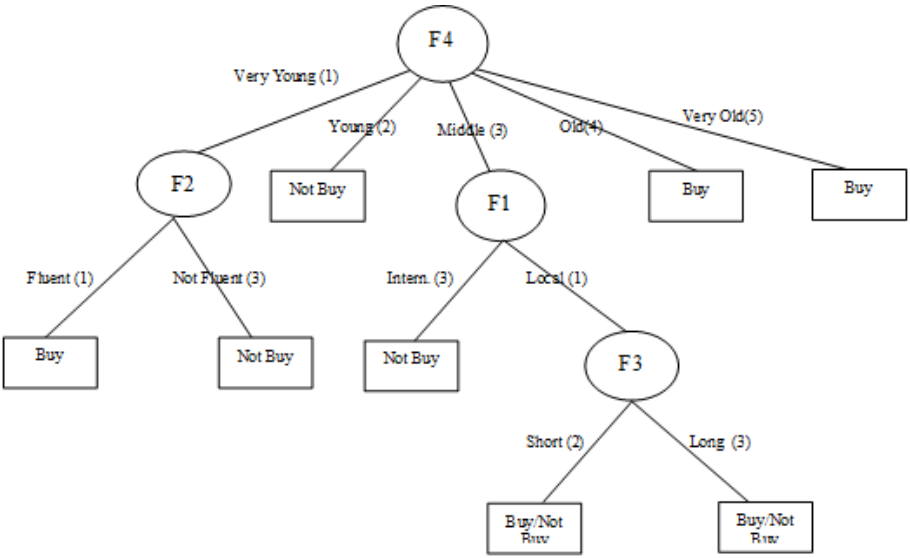
Atribut F3 memiliki dua jenis value yaitu long sebanyak 2 buah dan short sebanyak 2 buah. Dengan demikian perhitungan nilai entropy untuk setiap value atribut diperoleh sebagai berikut :

$$Entropy\ (S, F3_{short}) = -\frac{1}{2} log_2\left(\frac{1}{2}\right) - \frac{1}{2} log_2\left(\frac{1}{2}\right) = 1$$

$$Entropy\ \left(S, F3_{long}\right) = -\frac{1}{2} log_2\left(\frac{1}{2}\right) - \frac{1}{2} log_2\left(\frac{1}{2}\right) = 1$$

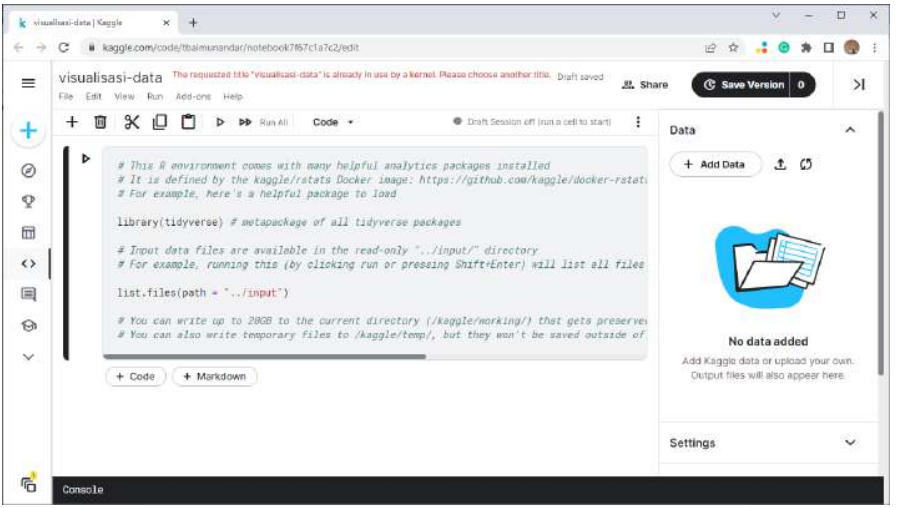
Karena setiap value atribut memiliki nilai entropy yang sama maka keputusan untuk leave paling akhir memiliki kemungkinan dua duanya, yaitu Buy atau Not Buy.

Langkah ketujuh, gambarkan graf akhir pohon keputusan



8.2. Pohon Keputusan Menggunakan R

Untuk memulai visualisasi data menggunakan Bahasa R, langkah pertama adalah login ke dalam kaggle.com menggunakan akun masing-masing. Kemudian pada manu navigator bagian kiri pilih menu < > **Code** kemudian klik tombol **New Notebook** sampai terbuka halaman seperti pada Gambar di bawah. Untuk lebih jelas bagaimana membuat akun dan membuka notebook baru silahkan pelajari Bab 5.



Langkah selanjutnya adalah menyiapkan dataset menggunakan perintah "read.csv" pastikan data sudah diupload ke dalam kaggle dengan format *.csv. Pada pembahasan buku ini, datasets yang digunakan adalah **dataset-pinjaman-nasabah.csv**. Dataset disimpan ke dalam variable **dataHistori.** Kemudian tampilkan 5 data dari urutan 20 sampai 25 menggunakan pemanggilan urutan baris data. Berikut baris kode yang digunakan :

```
dataHistori=read.csv('../input/pinjamankreditnasabah/dataset-pinjaman-nasabah.csv',sep=',')
dataHistori[20:25,]
```

A data.frame: 6 × 13

	ID_Nasabah	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman
	<chr>	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>
20	LP001041	Male	Yes	0	Graduate		2600	3500	115
21	LP001043	Male	Yes	0	Not Graduate	No	7660	0	104
22	LP001046	Male	Yes	1	Graduate	No	5955	5625	315
23	LP001047	Male	Yes	0	Not Graduate	No	2600	1911	116
24	LP001050		Yes	2	Not Graduate	No	3365	1917	112
25	LP001052	Male	Yes	1	Graduate		3717	2925	151

+ Code + Markdown

Pada bari kode di atas, perintah kedua **dataHistori[20:25,]** merupakan perintah untuk menampilkan data dari nomor urut 20 sampai 25. Hasil memperlihatkan ada beberapa value atribut memiliki nilai kosong (empty). Perhatikan variable **JenisKelamin** pada nomor data 24, dan variable **Wiraswasta** nomor data 20 dan 25. Keduanya memiliki value attribute kosong (empty). Oleh karenanya diperlukan penangan berupa pre-processing untuk mengganti empty value pada variable menjadi N/A. Penggantian value ini bertujuan untuk mempermudah pre-processing saat menghapus data dengan nilai N/A.

Panggil ulang dataset pinjaman nasabah kemudian tambahkan perintah **na.string=c("")** untuk mengganti value atribute yang bernilai null atau kosong menjadi NA. Dengan demikian lebih mudah untuk dilakukan pre-processing. Berikut baris kode pemanggilan dataset yang dimodifikasi dengan **na.string=c(“”).**

```
dataHistori2=read.csv('../input/pinjamankreditnasabah/dataset-pinjaman-nasabah.csv',
                      sep=',', na.string=c(""))
dataHistori2[20:25,]
```

A data.frame: 6 × 13

	ID_Nasabah	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman
	<chr>	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>
20	LP001041	Male	Yes	0	Graduate	NA	2600	3500	115
21	LP001043	Male	Yes	0	Not Graduate	No	7660	0	104
22	LP001046	Male	Yes	1	Graduate	No	5955	5625	315
23	LP001047	Male	Yes	0	Not Graduate	No	2600	1911	116
24	LP001050	NA	Yes	2	Not Graduate	No	3365	1917	112
25	LP001052	Male	Yes	1	Graduate	NA	3717	2925	151

+ Code + Markdown

Perhatikan hasil pemanggilan dataset setelah diberlakukan perintah **na.string=c(""),** ada perubahan value atribut data instance ke 24 pada variable JenisKelamin, data instance ke 20 dan 25 pada

variable Wiraswasta terisi nilai NA, sebelum diberlakukan perintah, berisi value kosong.

Tahap berikutnya adalah melakukan data cleaning untuk instance yang memiliki value NA. Gunakan perintah **na.omit(dataset)** untuk menghapus semua data instance yang memiliki value NA. Konsekuensi dari pre-processing data cleaning ini adalah kita akan kehilangan data bisa jadi sangat banyak. Tergantung data instance yang ada pada datasets yang memilik value NA. Konsekuensi lainnya adalah kita akan kehilangan beberapa informasi bahkan bisa jadi banyak tergantung jumlah data instance yang di bersihkan. Meskipun demikian, Sebagian ilmuwan data menganggap bahwa data cleaning berupa penghapusan data dengan value NA atau sejenis bisa jadi tidak dibutuhkan. Alternatif lainnya adalah dengan mengganti value yang lebih representative. Berikut adalah baris kode untuk data cleaning :

```
dataFix=na.omit(dataHistori2)
dataFix[19:25,]
```

A data.frame: 7 × 13

	ID_Nasabah	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman
	<chr>	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>
23	LP001047	Male	Yes	0	Not Graduate	No	2600	1911	116
26	LP001066	Male	Yes	0	Graduate	Yes	9560	0	191
27	LP001068	Male	Yes	0	Graduate	No	2799	2253	122
28	LP001073	Male	Yes	2	Not Graduate	No	4226	1040	110
29	LP001086	Male	No	0	Not Graduate	No	1442	0	35
32	LP001095	Male	No	0	Graduate	No	3167	0	74
33	LP001097	Male	No	1	Graduate	Yes	4692	0	106

+ Code + Markdown

Hasil data cleaning disimpan pada variable baru yakni **dataFix**. Setelah itu ditampilkan hasil nya dengan memanggil data instance ke 19 sampai 25 menggunakan **dataFix[19:25,].** Hasil data cleaning memperlihatkan data instance ke 22, 24, 25, 25, 30 dan 31 sudah terhapus. Pada kasus ini, kita akan membangun model klasifikasi

menggunakan algoritme pohon keputusan, untuk variable **ID_Nasabah** akan dihilangkan karena tidak berpengaruh terhadap proses pembelajaran algoritma.

```
# Hilangkan variabel ID_Nasabah dari dataset, karena tidak memberikan informasi apapun untuk
# membangun model klasifikasi dengan Pohon Keputusan. Simpan ke dalam variabel dataset baru
# dengan nama dataPre
dataPre=dataFix[,2:13]
head(dataPre)
```

A data.frame: 6 × 12

	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman	JangkaWaktuP
	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>	
2	Male	Yes	1	Graduate	No	4583	1508	128	
3	Male	Yes	0	Graduate	Yes	3000	0	66	
4	Male	Yes	0	Not Graduate	No	2583	2358	120	
5	Male	No	0	Graduate	No	6000	0	141	
6	Male	Yes	2	Graduate	Yes	5417	4196	267	
7	Male	Yes	0	Not Graduate	No	2333	1516	95	

Tahap berikutnya adalah pembagian dataset menjadi dua bagian. Bagi dataset ke dalam data training dan data testing dengan proporsi 80% untuk data training dan 20% untuk datatesting. Pengalokasian data training dan testing dilakukan secara random menggunakan fungsi set.seed(). Berikut adalah baris kode untuk melakukan pembagian data dan pengalokasian data secara random.

```
set.seed(1234)
sTrain=sample(nrow(dataPre), nrow(dataPre)*0.8)
dataTrain = dataPre[sTrain,]
dataTesting = dataPre[-sTrain,]
```

+ Code + Markdown

Perintah **dataPre[sTrain,]** merupakan instruksi untuk mengalokasikan data ke dalam data training dengan proporsi 80% sesuai pada baris perintah kedua. Sedangkan pada perintah **dataPre[-sTrain,]** merupakan pengalokasian sisa data yang sudah dimasukan ke dalam data training untuk dijadikan data testing. Symbol minus (-)

mengindikasikan pengurangan dataset setelah 80% dialokasikan untuk data training.

Untuk data training disimpan dalam variable **dataTrain** sedangkan data testing disimpan dalam **dataTesting.** Data training akan digunakan untuk melatih algoritma pohon keputusan sampai menghasilkan model klasifikasi. Sementara data testing digunakan untuk mengevaluasi model klasifikasi dari pohon keputusan. Kita juga bisa memeriksa proporsi kelas target untuk data keseluruhan, data training dan data testing. Baris kode berikut merupakan perintah untuk memeriksa proporsi kelas target.

```
[23]: # Periksa proporsi kelas target dari keseluruhan data
      prop.table(table(dataFix$StatusPinjaman))
```

```
        N         Y
0.3083333 0.6916667
```

+ Code + Markdown

```
[24]: # Periksa proporsi kelas target untuk data training
      prop.table(table(dataTrain$StatusPinjaman))
```

```
        N         Y
0.3072917 0.6927083
```

```
▷ # Periksa proporsi kelas target untuk data testing
  prop.table(table(dataTesting$StatusPinjaman))
```

```
    N     Y
0.3125 0.6875
```

Berdasarkan proporsi di atas, dapat diketahui bahwa proporsi data paling banyak menampilkan kelas target Y dibandingkan N, baik untuk keseluruhan data, data training maupun data testing. Informasi

ini mengindetifikasi awal bahwa model yang terbentuk akan cenderung lebih mengenali pola klasifikasi ke keputusan Y dibandingkan N. Idealnya, dataset yang dimiliki harusnya memiliki kelas target yang imbang sehingga model yang dihasilkan mampu mengenali kelas dengan adil.

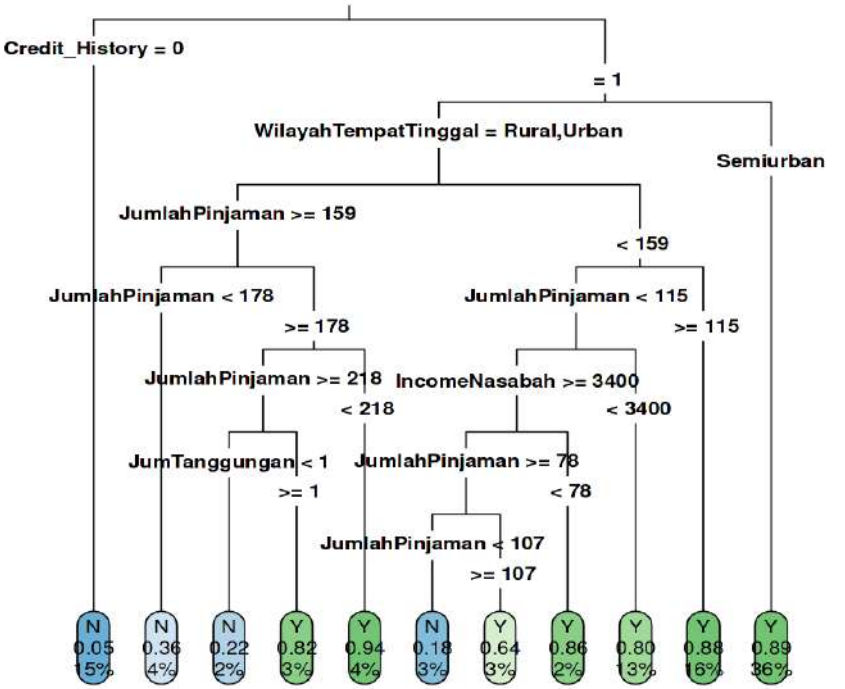
Tahap berikutnya adalah membangun model klasifikasi dengan melatih algoritma pohon keputusan menggunakan data training. Gunakan fungsi **rpart()** untuk membangun model klasifikasi. Oleh karenanya kita membutuhkan **library (rpart)** agar fungsi dapat dipanggil. Selain itu, untuk memvisualisasikan pohon keputusan yang terbentuk, kita bisa menggunakan fungsi **rpart.plot()** dengan memanggil **library(rpart.plot).**

rpart() sendiri merupakan kependekan dari *Recursive Partitioning and Regression Trees*. Syntax penulisan fungsi rpart adalah *rpart(formula, data, weights, subset, ...)*. Formula diisikan persamaan yang memuat variabel respon (dependen) dan variabel prediktor (independen). Sedangkan data diisi dengan data training yang digunakan untuk melatih algoritma. Berikut adalah perintah untuk membangun model klasifikasi yang diawali dengan memanggil **library(rpart)** dan **library(rpart.plot).**

Baris ketiga pada kode di atas merupakan perintah untuk melatih algoritma pohon keputusan menggunakan fungsi **rpart().** Formula yang digunakan pada model ini adalah **StatusPinjaman ~.** Sedangkan

untuk parameter **data=dataTrain** merupakan data latih yang digunakan untuk melatih algoritme pohon keputusan. Untuk menampilkan pohon keputusan yang terbentuk gunakan fungsi **rpart.plot()** sebagai berikut:

```
rpart.plot(modelDT, type=3)
```



Pada pohon keputusan yang terbentuk diketahui bahwa root node nya adalah variabel Credit_History. Ini berarti bahwa setiap ada data baru yang akan diklasifikasikan, yang pertama kali dilihat adalah variabel Credit_History yang dimiliki nasabah. Jika nasabah tidak memiliki Credit_History, maka keputusannya langsung di tolak (N), sedangkan juika Credit_History ada (1) maka variabel yang dicek

selanjutnya adalah WilayahTempatTinggal. Jika WilayahTempatTinggal nya Rural, Urban , maka variabel JumlahPinjaman akan dicek, sedangkan jika WilayahTempatTiggal Semi Urban maka pengajuan langsung diterima (Y). Begitu seterusnya membaca pohon keputusan yang terbentuk untuk mengklasifikasikan data baru.

```
library(caret)
predikTrain=predict(modelDT, dataTesting, type="class")
confusionMatrix(table(predikTrain, dataTesting$StatusPinjaman))
```

```
Confusion Matrix and Statistics


predikTrain  N  Y
          N 12  9
          Y 18 57

               Accuracy : 0.7188
                 95% CI : (0.6178, 0.8058)
    No Information Rate : 0.6875
    P-Value [Acc > NIR] : 0.2944

                  Kappa : 0.2871

 Mcnemar's Test P-Value : 0.1237

            Sensitivity : 0.4000
            Specificity : 0.8636
         Pos Pred Value : 0.5714
         Neg Pred Value : 0.7600
             Prevalence : 0.3125
         Detection Rate : 0.1250
   Detection Prevalence : 0.2188
      Balanced Accuracy : 0.6318

       'Positive' Class : N
```

Setelah model klasifikasi terbentuk, tahap selanjutnya adalah mengevaluasi model menggunakan data training. Evaluasi model dilakukan untuk mengukur akurasi yang dimiliki. Langkah pertama adalah menggunakan data testing untuk diujikan ke dalam model. Pada pengujian ini dilakukan prediksi terhadap data testing

menggunakan model. Fungsi **predict()** digunakan untuk melakukan prediksi klasifikasi. Selain itu untuk menampilkan akurasi hasil prediksi, dilakukan dengan memanggil **confussionMatrix()** menggunakan library(caret).

Pada baris kode diatas, hasil prediksi terhadap data testing disimpan dalam variabel predikTrain, model yang digunakan adalah modelDT sesuai dengan hasil pada tahap sebelumnya. Fungsi **confusionMatrix()** digunakan untuk menampilkan akurasi hasil prediksi terhadap kelas target dari data testing. Hasil evaluasi model memperlihatkan bahwa akurasi yang dimiliki adalah 71.88%.

Hasil dari Confusion Matrix juga memperlihatkan informasi nilai True Negatif sebanyak 12, True Positif Sebanyak 57, sedangkan False Positif sebanyak 9 dan False Negatif sebanyak 18. Nilai ini dapat digunakan lebih lanjut untuk mengukur nilai Recall dan Precision. Dari rangkuman confusion matrix juga kita dapat mengetahui nilai Kappa sebesar 0.2871. Nilai kappa tersebut masuk dalam kategori lumayan akurat atau lumayan konsisten. Kappa sendiri biasanya digunakan untuk mengukur konsistensi dari dua metode pengkuran terutama untuk data kategorik. Nilai kappa yang baik adalah mendekati 1, artinya, jika nilai kappa mendekati 1, maka indikator penggunaan metode pengukuran saling konsisten, dan biasanya diikuti dengan naiknya akurasi model.

Jika model sudah sesuai kebutuhan, maka model klasifikasi yang terbentuk dapat digunakan untuk mengklasifikasi data baru yang belum memiliki label / kelas. Misalnya jika anda sebagai seorang credit analist dihadapkan pada ratusan calon debitur yang memerlukan penstatusan diterima (Y) atau ditolak (N), anda dapat menggunakan model dengan memasukan data baru ke dalam model. Pada pembahasan buku ini, digunakan datasets baru dengan nama **Data-Baru-Calon-Debitur.csv** yang belum memiliki kelas.

