BAB II
TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Beberapa penelitian terkait dengan menggunakan teknik Data Mining metode klasifikasi untuk memprediksi penyakit stroke dapat dilihat pada tabel 2.

Tabel 2.1. Berbagai Teknik Data Mining yang digunakan dalam prediksi penyakit stroke

Judul	Penulis	Tahun	Metode	Hasil
Klasifikasi Tingkat Risiko Penyakit Stroke Menggunakan Metode *GA-Fuzzy Tsukamoto*	Vina Adelinal	2018	*GA-Fuzzy Tsukamoto*	Diketahui bahwa akurasi sistem menggunakan metode *Fuzzy Tsukamoto-GA* dari hasil dilakukannya pengoptimalan batasan fungsi keanggotaan adalah 86.66% yang didapatkan
Penerapan Metode Adaboost Untuk Mengoptimasi Prediksi Penyakit Stroke Dengan *Algoritma Naïve Bayes*	Agus Byna	2020	Metode *Adaboost* Dengan *Algoritma Naïve Bayes*	Hasil pengujian yang pertama menggunakan *Naive Bayes* dengan data split 80/20 memberikan akurasi 0,976 memiliki diagnosis klasifikasi yang sangat baik. Setelah di optimasi dengan *Adaboost* dengan data split 70/30 memberikan akurasi 0,981.
Perbandingan Klasifikasi Metode *Naive Bayes* dan Metode *Decision Tree Algoritma (J48)* pada Pasien Penderita Penyakit	Irene Lishania	2019	Metode *Naive Bayes* dan Metode *Decision Tree*	Dari 12 atribut yang terdapat dalam dataset yaitu Jenis, kelamin, tekanan darah, diabetes melitus, dislipidemia, kadar asam urat, penyakit jantung, status stroke. Penelitian ini

Stroke di RSUD Abdul Wahab Sjahranie Samarinda				menggunakan perbandingan dengan metode naive Bayes adalah 81,25% dan metode decision tree algoritma (J48) diperoleh tingkat akurasi sebesar 87,5%.
Komparasi Penerapan Metode *Bagging* dan *Adaboost* pada Algoritma C4.5 untuk Prediksi Penyakit Stroke	Nur Diana Saputri	2022	Metode *bagging* Dan *adaboost* pada *algoritma C4.5*	Hasil pengujian klasifikasi menggunakan *Confusion Matrix* dan *k-fold cross validation* untuk algoritma C4.5 menghasilkan akurasi sebesar 92.87%. Kemudian hasil akurasi dari algoritma C4.5 dengan metode bagging meningkat menjadi 95.02% dan ketika dikombinasikan dengan metode *Adaboost* nilai akurasinya juga meningkat menjadi 94.63%.
Klasifikasi Penyakit Stroke Menggunakan Algoritma *K-Nearest Neighbor (KNN)*	Zuriati Z1	2023	Algoritma *K-Nearest Neighbor (KNN)*	Penerapan Algoritma KNN dan Evaluasi kinerja KNN dengan confusion matrix dan penghitungan akurasi. Performa algoritma KNN terbaik didapatkan dengan nilai k=5 dan akurasi 93.54%
Stroke Prediction Using Machine Learning Classification Methods	Hamza Al-Zubaidi, et al	2022	Machine Learning Classification	beberapa model diproduksi menggunakan algoritma yang berbeda (pengklasifikasi), tetapi model yang menghasilkan akurasi, presisi, recall, dan skor F1 terbaik sebesar 94%-95% didasarkan pada pengklasifikasi Random Forest.

A comparative analysis of machine learning classifiers for stroke prediction: A predictive analytics approach	Nitish Biswas, et al	2022	Support Vector Machine, Random Forest, K-nearest Neighbor, Decision Tree, Naïve Bayes, Voting Classifier, AdaBoost, Gradient Boosting, Multi-Layer Perception, and Nearest Centroid	Hasilnya menunjukkan Support Vector Machine memiliki akurasi tertinggi sebesar 99,99%, dengan nilai recall sebesar 99,99%, nilai precision sebesar 99,99%, dan F1-measure sebesar 99,99%. Random Forest mencapai akurasi tertinggi kedua sebesar 99,87%, dengan kesalahan sebesar 0,001%.