Untuk dapat melakukan klasifikasi terhadap data baru, langkah pertama adalah memanggil dataset seperti biasa menggunakan perintah **read.csv()** dan pastikan dataset **Data-Baru-Calon-Debitur.csv** sudah ditambahkan pada Kaggle anda. Silahkan pelajari lagi Bab 5 untuk memasukan dataset ke dalam platform Kaggle. Simpan datasets baru ke dalam variable **dataBaru** kemudian tampilkan isinya dengan memanggil nama variable datasets baru terebut. Baris kode berikut merupakan proses pemanggilan datasets baru :

```
dataBaru=read.csv('../input/datauntukprediksi/Data-Baru-Calon-Debitur.csv', sep=',')
dataBaru
```

A data.frame: 12 × 13

ID_Nasabah	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman	JangkaWaktuPinjaman	Credit
<chr>	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>	<int>	
LP003002	Male	No	0	Not Graduate	No	3744	0	5000	360	
LP003003	Male	Yes	1	Not Graduate	No	3855	200	6000	240	
LP003004	Female	Yes	0	Graduate	Yes	4908	3500	7600	120	
LP003005	Female	Yes	0	Not Graduate	No	5802	2600	5500	240	
LP003006	Male	No	0	Not Graduate	No	4596	0	3000	360	
LP003007	Male	Yes	2	Graduate	Yes	5161	1200	1200	360	
LP003008	Male	Yes	0	Graduate	Yes	5200	1516	4500	240	
LP003009	Female	No	0	Graduate	Yes	3714	0	4000	360	
LP003010	Male	No	2	Not Graduate	No	4152	0	3500	120	
LP003011	Female	Yes	1	Graduate	No	3782	5000	2750	120	
LP003012	Male	Yes	2	Graduate	Yes	5972	250	400	360	

Ada sebanyak 12 data instance baru yang belum memiliki label / kelas. Seperti pada tahap sebelumnya, untuk variable **ID_Nasabah** akan dihilangkan karena tidak memberikan pengaruh apapun. Sehingga datasets yang ada hanya tinggak 12. Berikut baris kode untuk menghilangkan variable pertama (**ID_Nasabah).**

```
# Hilangkan variabel I_Nasabah
dataPrediksi=dataBaru[,2:13]
head(dataPrediksi)
```

A data.frame: 6 × 12

	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman	Jangk
	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>	
1	Male	No	0	Not Graduate	No	3744	0	5000	
2	Male	Yes	1	Not Graduate	No	3653	200	6000	
3	Female	Yes	0	Graduate	Yes	4908	3500	7600	
4	Female	Yes	0	Not Graduate	No	5802	2600	5500	
5	Male	No	0	Not Graduate	No	4596	0	3000	
6	Male	Yes	2	Graduate	Yes	5161	1200	1200	

+ Code + Markdown

Gunakan **modelDT** yang sudah dibuat sebelumnya untuk memprediksi data baru sehingga diperoleh class nya. Fungsi **predict()** digunakan pada tahap ini, hanya saja untuk data diganti dengan variable **dataPrediksi** sesuai dengan datasets baru yang belum memiliki kelas. Baris kode untuk melakukan prediksi terhadap data baru sebagai berikut :

```
hasilPrediksiM=predict(modelDT, dataPrediksi, type="class")
data.frame(hasilPrediksiM)
```

A data.frame: 12 × 1

	hasilPrediksiM
	<fct>
1	N
2	N
3	Y
4	Y
5	N
6	Y
7	N
8	Y
9	N
10	N
11	Y
12	Y

Parameter **type** pada fungsi **predict()** digunakan untuk menyajikan hasil klasifikasi dalam bentuk kelas kategori sesuai yang ada pada dataset. Pada baris kode di atas, type yang digunakan adalah **'class'** karena digunakan untuk menyajikan hasil prediksi klasifikasi dalam bentuk kelas sesuai yang ada pada model. Selain type **class**, kita juga bisa menggunakan type **prob** untuk menyajikan hasil klasifikasi dalam bentuk matrik probabilitas antar kelas yang ada. Adapun untuk mengindikasikan suatu data instance memiliki kelas tertentu dapat dilihat berdasarkan nilai probabilitas paling tinggi ada pada kelas yang mana. Berikut adalah baris kode untuk menyajikan hasil prediksi dalam bentuk matriks probabilitas antar kelas.

```
hasilPrediksiM=predict(modelDT, dataPrediksi, type="prob")
data.frame(hasilPrediksiM)
```

A data.frame: 12 × 2

	N	Y
	<dbl>	<dbl>
1	0.9464286	0.05357143
2	0.9464286	0.05357143
3	0.1071429	0.89285714
4	0.1071429	0.89285714
5	0.9464286	0.05357143
6	0.1818182	0.81818182
7	0.7777778	0.22222222
8	0.1071429	0.89285714
9	0.9464286	0.05357143
10	0.9464286	0.05357143
11	0.1818182	0.81818182
12	0.1818182	0.81818182

Interpretasi nilai probabilitas di atas adalah : Misal pada data instance pertama, nilai probabilitas kelas N sebesar 0.94, sedangkan untuk kelas Y sebesar 0.05. Dengan demikian, data instance pertama masuk ke dalam kelas N, atau pengajuan peminjaman kredit yang dilakukan oleh nasabah pertama ditolak. Demikian seterusnya untuk data instance yang lain.

BAB IX
ALGORITME NAIVE BAYES

9.1. Dasar Metode Bayes

Teori Bayess merupakan teknik yang digunakan untuk kebutuhan pengenalan pola dan klasifikasi suatu objek. Teori ini bekerja berdasarkan asumsi bahwa pola klasifikasi didasarkan atas nilai-nilai probabilistik yang dimiliki suatu objek berdasarkan pola alami dan fitur yang dimilikinya.

Probabilitas P(X) merupakan peristiwa Xyang dapat terjadi sebanyak n kali diantara N peristiwa dan terjadi saling ekslusif (saling asing/terjadinya peristiwa yang satu mencegah terjadinya peristiwa yang lain) dan masing-masing terjadi dengan kesempatan yang sama dan dapat diformulasikan sebagai berikut :

$$P(X) = \frac{n}{N}\ dimana : 0 \leq P(X) \leq 1$$

Jika probabilitas kejadian P(X) = 0, artinya bahwa peristiwa E pasti tidak terjadi, sedangkan jika P(X) = 1, maka dapat dipastikan peristiwa E terjadi. Selain itu, jika É menyatakan suatu kejadian dan bukan merupakan peristiwa E, maka diperoleh :

P(É) = 1 – P(X) atau berlaku hubungan
P(X) + P(É) = 1

Contoh sederhana misalnya, peluang kejadian munculnya Muka pada saat pelemparan koin adalah 1/2. Maka nilai P(Muka) = ½, sedangkan nilai $P(\dot{Muka})$ adalah ½ diperoleh dari $P(\dot{Muka}) = 1 - P(Muka)$. Contoh lainnya adalah Peluang munculnya angka 5 pada

saat pelemparan mata dadu adalah 1/6, sedangkan peluang munculnya bukan angka 5 pada saat pelemparan mata dadu adalah 5/6.

9.2. Probabilitas Bersyarat

Jika P(X) menyatakan probabilitas kejadian X, P(Y) menyatakan probabilitas kejadian Y, dan probabilitas X dan Y terjadi bersama-sama disimbolkan oleh P(X∩Y) atau P(Y∩X). Sebuah kondisi probabilitas bersyarat berlaku jika kejadian X terjadi kalau kejadian Y terjadi terlebih dahulu yang disimbolkan dengan P(X|Y), sehingga nilainya probabilitas yang dihasilkan diperoleh melalui persamaan sebagai berikut :

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)}$$

Sebaliknya, jika kejadian Y terjadi kalau kejadian X terjadi terlebih dahulu, maka nilai probabilitas P(Y|X) dapat diperoleh dengan persamaan :

$$P(Y|X) = \frac{P(Y \cap X)}{P(X)}$$

Berdasarkan kejadian probabilitas bersyarat dimana jika X dan Y terjadi bersama sama, maka dapat dinyatakan bahwa P(X∩Y) = P(Y∩X), maka dapat diperoleh persamaan sebagai berikut :

$$P(X \cap Y) = P(X|Y)xP(Y)$$

dan

$$P(Y \cap X) = P(Y|X)xP(X)$$

Dengan demikian diperoleh :

$$P(X \cap Y) = P(Y \cap X)$$
$$P(X|Y)xP(Y) = P(Y|X)xP(X)$$

$$P(X|Y) = \frac{P(Y|X)xP(X)}{P(Y)}$$

atau

$$P(Y|X) = \frac{P(X|Y)xP(Y)}{P(X)}$$

Inilah yang menjadi dasar dari persamaan teorma bayess.

9.3. Persamaan Teorema Bayess

Bedasarkan hasil dekomposisi persamaan probabilitas bersyarat, persamaan teorema Bayess dapat dinyatakan sebagai berikut :

$$P(X|Y) = \frac{P(Y|X)xP(X)}{P(Y)}$$

Dimana Y merupakan data dengan class yang belum diketahui, X merupakan hipotesis data atau suatu class yang spesifik, P(X|Y) merupakan probabilitas hipotesis berdasarkan kondisi tertentu (posteriori probability), P(X) merupakan probabilitas hipotesis (prior probability), P(Y|X) merupakan probabilitas berdasarkan kondisi pada hipotesis dan P(Y) merupakan probabilitas dari X.

Persamaan teorema Bayess di atas memperlihatkan bahwa peluang sebuah sample menjadi bagian dari class X (*posterior*) merupakan peluang munculnya kelas X sebelum munculnya sampel (prior) dikalikan dengan peluang munculnya karakteristik sample pada class X (*likelihood*) dibagi dengan kemunculan karakteristik sample secara global (*evidence*). Penjabaran lebih lanjut terhadap persamaan Bayess dilakukan dengan menjabarkan kondisi data Y dengan jumlah sample tertentu dan belum memiliki class X, sehingga dapat dijabarkan sebagai berikut :

$$P(X_i|Y_1, Y_2, \dots Y_n) = \frac{P(Y_1, Y_2, \dots Y_n|X_1)xP(X_i)}{\sum_{k=1}^{n} P(Y_1, Y_2, \dots Y_k|X_k)xP(X_k)}$$

Dimana X_i merupakan class spesifik ke-i dan Y_n atau Y_k merupakan sejumlah data yang belum memiliki class.

9.4. Contoh Kasus Klasifikasi dengan Naive Bayess

Diberikan sejumlah data keputusan seseorang melakuan pembelian komputer seperti diperlihatkan pada Tabel 9.1.

Tabel 9.1 Data Keputusan Pembelian Komputer

age	income	student	credit_rating	buys_computer
≤ 30	High	No	Fair	No
≤ 30	High	No	Excellent	No
31...40	High	No	Fair	Yes
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31...40	Low	Yes	Excellent	Yes
≤ 30	Medium	No	Fair	No
≤ 30	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
≤ 30	Medium	Yes	Exellent	Yes
31...40	Medium	No	Exellent	Yes
31...40	High	Yes	Fair	Yes
>40	Medium	No	Ecellent	No

Jika diketahui **Data TestingP = (age** ≤ 30, **income** = medium, **student** = yes, **credit_rating** = fair) maka tentukan apakah calon pelanggan tersebut akan membeli komputer atau tidak.

Langkah pertama adalah menghitung nilai probabilitas untuk semua class target (buys_computer) dengan persamaan $P\left(buys_{computer} = \frac{Yes}{No}\right) = \frac{n\left(\frac{Yes}{No}\right)}{N}$ dimana *n* merupakan jumlah data sample untuk class *Yes* atau *No* yang akan dihitung nilai probabilitasnya, sedangkan *N* merupakan jumlah data sample keseluruhan. Dengan demikian diperoleh nilai probabilitas untuk masing-masing class target sebagai berikut :

P(buys_computer = Yes) = 9/14 = 0.64
P(buys_computer = No) = 5/14 = 0.36

Langkah kedua adalah menghitung nilai probabilitas untuk setiap variabel pada **Data Testing** terhadap class target. Tahap ini menghasilkan nilai-nilai probabilitas berupa P(age ≤ 30 | buys_computer = Yes), P(age ≤ 30 | buys_computer = No), P(income=medium | buys_computer = Yes), P(income = medium | buys_computer = No), P(student = Yes | buys_computer = Yes), P(student = Yes | buys_computer = No), P(credit_rating = Fair | buys_computer = Yes) dan P(credit_rating = Fair | buys_computer = No). Hasil perhitungan nilai probabilitas setiap variabel **Data Testing** diperoleh sebagai berikut :

P(age ≤ 30 | buys_computer = Yes) = 2/9 = 0.22
P(age ≤ 30 | buys_computer = No) = 3/5 = 0.6
P(income=medium | buys_computer = Yes) = 4/9 = 0.44
P(income = medium | buys_computer = No) = 2/5 = 0.4
P(student = Yes | buys_computer = Yes) = 6/9 = 0.67
P(student = Yes | buys_computer = No) = 1/5 = 0.2
P(credit_rating = Fair | buys_computer = Yes) = 6/9 = 0.67
P(credit_rating = Fair | buys_computer = No) = 2/5 = 0.5

Langkah ketiga, menghitung nilai probabilitas **Data Testing** terhadap setiap class target. Berikut hasil perhitungan nilai probabilitas untuk setiap class target :

Probabilitas untuk class target buys_computer = Yes :

Dari persamaan :

$$P(X_i|Y_1, Y_2, \dots Y_n) = \frac{P(Y_1, Y_2, \dots Y_n|X_1)xP(X_i)}{\sum_{k=1}^{n} P(Y_1, Y_2, \dots Y_k|X_k)xP(X_k)}$$

Diperoleh :

$$P\left(buys_{computer} = Yes \middle| age \leq 30, income = medium, student = Yes, credit_{rating} =\right.$$

$$= \frac{P\left(age \leq 30, income\right.}{\left(P\left(age \leq 30, income = medium, student = Yes, credit_{rating} = Fair \middle| buys_{comput}\right.\right.}$$

$$P\left(buys_{computer} = Yes \middle| age \leq 30, income = medium, student = Yes, credit_{rating} =\right.$$

$$= \frac{0.22\ x\ 0.44\ x\ 0.67\ x\ 0.67\ x\ 0.64}{(0.22\ x\ 0.44\ x\ 0.67\ x\ 0.67\ x\ 0.64) + (0.6\ x\ 0.4\ x\ 0.2\ x\ 0.5\ x\ 0.36)}$$

$$P\left(buys_{computer} = Yes \middle| age \leq 30, income = medium, student = Yes, credit_{rating} =\right.$$

$$= \frac{0.0278}{(0.0278) + (0.00864)} = \frac{0.0278}{0.0364} = 0.763$$

Probabilitas untuk class target buys_computer = No :

Dari persamaan :

$$P(X_i|Y_1, Y_2, \dots Y_n) = \frac{P(Y_1, Y_2, \dots Y_n|X_1)xP(X_i)}{\sum_{k=1}^{n} P(Y_1, Y_2, \dots Y_k|X_k)xP(X_k)}$$

Diperoleh :

$$P\left(buys_{computer} = No \middle| age \leq 30, income = medium, student = Yes, credit_{rating} =\right.$$

$$= \frac{P\left(age \leq 30, income\right.}{\left(P\left(age \leq 30, income = medium, student = Yes, credit_{rating} = Fair \middle| buys_{comput}\right.\right.}$$

$$P\left(buys_{computer} = No \middle| age \leq 30, income = medium, student = Yes, credit_{rating} =\right.$$

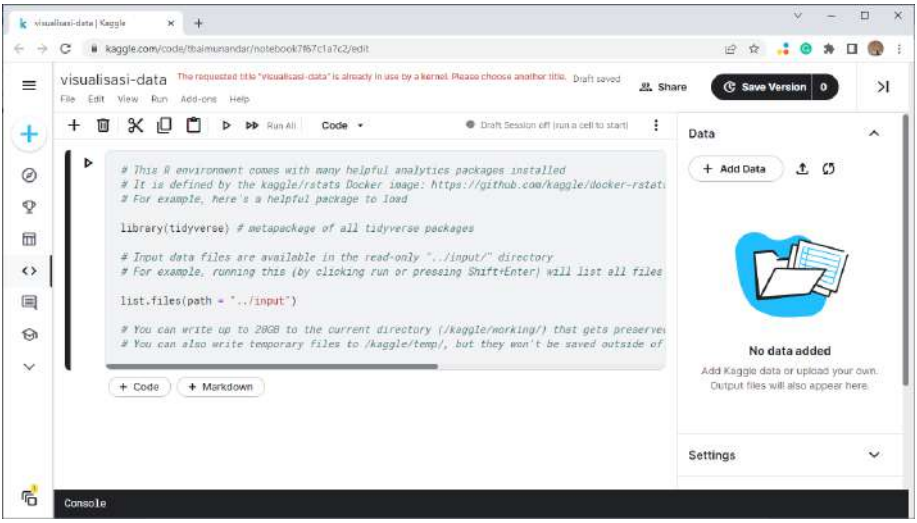
$$= \frac{0.6\ x\ 0.4\ x\ 0.2\ x\ 0.5\ x\ 0.36}{(0.6\ x\ 0.4\ x\ 0.2\ x\ 0.5\ x\ 0.36) +\ (0.22\ x\ 0.44\ x\ 0.67\ x\ 0.67\ x\ 0.64)}$$

$$P(buys_{computer} = No | age \leq 30, income = medium, student = Yes, credit_{rating}$$
$$= \frac{0.00864}{(0.00864) + (0.0278)} = \frac{0.00864}{0.0364} = 0.237$$

Dengan demikian, berdasarkan hasil perhitungan menggunakan Naive Bayess, dapat diketahui bahwa **Data TestingP = (age** $\leq$ 30, **income** = medium, **student** = yes, **credit_rating** = fair) masuk ke dalam class target **buys_computer = Yes**. Hal ini dilihat dari nilai probabilitas untuk class target **buys_computer = Yes** lebih tinggi dibandingkan dengan nilai probabilitas untuk class target **buys_computer = No.**

9.5. Algoritme Naïve Bayes Menggunakan R

Untuk memulai visualisasi data menggunakan Bahasa R, langkah pertama adalah login ke dalam kaggle.com menggunakan akun masing-masing. Kemudian pada manu navigator bagian kiri pilih menu < > **Code** kemudian klik tombol **New Notebook** sampai terbuka halaman seperti pada Gambar di bawah. Untuk lebih jelas bagaimana membuat akun dan membuka notebook baru silahkan pelajari Bab 5.



Menyiapkan dataset

Langkah selanjutnya adalah menyiapkan dataset menggunakan perintah "**read.csv**" pastikan data sudah diupload ke dalam kaggle dengan format *.csv. Data studi kasus yang digunakan pada pembahasan buku ini adalah **dataset-pinjaman-nasabah.csv.** Pada fungsi **read.csv()** untuk ditambahkan parameter **na.string(c=(“ ”))** untuk mengatasi value variable yang bernilai kosong (empty). Dengan menambahkan parameter tersebut, setiap value variable yang memiliki nilai kosong akan diisi dengan value = NA. Pengisian value variable yang kosong ini dimaksudkan untuk mempermudah proses analisis data terutama pada tahap data cleaning. Pada tahap data cleaning dilakukan penghapusan terhadap data instance yang memiliki niilai NA, dengan demikian dataset yang digunakan sudah betul-betul siap pakai. Perintah head digunakan untuk menampilkan 6 buah data pertama. Berikut baris kode yang digunakan :

```
data=read.csv('../input/pinjamankreditnasabah/dataset-pinjaman-nasabah.csv',
              sep=',', na.string=c(""))
head(data)
```

A data.frame: 6 × 13

	ID_Nasabah	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman
	<chr>	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>
1	LP001002	Male	No	0	Graduate	No	5849	0	NA
2	LP001003	Male	Yes	1	Graduate	No	4583	1508	128
3	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66
4	LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120
5	LP001008	Male	No	0	Graduate	No	6000	0	141
6	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267

+ Code + Markdown

Proses Data Cleaning

Langkah berikutnya adalah melakukan pembersihan data terhadap value variabel yang memiliki nilai NA. Pembersihan data menggunakan fungsi **na.omit()** kemudian datasets setelah pembersihan disimpan dalam variabel **dataClean**. Berikut proses pembersihan dataset pada R.

```
dataClean=na.omit(data)
head(dataClean)
```

A data.frame: 6 × 13

	ID_Nasabah	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman
	<chr>	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>
2	LP001003	Male	Yes	1	Graduate	No	4583	1508	128
3	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66
4	LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120
5	LP001008	Male	No	0	Graduate	No	6000	0	141
6	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267
7	LP001013	Male	Yes	0	Not Graduate	No	2333	1516	95

+ Code + Markdown

Untuk tahap pembersihan data selanjutnya adalah dengan menghilangkan variable **ID_Nasabah**. Variable ini dihilangkan karena memang tidak memiliki pengaruh signifikan terhadap proses pembelajaran algoritma naïve bayes. Hasil pembersihan data bagian kedua disimpan dalam variable **dataClean2** sebagai berikut :

```
dataClean2=dataClean[,2:13]
head(dataClean2)
```

A data.frame: 6 × 12

	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman	JangkaWaktul
	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>	
2	Male	Yes	1	Graduate	No	4583	1508	128	
3	Male	Yes	0	Graduate	Yes	3000	0	66	
4	Male	Yes	0	Not Graduate	No	2583	2358	120	
5	Male	No	0	Graduate	No	6000	0	141	
6	Male	Yes	2	Graduate	Yes	5417	4196	267	
7	Male	Yes	0	Not Graduate	No	2333	1516	95	

+ Code + Markdown

Pembagian Dataset menjadi Data Training dan Testing

Tahap berikutnya adalah membagi dataset ke dalam data training dan data testing. Proporsi untuk data training sebanyak 80% sedangkan untuk data testing sebanyak 20%. Fungsi **set.seed()** digunakan untuk memanggil sejumlah data secara random yang akan dialokasikan untuk data training dan testing. Fungsi **sample()** digunakan untuk membuat sample yang diambil dari variabel **dataClean2** sebanyak 80%. Sedangkan variabel **dataTrain** digunakan untuk menampung sejumlah data untuk pelatihan algoritma yang diambil dari variabel **sampleTrain**, sedangkan variabel **dataTesting** digunakan untuk menampung sisa dari data pada **dataClean2** setelah dikurangi dengan 80% alokasi ke data training (**-sampleTrain**).