Dari hasil review beberapa jurnal diatas baik jurnal nasional dan internasional maka dapat disimpulkan untuk mendapatkan akurasi yang baik maka data yang dibutuhkan pada penelitian yaitu minimal 500 record data, dan metode yang digunakan untuk beberapa jurnal yang sudah di review tingkat akurasinya cenderung lebih tinggi menggunakan algoritma *K-Nearest Neighbor* dan *Optimasi* Teknik *Bagging*

Tabel 2. 2 Hasil Akurasi Penelitian Sebelumnya

No	Jurnal	Algoritma	Akurasi
1	Vina Adelinal	*GA-Fuzzy Tsukamoto*	86.66%
2	Agus Byna	Metode *Adaboost* Dengan *Algorit ma Naïve Bayes*	*Naive Bayes 0.976 Adaboost 0.981*
3	Irene Lishania	Metode *Naive Bayes* dan Metode *Decision Tree*	*Naive Bayes* 81.25% *Decision tree* 87.05%
4	Nur Diana Saputri	Metode *bagging* Dan *adaboost* pada *algoritma C4.5*	Algoritma C4.5 92.87 % Metode *bagging* 95.02%
5	Zuriati Z1	Algoritma *K-Nearest Neighbor (KNN)*	93.54%
6	Hamza Al-Zubaidi, et al	Stroke Prediction Using Machine Learning Classification Methods	Menghasilkan akurasi, presisi, recall, dan skor F1 terbaik sebesar 94%-95%
7	Nitish Biswas, et al	A comparative analysis of machine learning classifiers for stroke prediction: A predictive analytics approach	Nitish Biswas, et al

2.2. Landasan Teori

2.2.1. Penyakit Stroke

Stroke adalah penyakit serebrovaskuler (pembuluh darah otak) yang ditandai dengan kematian jaringan otak (infarkserebral) yang terjadi karena berkurangnya aliran darah dan oksigen ke otak. Berkurangnya aliran darah dan oksigen ini bisa dikarenakan adanya sumbatan, penyempitan atau pecahnya pembuluh darah [5] Stroke merupakan masalah kesehatan global dan salah satu penyebab utama

kecacatan orang dewasa [6]. Stroke terjadi bila aliran darah ke berbagai area otak terganggu atau berkurang. Sel-sel di area otak tersebut tidak mendapatkan nutrisi dan oksigen sehingga mengakibatkan kematian sel. Deteksi awal dan penanganan yang tepat diperlukan untuk meminimalkan kerusakan lebih lanjut di area otak yang terkena dan komplikasi pada bagian tubuh lainnya [6].

Menurut WHO stroke adalah adanya tanda-tanda klinik yang berkembang cepat akibat gangguan fungsi otak fokal (atau global) dengan gejala-gejala yang berlangsung selama 24 jam atau lebih yang menyebabkan kematian tanpa adanya penyebab lain yang jelas selain vaskuler [6]. Stroke dibedakan menjadi dua, yaitu:

a) Stroke Ischemic

   Stroke ischemic terjadi karena adanya sumbatan pembuluh darah oleh thromboembolic yang mengakibatkan daerah di bawah sumbatan tersebut mengalami ischemic [7] Stroke iskemik disebabkan karena tersumbatnya arteri servikal atau serebral mengakibatkan matinya jaringan otak karena aliran darah terganggu di bagian otak.5 Aterosklerosis merupakan salah satu faktor resiko penyebab stroke iskemik, aterosklerosis ditandai dengan adanya penebalan dinding arteri akibat penyumbatan kolesterol di tunika intima. Penumpukan trombus, hiperkolesterolemia, dan radikal bebas menjadi penyebab aterosklerosis.

b) Stroke Hemoragik yang terjadi akibat adanya mycroaneurisme yang pecah atau pecahnya pembuluh darah dalam otak yang menyebabkan perdarahan dan terhentinya asupan nutrisi dan oksigen pada area tertentu di dalam otak.