Perlu diperhatikan bahwa proporsi untuk data training dan testing memang tidak ada ketentuan baku. Namun yang pasti, jumlah data training harus lebih banyak dibandingkan data testing. Hal ini

dikarenakan data testing dipakai untuk melatih algoritma sehingga lebih banyak data yang dipelajari oleh algoritma maka semakin banyak yang bisa dikenali oleh algoritma ketika berhadapa dengan dataset baru. Selain proporsi dataset, proporsi dari label / kelas dari variable target juga mempengaruhi model yang terbentuk dari algoritma. Kemampuan mengenali kelas tertentu ditentukan imbang tidaknya kelas yang dimiliki. Berikut proses pembagian datasets ke dalam data training dan data testing.

```
set.seed(1234)
sampleTrain=sample(nrow(dataClean2), nrow(dataClean2)*0.8)
dataTrain = dataClean2[sampleTrain,]
dataTesting = dataClean2[-sampleTrain,]
```

Gunakan perintah **head**() untuk mengecek 6 data pertama baik untuk data training maupun data testing.

```
head(dataTrain)
```

A data.frame: 6 × 12

	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman
	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>
366	Male	No	0	Not Graduate	No	6216	0	133
429	Male	Yes	0	Graduate	No	2920	16	87
520	Female	No	0	Not Graduate	No	3400	0	95
136	Male	Yes	3	Graduate	No	4000	7750	290
146	Female	Yes	0	Graduate	No	2330	4486	100
503	Male	Yes	2	Graduate	No	4865	5624	208

```
head(dataTesting)
```

A data.frame: 6 × 12

	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	JumlahPinjaman
	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>
2	Male	Yes	1	Graduate	No	4583	1508	128
6	Male	Yes	2	Graduate	Yes	5417	4196	267
8	Male	Yes	3	Graduate	No	3036	2504	158
9	Male	Yes	2	Graduate	No	4006	1526	168
10	Male	Yes	1	Graduate	No	12841	10968	349
13	Male	Yes	2	Graduate	No	3073	8106	200

Membangun Model Klasifikasi Dengan Algoritma Naïve Bayes

Langkah berikutnya adalah membangun model klasifikasi menggunakan algoritma Naive Bayes dengan memanfaatkan fungsi **naiveBayes()** yang ada pada **library(e1071)**. Library ini juga memuat metode lain seperti Fourier Transform, Fuzzy Clustering, Support Vector Machine, Shortest Path Computation, Bagged Clustering, Naive Bayes dan Generalized k-Nearest Neighbour. Berikut adalah baris kode untuk membangun model klasifikasi dengan naïve Bayes.

```
library(e1071)
modelBayes=naiveBayes(StatusPinjaman~., dataTrain, laplace=1)
modelBayes
```

Output dari perintah di atas :

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        N         Y
0.3072917 0.6927083

Conditional probabilities:
   JenisKelamin
Y       Female      Male
  N 0.2118644 0.8050847
  Y 0.1691729 0.8383459

   StatusPernikahan
Y         No       Yes
  N 0.4237288 0.5932203
  Y 0.3270677 0.6804511

   JumTanggungan
Y       [,1]    [,2]
  N 0.7203390 1.020186
  Y 0.7857143 1.025681

   Pendidikan
Y    Graduate Not Graduate
```

```
  N 0.7796610    0.2372881
  Y 0.8082707    0.1992481

   Wiraswasta
Y         No       Yes
  N 0.8728814 0.1440678
  Y 0.8759398 0.1315789

   IncomeNasabah
Y       [,1]     [,2]
  N 5653.059 7935.674
  Y 5269.492 4898.148

   IncomePasangan
Y       [,1]     [,2]
  N 1868.568 3943.104
  Y 1414.955 2005.318

   JumlahPinjaman
Y       [,1]     [,2]
  N 154.3136 93.76099
  Y 139.0263 76.15854

   JangkaWaktuPinjaman
Y       [,1]     [,2]
  N 340.4746 78.05003
  Y 341.5940 62.67196

   Credit_History
Y       [,1]     [,2]
  N 0.5508475 0.4995290
  Y 0.9887218 0.1057973

   WilayahTempatTinggal
Y       Rural Semiurban     Urban
  N 0.3813559 0.2542373 0.3898305
  Y 0.2556391 0.4812030 0.2744361
```

Evaluasi Model

Langkah berikutnya adalah mengevaluasi model yang sudah dibangun menggunakan data testing. Gunakan fungsi **predict()** untuk mengevaluasi model, kemudian tampilkan confusion matrix serta informasi akurasinya. Berikut adalah perintah untuk melakukan evaluasi model naive bayes.

```
evalNB=predict(modelBayes, dataTesting, type="class")
confusionMatrix(table(evalNB, dataTesting$StatusPinjaman))
```

```
Confusion Matrix and Statistics


evalNB  N  Y
     N 12  7
     Y 18 59

               Accuracy : 0.7396
                 95% CI : (0.64, 0.8238)
    No Information Rate : 0.6875
    P-Value [Acc > NIR] : 0.1609

                  Kappa : 0.3266

 Mcnemar's Test P-Value : 0.0455

            Sensitivity : 0.4000
            Specificity : 0.8939
         Pos Pred Value : 0.6316
         Neg Pred Value : 0.7662
             Prevalence : 0.3125
         Detection Rate : 0.1250
   Detection Prevalence : 0.1979
      Balanced Accuracy : 0.6470

       'Positive' Class : N
```

Hasil evaluasi model terhadap data testing diperoleh akurasi sebesar 73.96%. Sedangkan tabel confusion matrix memperlihatkan jumlah True Negatif sebanyak 12, True Positif sebanyak 59, False Positif sebanyak 7 dan False Negatif sebanyak 18. Nilai-nilai ini dapat digunakan untuk menghitung Recall dan Precision sehingga dapat mendukung tingkat akurasi yang dihasilkan model.

Penggunaan Model Naïve Bayes untuk Prediksi Kelas Baru

Panggil datasets baru yang belum memiliki kelas, pada kasus pembahasan buku ini, digunakan dataset Data-Baru-Calon-Debitur.csv. Data calon debitur terdiri atas dua belas orang dengan variable yang sama dengan datasets saat membangun model. Yang

membedakan adalah datasets baru ini belum memiliki label / kelas. Berikut pemanggilan datasets calon debitur baru menggunakan fungsi **read.csv().**

```
dataBaru=read.csv('../input/datauntukprediksi/Data-Baru-Calon-Debitur.csv', sep=',')
head(dataBaru)
```

A data.frame: 6 × 13

	ID_Nasabah	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasa
	<chr>	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<
1	LP003002	Male	No	0	Not Graduate	No	3744	
2	LP003003	Male	Yes	1	Not Graduate	No	3653	
3	LP003004	Female	Yes	0	Graduate	Yes	4908	
4	LP003005	Female	Yes	0	Not Graduate	No	5802	
5	LP003006	Male	No	0	Not Graduate	No	4596	
6	LP003007	Male	Yes	2	Graduate	Yes	5161	

Hilangkan variable **ID_Nasabah** kemudian simpan datasets ke dalam variable baru dengan nama **dataPrediksi**. Berikut baris kodenya.

```
dataPrediksi=dataBaru[,2:13]
head(dataPrediksi)
```

A data.frame: 6 × 12

	JenisKelamin	StatusPernikahan	JumTanggungan	Pendidikan	Wiraswasta	IncomeNasabah	IncomePasangan	Jumlah
	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	
1	Male	No	0	Not Graduate	No	3744	0	
2	Male	Yes	1	Not Graduate	No	3653	200	
3	Female	Yes	0	Graduate	Yes	4908	3500	
4	Female	Yes	0	Not Graduate	No	5802	2600	
5	Male	No	0	Not Graduate	No	4596	0	
6	Male	Yes	2	Graduate	Yes	5161	1200	

Gunakan variabel **dataPrediksi** untuk melakukan prediksi kelas yang ada di dalam nya menggunakan **modelBayes**. Dalam hal ini fungsi **predict()** digunakan untuk menjalankan proses prediksi data baru berdasarkan model yang ada. Hasil prediksi disimpan pada variabel **hasilPrediksi**. Untuk parameter **type**, pilih **'class'** untuk menampilkan hasil prediksi dalam bentuk kelas. Berikut baris kodenya.

```
hasilPrediksi=predict(modelBayes,dataPrediksi, type='class')
data.frame(hasilPrediksi)
```

A data.frame: 12 × 1

hasilPrediksi <fct>
N
N
Y
Y
N
N
Y
Y
Y
N
Y
N

Selain menyajikan hasil prediksi dalam **type = ‘class’**, kita juga dapat menyajikan hasil prediksi dalam bentuk nilai probabilitas, yaitu dengan menggunakan **type ='raw’**. Dengan menggunakan **type = ‘raw’** maka akan disajikan data frame sebanyak jumlah kelas yang tersedia disertai dengan nilai probabilitas setiap kelas. Pada pembahasan buku ini, karena kelas **StatusPinjaman** hanya terdiri atas Y dan N, maka yang disajikan hanya dua kolom yang merepresentasikan kelas variable **StatusPinjaman** beserta nilai probabilitas yang dimilikinya. Berikut baris kode untuk menyajikan

hasil prediksi ke dalam **tipe = 'raw'** yang disimpan pada variable **hasilPrediksi2.**

```
hasilPrediksi2=predict(modelBayes,dataPrediksi, type='raw')
data.frame(hasilPrediksi2)
```

A data.frame: 12 × 2

N	Y
<dbl>	<dbl>
1.000000e+00	3.070591e-18
1.000000e+00	3.139016e-18
1.331670e-01	8.668330e-01
2.140218e-02	9.785978e-01
1.000000e+00	1.557427e-124
1.000000e+00	4.903272e-14
3.587688e-02	9.641231e-01
1.759332e-02	9.824067e-01
3.509074e-258	1.000000e+00
1.000000e+00	8.464941e-108
1.599995e-01	8.400005e-01
1.000000e+00	1.028255e-45

Untuk membaca tabel pada hasil di atas, bisa dilihat berdasarkan nilai probabilitas tertinggi. Jika nilai probabilitas tertinggi berada pada Kolom N, maka sudah dipastikan calon debitur di tolak pengajuannya. Begitu juga sebaliknya.

BAB X
ALGORITME PARTITIONING ARROUND MEDOIDS

10.1. Konsep Partitioning Around Medoids / K-Medoid

K-Medoids merupakan teknik klaster yang menggunakan objek sebagai perwakilan (medoid) bagi titik pusat klaster. K-Medoids mirip dengan K-means, hanya saja pada K-means penentuan pusat klaster dilakukan dengan menghitung nilai rata-rata suatu objek (*mean*) sehingga dapat ditentukan pusat klasternya, sedangkan pada K-medoids tidak. Algoritma K-medoid adalah sebagai berikut (Pramesti dkk, 2017) :

1. Inisialisasi pusat klaster sebanyak k (jumlah klaster)
2. Alokasikan setiap data (objek) ke klaster terdekat menggunakan persamaan ukuran jarak *Euclidian Distance* dengan persamaan :

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

3. Pilih secara acak objek pada masing-masing klaster sebagai kandidat medoid baru
4. Hitung jarak setiap objek yang berada pada masing-masing klaster dengan kandidat medoid baru
5. Hitung total simpangan (S) dengan menghitung nilai total distance baru - total distance lama. Jika S < 0, maka tukar objek dengan data klaster untuk membentuk sekumpulan k objek baru sebagai medoid.
6. Ulangi langkah 3 sampai 5 hingga tidak terjadi perubahan medoid, sehingga didapatkan klaster beserta anggota klaster.

10.2. Studi Kasus

Diberikan data dummy sebagai berikut :

X_1	2	2
X_2	6	4
X_3	4	5
X_4	8	2
X_5	4	6
X_6	3	3

Berdasarkan data pada tabel di atas, kelompokan ke dalam dua buah klaster menggunakan k-medoid dengan k=2.

Penyelesaian :

Langkah 1 : Tentukan pusat klaster awal sebanyak k=2, ditentukan secara random atau pemilihan. Dalam hal ini ditentukan pusat klaster 1 (c1 = {2,2} diambil dari X_1) dan klaster 2 (c2 = {4,6} diambil dari X_5).

Langkah 2 : Hitung jarak setiap objek data terhadap pusat klaster terpilih. Perhitungan jarak dilakukan dengan persamaan Euclidian Distance sebagai berikut :

Data objek		**Jarak Data ke**		**Klaster**
i	**Xi**	**c1 = {2,2}**	**c2 = {4,6}**	
1	(2,2)	0	4.5	
2	(6,4)			
3	(4,5)			
4	(8,2)			
5	(4,6)			
6	(3,3)			

Berikut perhitungan jarak objek terhadap pusat klaster untuk data X_1 :

X_1 ke C1 :

$$d(X1, C1) = \sqrt{(2-2)^2 + (2-2)^2} = 0$$

X_1 ke C2 :

$$d(X1, C2) = \sqrt{(2-4)^2 + (2-6)^2}$$

$$d(X1, C2) = \sqrt{4 + 16} = 4.47$$

Lakukan perhitungan jarak untuk objek yang lain dengan cara yang sama sehingga diperoleh jarak antar objek keseluruhan sebagai berikut :

Data objek		**Jarak Data ke**		**Klaster**
i	**Xi**	**c1 = {2,2}**	**c2 = {4,6}**	
1	(2,2)	**0.0**	4.5	1
2	(6,4)	4.4	**2.8**	2
3	(4,5)	3.6	**1.0**	2
4	(8,2)	6.0	**5.6**	2
5	(4,6)	4.4	**0.0**	2
6	(3,3)	**1.4**	3.1	1
Cost		**1.4**	**9.4**	

Langkah selanjutnya adalah menentukan klaster untuk masing-masing objek data. Caranya adalah dengan melihat jarak terpendek yang dimiliki objek data terhadap pusat klaster masing-masing. Pada tabel di atas terlihat bahwa dari enam objek data ada empat objek data yang menjadi anggota klaster 2 dan dua objek data yang menjadi anggota klaster 1.

Kalster 1 = {(2,2); (3,3)}
Klaster 2 = {(6,4); (4,5); (8,2); (4,6)}

Selanjutnya adalah menghitung total *cost* yang dimiliki setiap klaster. Caranya adalah dengan menjumlahkan data jarak keseluruhan dari setiap objek data yang ada pada masing-masing klaster. Berdasarkan hasil klaster awal dapat diketahui total *cost* untuk masing-masing klaster sebagai berikut :

$$Klaster\ 1 = \underbrace{0.0 + 1.4 = 1.4}$$
$$Klaster\ 2 = \underbrace{2.8 + 1.0 + 5.6 + 0.0 = 9.4}$$

Total *cost* klaster adalah penjumlahan dari *cost Klaster 1 + cost Klaster 2* sehingga dengan demikian total *cost* untuk klaster tahap pertama ini sebesar 10.8. Selanjutnya lakukan pengklasteran ulang untuk mengetahui total *cost* dari proses klaster kedua, kemudian dibandingkan. Jika total *cost* akhir lebih besar dari total *cost* awal, maka pengelompokan selesai. Sebaliknya jika total *cost* akhir lebih kecil dari total *cost* awal, maka pengklasteran terus dilakukan sampai memenuhi syarat *current total cost > previous total cost.*

Selanjutnya, tentukan pusat klaster baru (medoid) untuk perhitungan jarak objek data yang baru. Caranya adalah dengan melihat total *cost* akhir dari setiap klaster. Ambil total *cost* dengan nilai terkecil. Pada studi kasus ini, nilai *cost* terkecil dimiliki oleh klaster 1. Selanjutnya tentukan pusat klaster baru untuk C1, dalam hal ini misal kita pilih X_3 (4,5) sebagai pusat klaster baru untuk C1. Kemudian lakukan perhitungan jarak kembali untuk data hasil klaster pertama. Hasil perhitungan jarak klaster kedua diperoleh sebagai berikut :

Data objek		**Jarak Data ke**		**Klaster**
i	**Xi**	**c1 = {4,5}**	**c2 = {4,6}**	
1	(2,2)	**3,6**	4,5	1
2	(6,4)	**2,2**	2,8	1
3	(4,5)	**0,0**	1,0	1
4	(8,2)	**5,0**	5,7	1
5	(4,6)	1,0	**0,0**	2
6	(3,3)	**2,2**	3,2	1
Cost		**13.1**	**0.0**	

Berdasarkan hasil klaster awal dapat diketahui total *cost* untuk masing-masing klaster sebagai berikut :

$$Klaster\ 1 = \underbrace{3.6 + 2.2 + 0.0 + 2.2 = 13.1}$$

$$Klaster\ 2 = \underbrace{0.0}$$

Total *cost* klaster adalah penjumlahan dari *cost Klaster 1* + *cost Klaster 2* sehingga dengan demikian total *cost* untuk klaster tahap kedua ini sebesar 13.1.

Pada tahap klaster sebelumnya diketahui total *cost* yang dihabiskan adalah 10.8, sedangkan pada tahap kedua, total *cost* yang dihabiskan sebesar 13.1, dengan demikian hasil klaster optimal ditunjukan oleh tahap klaster awal dengan nilai centroid {2,2} dan {4,6}. Dengan demikian klaster yang terbentuk adalah klaster dengan total *cost* paling kecil yaitu dari hasil klaster tahap pertama :

Klaster 1 = {(2,2); (3,3)}

Klaster 2 = {(6,4); (4,5); (8,2); (4,6)}

10.3. Algoritme PAM Menggunakan R

Untuk dapat melakukan analisis klaster menggunakan *partitioning around medoid* (PAM), hal yang pertama adalah siapkan datasets yang akan dikelompokan. Pada pembahasan buku ini, untuk teknik pengelompokan data (Bab 9 sampai 11) akan digunakan data Pendapatan Domestik Regional Bruto (PDRB) 41 Kabupaten di Pulau Jawa Bagian Barat. Diambil dari 3 Provinsi yakni Jawa Barat, DKI Jakarta dan Provinsi Banten. Sedangkan untuk data diambil berdasarkan data PDRB tahun 2010 – 2021. Dataset disimpan dalam bentuk *.csv dengan nama **pdrbgabungan.csv.** Hal yang perlu diperhatikan saat melakukan analisis klaster adalah, pastikan bahwa data yang dimiliki bertipe numerik. Berikut pemanggilan data PDRB dalam bahasa R.

```
data=read.csv('../input/pdrbjawabagianbarat/pdrb-gabungan.csv', sep=',')
head(data)
```

A data.frame: 6 × 13

	Nama.Wilayah	Year.2010	Year.2011	Year.2012	Year.2013	Year.2014	Year.2015	Year.2016	Year.2017	Year.2018	Year.2019	Year.2020	Year.2021
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Kabupaten Lebak	12572538	13325629	14006209	14887984	15756247	16733238	17665397	18683740	19735870	20810490	20610990	21245040
2	Kabupaten Pandeglang	12279542	12964402	13738882	14387883	15097105	15974129	16855618	17866428	18812932	19644125	19541488	20127757
3	Kabupaten Serang	33641000	35905370	37849640	40246690	42541180	44454580	46715180	49154636	51754320	54347488	53055563	54992522
4	Kabupaten Tangerang	18549119	62022491	65848281	70065983	74101232	78093560	82183596	86964027	92011405	97142198	93544934	97809902
5	Kota Cilegon	44676529	47633318	51300206	54732934	57261923	59982732	62981047	66444529	70502082	74228641	73534471	77071368
6	Kota Serang	12549572	13595691	14604637	15670784	16745084	17808478	18935486	20153023	21482093	22813096	22517969	23374085

Baris kode di atas merupakan perintah untuk memanggil dataset yang disimpan pada variable **data,** kemudian ditampilkan hasilnya sebanyak enam data instance pertama menggunakan perintah **head().** Langkah berikutnya adalah melakukan pre-processing data dengan menghilangkan data bernilai N/A (jika ada). Gunakan perintah **na.omit()** untuk menghilangkan data instance yang memiliki value N/A.

Jika pre-processing dilakukan secara manual, misalnya diperiksa menggunakan aplikasi lain, maka perintah **na.omit()** tidak diperlukan. Artinya bahwa data yang dipanggil dan disimpan dalam platform Kaggle adalah data yang sudah bersih dan tidak memuat empty attribute value, NA value, redudance dan sebagainya. Dengan demikian data sudah siap pakai untuk kebutuhan analisi klaster. Meskipun demikian, Bahasa R tetap menyediakan fungsi pre-processing untuk menghilangkan data yang memiliki value N/A tersebut. Apalagi jika data nya dalam jumlah yang sangat besar. Berikut perintah untuk removal data instance dengan N/A value kemudian diperiksa struktur dan jumlah observasi data setelah dilakukan **na.omit()** menggunakan **str()** sebagai berikut :

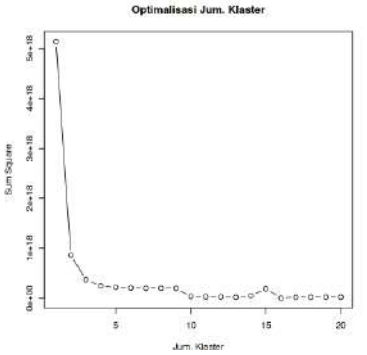
```
dataClean=na.omit(data)
str(dataClean)
```

```
'data.frame':   41 obs. of  13 variables:
 $ Nama.Wilayah: chr  "Kabupaten Lebak" "Kabupaten Pandeglang" "Kabupaten Serang" "Kabupaten Tangerang" ...
 $ Year.2010   : num  12572538 12279542 33841000 18549119 44676529 ...
 $ Year.2011   : num  13325629 12984402 35905370 62022491 47633318 ...
 $ Year.2012   : num  14006209 13738882 37849640 65848281 51300206 ...
 $ Year.2013   : num  14887984 14387883 40246690 70065983 54732934 ...
 $ Year.2014   : num  15756247 15097105 42541180 74101232 57261923 ...
 $ Year.2015   : num  16733238 15974129 44454580 78093560 59982732 ...
 $ Year.2016   : num  17665397 16855618 46715180 82183596 62981047 ...
 $ Year.2017   : num  18683740 17866428 49154636 86964027 66444529 ...
 $ Year.2018   : num  19735870 18812932 51754320 92011405 70502082 ...
 $ Year.2019   : num  20810490 19644125 54347488 97142198 74228641 ...
 $ Year.2020   : num  20610990 19541488 53055563 93544934 73534471 ...
 $ Year.2021   : num  21245040 20127757 54992522 97809902 77071368 ...
```

Berdasarkan hasil pembersihan data, jumlah data masih seperti apa adanya yakni 41 data instance dengan 13 variabel. Ini berarti dataset yang digunakan sudah siap digunakan. Untuk dapat menggunakan fungsi partitioning around medoid, pada Bahasa R dibutuhkan **library(cluster)** yang di dalamnya memuat fungsi **pam().** Berikut langkah-langkah analisis klaster dengan PAM.

1. Untuk melakukan klasterisasi khususnya menggunakan PAM dan k-means, hal yang paling krusial adalah menentukan jumlah klaster optimal. Banyak dari analis data yang sering kesulitan dalam menentukan jumlah klaster optimal saat akan mengelompokan data tertentu. Pada bahasa R, penentuan jumlah klaster optimal dapat dilakukan dengan menghitung nilai Within Sum Square (WWS) sebagai berikut :

```
D1 =(nrow(dataClean[,2:13])-1)*sum(apply(dataClean[,2:13],2,var))
for (i in 2:20) D1[i]<-sum(kmeans(dataClean[,2:13],centers=i)$withinss)
plot(1:20, D1, type = "b", main = "Optimalisasi Jum. Klaster", xlab = "Jum. Klaster", ylab = "Sum Square")
```



Seperti diperlihatkan pada Grafik di atas, terjadi penurunan sangat drastis dari nilai within sum square saat jumlah klaster meningkat dari 1 klaster menjadi 4 atau 5. Dengan demikian, pada turunan klaster k = 4 memperlihatkan kestabilan nilai WWS nya. Jumlah klaster dalam hal ini dapat dinyatakan dengan melihat keseimbangan antara k dan WWS. Dengan demikian, pada kasus ini jumlah klaster yang akan digunakan adalah k = 4.

2. Setelah jumlah klaster diketahui, langkah berikutnya adalah membangun klaster dengan fungsi k-means menggunakan R Studio. Dalam hal ini fungsi pam yang sudah tersedia pada bahasa R digunakan sebagai berikut :

   Perintah dasar PAM pada R

   pam(x, k, diss = inherits(x, "dist"), metric = c("euclidean", "manhattan"))

   Keterangan :

   x : matrik data dalam bentuk numerik atau data yang dapat dikonversi ke dalam bentuk matrik.

   k : jumlah klaster, bisa disebut jumlah k atau jumlah

initial pusat klaster
metric : jenis perhitungan jarak, pilih salah satu

Berdasarkan data yang ada dan jumlah klaster (k) yang telah ditentukan, untuk mengelompokan data dengan PAM dilakukan dengan menuliskan perintah sebagai berikut :

```
library(cluster)
klasterPAM=pam(dataClean[,2:13],4, diss = inherits(dataClean[,2:13], "dist"),metric = c("euclidean"))
klasterPAM
```

```
Medoids:
   ID Year.2010 Year.2011 Year.2012 Year.2013 Year.2014 Year.2015 Year.2016
19 19  13718207  14435293  15215686  16028527  16839416  17779913  18844974
35 35  41283495  43946084  46907333  49741127  52534090  55456075  58831077
12 12 241225134 258049287 275317710 292545983 310185286 329155038 349251708
24 24  99641320 106174676 111424084 120294864 126748693 132453568 141125537
   Year.2017 Year.2018 Year.2019 Year.2020 Year.2021
19  19826748  20878689  22001240  21970410  22774934
35  62202006  65845090  69404620  67619240  69796930
12 371253514 394429958 421300649 419262189 429398830
24 148358445 157317842 163946848 157710593 166941492
Clustering vector:
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 1  1  2  2  2  1  2  2  1  3  3  3  3  3  2  1  4  4  1  1  1  1  2  4  1  1
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
 1  2  1  2  1  1  4  1  2  1  1  1  2  1  1
Objective function:
   build     swap
61770303 53099489

Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
 [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

Pada baris perintah di atas. library(cluster) digunakan untuk memanggil package cluster analisis yang di dalamnya memuat fungsi klaster PAM. Untuk baris kedua dengan nama variabel klasterPAM adalah perintah untuk melakukan analisis klaster menggunakan fungsi pam() dimana parameter data diambil dari variabel dataClean dengan mengambil variabel ke 2 sampai ke 13 karena bertipe numerik, sementara variabel ke 1 bertipe character (berupa nama wilayah), paramter perhitungan jarak dilakukan dengan pendekatan euclidean. Hasil klaster kemudian ditampilkan dengan memanggil variabel klasterPAM sehingga menampilkan informasi medoids, clustering vector, objective function dan available components. Perhatikan pada dua hasil pertama yaitu medoids dan clustering vector. Medoids

memperlihatkan nilai akhir medoids untuk semua variabel untuk sebanyak klaster yang dibentuk. Pada medoid ini juga diperoleh informasi data instance ke berapa yang dipilih secara random menjadi medoids sampai dengan iterasi terakhir. Sedangkan untuk komponen clustering vector, menyajikan informasi keanggotaan klaster setiap data instance. Baris pertama menunjukan nomor urut data instance, sementara baris kedua menunjukan klaster tempat data instance tersebut berada.

Kita juga bisa menyajikan informasi nama wilayah terklaster ke dalam klaster mana saja dengan memodifikasi variabel klasterPAM digabungkan dengan variabel Nama Wilayah pada dataset awal. Gunakan fungsi row.names() untuk memanggil data instance variabel Nama Wilayah kemudian gunakan perintah cbind() untuk menggabungkan keduanya. Berikut perintahnya :

```
wil=row.names=c(data[,1])
wil
```

'Kabupaten Lebak' · 'Kabupaten Pandeglang' · 'Kabupaten Serang' · 'Kabupaten Tangerang' · 'Kota Cilegon' · 'Kota Serang' · 'Kota Tangerang' · 'Kota Tangerang Selatan' · 'Kabupaten Kepulauan Seribu' · 'Kota Jakarta Barat' · 'Kota Jakarta Pusat' · 'Kota Jakarta Selatan' · 'Kota Jakarta Timur' · 'Kota Jakarta Utara' · 'Kabupaten Bandung ' · 'Kabupaten Bandung Barat ' · 'Kabupaten Bekasi' · 'Kabupaten Bogor ' · 'Kabupaten Ciamis' · 'Kabupaten Cianjur ' · 'Kabupaten Cirebon ' · 'Kabupaten Garut ' · 'Kabupaten Indramayu ' · 'Kabupaten Karawang ' · 'Kabupaten Kuningan ' · 'Kabupaten Majalengka' · 'Kabupaten Pangandaran' · 'Kabupaten Purwakarta' · 'Kabupaten Subang ' · 'Kabupaten Sukabumi' · 'Kabupaten Sumedang ' · 'Kabupaten Tasikmalaya ' · 'Kota Bandung ' · 'Kota Banjar ' · 'Kota Bekasi ' · 'Kota Bogor ' · 'Kota Cimahi ' · 'Kota Cirebon ' · 'Kota Depok ' · 'Kota Sukabumi' · 'Kota Tasikmalaya '

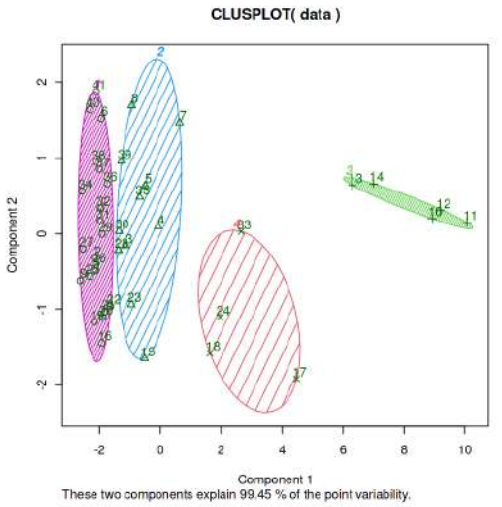
```
lengkap=cbind(wil, klasterPAM$cluster)
lengkap[1:10,]
```

A matrix: 10 × 2 of type chr

	wil	
1	Kabupaten Lebak	1
2	Kabupaten Pandeglang	1
3	Kabupaten Serang	2
4	Kabupaten Tangerang	2
5	Kota Cilegon	2
6	Kota Serang	1
7	Kota Tangerang	2
8	Kota Tangerang Selatan	2
9	Kabupaten Kepulauan Seribu	1
10	Kota Jakarta Barat	3

Hasil penggabungan terlihat informasi yang lebih representatif terhadap luaran klaster. Contoh : Kabupaten Lebak di kanannya terdapat informasi angka 1, artinya bahwa Kabupaten Lebak terkelompok ke dalam klaster 1 dan seterusnya. Untuk visualisasi hasil klaster kita juga dapat menyajikannya ke dalam bentuk grafik menggunakan fungsi clusplot() sebagai berikut :

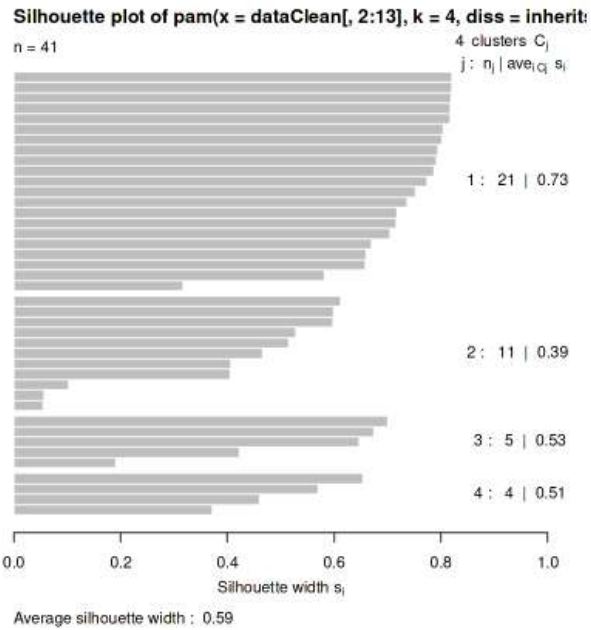
```
clusplot(data, klasterPAM$cluster, color=TRUE, shade=TRUE,labels=2, lines=0)
```



Baris perintah diatas memanggil fungsi clusplot dengan parameter data diambil data dataset awal saat dipanggil, dan parameter model klasterPAM yang dibuat pada tahap sebelumnya, sisanya adalah pengaturan parameter berupa warna,, arsisan, label dan tebal baris. Angka yang ditunjukan pada setiap plot klaster merupakan informasi nomor data instance.

Evaluasi klaster dapat dilakukan dengan melihat koefisien silhouette. Koefisien silhouette memiliki rentang nilai antara -1 sampai +1, semakin mendekati +1 data dikatakan bahwa objek data terklaster dengan baik, begitu juga sebaliknya. Luaran klaster PAM sebetulnya sudah secara otomatis menghitung koefisien silhouette yang dimaksud. Untuk mengetahui distribusi koefisien silhouette, kita dapat menggunakan fungsi **plot(silhouette())** sebagai berikut :

```
plot(silhouette(klasterPAM))
```



Hasil visualisasi koefisien silhouette memperlihatkan bahwa setiap objek data memiliki nilai silhouette beragam dan sebagian besar mendekati +1. Meskipun demikian ada beberapa objek data yang menjauhi +1 seperti yang terjadi pada klaster 2 ada tiga objek data yang mendekati 0 dan menjauhi +1.

BAB XI
ALGORITME K-MEANS

11.1. Teori Algoritme K-means

K-means merupakan metode pengelompokan data yang masuk ke dalam kategori unsupervised atau tidak terbimbing. Metode ini mengelompokan data berdasarkan fitur-fitur mirip yang dimiliki oleh setiap objek data ke dalam kelompok tertentu (Mirkes, 2011). Sebelum proses pengelompokan dilakukan, terlebih dahulu ditentukan jumlah klaster yang akan didapatkan serta jumlah iterasi (perulangan) proses klaster nya. Objek data dikelompokan berdasarkan similarity yang dimilikinya dengan melihat jarak antar objek data dengan pusat klaster (*centroid*). Perhitungan jarak dapat dilakukan dengan menggunakan metode-metode perhitungan jarak, salah satunya adalah menggunakan euclidean distance. Berikut ini adalah algoritma dari K-means :

1. Tentukan jumlah klaster sebanyak K
2. Tentukan pusat klaster (centroid) secara acak sebanyak K
3. Hitung jarak antar objek data menggunakan Euclidean Distance sebagai berikut :

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$

Dimana, $d(x_i,x_j)$ merupakan jarak antara objek data x_i dan x_j, n adalah dimensi atau jumlah data, x_i merupakan koordinat objek data x_i pada dimensi ke k, dan x_j adalah koordinat objek data x_i pada dimensi ke k.

4. Kelompokan data berdasarkan nilai jarak yang sudah dihitung pada tahap 3 sehingga terkelompok menjadi K buah klaster.
5. Perbaharui nilai pusat klaster dengan persamaan sebagai berikut :

$$\mu_k = \frac{1}{N_k} \sum_{q=1}^{N_k} x_q$$

   Dimana μ_k merupakan titik centroid dari klaster ke K, N_k merupakan banyaknya data pada klaster ke K dan x_q adalah data ke – q pada klaster ke K.
6. Ulangi langkah ke 2 sampai 5 sampai nilai dari pusat klaster tidak berubah lagi, atau anggota klaster tidak mengalami perubahan.

11.2. Studi kasus

Diberikan sejumlah data dummy sebagai berikut :

No.	Objek Data		
	X1	X2	X3
1	8	10	20
2	10	6	6
3	12	8	10
4	10	12	18
5	20	10	26
6	2	20	12
7	4	12	6
8	6	8	20

Kelompokan data pada table di atas ke dalam tiga buah klaster. Diketahui nilai pusat klaster awal masing-masing sebagai berikut : C1 = {10,6,6}; C2 = {10,12,18) dan C3 = {6,8,20}.

Penyelesaian :

Diketahui jumlah K = 3

Pusat klaster adalah C1 = {10,6,6}; C2 = {10,12,18) dan C3 = {6,8,20}.

Langkah selanjutnya adalah menghitung jarak antar objek data berdasarkan ketiga pusat klaster yang sudah ditentukan. Berikut ilustrasi perhitungan jarak antar objek data pertama terhadap semua pusat klaster menggunakan Euclidian distance :

Dalam hal ini x_i merupakan titik koordinat untuk objek data yang akan di klaster, dan x_j merupakan pusat klaster (centroid) yang telah ditentukan sebelumnya.