Kondisi ini selanjutnya akan merusak sel-sel dan jaringan otak [7]. Stroke hemoragik memiliki morbiditas dan mortalitas yang lebih beresiko dibanding stroke iskemik. [12] Stroke hemoragik memiliki 2 tipe, yang pertama adalah perdarahan intraserebral (ICS) yang merupakan perdarahan yang bukan disebabkan oleh trauma tetapi pada pembuluh darah bagian parenkim otak mengalami perdarahan. Tipe yang kedua adalah subarachnoid hemorrhage (SAH) yang merupakan keadaan akut karena terjadi perdarahan diluar pembuluh darah otak, dimana pecahnya pembuluh darah disekitar permungkaan otak.

a. Faktor-faktor Penyakit Stroke

Faktor resiko stroke adalah kondisi atau penyakit atau kelainan yang terdapat pada seseorang yang memiliki potensi untuk memudahkan orang tersebut mengalami serangan stroke pada suatu saat. Jika seseorang terdapat faktor-faktor risiko untuk terjadinya serangan stroke disebut sebagai stroke prone profile. Terdapat dua macam faktor risiko penyakit stroke, yaitu faktor risiko yang dapat diubah atau dikendalikan dan faktor risiko yang tidak dapat diubah atau dikendalikan. Faktor risiko yang tidak dapat diubah atau dikendalikan meliputi: [7]

1. Usia

   Stroke sering terjadi pada orang yang telah lanjut usia (tua). Setiap penambahan 10 tahun setelah usia 55 tahun, terdapat peningkatan risiko penyakit stroke sebanyak dua kali lipat.

2. Jenis Kelamin

Stroke lebih mungkin pada pria dibandingkan pada wanita. Namun, lebih dari separuh kematian stroke total yang terjadi pada wanita. Penggunaan pil KB dan kehamilan meningkatkan risiko stroke bagi perempuan.

3. Ras

   Kematian akibat penyakit stroke lebih banyak terjadi pada orang Afrika, Amerika daripada orang kulit putih. Hal ini dikarenakan mereka mempunyai risiko lebih tinggi menderita tekanan darah tinggi, diabetes, dan obesitas. Sedangkan faktor risiko yang dapat diubah atau dikendalikan

4. Tekanan Darah

   Tekanan darah adalah tekanan yang terjadi pada pembuluh darah arteri saat darah dipompa oleh jantung untuk dialirkan ke seluruh tubuh. Tekanan darah sistolik adalah tekanan darah yang terjadi pada saat otot jantung berkontrakasi. Sedangkan tekanan darah diastolik adalah tekanan darah yang terjadi pada saat otot jantung beristirahat atau tidak sedang berkontraksi

5. Kadar Gula Darah

   Gula darah adalah bahan bakar tubuh yang dibutuhkan untuk kerja otak, sistem saraf, dan jaringan tubuh yang lain. Gula darah yang terdapat di dalam tubuh dihasilkan oleh makanan yang mengandung karbohidrat, 14 protein, dan lemak. Rata-rata, kadar gula darah normal adalah sebagai berikut:

   a. Gula darah 8 jam sebelum makan atau setelah bangun pagi (70- 110 mg/dl).

b. Gula darah 2 jam setelah makan (100-150 mg/dl).

c. Gula darah acak (70-125 mg/dl).

6. Kadar Kolesterol

   Total Kolesterol total merupakan kadar keseluruhan kolesterol yang beredar dalam tubuh manusia. Kolesterol adalah lipid amfipatik dan merupakan kompenen struktural esensial pada membran plasma. Senyawa kolesterol total ini disintesis dibanyak jaringan dari asetil-KoA dan merupakan prekusor utama semua steroid lain di dalam tubuh termasuk kortikosteroid, hormone seks, asam empedu, dan vitamin D.

7. Low Density Lipoprotein (LDL)

   Kolesterol LDL disebut sebagai kolesterol jahat disebabkan peranannya membawa kolesterol total ke banyak jaringan di dalam tubuh. Sehingga memberikan peluang terjadinya penumpukan kolesterol di berbagai jaringan tubuh, termasuk diantaranya dalam pembuluh darah.