Data 1 {8,10,20} terhadap C1 = {10,6,6}

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{3} (x_{ik} - x_{jk})^2}$$

$$= \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + (x_{i3} - x_{j3})^2}$$

$$d(x_i, x_j) = \sqrt{(8-10)^2 + (10-6)^2 + (20-6)^2}$$

$$d(x_i, x_j) = \sqrt{(-2)^2 + 4^2 + (14)^2}$$

$$d(x_i, x_j) = \sqrt{4 + 16 + 196} = 14.7$$

Data 1 {8,10,20} terhadap C2 = {10,12,18}

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{3} (x_{ik} - x_{jk})^2}$$

$$= \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + (x_{i3} - x_{j3})^2}$$

$$d(x_i, x_j) = \sqrt{(8-10)^2 + (10-12)^2 + (20-18)^2}$$

$$d(x_i, x_j) = \sqrt{(-2)^2 + (-2)^2 + (2)^2}$$

$$d(x_i, x_j) = \sqrt{4 + 4 + 4} = 3.4$$

Data 1 {8,10,20} terhadap C3 = {6,8,20}

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{3} (x_{ik} - x_{jk})^2}$$

$$= \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + (x_{i3} - x_{j3})^2}$$

$$d(x_i, x_j) = \sqrt{(8-6)^2 + (10-8)^2 + (20-20)^2}$$

$$d(x_i, x_j) = \sqrt{(2)^2 + 2^2 + (0)^2}$$

$$d(x_i, x_j) = \sqrt{4+4} = 2.8$$

Lakukan perhitungan yang sama untuk semua data (data 2 sampai ke 8) sehingga diperoleh informasi jarak antar data sebagai berikut :

No.	Objek Data			Jarak Data ke Pusat Klaster		
	X1	X2	X3	C1 = {10,6,6}	C2 = {10,12,18}	C3 = {6,8,20}
1	8	10	20	14.7	3.4	2.8
2	10	6	6	0.0	13.4	14.7
3	12	8	10	4.9	9.1	11.6
4	10	12	18	13.4	0.0	7.5
5	20	10	26	22.7	12.9	15.3
6	2	20	12	17.2	12.8	14.9
7	4	12	6	8.5	13.4	14.7
8	6	8	20	14.7	6.0	0.0

Langkah selanjutnya adalah mengelompokan setiap objek data ke dalam klaster yang ditentukan berdasarkan jarak yang sudah diketahui, sehingga diperoleh sebagai berikut :

No.	Objek Data			Jarak Data ke Pusat Klaster			Klaster
	X1	X2	X3	C1 = {10,6,6}	C2 = {10,12,18}	C3 = {6,8,20}	
1	8	10	20	14.7	3.4	2.8	3

2	10	6	6	0.0	13.4	14.7	1
3	12	8	10	4.9	9.1	11.6	1
4	10	12	18	13.4	0.0	7.5	2
5	20	10	26	22.7	12.9	15.3	2
6	2	20	12	17.2	12.8	14.9	2
7	4	12	6	8.5	13.4	14.7	1
8	6	8	20	14.7	6.0	0.0	3

Hasil pengelompokan awal dapat dilihat bahwa ada sebanyak tiga objek data yang masuk ke dalam klaster 1, tiga buah objek data pada klaster 2 dan 2 buah objek data pada klaster 3. Langkah selanjutnya adalah menghitung ulang pusat klaster (centroid) baru dengan mencari nilai rata-rata untuk setiap klaster yang terbentuk. Perhitungan pusat klaster baru dilakukan dengan persamaan pada tahap ke – 5 algoritma K-means. Hasil perhitungan pusat klaster baru diperoleh sebagai berikut :

C1 dengan anggota klaster = ({10,6,6};{12,8,10};{4,12,6}) diperoleh pusat klaster baru sebagai berikut :

$$C1 = \left\{\left(\frac{10+12+4}{3}\right), \left(\frac{6+8+12}{3}\right), \left(\frac{6+10+6}{3}\right)\right\}$$

$$C1 = \left\{\left(\frac{26}{3}\right), \left(\frac{26}{3}\right), \left(\frac{22}{3}\right)\right\}$$

$$C1 = \{8.7,8.7,7.3\}$$

C2 dengan anggota klaster = ({10,12,18};{20,10,26};{2,20,12}) diperoleh pusat klaster baru sebagai berikut :

$$C2 = \left\{\left(\frac{10+20+2}{3}\right), \left(\frac{12+10+20}{3}\right), \left(\frac{18+26+12}{3}\right)\right\}$$

$$C2 = \left\{\left(\frac{32}{3}\right), \left(\frac{32}{3}\right), \left(\frac{56}{3}\right)\right\}$$

$$C3 = \{10.7,10.7,18.7\}$$

C3 dengan anggota klaster = ({8,10,20}; {6,8,20}) diperoleh pusat klaster baru sebagai berikut :

$$C3 = \left\{\left(\frac{8+6}{2}\right), \left(\frac{10+8}{2}\right), \left(\frac{20+20}{2}\right)\right\}$$
$$C3 = \left\{\left(\frac{14}{2}\right), \left(\frac{18}{2}\right), \left(\frac{40}{2}\right)\right\}$$
$$C3 = \{7,9,20\}$$

Selanjutnya hitung ulang jarak antar data berdasarkan nilai pusat klaster yang baru untuk mengelompokan ulang setiap objek data kemudian dibandingkan hasilnya dengan klaster yang terbentuk sebelumnya. Berikut hasil perhitungan jarak antar objek data terhadap nilai pusat klaster yang baru :

No.	Objek Data			Jarak Data ke Pusat Klaster		
	X1	X2	X3	C1 = {8.7,8.7,7.3}	C2 = {10.7,10.7,18.7}	C3 = {7,9,20}
1	8	10	20	12.8	3.1	1.4
2	10	6	6	3.3	13.6	14.6
3	12	8	10	4.3	9.2	11.2
4	10	12	18	11.3	1.6	4.7
5	20	10	26	21.9	11.8	14.4
6	2	20	12	14.0	14.4	14.5
7	4	12	6	5.9	14.4	14.6
8	6	8	20	13.0	5.6	1.4

Kelompokan ulang berdasarkan jarak objek data yang diperoleh, sehingga diperoleh sebagai berikut :

No.	Objek Data			Jarak Data ke Pusat Klaster			Klaster
	X1	X2	X3	C1 = {8.7,8.7,7.3}	C2 = {10.7,10.7,18.7}	C3 = {7,9,20}	
1	8	10	20	12.8	3.1	1.4	3
2	10	6	6	3.3	13.6	14.6	1
3	12	8	10	4.3	9.2	11.2	1
4	10	12	18	11.3	1.6	4.7	2
5	20	10	26	21.9	11.8	14.4	2

6	2	20	12	14.0	14.4	14.5	1
7	4	12	6	5.9	14.4	14.6	1
8	6	8	20	13.0	5.6	1.4	3

Hasil pengelompokan iterasi kedua terlihat ada perubahan anggota klaster, Klaster 1 pada iterasi pertama berisi 3 anggota berubah menjadi 4 anggota klaster pada iterasi kedua. Karena anggota klaster dari iterasi pertama dan kedua mengalami perubahan, maka lakukan pengelompokan ulang ke iterasi ketiga. Langkah kedua sampai lima pada algoritma K-means diulang. Berikut adalah pusat klaster yang baru untuk iterasi ketiga :

C1 dengan anggota klaster = ({10,6,6};{12,8,10};{2,20,12}; {4,12,6}) diperoleh pusat klaster baru sebagai berikut :

$$C1 = \left\{\left(\frac{10+12+2+4}{4}\right), \left(\frac{6+8+20+12}{4}\right), \left(\frac{6+10+12+6}{4}\right)\right\}$$

$$C1 = \left\{\left(\frac{28}{4}\right), \left(\frac{46}{4}\right), \left(\frac{34}{4}\right)\right\}$$

$$C1 = \{7,11.5,8.5\}$$

C2 dengan anggota klaster = ({10,12,18};{20,10,26}) diperoleh pusat klaster baru sebagai berikut :

$$C2 = \left\{\left(\frac{10+20}{2}\right), \left(\frac{12+10}{2}\right), \left(\frac{18+26}{2}\right)\right\}$$

$$C2 = \left\{\left(\frac{30}{2}\right), \left(\frac{22}{2}\right), \left(\frac{44}{2}\right)\right\}$$

$$C3 = \{15,11,22\}$$

C3 dengan anggota klaster = ({8,10,20}; {6,8,20}) diperoleh pusat klaster baru sebagai berikut :

$$C3 = \left\{\left(\frac{8+6}{2}\right), \left(\frac{10+8}{2}\right), \left(\frac{20+20}{2}\right)\right\}$$

$$C3 = \left\{\left(\frac{14}{2}\right), \left(\frac{18}{2}\right), \left(\frac{40}{2}\right)\right\}$$

$$C3 = \{7,9,20\}$$

Selanjutnya hitung ulang jarak antar data berdasarkan nilai pusat klaster yang baru untuk mengelompokan ulang setiap objek data kemudian dibandingkan hasilnya dengan klaster yang terbentuk sebelumnya. Berikut hasil perhitungan jarak antar objek data terhadap nilai pusat klaster yang baru :

No.	Objek Data			Jarak Data ke Pusat Klaster		
	X1	X2	X3	C1 = {7,11.5,8.5}	C2 = {15,11,22}	C3 = {7,9,20}
1	8	10	20	11.6	7.3	1.4
2	10	6	6	6.7	17.5	14.6
3	12	8	10	6.3	12.7	11.2
4	10	12	18	10.0	6.5	4.7
5	20	10	26	21.9	6.5	14.4
6	2	20	12	10.5	18.7	14.5
7	4	12	6	3.9	19.4	14.6
8	6	8	20	12.1	9.7	1.4

Kelompokan ulang berdasarkan jarak objek data yang diperoleh, sehingga diperoleh sebagai berikut :

No.	Objek Data			Jarak Data ke Pusat Klaster			Klaster
	X1	X2	X3	C1 = {8.7,8.7,7.3}	C2 = {10.7,10.7,18.7}	C3 = {7,9,20}	
1	8	10	20	11.6	7.3	1.4	3
2	10	6	6	6.7	17.5	14.6	1
3	12	8	10	6.3	12.7	11.2	1
4	10	12	18	10.0	6.5	4.7	3
5	20	10	26	21.9	6.5	14.4	2
6	2	20	12	10.5	18.7	14.5	1
7	4	12	6	3.9	19.4	14.6	1
8	6	8	20	12.1	9.7	1.4	3

Selanjutnya bandingkan antara anggota klaster pada iterasi saat ini dengan iterasi sebelumnya, karena masih terlihat perbedaan anggota klaster, maka pengelompokan ulang terus dilakukan sampai iterasi tertentu. Berikut ini adalah hasil pengelompokan pada iterasi keempat dimana tidak terjadi lagi perubahan anggota klaster hasil dari iterasi ketiga dan keempat.

No.	Objek Data			Jarak Data ke Pusat Klaster			Klaster
	X1	X2	X3	C1 = {8.7,8.7,7.3}	C2 = {10.7,10.7,18.7}	C3 = {7,9,20}	
1	8	10	20	11.6	13.4	0.7	3
2	10	6	6	6.7	22.7	14.1	1
3	12	8	10	6.3	18.0	10.3	1
4	10	12	18	10.0	13.0	3.1	3
5	20	10	26	21.9	0.0	13.7	2
6	2	20	12	10.5	24.9	13.8	1
7	4	12	6	3.9	25.7	14.1	1
8	6	8	20	12.1	15.4	2.9	3

Dengan demikian, hasil pengelompokan data menggukan K-means diperoleh

Klaster 1 = {Data 2, Data 3, Data 6, Data 7}
Klaster 2 = {Data 5}
Klaster 3 = {Data 1, Data 4}

11.3. Algoritme K-means Menggunakan R

Sama seperti algoritme PAM, pada k-means hampir tidak ada perbedaan. Diawali dengan pemanggilan datasets, pre-processing, normalisasi (jika diperlukan), penentuan jumlah klaster, proses klastering sampai dengan evaluasi hasil klaster. Fungsi kmeans() digunakan pada teknik ini yang merupakan bagian dari paket library(cluster). Berikut tahapan analisis klaster dengan k-means menggunakan R :

1. Siapkan dataset (untuk dataset yang digunakan sama dengan analisis PAM pada bab 10), kemudian panggil menggunakan perintah read.csv() dan tampilkan enam data pertama menggunakan perintah head() sebagai berikut :

```
ambilData=read.csv('../input/pdrbjawabagianbarat/pdrb-gabungan.csv',
                   sep=',', na.string=c('"))
head(ambilData)
```

2. Hasil perintah head()

A data.frame: 6 × 13

	Nama.Wilayah	Year.2010	Year.2011	Year.2012	Year.2013	Year.2014	Year.2015	Year.2016	Year.2017	Year.2018	Year.2019	Year.2020	Year.2021
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Kabupaten Lebak	12572538	13325629	14006209	14887984	15756247	16733238	17665397	18683740	19735870	20810490	20610990	21245040
2	Kabupaten Pandeglang	12279542	12984402	13738882	14387883	15097105	15974129	16855618	17866428	18812932	19644125	19541488	20127757
3	Kabupaten Serang	33841000	35905370	37849640	40246690	42541180	44454580	46715180	49154636	51754320	54347488	53055563	54992522
4	Kabupaten Tangerang	18549119	62022491	65848281	70065983	74101232	78093560	82183596	86964027	92011405	97142198	93544934	97809902
5	Kota Cilegon	44676529	47633318	51300206	54732934	57261923	59982732	62981047	66444529	70502082	74228641	73534471	77071368
6	Kota Serang	12549572	13595691	14604637	15670784	16745084	17808478	18935486	20153023	21482093	22813096	22517969	23374085

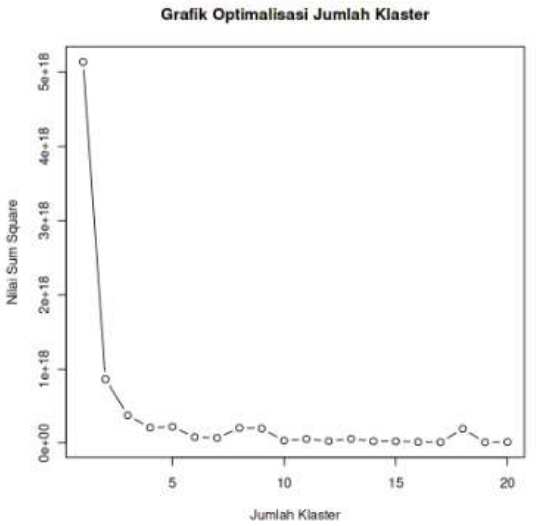
3. Lakukan data cleaning menggunakan perintah na.omit() untuk menghilangkan data instance yang memiliki value N/A sebagai berikut ;

```
ambilData=na.omit(ambilData)
```

4. Untuk melakukan klasterisasi khususnya menggunakan k-means, hal yang paling krusial adalah menentukan jumlah klaster optimal. Banyak dari analis data yang sering kesulitan dalam menentukan jumlah klaster optimal saat akan mengelompokan data tertentu. Pada bahasa R, penentuan jumlah klaster optimal dapat dilakukan

dengan menghitung nilai Within Sum Square (WWS) sebagai berikut :

```
D1 =(nrow(ambilData[,2:13])-1)*sum(apply(ambilData[,2:13],2,var))
for (i in 2:20) D1[i]<-sum(kmeans(ambilData[,2:13],centers=i)$withinss)
plot(1:20, D1, type = "b", main = "Grafik Optimalisasi Jumlah Klaster",
     xlab = "Jumlah Klaster", ylab = "Nilai Sum Square")
```



Seperti diperlihatkan pada Grafik di atas, terjadi penurunan sangat drastis dari nilai within sum square saat jumlah klaster meningkat dari 1 klaster menjadi 4 atau 5. Dengan demikian, pada turunan klaster k = 4 memperlihatkan kestabilan nilai WWS nya. Jumlah klaster dalam hal ini dapat dinyatakan dengan melihat keseimbangan antara k dan WWS. Dengan demikian, pada kasus ini jumlah klaster yang akan digunakan adalah k = 4

5. Setelah jumlah klaster diketahui, langkah berikutnya adalah membangun klaster dengan fungsi k-means menggunakan R Studio. Dalam hal ini fungsi kmeans yang sudah tersedia pada bahasa R digunakan sebagai berikut :

Perintah dasar k-medoid R Studio

kmeans(x, centers, iter.max = 10, nstart = 1, algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"), trace=FALSE)

Keterangan :

x	:	matrik data dalam bentuk numerik atau data yang dapat dikonversi ke dalam bentuk matrik
centers	:	jumlah klaster, bisa disebut jumlah k atau jumlah initial pusat klaster
iter.max	:	jumlah maksimum iterasi yang digunakan untuk menghasilkan bentuk klaster optimal
nstart	:	jumlah random yang digunakan untuk melakukan klasterisasi berdasarkan k
algorithm	:	jenis algoritma k-means yang akan digunakan

6. Berdasarkan data yang ada dan jumlah klaster (k) yang telah ditentukan, untuk mengelompokan data dengan k-means dilakukan dengan menuliskan perintah sebagai berikut :

```
klaster=kmeans(ambilData[,2:13],centers=4, nstart=4, iter.max = 10)
print(klaster)
```

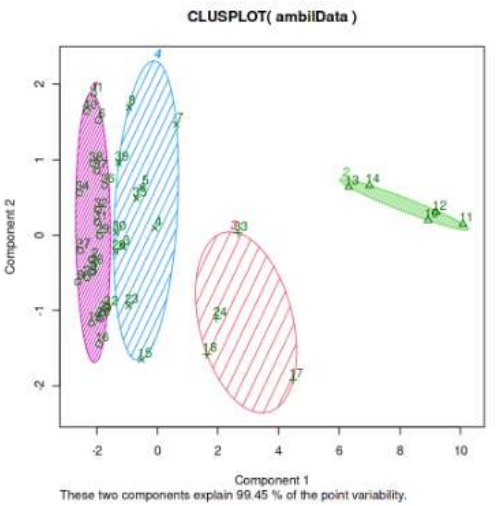
```
K-means clustering with 4 clusters of sizes 21, 5, 4, 11

Cluster means:
  Year.2010 Year.2011 Year.2012 Year.2013 Year.2014 Year.2015 Year.2016
1  12579917  13884402  14650214  15420415  16236685  17106107  18111072
2 214319782 231114857 248005150 266262952 285867062 306752647 326372472
3 112268899 119831671 127655779 136548047 145053178 153122102 162524197
4  34726179  43989648  46837246  49713456  52590301  55303884  58210339
  Year.2017 Year.2018 Year.2019 Year.2020 Year.2021
1  19129558  20208750  21343850  21123167  22032759
2 349430503 373927746 398236974 390079295 405603494
3 172243864 183157168 192627267 186977860 194691687
4  61269996  64795647  68216133  66349610  68813875

Clustering vector:
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 1  1  4  4  4  1  4  4  1  2  2  2  2  2  4  1  3  3  1  1  1  1  4  3  1  1
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
 1  4  1  4  1  1  3  1  4  1  1  1  4  1  1

Within cluster sum of squares by cluster:
[1] 1.724228e+16 1.408127e+17 5.010995e+16 3.851900e+16
 (between_SS / total_SS =  95.2 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

Perintah pengelompokan data diatas menggunakan parameter data yang diambil dari ambilData berdasarkan variabel ke 2 sampai 13 (bertipe numerik), jumlah k = 4 sekaligus sebagai centroid data, jumlah pengambilan data random (nstart) sama dengan jumlah parameter centers, sedangkan untuk iterasi yang digunakan adalah 10 perulangan. Perhatikan pada dua hasil pertama yaitu cluster means dan clustering vector. cluster means memperlihatkan nilai akhir cluster means untuk semua variabel sebanyak klaster yang terbentuk. Sedangkan untuk komponen clustering vector, menyajikan informasi keanggotaan klaster setiap data instance. Baris pertama menunjukan nomor urut data instance, sementara baris kedua menunjukan klaster tempat data instance tersebut berada.

7. Visualisasi hasil klaster ke dalam plot

Kita dapat memvisualisasikan hasil klaster ke dalam plot menggunakan fungsi clusplot() sebagai berikut :

```
library(cluster)
clusplot(ambilData, klaster$cluster, color=TRUE, shade=TRUE,labels=2, lines=0)
```



8. Informasi klaster untuk masing-masing data instance (dalam hal ini Wilayah) juga dapat dilakukan sebagai berikut :

Pisahkan variable Nama Wilayah dari datasets menggunakan fungsi **row.names()** sebagai berikut :

```
wil=row.names=c(ambilData[,1])
wil
```

Kemudian gabungkan dengan hasil klaster menggunakan perintah cbind() sebagai berikut :