8. Asam Urat

   Penyakit asam urat adalah penyakit yang timbul akibat kadar asam urat darah yang berlebihan. Yang menyebabkan kadar asam urat darah 15 berlebihan adalah produksi asam urat di dalam tubuh lebih banyak dari pembuangannya. Organ yang bisa terserang adalah sendi, otot, jaringan di sekitar sendi, telinga, kelopak mata, jantung, ginjal, dan lain-lain.

9. Blood Urea Nitrogen (BUN)

   Blood Urea Nitrogen (BUN) dapat didefinisikan sebagai jumlah nitrogen urea yang hadir dalam darah. Urea adalah produk limbah yang dibentuk

dalam tubuh selama proses pemecahan protein. Selama metabolisme protein, protein diubah menjadi asam amino yang juga menghasilkan amonia. Urea tidak lain adalah substansi yang dibentuk oleh beberapa molekul amonia. Metabolisme protein berlangsung dalam hati dan dengan demikian urea juga diproduksi oleh hati. Selanjutnya, urea ditransfer ke ginjal melalui aliran darah dan dikeluarkan dari tubuh dalam bentuk urin. Dengan demikian, setiap disfungsi ginjal akan menyebabkan kadar tinggi atau rendah BUN dalam darah.

10. Kreatinin (*creatinine*)

    Kreatinin adalah produk penguraian dari kreatin fosfat dalam metabolisme otot dan dihasilkan dari kreatin (creatine). Kreatinin pada dasarnya merupakan limbah kimia yang selanjutnya diangkut ke ginjal melalui aliran darah untuk dikeluarkan melalui urin. Kadar kreatinin dapat diukur dalam urin serta darah. Tingkat kreatinin dalam darah umumnya tetap normal. Dengan demikian, ginjal yang berfungsi normal juga akan menunjukkan tingkat normal kreatinin dalam darah. Tapi ketika ginjal tidak berfungsi dengan baik, jumlah kreatinin dalam darah akan meningkat.

2.2.2. Data Mining

Data mining adalah bidang untuk memeriksa database yang sudah ada sebelumnya untuk menggali informasi baru. Bidang ini membentuk dasar Analisa dan digunakan untuk membuat prediksi untuk berbagai bidang seperti pemasaran,

keuangan, Medis, cuaca, dll. Sering kali, istilah penambangan data ditafsirkan untuk menemukan data yang relevan dari kumpulan data yang sudah ada sebelumnya, itu adalah ekstraksi informasi atau pola yang relevan untuk membuat prediksi. Data mining adalah bidang yang pasti digunakan di bidang Medis [8]. Dalam data mining, tugas dapat dikategorikan menjadi dua jenis yaitu Prediktif dan Deskriptif.

1. Prediktif

   Dengan bantuan pendekatan ini, pengguna dapat membuat prediksi tentang nilai data yang digunakan dalam berbagai database dengan bantuan hasil yang sudah diketahui dari beberapa data yang berbeda atau berdasarkan data historis. Prediktif dapat dicirikan lebih lanjut menjadi empat bagian lain yang tercantum di bawah ini:

   a. Klasifikasi : Pada dasarnya bertanggung jawab untuk menemukan fungsi yang mengklasifikasikan item data ke dalam salah satu dari beberapa kelas yang telah ditentukan sebelumnya. Ini dapat dipahami lebih lanjut seperti ini:
   b. Regresi : Regresi adalah fungsi data mining yang digunakan untuk memprediksi angka. Misalnya, model regresi dapat digunakan untuk memprediksi nilai barang antik tertentu berdasarkan tampilannya, berapa banyak penambahan atau penghapusan yang dilakukan, kondisi penyimpanannya, dan faktor lainnya. Umumnya, tugas regresi dimulai dengan sekumpulan data di mana nilai target sudah dikenal atau dipelajari.

c. Analisis Deret Waktu : Deret waktu dapat didefinisikan sebagai rangkaian atau rangkaian titik data yang diturunkan pada titik waktu yang berbeda. Ini biasanya diturunkan dalam interval waktu yang teratur seperti detik, jam, hari, bulan, tahun, dan seterusnya.

d. Prediksi : Prediksi adalah teknik data mining yang dapat digunakan untuk mengekstrak model dan mewakili kelas data untuk memprediksi tren data masa depan. Prediksi dapat didefinisikan sebagai output dari suatu algoritma setelah diinstruksikan untuk beroperasi pada kumpulan data historis dan kemudian diterapkan pada data baru. Misalnya, meramalkan cuaca adalah contoh prediksi data mining yang paling populer. Dengan kata lain, itu juga dapat didefinisikan sebagai ramalan atau ramalan. Contoh lain dari prediksi bisa menjadi paranormal yang memberi tahu seseorang tentang masa depannya.

2. Deskriptif

Jenis tugas data mining yang kedua adalah tugas Deskriptif. Jenis ini mencakup fungsi-fungsi berikut:

a. Aturan Asosiasi : Dalam data mining, aturan asosiasi dapat digunakan untuk mengungkap asosiasi atau koneksi di antara berbagai set item yang berbeda. Asosiasi juga dapat digunakan untuk mendapatkan hubungan antara objek yang berbeda.

b. Pengelompokan : Dalam data mining, proses clustering dapat digunakan untuk mendapatkan objek data yang memiliki beberapa kesamaan. Kesamaan tersebut dapat dimanifestasikan atas dasar faktor yang berbeda

seperti perilaku pembelian, daya tanggap terhadap tindakan tertentu, lokasi geografis dan sebagainya. Dengan mengelompokkan informasi ini, akan sangat membantu untuk memahami pelanggan dengan lebih baik dan akibatnya memberikan layanan yang lebih baik kepada pelanggan.

c. Ringkasan : Peringkasan dapat didefinisikan sebagai proses generalisasi data. Dalam proses ini, sekumpulan data yang relevan dan penting diringkas yang kemudian menghasilkan sekumpulan data yang lebih kecil yang memberikan informasi yang terkumpul dari data tersebut. Informasi yang diringkas ini dapat berguna untuk penjualan atau hubungan pelanggan.

d. Penemuan Urutan: Sequence discovery atau sequential pattern mining, adalah teknik data mining yang digunakan untuk menemukan pola yang relevan dan penting dalam data sekuensial. Program penambangan ini menilai kriteria tertentu yang merupakan frekuensi kemunculan, durasi, atau nilai dalam satu set urutan untuk menemukan pola tersembunyi atau tersembunyi.

e. Model prediktif bekerja dengan membuat prediksi tentang nilai data, yang menggunakan hasil yang diketahui yang ditemukan dari kumpulan data yang berbeda. Tugas-tugas yang termasuk dalam model data mining prediktif meliputi klasifikasi, prediksi, regresi dan analisis deret waktu. Model datamining prediktif memprediksi hasil masa depan berdasarkan catatan masa lalu yang ada dalam database atau dengan jawaban yang diketahui. Model deskriptif sebagian besar mengidentifikasi pola atau

hubungan dalam kumpulan data. Ini berfungsi untuk mengeksplorasi properti dari data yang diperiksa sebelumnya dan bukan untuk memprediksi properti baru.

2.2.3. Klasifikasi

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi. Pengertian Prediksi sama dengan ramalan atau perkiraan. Menurut kamus besar bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan nilai pada masa yang akan datang dengan menggunakan data masa lalu. Prediksi menunjukkan apa yang akan terjadi pada suatu keadaan tertentu dan merupakan input bagi proses perencanaan dan pengambilan keputusan.