```
lengkap=cbind(wil, klaster$cluster)
lengkap[1:10,]
```

A matrix: 10 × 2 of type chr

	wil	
1	Kabupaten Lebak	1
2	Kabupaten Pandeglang	1
3	Kabupaten Serang	4
4	Kabupaten Tangerang	4
5	Kota Cilegon	4
6	Kota Serang	1
7	Kota Tangerang	4
8	Kota Tangerang Selatan	4
9	Kabupaten Kepulauan Seribu	1
10	Kota Jakarta Barat	2

Hasilnya dapat kita lihat untuk 10 data pertama, Kabupaten Lebak masuk ke dalam klaste 1, Kabupaten Pandeglang ke dalam Klaster 1, Kabupaten Serang ke dalam klaster 4 dan seterusnya.

BAB XII
HIERARCHICAL AGGLOMERATIVE CLUSTERING

12.1 Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering (HAC) adalah metode pengelompokan data yang menggabungkan 2 buah klaster (setiap objek dipandang sebagai single klaster) yang memiliki kemiripan. Bentuk klaster berupa rangkaian penurunan jumlah kelompok pada setiap tahapan membentuk sebuah hirarki layaknya susunan struktur organisasi. Metode HAC tidak seperti metode lainnya yang harus ditentukan jumlah klaster yang akan dibentuk sejak awal. Pada HAC jumlah klaster tidak pernah ditentukan secara spesifik. Meskipun demikian, pada beberapa kasus penentuan jumlah klaster pada HAC juga sering dilakukan dengan menggabungkan metode lain di dalamnya. Berikut ini adalah algoritme HAC :

1. Input :
   E = {e1,e2,...,en}; (himpunan objek data)
2. Output :
   C = {c1,c2,...,cn}; (himpunan cluster)
3. Foreach{e1,e2} | ei ϵ E, $1 \leq i \leq n$do
   D(e1,e2) $\leftarrow$ sqrt(sqr(sum(e1 – e2))); (hitung jarak antar objek data, misal dengan Eucledian distance)
4. End;
5. Tentukan matrik kedekatan berdasarkan jarak D untuk semua himpunan E;
6. Tentukan himpunan cluster berdasarkan cluster singleton, dimana setiap himpunan cluster merepresentasikan input E;

7. Repeat
   d(e1,e2) ← min(D(e1,e2)); (gabungkan dua cluster terdekat, misal dengan single linkage)
   Update matriks kedekatan dengan distance D baru antara cluster yang baru terbentuk dengan cluster asli E;
8. Until jarak D(e1,e2) = 1;

Teknik pengelompokan hirarki (Single Linkage, Complete Linkage, Average Linkage, Average Group Linkage) dibedakan berdasarkan konsep jarak antar kelompok, penentuan jarak antar kelompok untuk teknik-teknik tersebut seperti diperlihatkan pada Tabel 12.1.

Tabel 12. 1 Teknik Pengelompokan HAC Berdasarkan Nilai Jarak

No.	**Metode**	**Jarak antara kelompok (i,j) dengan k**
1	*Single linkage*	$d_{(i,j)k} = \min(d_{ik}, d_{jk})$
2	*Complete linkage*	$d_{(i,j)k} = \max(d_{ik}, d_{jk})$
3	*Average linkage*	$d_{(i,j)k} = average(d_{ik}, d_{jk})$

12.2 Studi Kasus

Diberikan data dummy sebagai berikut :

No.	**X1**	**X2**
1	10	12
2	6	3
3	8	5
4	4	6

Kelompokan data di atas menggunakan metode single, complete dan average linkage. Tahap pertama adalah menghitung jarak antara objek data menggunakan salah satu metode pengukuran jarak, pada kasus ini digunakan Euclidian Distance. Berikut perhitungan jarak untuk keempat data tersebut :

$$d(x_{ik}, x_{jk}) = \sqrt{\sum_{k=1}^{n}(x_{jk} - x_{ik})^2}$$

$$d(Data\ 1, Data\ 1) = \sqrt{(12-12)^2 + (10-10)^2} = 0$$

$$d(Data\ 1, Data\ 2) = \sqrt{(12-3)^2 + (10-6)^2} = \sqrt{(9)^2 + (4)^2} = \sqrt{97} = 9.8$$

$$d(Data\ 1, Data\ 3) = \sqrt{(12-5)^2 + (10-8)^2} = \sqrt{(7)^2 + (2)^2} = \sqrt{53} = 7.3$$

$$d(Data\ 1, Data\ 4) = \sqrt{(12-6)^2 + (10-4)^2} = \sqrt{(6)^2 + (6)^2} = \sqrt{72} = 8.5$$

$$d(Data\ 2, Data\ 3) = \sqrt{(3-5)^2 + (6-8)^2} = \sqrt{(-2)^2 + (-2)^2} = \sqrt{8} = 2.8$$

$$d(Data\ 2, Data\ 4) = \sqrt{(3-6)^2 + (6-4)^2} = \sqrt{(-3)^2 + (2)^2} = \sqrt{13} = 3.6$$

$$d(Data\ 3, Data\ 4) = \sqrt{(5-6)^2 + (8-4)^2} = \sqrt{(-1)^2 + (4)^2} = \sqrt{17} = 4.1$$

Urutkan jarak yang sudah dihitung ke dalam matrik jarak untuk mempermudah pengelompokan data. Berikut tabel matrik jarak yang dimaksud :

D_{Euc}	1	2	3	4
1	0	9.8	7.3	8.5
2	9.8	0	2.8	3.6
3	7.3	2.8	0	4.1
4	8.5	3.6	4.1	0

12.3 Metode Single Linkage

Untuk mengelompokan data berdasarkan pendekatan Single Linkage, gunakan persamaan sebagai berikut :

$$d_{(i,j)k} = \min(d_{ik}, d_{jk})$$

Berdasarkan data jarak yang sudah dihitung dan memperlakukan setiap objek data sebagai single klaster,pilih jarak dua kelompok terkecil dengan mengelompokan masing-masing dua buah data :

$$\min(D_{Euc}) = \min(1,2) = 9.8$$
$$\min(D_{Euc}) = \min(1,3) = 7.3$$
$$\min(D_{Euc}) = \min(1,4) = 8.5$$
$$\min(D_{Euc}) = \min(2,3) = 2.8$$
$$\min(D_{Euc}) = \min(2,4) = 3.6$$
$$\min(D_{Euc}) = \min(3,4) = 4.1$$

dengan demikian terpilih kelompok 2 dan 3, sehingga keduanya digabungkan menjadi satu buah klaster. Kemudian lakukan perhitungan ulang jarak antara kelompok yang sudah terbentuk (2 dan 3) dengan kelompok lain yaitu 1 dan 4, sehingga diperoleh sebagai berikut :

$$\mathrm{D}_{(23),1} = \min(D_{21}, D_{31}) = \min(9.8\ , 7.3) = 7.3$$
$$\mathrm{D}_{(23),4} = \min(D_{24}, D_{34}) = \min(3.6\ , 4.1) = 3.6$$

Dengan demikian matrik jarak menjadi sebagai berikut :

D_{Euc}	1	(2,3)	4
1	0	7.3	8.5
(2,3)	7.3	0	3.6
4	8.5	3.6	0

Selanjutnya pilih jarak dua kelompok yang terkecil sehingga diperoleh :

$$\min(D_{Euc}) = \min(D_{(23),1}) = 7.3$$
$$\min(D_{Euc}) = \min(D_{(23),4}) = 3.6$$
$$\min(D_{Euc}) = \min(D_{14}) = 8.5$$

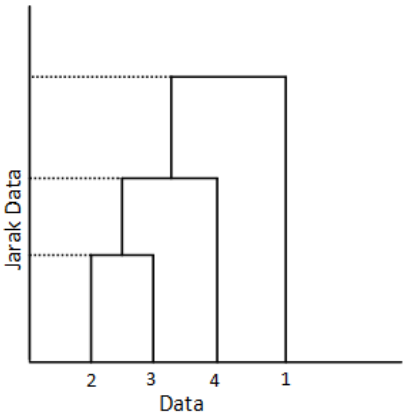
Dengan demikian terpilih kelompok (2,3) dan 4 dengan jarak terkecil, sehingga keduanya digabungkan menjadi satu buah klaster. Selanjutnya hitung ulang jarak antara kelompok yang sudah terbentuk (2,3) dan 4 dengan kelompok lain yaitu klaster 1, sehingga diperoleh sebagai berikut :

$$D_{(2,3,4),1} = \min(D_{21}, D_{31}, D_{41}) = \min(9.8\ , 7.3\ , 8.5) = 8.5$$

Dengan demikian matrik jarak menjadi sebagai berikut :

D_{Euc}	((2,3),4)	1
((2,3),4)	0	8.5
1	8.5	0

Tahap terakhir, gabungkan klaster (2,3,4) dengan klaster 1. Sehingga terbentuk dendrogram klaster hirarki sebagai berikut :



12.4 Metode Complete Linkage

D_{Euc}	1	2	3	4
1	0	9.8	7.3	8.5
2	9.8	0	2.8	3.6
3	7.3	2.8	0	4.1
4	8.5	3.6	4.1	0

Untuk mengelompokan data berdasarkan pendekatan Complete Linkage, gunakan persamaan sebagai berikut :

$$d_{(i,j)k} = \max(d_{ik}, d_{jk})$$

Berdasarkan data jarak yang sudah dihitung dan memperlakukan setiap objek data sebagai single klaster,pilih jarak dua kelompok terbesar dengan mengelompokan masing-masing dua buah data :

$$max\ (D_{Euc}) = max\ (1,2) = 9.8$$
$$max\ (D_{Euc}) = max\ (1,3) = 7.3$$
$$max\ (D_{Euc}) = max\ (1,4) = 8.5$$
$$max\ (D_{Euc}) = max\ (2,3) = 2.8$$
$$max\ (D_{Euc}) = max\ (2,4) = 3.6$$
$$max\ (D_{Euc}) = max\ (3,4) = 4.1$$

dengan demikian terpilih kelompok 1 dan 2, sehingga keduanya digabungkan menjadi satu buah klaster. Kemudian lakukan perhitungan ulang jarak antara kelompok yang sudah terbentuk (1 dan 2) dengan kelompok lain yaitu 3 dan 4, sehingga diperoleh sebagai berikut :

$$D_{(12),3} = \max(D_{13}, D_{23}) = \max(7.3\ , 2.8) = 7.3$$
$$D_{(12),4} = \max(D_{14}, D_{24}) = \max(8.5\ , 3.6) = 8.5$$

Dengan demikian matrik jarak menjadi sebagai berikut :

D_{Euc}	(1,2)	3	4
(1,2)	0	7.3	8.5
3	7.3	0	4.1
4	8.5	4.1	0

Selanjutnya pilih jarak dua kelompok yang terbesar sehingga diperoleh :

$$\max(D_{Euc}) = \max\left(D_{(12),3}\right) = \max\left(D_{(13)}, D_{(23)}\right) = 7.3$$
$$\max(D_{Euc}) = \max\left(D_{(12),4}\right) = \max\left(D_{(14)}, D_{24}\right) = 8.5$$
$$\max(D_{Euc}) = \max(D_{34}) = 4.1$$

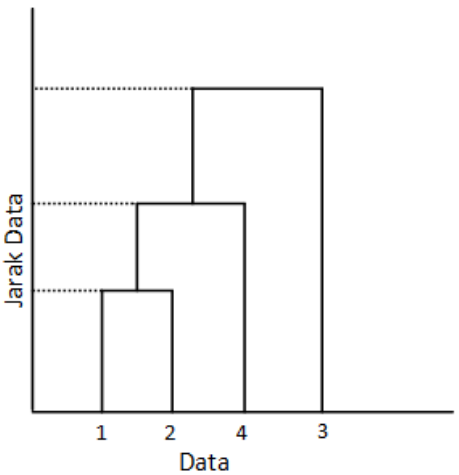
Dengan demikian terpilih kelompok (1,2) dan 4 dengan jarak terbesar, sehingga keduanya digabungkan menjadi satu buah klaster. Selanjutnya hitung ulang jarak antara kelompok yang sudah terbentuk (1,2) dan 4 dengan kelompok lain yaitu klaster 3, sehingga diperoleh sebagai berikut :

$$D_{(1,2,4),3} = \max(D_{13}, D_{23}, D_{43}) = \max(7.3\ , 2.8\ , 4.1) = 7.3$$

Dengan demikian matrik jarak menjadi sebagai berikut :

D_{Euc}	((1,2),4)	3
((1,2),4)	0	7.3
3	7.3	0

Tahap terakhir, gabungkan klaster ((1,2),4) dengan klaster 3. Sehingga terbentuk dendrogram klaster hirarki sebagai berikut :



12.5 Metode Average Linkage

D_{Euc}	1	2	3	4
1	0	9.8	7.3	8.5
2	9.8	0	2.8	3.6
3	7.3	2.8	0	4.1
4	8.5	3.6	4.1	0

Untuk mengelompokan data berdasarkan pendekatan Average Linkage, gunakan persamaan sebagai berikut :

$$d_{(i,j)k} = average\ (d_{ik}, d_{jk})$$

Berdasarkan data jarak yang sudah dihitung dan memperlakukan setiap objek data sebagai single klaster,pilih jarak dua kelompok terbesar dengan mengelompokan masing-masing dua buah data :

$$average\ (D_{Euc}) = average\ (1,2) = 9.8$$
$$average\ (D_{Euc}) = average\ (1,3) = 7.3$$
$$average\ (D_{Euc}) = average\ (1,4) = 8.5$$
$$average\ (D_{Euc}) = average\ (2,3) = 2.8$$
$$average\ (D_{Euc}) = average\ (2,4) = 3.6$$
$$average\ (D_{Euc}) = average\ (3,4) = 4.1$$

Dalam hal ini, data rata-rata jarak pengelompokan pertama diambil dari nilai rata-rata terkecil, dengan demikian terpilih kelompok 2 dan 3 sebagai group klaster awal, sehingga keduanya digabungkan menjadi satu buah klaster. Kemudian lakukan perhitungan ulang jarak antara kelompok yang sudah terbentuk (2 dan 3) dengan kelompok lain yaitu 1 dan 4, sehingga diperoleh sebagai berikut :

$$D_{(23),1} = average\ (D_{21}, D_{31}) = average\ (9.8\ , 7.3) = \frac{9.8 + 7.3}{2}$$
$$= 8.55$$

$$D_{(23),4} = average\ (D_{24}, D_{34}) = average\ (3.6\ , 4.1) = \frac{3.6 + 4.1}{2}$$
$$= 3.85$$

Dengan demikian matrik jarak menjadi sebagai berikut :

D_{Euc}	1	(2,3)	4
1	0	8.55	8.5
(2,3)	8.55	0	3.85
4	8.5	3.85	0

Selanjutnya pilih jarak dua kelompok yang terbesar sehingga diperoleh :

$$average\ (D_{Euc}) = average\ \left(D_{(23),1}\right) = 8.55$$
$$average\ (D_{Euc}) = average\ \left(D_{(23),4}\right) = 3.85$$
$$average\ (D_{Euc}) = average\ (D_{14}) = 8.5$$

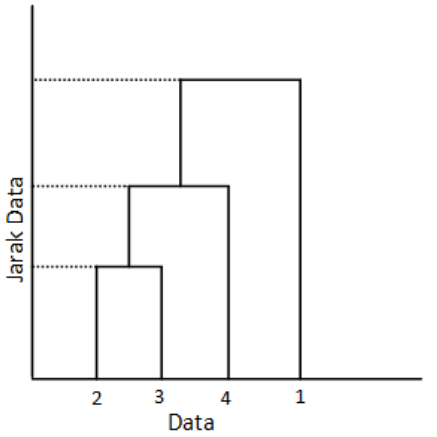
Dengan demikian terpilih kelompok (2,3) dan 4 dengan jarak rata-rata terkecil, sehingga keduanya digabungkan menjadi satu buah klaster. Selanjutnya hitung ulang jarak antara kelompok yang sudah terbentuk (2,3) dan 4 dengan kelompok lain yaitu klaster 1, sehingga diperoleh sebagai berikut :

$$D_{(2,3,4),1} = \text{average}(D_{21}, D_{31}, D_{41}) = \text{average}(9.8\ , 7.3\ , 8.5)$$
$$= \frac{9.8 + 7.3 + 8.5}{3} = 8.5$$

Dengan demikian matrik jarak menjadi sebagai berikut :

D_{Euc}	((2,3),4)	1
((2,3),4)	0	8.5
1	8.5	0

Tahap terakhir, gabungkan klaster ((2,3),4) dengan klaster 1. Sehingga terbentuk dendrogram klaster hirarki sebagai berikut :



12.6 Hierarchical Agglomerative Clustering Menggunakan R

Untuk melakukan klastering data menggunakan hierarchical agglomerative clustering (HAC) menggunakan R, kita dapat menggunakan fungsi hclust(). Namun sebelumnya kita harus melakukan perhitungan jarak antar data menggunakan fungsi dist(). Berikut tahapan klastering data menggunakan HAC pada R. Pada pembahasan buku ini kita akan fokus pada teknik single, average dan complete linkage.

Langkah awal adalah memanggil dataset dengan perintah **read.csv()** serta memeriksa isi data dengan perintah **head()** sebagai berikut :

```
data=read.csv('../input/pdrbjawabagianbarat/pdrb-gabungan.csv', sep=',')
head(data)
```

A data.frame: 6 × 13

	Nama.Wilayah	Year.2010	Year.2011	Year.2012	Year.2013	Year.2014	Year.2015	Year.2016	Year.2017	Year.2018	Year.2019	Year.2020	Year.2021
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Kabupaten Lebak	12572538	13325629	14006209	14887964	15756247	16733238	17665397	18683740	19735870	20810490	20610990	21245040
2	Kabupaten Pandeglang	12279542	12984402	13738882	14387883	15097105	15974129	16855618	17866428	18812932	19644125	19541488	20127757
3	Kabupaten Serang	33841000	35905370	37849640	40246690	42541180	44454580	46715180	49154636	51754320	54347488	53055563	54992522
4	Kabupaten Tangerang	18549119	62022491	65848281	70065983	74101232	78093560	82183596	86964027	92011405	97142198	93544934	97809902
5	Kota Cilegon	44676529	47633318	51300206	54732934	57261923	59982732	62981047	66444529	70502082	74228641	73534471	77071368
6	Kota Serang	12549572	13595691	14604637	15670784	16745084	17808478	18935486	20153023	21482093	22813096	22517969	23374085

Tahap berikutnya adalah melakukan pre-processing data dengan menghilangkan value dengan nilai N/A menggunakan **na.omit().** Namun pada pembahasan buku ini, proses pembersihan data sudah dilakukan sebelum masuk tahap klastering dengan R, sehingga data cleaning dilewati. Tahp selanjutnya adalah menghitung jarak antar data menggunakan fungsi **dist().** Secara umum syntax penulisan fungsi tersebut adalah :

dist(data, method = "euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski", "pearson", "spearman" or "kendall")

Berikut adalah baris kode untuk melakukan perhitungan jarak antar data menggunakan fungsi **dist()** dengan metode Euclidean :

```
#Hitung Jarak Antara Data
jarak=dist(data[,2:13], method = "euclidean")
```

Pada baris kode di atas, hasil perhitungan jarak disimpan pada variable **jarak** dimana variable yang dihitung adalah urutan dari ke 2

sampai variable ke 13. Adapun metode yang digunakan adalah Euclidean. Anda dapat mengganti metode perhitungan jarak sesuai kebutuhan dengan jenis lainnya pada parameter **method.** Metode lainnya yang digunakan adalah **"maximum", "manhattan", "canberra", "binary", "minkowski", "pearson", "spearman" dan "kendall".**

Untuk mengelompokan dataset jarak secara hirarki kita dapat gunakan fungsi hclust() sebagai berikut:

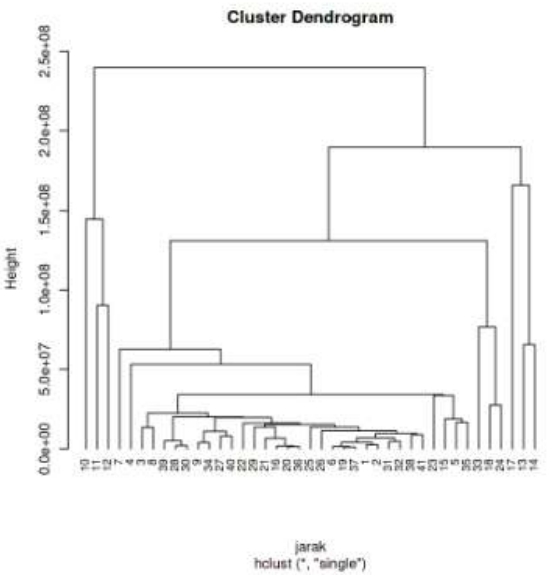
hclust(datajarak, method = "complete", "average", "single", "ward")

Metode atau teknik HAC yang dapat digunakan pada R adalah "complete", "average", "single" dan "ward". Untuk mengganti metode klaster HAC pada dataset, ganti nama pada parameter **method** pada fungsi *hclust* sesuai dengan kebutuhan. Berikut pembahasan pengelompokan data hirarki menggunakan tiga teknik, yaitu single, average dan complete.

1. Klaster data dengan single linkage

Untuk mengelompokan data jarak ke dalam teknik single linkage, kita dapat mengganti value pada parameter method menjadi "single". Sebagai informasi, data jarak sudah dilakukan perhitungan pada tahap sebelumnya. Pada pembahasan buku ini data jarak yang digunakan adalah jarak antar data yang disimpan pada variable **jarak.** Data jarak ini juga digunakan untuk membahas pengelompokan data menggunakan average dan complete linkage. Berikut baris kode dan hasil pengelompokan menggunakan single linkage :

```r
# HAC menggunakan Single Linkage
hac1 <- hclust(jarak, method = "single" )

# # Menampilkan dendrogram
plot(hac1, cex = 0.8, hang = -1)
```



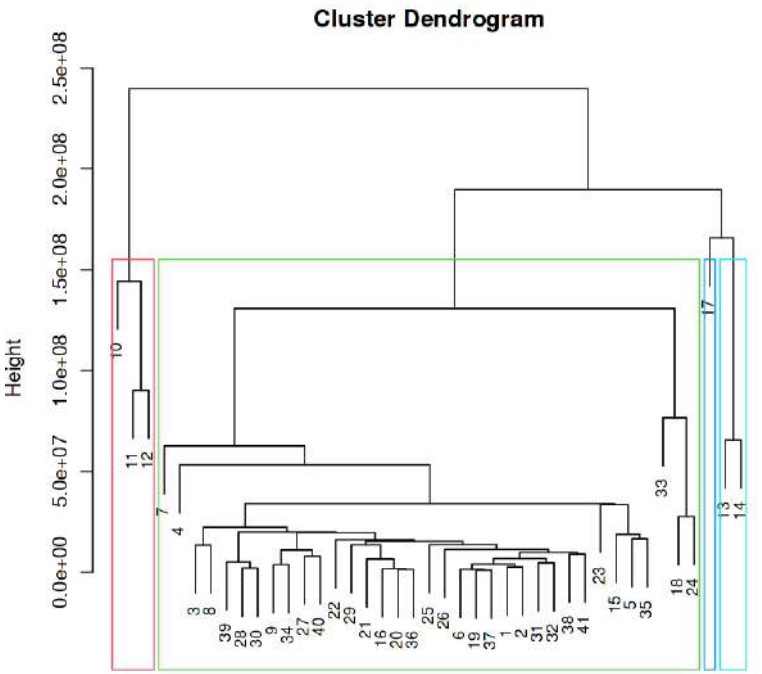
Hasil klaster HAC divisualiasikan dalam bentuk tree yang disebut dengan dendrogram. Angka yang ditunjukan pada sumbu x merepresentasikan indeks data instance yang diklasterkan.

Untuk mempermudah interpretasi klaster HAC, kita juga dapat membagi klaster ke dalam jumlah tertentu kemudian memberikan garis diantara klaster tersebut. Bisanya garis tersebut berbentuk kotak yang diberi warna. Berikut adalah baris kode untuk membagi klaster HAC ke dalam 4 buah group klaster yang diinisiasi dari k = 4 pada parameter rect.hclust(). Dengan cara ini, kita dapat

menginterpretasikan hasil klaster berdasarkan sub klaster yang terbentuk bersama dengan anggota klaster di dalamnya.

```
plot(hac1, cex = 0.8)
rect.hclust(hac1, k = 4, border = 2:5)
```

Berikut adalah visualisasi dari hasil klaster HAC yang sudah dilakukan pemisahan ke dalam empat group (k = 4) :



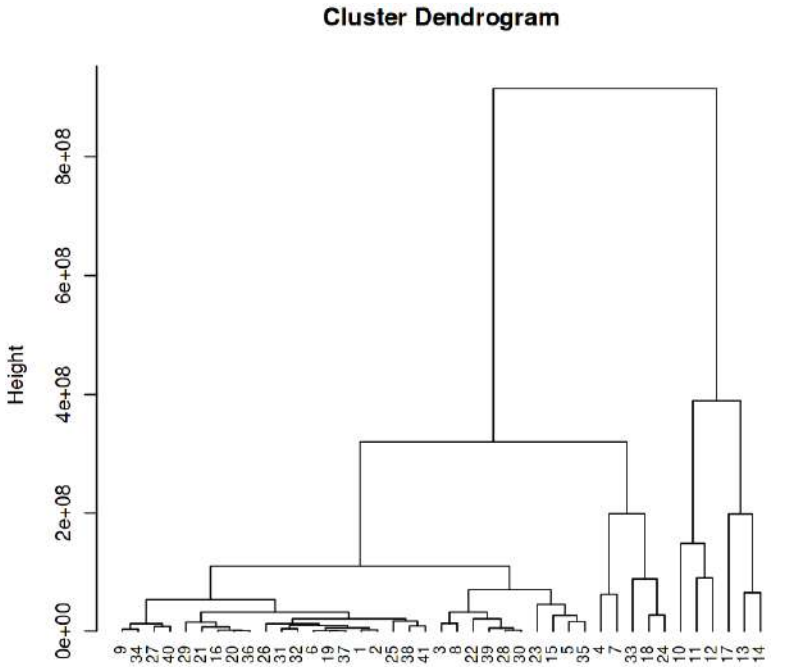
2. Klaster data dengan average linkage

Untuk melakukan klastering data jarak mengguankan metode average linkage, kita dapat menggunakan cara yang sama dengan teknik single, hanya saja pada parameter methode, kita ganti dengan

"average" kemudian visualisasikan dalam bentuk dendrogram. Adapun baris perintahnya sebagai berikut :

```
# HAC menggunakan Average Linkage
hc2 <- hclust(jarak, method = "average" )

# Menampilkan dendrogram
plot(hc2, cex = 0.8, hang = -1)
```

Hasil dendrogram untuk klaster average lingkage seperti diperlihatan pada Gambar di bawah ini :



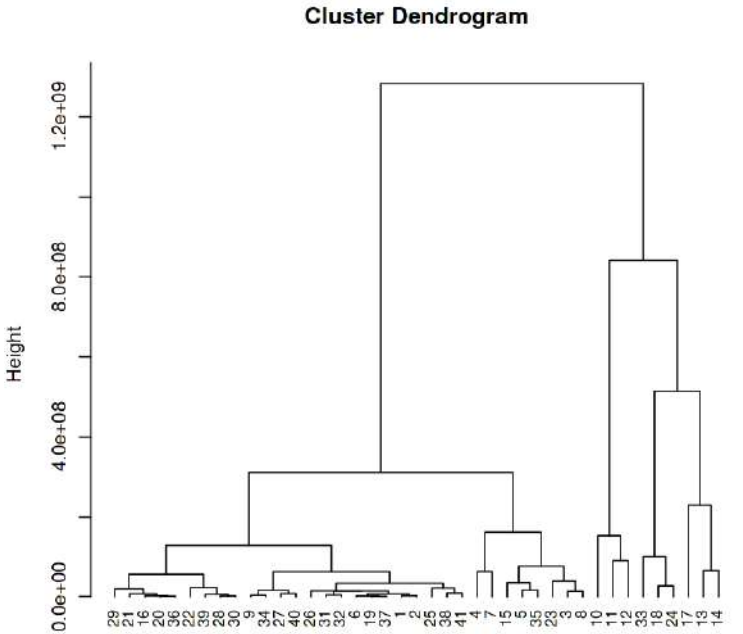
Untuk proses pembagian klaster ke dalam sub group, kita juga dapat melakukan hal yang sama seperti pada teknik single linkage.

3. Klaster data dengan complete linkage

Pengelompokan data jarak menggunakan teknik complete juga sama seperti menggunakan single dan average linkage. Yang dimodifikasi adalah parameter dari **method** diganti dengan "complete". Berikut adalah baris kode untuk melakukan pengelompokan data menggunakan complete linkage:

```
# HAC menggunakan Complete Linkage
hc3 <- hclust(jarak, method = "complete" )

# Menampilkan dendrogram
plot(hc3, cex = 0.8, hang = -1)
```

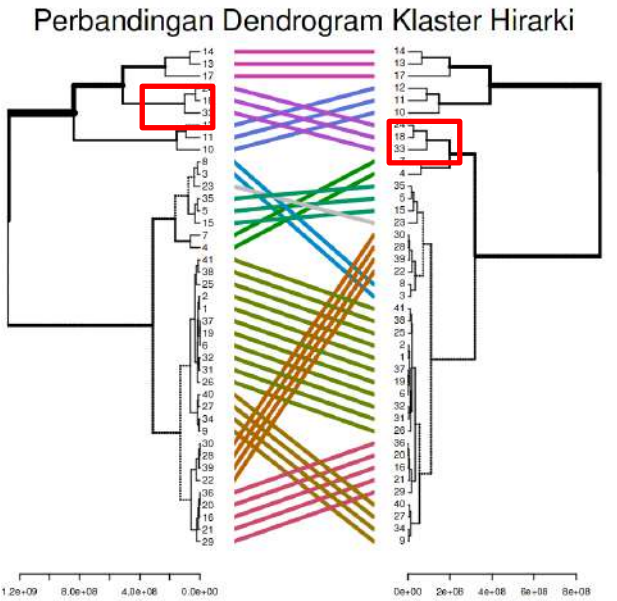
Adapun visualisasi dendrogram untuk baris perintah di atas adalah sebagai berikut :



Kita juga dapat membandingkan hasil pengelompokan HAC dari dua teknik yang berbeda misalnya antara teknik complete dengan average, atau average dengan single dan seterusnya. Fungsinya adalah untuk melihat adakah perubahan posisi anggota klaster ketika dikelompokan dengan dua teknik berbeda. Untuk menampilkan hasil perbandingan dua buah teknik klaster kita dapat menggunakan fungsi **tanglegram**(). Misalnya kita akan membandingkan hasil klaster antara teknik "complete" dengan "average", maka kita tuliskan baris kode sebagai berikut :

```
hca11 =hclust(jarak, method = "complete")
hca21 =hclust(jarak, method = "average")
dendro1 =as.dendrogram (hca11)
dendro2 = as.dendrogram (hca21)
tanglegram(dendro1, dendro2, main = "Perbandingan Dendrogram Klaster Hirarki")
```

Variabel **hca11** menampung hasil klaster untuk teknik "**complete**", sedangkan untuk **hca21** menampung hasil klaster dengan teknik "**average**". Kedua hasil klaster kemudian divisualisaikan ke dalam bentuk dendrogram dan disimpan masing-masing ke dalam variabel **dendro1** dan **dendro2**. Langkah berikutnya adalah memvisualisasikan perbedaan hasil klaster kedua teknik menggunakan fungsi **tanglegram**() sehingga menghasilkan visualisasi sebagai berikut :



Hasil perbandingan klaster menggunakan **tanglegram()** memperlihatkan bagaimana posisi suatu objek bergeser setelah diklasterkan dengan teknik berbeda. Perhatikan group klaster data instance dengan indeks 24, 18, dan 33 seperti pada gambar yang diberi kotak merah, ketika diklasterkan dengan teknik average linkage, posisi nya berpindah seperti pada dendrogram bagian kanan. Begitu seterusnya untuk data objek yang lain.

4. Menentukan teknik HAC Terbaik Untuk Digunakan

Untuk memilih teknik HAC mana yang bisa kita gunakan untuk analisis klaster, kita dapat menghitung koefisien agglomerative untuk setiap teknik HAC. Semakin mendekati +1 maka teknik tersebut semakin baik. Kita bisa menghitung nilai variance dari setiap teknik untuk kemudian menghitung koefisien agglomerative. Untuk mencari nilai koefisien agglomerative kita dapat mengkombinasikan proses

klaster setiap teknik HAC dengan fungsi perhitungan koefisiennya. Berikut adalah baris kode yang bisa digunakan :

```
# Definisikan teknik linkage yang akan dibandingkan
m = c( "average", "single", "complete")
names(m) = c( "average", "single", "complete")

# Buat fungsi untuk menghitung agglomerative coefficient
ac <- function(x) {
  agnes(data[,2:13], method = x)$ac
}

# Hitung agglomerative coefficient untuk setiap teknik linkage HAC
sapply(m, ac)
```

average: 0.967663781391704 **single:** 0.886008173135606 **complete:** 0.975720221605878

Pada fungsi perhitungan agglomerative coefficient, data yang digunakan adalah data awal saat dipanggil. Sementara untuk variabel yang digunakan adalah variabel ke 2 sampai 13 sesuai dengan proses klaster sebelumnya. Hasil perhitungan koefisien agglomerative memperlihatkan bahwa average memiliki koefisien sebesar 0.967, single linkage sebesar 0.886 dan complete linkage sebesar 0.975. Dengan kata lain, berdasarkan perhitungan koefisien agglomerative, pada kasus pembahasan ini, teknik complete linkage lebih baik dibandingkan teknik lainnya.

BAB XIII
EVALUASI MODEL

Pada data mining, model yang terbentuk tentu saja harus dievaluasi untuk mengetahui seberapa prediktif model yang dihasilkan. Selain itu, perlu diketahui bahwa akurasi yang dihasilkan pada data training bukan merupakan indikator yang dapat dijadikan sebagai alat evaluasi kinerja model yang terbentuk. Pada konsep klasifikasi sejumlah data yang besar dibagi ke dalam dua jenis data, yaitu data training dan data testing. Data training yang besar tentu saja akan menghasilkan kinerja model yang lebih baik. Hal ini karena karakteristik data yang dimiliki lebih beragam dibandingkan dengan data training yang berjumlah terbatas. Hal yang sama juga berlaku untuk data testing, semakin besar data testing yang dimiliki untuk diujikan ke dalam model yang terbentuk, semakin akurat estimasi kinerja model yang terbentuk.

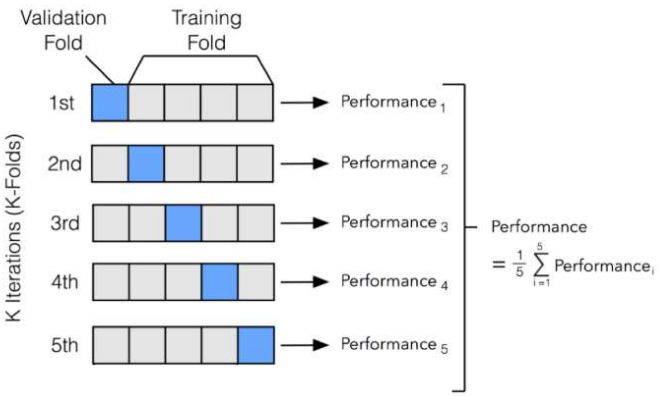
13.1 Prosedur Estimasi Kinerja Model

1. Metode Hold-out

Holdout merupakan metode yang digunakan untuk membagi data secara acak menjadi dua bagian yaitu, data training dan data testing. Secara umum metode ini membagi data dengan perbandingan 2:1 atau 3:2 untuk perbandingan data training dan data testing. Pembagian data training dan data testing sangat mempengaruhi bias setimasi dari model yang dihasilkan. Semakin banyak jumlah instance data testing semakin tinggi nilai bias estimasinya, sebaliknya semakin kecil jumlah instance data testing yang diambil dari data keseluruhan, maka akurasi yang dihasilkan semakin besar.

2. K-Fold Cross validation

Metode K-fold cross validation merupakan teknik yang digunakan untuk membagi data menjadi data training dan data testing dengan membaginya secara acak sebanyak K buah. Untuk membagi data menggunakan K-Fold Cross Validation, tahap pertama adalah dengan membagi data menjadi k bagian dengan ukuran sama, selanjutnya pada tahap kedua gunakan setiap bagian tadi untuk proses testing, sedangkan sisanya sebagai data training. Keunggulan dari metode ini adalah minimnya masalah yang muncul pada saat proses pembagian data. Data keseluruhan hanya akan menjadi data testing sebanyak satu kali dan menjadi data training sebanyak K-1 kali. Oleh karena ada proses pengujian data training sebanyak K kali, maka kelemahan dari algoritma ini adalah dari sisi waktu komputasi, sebab proses pembelajaran harus dilakukan sebanyak K.



3. Bootstrap

Metode Bootstrap merupakan pengukuran akurasi menggunakan sampling dengan proses penggantian untuk membentuk training set.

Sejumlah data yang terdiri dari n instance dilakukan sampling dengan cara menggantinya sebanyak n kali sehingga membentuk data training. Data testing sendiri terbentuk berdasarkan data instance yang tidak terdapat pada data training yang sudah terbentuk. Meski demikian, metode bootstrap ini lebih cocok diimplementasikan untuk data dengan ukuran kecil.

13.2 Pengukuran Estimasi Kinerja Model Supervised

1. Success Rate (SR)

SR merupakan ukuran pengukuran kinerja yang biasanya digunakan pada teknik klasifikasi. Seberapa sukses model yang terbentuk mampu mengklasifikasikan sejumlah data berdasarkan aturan yang dihasilkan dari model yang terbentuk. Ukuran kesuksesan dilihat jika kelas suatu data dapat diprediksi dengan benar, sebaliknya jika kelas suatu data tidak dapat diprediksi dengan benar, maka dikatakan gagal/Error. Untuk mengukur nilai SR digunakan pendekatan confusion matrix sebagai berikut :

		Predicted Class	
		Yes	**No**
Actual Class	**Yes**	True positive (TP)	False negative (FN)
	No	False positive (FP)	True negative (TN)

True Positive Rate = TP / (TP + FN)

False Positive Rate = FP / (FP + TN)

Success Rate = (TP + TN) / (TP + TN + FP + FN)

Error Rate = 1 – Success Rate

2. Kappa Statistic

Kappa statistik merupakan metode yang digunakan untuk membandingkan tingkat akurasi data yang teramati dengan data hasil prediksi. Kappa biasanya digunakan untuk mengevaluasi hasil klasifikasi yang terbentuk. Interval nilai Kappa berkisar antara -1 sampai +1. Jika nilai Kappa = 1 maka sudah dipastikan hasil klasifikasi sangat akurat, semakin menjauhi angka 1 dapat dikatakan bahwa akurasi semakin berkurang. Berikut adalah persamaan untuk menghitung nilai Kappa.

$$Kappa = \frac{observed\ accuracy - expected\ accuracy}{(1 - expected\ accuracy)}$$

Untuk mencari nilai akurasi observed data dapat dihitung berdasarkan ilustrasi sebagai berikut:

Jika diketahui matrik data sebagai berikut :

	X	Y
X	a	b
Y	c	d

Maka nilai akurasi observed data dapat dihitung sebagai berikut :

$$p_0 = \frac{a + d}{a + b + c + d}$$

Sedangkan untuk mencari nilai akurasi expected dapat dihitung dengan persamaan sebagai berikut :

$$p_X = \frac{a + b}{a + b + c + d} x \frac{a + c}{a + b + c + d}$$

$$p_Y = \frac{c+d}{a+b+c+d} x \frac{b+d}{a+b+c+d}$$

Dengan demikian, diperoleh :

$$expected\ accuracy = p_X + p_Y$$

Contoh kasus :

Jika diketahui matrik data observasi dan data predisi hasil klasifikasi sebagai berikut :

	Diterima	Ditolak
Diterima	30	15
Ditolak	20	35

Jika diketahui bahwa kolom merupakan klasifikasi data observasi sedangkan baris menunjukan data prediksi, maka hitunglah nilai Kappa untuk hasil klasifikasi pada tabel di atas.

Penyelesaian :

Diketahui klasifikasi data observasi : Diterima (50), Ditolak (50)
Diketahui klasifikasi data prediksi : Diterima (45), Ditolak (55)
Total data keseluruhan : 100
Observed Accuracy : ((30 + 35) / 100) = 0.65

$$p_{\text{Diterima}} = \frac{30+15}{100} x \frac{30+20}{100} = 0.45\ x\ 0.6 = 0.27$$

$$p_{\text{Ditolak}} = \frac{20+35}{100} x \frac{35+15}{100} = 0.55\ x\ 0.5 = 0.275$$

Expected Accuracy : $p_{\text{Diterima}} + p_{\text{Ditolak}} = 0.27 + 0.275 = 0.54$

Nilai Kappa diperoleh sebagai berikut :

$$Kappa = \frac{observed\ accuracy - expected\ accuracy}{(1 - expected\ accuracy)}$$

$$Kappa = \frac{0.65 - 0.54}{(1 - 0.54)} = \frac{0.11}{0.46} = 0.24$$

3. Mean Squared Error (MSE)

Metode Mean Squared Error biasanya digunakan untuk mengukur kinerja dari masalah prediksi menggunakan regresi. Dapat juga digunakan untuk mengukur kinerja hasil prediksi menggunakan pendekatan lainnya. Ilustrasi perhitungan mean squared error dapat diperlihatkan sebagai berikut:

Nilai-nilai target sebenarnya : $a_1, a_2, ..., a_n$
Nilai-nilai target hasil prediksi : $p_1, p_2, ..., p_n$

dengan demikian nilai *Mean Squared Error* dapat diperoleh sebagai berikut :

$$MSE = \frac{1}{n}\sum_{i=1}^{m}(p_i - a_i)^2 = \frac{(p_1 - a_1)^2 + (p_2 - a_2)^2 + \cdots + (p_i - a_i)^2}{n}$$

Contoh Perhitungan MSE :

No	Hasil Aktual	Prediksi
1	60 a1	62 p1
2	40 a2	38 p2
3	70 a3	65 p3
4	80 a4	84 p4

$$MSE = \frac{(62-60)^2 + (38-40)^2 + (65-70)^2 + (84-80)^2}{4}$$

$$MSE = \frac{(2)^2 + (2)^2 + (-5)^2 + (4)^2}{4}$$

$$MSE = \frac{4+4+25+16}{4} = \frac{49}{4} = 12{,}25$$

4. Root Mean Squared Error (RMSE)

Root mean squared eror (RMSE) merupakan pendekatan yang digunakan untuk mengukur akurasi regresi dan prediksi melalui perhitungan besaran kesalahan rata-rata dalam skala yang sama. Dengan kata lain RMSE adalah nilai MSE yang dikuadratkan. Berikut persamaan untuk mencari nilai RMSE :

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{m}(p_i - a_i)^2} = \sqrt{\frac{(p_1-a_1)^2 + (p_2-a_2)^2 + \cdots + (p_i-a_i)^2}{n}}$$

Contoh Perhitungan RMSE :

No	Hasil Aktual	Prediksi
1	60 a1	62 p1
2	40 a2	38 p2
3	70 a3	65 p3
4	80 a4	84 p4

$$RMSE = \sqrt{\frac{(62-60)^2 + (38-40)^2 + (65-70)^2 + (84-80)^2}{4}}$$

$$RMSE = \sqrt{\frac{(2)^2 + (2)^2 + (-5)^2 + (4)^2}{4}}$$

$$RMSE = \sqrt{\frac{4 + 4 + 25 + 16}{4}} = \sqrt{\frac{49}{4}} = \sqrt{12{,}25} = 3.5$$

5. Mean Absolute Error (MAE)

MAE merupakan alat ukur yang digunakan untuk menguji akurasi dari kebenaran dua buah variabel (prediksi dan pengamatan) yang bersifat kontinyu. Untuk menghitung nilai MAE dapat dilakukan dengan persamaan sebagai berikut :

$$MAE = \frac{1}{n}\sum_{i=1}^{m}|p_i - a_i| = \frac{|p_1 - a_1| + |p_2 - a_2| + \cdots + |p_i - a_i|}{n}$$

Contoh Perhitungan MAE :

No	Hasil Aktual	Prediksi
1	60 a1	62 p1
2	40 a2	38 p2
3	70 a3	65 p3
4	80 a4	84 p4

$$MAE = \frac{|62 - 60| + |38 - 40| + |65 - 70| + |84 - 80|}{4}$$

$$MAE = \frac{|2| + |2| + |-5| + |4|}{4}$$

$$MAE = \frac{2+2+5+4}{4} = \frac{13}{4} = 3.25$$

6. Relative Squared Error (RSE)

RSE merupakan nilai kesalahan relatif yang diperoleh dari nilai kesalahan kuadrat total yang berbanding terbalik dengan kesalahan kuadrat total dari prediktornya. Persamaan untuk mengukur nilai RSE adalah sebagai berikut :

$$RSE = \frac{\sum_{i=1}^{m}(p_i - a_i)^2}{\sum_{i=1}^{m}(\bar{a} - a_i)^2}$$

Contoh Perhitungan RSE :

No	Hasil Aktual	Prediksi
1	60 a1	62 p1
2	40 a2	38 p2
3	70 a3	65 p3
4	80 a4	84 p4
Rata2	62.5	

$$RSE$$

$$= \frac{(62-60)^2 + (38-40)^2 + (65-70)^2 + (84-80)^2}{(62.5-60)^2 + (62.5-40)^2 + (62.5-70)^2 + (62.5-80)^2}$$

$$RSE = \frac{(2)^2 + (2)^2 + (-5)^2 + (4)^2}{(2.5)^2 + (22.5)^2 + (-7.5)^2 + (-17.5)^2}$$

$$RSE = \frac{4+4+25+16}{6.25+506.25+56.25+306.25} = \frac{49}{875} = 0.056$$

7. Relative Absolute Error (RAE)

Pada dasarnya RAE sangat mirip dengan RSE, namun dalah RAE, kesalahan yang dimaksud adalah kesalahan absolut total, bukan kesalahan kuadrat total. Nilai RAE diperoleh dengan membagi nilai kesalahan absolut terhadap total kesalahan absolut prediktornya. Persamaan untuk mengukur RAE adalah sebagai berikut :

$$RAE = \frac{\sum_{i=1}^{n}|p_i - a_i|}{\sum_{i=1}^{n}|\bar{a} - a_i|}$$

Contoh Perhitungan RAE :

No	Hasil Aktual	Prediksi
1	60 a1	62 p1
2	40 a2	38 p2
3	70 a3	65 p3
4	80 a4	84 p4

$$MAE = \frac{|62 - 60| + |38 - 40| + |65 - 70| + |84 - 80|}{|62.5 - 60| + |62.5 - 40| + |62.5 - 70| + |62.5 - 80|}$$

$$MAE = \frac{|2| + |2| + |-5| + |4|}{|2.5| + |22.5| + |-7.5| + |-17.5|}$$

$$MAE = \frac{2 + 2 + 5 + 4}{50} = \frac{13}{50} = 0.26$$

13.3 Pengukuran Estimasi Kinerja Model Unsupervised

Clustering merupakan proses pengelompokan tidak terbimbing dimana setiap kelompok yang terbentuk tidak menunjuk pada label/class kelompok tertentu sebagaimana pada konsep supervised. Oleh karenanya sangat penting untuk melakukan evaluasi terhadap algoritma clustering yang digunakan. Pada clustering sulit untuk menentukan metrik yang sesuai untuk mengukur apakah konfigurasi cluster yang ditetapkan dapat diterima atau tidak. Meskipun demikian beberapa pendekatan untuk validasi pengelompokan dengan teknik clustering telah banyak dikembangkan. Banyak praktisi dan ilmuwan menggunakan indeks validitas (validity indices) untuk mengukur kinerja algoritma clustering.

Indeks validitas dibunakan untuk mengukur "goodness - kebaikan" hasil pengelompokan yang dilakukan, dibandingkan dengan algoritma lainnya, atau dapat juga dengan algoritma yang sama namun dengan konfigurasi parameter yang berbeda. Meski demikian indeks validitas ini cocok untuk mengukur pengelompokan yang bersifat "crisp - tegas", atau pengelompokan dengan partisi/kelompok yang tidak tumpang tindih (misalnya : hierarchical clustering). Adapun beberapa indeks validitas yang dapat digunakan untuk mengukur kinerja algoritma clustering antara lain :

1. Silhouette coefficient

Silhouette coefficient digunakan untuk mengukur seberapa dekat suatu titik dalam satu cluster dengan titik lain di cluster tetangganya. Interval nilai koefisien Silhouette adalah antara -1 sampai +1. Jika nilai koefisien Silhouette bernilai 1 maka cluster terpisah satu sama lain dan dibedakan dengan sangat jelas antara satu cluster dengan cluster lainnya. Jika nilai koefisien Silhouette 0 dapat dikatakan bahwa jarak antar cluster tidak signifikan, dengan kata lain sulit untuk

membedakan apakah titik data masuk di cluster yang tepat atau tidak. Sedangkan jika nilai koefisien Silhouette -1 maka anggota/titik cluster berada pada hasil cluster yang salah. Adapun untuk menghitung koefisien Silhouette dapat dilakukan menggunakan persamaan sebagai berikut :

$$Silhouette_{Idx} = \frac{(b-a)}{\max(a,b)}$$

Dimana *b* merupakan rata-rata jarak intra-cluster yaitu jarak rata-rata antara setiap titik dalam suatu cluster. Sedangkan *a* adalah rata-rata jarak antar cluster yaitu jarak rata-rata antar semua cluster.

2. Indeks Dunn

Indeks Dunn merupakan pengukuran kinerja cluster berdasarkan jarak diameter cluster terkecil dan jarak antar cluster yang besar. Perhitungan indeks Dunn seperti diperlihatkan pada persamaan di bawah ini :

$$D = \min_{i=1\ldots n_c} \left\{ \min_{j=i+1\ldots n_c} \left(\frac{d(c_i, c_j)}{\max_{k=1\ldots n_c} (diam(c_k))} \right) \right\}$$

Dimana,

$$d(c_i, c_j) = \min_{x \in c_i, y \in c_j} \{d(x,y)\}$$

Dan

$$diam(c_k) = \max_{x,y \in c_i} \{d(x,y)\}$$

Keterangan :

n_c : Jumlah Klaster

d : Jumlah Dimensi
d(x,y) : Jarak antara dua elemen data
c_i : Klaster ke – i
c_k : Klaster ke – k
c_j : Klaster ke – j

3. Davies Bouldin Index

Indeks Davies-Bouldin didasarkan pada ukuran kesamaan cluster (R_{ij}) yang basisnya adalah ukuran dispersi / perpindahan cluster (s_i) dan ukuran ketidaksamaan cluster (d_{ij}). Ukuran kesamaan cluster (R_{ij}) dapat didefinisikan secara bebas tetapi harus memenuhi ketentuan sebagai berikut :

- $R_{ij} \geq 0$
- $R_{ij} = R_{ji}$
- Jika $s_i = 0$ dan $s_j = 0$ maka $R_{ij} = 0$
- Jika $s_j > s_k$ dan $d_{ij} = d_{ik}$ maka $R_{ij} > R_{ik}$
- Jika $s_j = s_k$ dan $d_{ij} < d_{ik}$ maka $R_{ij} > R_{ik}$

Dimana R_{ij} diperoleh dari persamaan sebagai berikut :

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

Dimana

$$d_{ij} = d(v_i, v_j)$$

Dan

$$s_i = \frac{1}{\|c_i\|} \sum_{x \in c_i} d(x, v_i)$$

dengan demikian indeks Davies – Bouldin dapat diperoleh dengan persamaan sebagai berikut :

$$DB = \frac{1}{n_c}\sum_{i=1}^{n_c} R_i$$

Dimana

$$R_i = \max_{j=1\ldots n_c, i\neq j}\left(R_{ij}\right)$$

4. RMSSTD (Root Mean Square Standar Deviation) Index

RMSSTD biasanya digunakan untuk mengukur homogenitas cluster, dimana semakin kecil nilai RMSSTD semakin baik homogenitas suatu cluster. Indeks RMSSTD ini selain digunakan untuk mengevaluasi algoritma pengelompokan yang bersifat "crisp - tegas" dapat juga digunakan untuk mengevaluasi algoritma hierarchical clustering. Secara formal nilai RMSSTD diperoleh dengan persamaan sebagai berikut :

$$RMSSTD = \sqrt{\frac{\sum_{j=1\ldots d}^{i=1\ldots n_c}\sum_{k=1}^{n_{ij}}\left(x_k - \bar{x_j}\right)^2}{\sum_{j=1\ldots d}^{i=1\ldots n_c}\left(n_{ij}-1\right)}}$$

5. RS (R Squared)Validity Index

Indeks RS digunakan untuk mengukur ketidaksamaan cluster dengan melakukan pengukuran derajat homogenitas antar kelompok. Nilai Indeks RS antara 0 sampai 1. Jika Indeks RS bernilai 0 maka tidak ada perbedaan antar cluster, dan jika bernilai 1 maka terdapat

perbedaan yang signifikan antar cluster. Adapun perhitungan indeks RS seperti diperlihatkan pada persamaan berikut :

$$RS = \frac{SS_t - SS_w}{SS_t}$$

Dimana

$$SS_t = \sum_{j=1}^{d} \sum_{k=1}^{n_j} \left(x_k - \overline{x_j}\right)^2$$

Dan

$$SS_w = \sum_{j=1\ldots d}^{i=1\ldots n_c} \sum_{k=1}^{n_{ij}} \left(x_k - \overline{x_j}\right)^2$$

6. SD Validity Index

Dasar dari indeks validitas SD adalah nilai rata-rata sebaran cluster dan total pemisahan cluster. Nilai sebaran dihitung dengan varians cluster dan varians dari datasetnya untuk mengukur homogenitas clusternya. Adapun varians dari data set dan cluster diperoleh dengan persamaan sebagai berikut:

- Varians dataset

$$\sigma_x^p = \frac{1}{n} \sum_{k=1}^{n} \left(x_k^p - \overline{x^p}\right)^2$$

$$\sigma(x) = \begin{bmatrix} \sigma_x^1 \\ \vdots \\ \sigma_x^d \end{bmatrix}$$

- Varians cluster

$$\sigma_{v_i}^{p} = \frac{1}{\|c_i\|}\sum_{k=1}^{n}\left(x_k^p - \overline{v_i^p}\right)^2$$

$$\sigma(v_i) = \begin{bmatrix}\sigma_{v_i}^{1}\\ \vdots \\ \sigma_{v_i}^{d}\end{bmatrix}$$

Adapun perhitungan nilai persebaran cluster dapat dihitung dengan persamaan sebagai berikut :

$$Scatt = \frac{1}{n_c}\sum_{i=1}^{n_c}\frac{\|\sigma(v_i)\|}{\|\sigma(x)\|}$$

Adapun total pemisahan cluster berdasarkan jarak titik pusat cluster dapat dihitung dengan mengukur cluster separation sebagai berikut :

$$Diss = \frac{\max\limits_{i,j=1\ldots n_c}\left(\|v_j - v_i\|\right)}{\min\limits_{i,j=1\ldots n_c}\left(\|v_j - v_i\|\right)}\sum_{k=1}^{n_c}\left(\sum_{j=1,j\neq 0}^{n_c}\left(\|v_j - v_i\|\right)\right)^{-1}$$

Dengan demikian, nilai indek SD dapat diperoleh dengan persamaan sebagai berikut :

$$SD = \alpha.Scatt + Diss$$

dimana α adalah faktor pembobotan yang sama dengan parameter Dis jika jumlah cluster maksimum. Indeks SD yang lebih rendah

berarti konfigurasi cluster yang lebih baik karena dalam hal ini cluster kompak dan terpisah.

7. S_Dbw Validity Index

Indeks S-Dbw merupakan teknik validasi cluster dengan cara mengukur varias dari intra-cluster dan antar-cluster. Varian intra cluster sendiri digunakan untuk mengukur rata-rata sebaran cluster. Adapun persamaan yang digunakan untuk mengukur varians intra-cluster adalah sebagai berikut:

$$Dens_{bw} = \frac{1}{n_c(n_c-1)} \sum_{i=1}^{n_c} \left(\sum_{j=1, i \neq j}^{n_c} \frac{density(u_{ij})}{max\{density(v_i), density(v_j)\}} \right)$$

dimana u_{ij} adalah titik tengah dari ruas garis yang ditentukan oleh pusat cluster v_i dan v_j. Fungsi kerapatan di sekitar titik didefinisikan sebagai berikut: fungsi ini menghitung jumlah titik dalam hyper-sphere yang jari-jarinya sama dengan deviasi standar rata-rata cluster. Adapaun nilai deviasi standar rata-rata cluster dapat dihitung dengan persamaan sebagai berikut :

$$stdev = \frac{1}{n_c} \sqrt{\sum_{i=1}^{n} \|\sigma(v_i)\|}$$

Oleh karenanya, nilai indeks S_Dbw kemudian dapat dihitung dengan persamaan sebagai berikut :

$$S_{Dbw} = Scatt + Dens_{bw}$$

Sedangkan nilai *Scatt* dari tahapan perhitungan indeks SD Validity. Dengan ketentuan bahwa semakin kecil nilai indeks *S_Dbw* maka hasil pengelompokan dinyatakan lebih baik.

DAFTAR PUSTAKA

Baesens, B., Roesch, D., & Scheule, H. (2016). Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS. John Wiley & Sons.

Cios, K.J., Pedrycz, W., Swiniarski, R.W., dan Kurgan, L.A., 2007, *Data Mining : A Knowledge Discovery Approach*, Springer Science+Business Media, LLC

Dunn, J. C., 1974, Well Separated Clusters and Optimal Fuzzy Partitions, *Journal of Cybernetica*, Vol. 4, pp. 95-104

E.M. Mirkes, *K-means and K-medoids applet*, University of Leicester, 2011, available at:http://www.math.le.ac.uk/people/ag153/homepage/Kmeans Kmedoids/Kmeans_Kmedoids.html

Emery, A., K., 2014, How to Visualize Qualitative Data, avalilable at https://depictdatastudio.com/how-to-visualize-qualitative-data/, akses tanggal 19 Maret 2021

Halkidi M., dan Vazirgiannis, M., 2001, Clustering Validity Assessment: Finding the Optimal Partitioning of a Data Set, *Proc. of ICDM 2001*, pp. 187-194

Halkidi, M., Batistakis, Y., dan Vazirgiannis, M., 2002, Cluster validity methods: part I, *SIGMOD Rec*., Vol. 31, No. 2, pp. 40-45

Halkidi, M., Batistakis, Y., dan Vazirgiannis, M., 2002, Cluster validity methods: part II, *SIGMOD Rec*., Vol. 31, No. 2, pp. 19-27

Han, J., Kamber, M., dan Pei, J., 2012, *Data Mining : Concept and Techniques*, Edisi Ketiga, Morgan Kaufmann Publishers, USA

https://majkamichal.github.io/naivebayes/reference/naive_bayes.html

https://www.kdnuggets.com/datasets/index.html

Kovács, F., Legány, C., dan Babos, A., *Cluster Validity Measurement Techniques*, Budapest University of Technology and Economics

Larose, D.T., 2005, *Discovering Knowledge in Data : An Introduction to Data Mining*, John Wiley & Sons, Inc., Canada

Larose, D.T., 2006, *Data Mining : Methods and Models*, John Wiley & Sons, Inc. Publication, New Jersey – Canada

Matias, M., 2021, Visualize It! A Comprehensive Guide to Data VisualizationVisualize It!, available at www.netquest.com, akses tanggal 19 Maret 2021

Olson, D.L dan Delen, D., 2008, *Advanced Data Mining Techniques*, Springer-Verlag Berlin Heidelberg

Pramesti, D.F., Furqon, M.T., dan Dewi, C., 2017, Implementasi Metode K-Medoids Clustering Untuk Pengelompokan Data Potensi Kebakaran Hutan/Lahan Berdasarkan Persebaran Titik Panas (Hotspot), *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 1, No. 9, Juni 2017, hlm. 723-732

Robbins, N., 2012, A Histogram is NOT a Bar Chart, available at https://www.forbes.com/sites/naomirobbins/2012/01/04/a-histogram-is-not-a-bar-chart/?sh=1d0318cf6d77, akses tanggal 18 Maret 2021

sas.com, Data Visualization : What it is and why it matters, available at https://www.sas.com/en_us/insights/big-data/data-visualization.html, akses tanggal 18 Maret 2021

Sharma, S., 1996, *Applied Multivariate Techniques*, John Wiley & Sons, Inc.,

tableau.com, Data visualization beginner's guide: a definition, examples, and learning resources, available at https://www.tableau.com/, akses tanggal 18 Maret 2021

Yi, M., 2019, A Complete Guide to Scatter Plots, available at https://chartio.com/learn/charts/what-is-a-scatter-plot/, akses tanggal 19 Maret 2021

Buku ini menjabarkan teori dasar dari data mining beserta model yang bisa dipilih untuk melakukan penambangan data. Buku ini juga dilengkai dengan konsep dasar visualisasi data yang mungkin diperlukan saat melakukan analisis. Beberapa teknik yang ada pada pembelajaran mesin (machine learning) dibahas untuk melakukan tugas-tugas data mining seperti analisis regresi, klasifikasi dan klastering data. Pada bab akhir dibahas teknik evaluasi model klasifikasi dan uji vaiditas hasil klaster yang dapat digunakan. Yang menarik dari buku ini adalah, setiap pembahasan tugas data mining menggunakan teknik pembelajaran mesin, disertai dengan teori dasar dilanjutkan implementasinya menggunakan R programming.

Secara umum buku ini memuat pengantar data mining, proses penemuan pengetahuan, pre-processing data, visualisasi data, pengenalan paltform Kaggle, analisis regresi, teknik pembelajaran terbimbing (supervised learning), teknik pembelajaran tidak terbimbing (unsupervised learning), evaluasi model, serta materi praktik menggunakan bahasa R pada platform Kaggle.

Tb Ai Munandar lahir di Pandeglang Provinsi Banten. Penulis menyelesaikan Pendidikan Dasar hingga menengah atas di Kabupaten Pandeglang (SDN Cening II, MTs Mathla'ul Anwar Pusat Menes dan SMA N 1 Pandeglang). Kemudian melanjutkan studi ke jenjang perguruan tinggi di Yogyakarta (S1 – Teknik Informatika Universitas PGRI Yogtakarta, S2 – Teknik Informatika Universitas Atma Jaya Yogyakarta dan S3 – Ilmu Komputer Universitas Gadjah Mada). Saat ini penulis tercatat sebagai dosen di program studi Informatika, Universitas Bhayangkara Jakarta Raya.

PT. BALE DAMAR PUBLISHING
Jl. KH. Syuhada Perum Griya Lestari Blok B.i No. 17
Kab. Serang, Provinsi Banten
Website www.bd-publishing.store
Email : info@bd-publishing.store

ISBN 978-623-09-1542-0
9 786230 915420