Prediksi bisa berdasarkan metode ilmiah ataupun subjektif belaka. Sebagai contoh, prediksi cuaca selalu berdasarkan data dan informasi terbaru yang didasarkan pengamatan termasuk oleh satelit. Begitupun prediksi gempa, gunung meletus ataupun bencana secara umum. Namun, prediksi seperti pertandingan sepakbola, olahraga, dll umumnya berdasarkan pandangan subjektif dengan sudut pandang sendiri yang memprediksinya. Permulaan awal, walaupun pengkajian yang mendalam mengenai alternatif masa depan adalah suatu disiplin baru,

barangkali orang telah menaruh perhatian besar tentang apa yang akan terjadi kemudian semenjak manusia mulai mengetahui sesuatu. Populasi para peramal pada zaman kuno dan abad pertengahan merupakan satu manifestasi dari keinginan tahu orang tentang masa depannya. Perhatian tentang masa depan ini berlangsung terus bahkan berkembang menjadi kolom astrologi yang disindikatkan pada tahun 1973.

Secara Eksplisit, pembahasan mengenai teori peramalan kebijakan sangatlah sedikit. Namun, secara implisit, peramalan kebijakan terkait menjadi satu dengan 6-7 proses analisa kebijakan. Karena didalam menganalisa kebijakan, untuk menformulasikan sebuah rekomendasi kebijakan baru, maka diperlukan adanya peramalan-peramalan atau prediksi mengenai kebijakan yang akan diberlakukan dimasa yang akan datang. Namun, satu dari sekian banyak prosedur yang ditawarkan oleh para pakar Dunn, masih memberikan pembahasan tersendiri mengenai peramalan kebijakan. Menurut Dunn, Peramalan Kebijakan (policy forecasting) merupakan suatu prosedur untuk membuat informasi factual tentang situasi social masa depan atas dasar informasi yang telah ada tentang masalah kebijakan.

2.2.4. K-Nearst Neighboard

Nearest Neighbor atau k-Nearest Neighbor (kNN) merupakan salah satu algoritme klasifikasi dalam data mining yang memanfaatkan data terdekat untuk melakukan prediksi pada data baru yang belum dikenal (data uji). [6] Algoritme ini bekerja dengan cara mencari sejumlah tetangga terdekat dari data uji dan

menentukan kelas data uji tersebut berdasarkan mayoritas kelas dari tetangga terdekat (data latih) yang ditemukan. Nearest Neighbor dapat digunakan untuk menangani berbagai jenis data, baik data numerik maupun kategorikal. Pada data kategorikal, perhitungan jarak perbedaan atau kesamaan tidak dapat dihitung menggunakan operasi matematik seperti yang dapat dilakukan pada data numerik. Diberikan dataset pelatihan D dan jarak ukuran:

a. *(xi, yi), i = 1, 2, …, N*

b. xi adalah data pelatihan dalam R^n

c. *yi* adalah kelas yang sesuai dari data *xi,* dan *yi {cj, j 1, 2, ... M}*

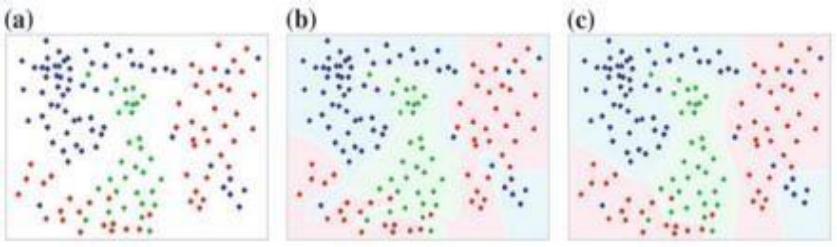
d. *dist (x - xi)*=|| *x - xi* ||

Data pengamatan baru x diklasifikasikan ke dalam salah satu kelas yj menggunakan algoritma berikut :

1. Masukkan data baru x
2. Hitung jarak x ke semua sampel pelatihan xi dalam kumpulan data: *dist (x - xi)*
3. Urutkan *dist (x - xi) (i = 1, 2, …, N)* dalam urutan menaik dan urutkan semua xi sesuai dengan: *xr1, xr2, …, xrk, …, xrN*
4. Untuk klasifikasi tetangga terdekat (NN) mengklasifikasikan *x* ke *yrl*
5. Untuk klasifikasi *K-NN*, klasifikasikan *x* ke kelas mayoritas *yrp* di antara data peringkat k teratas: { *xr1, xr2, …, xrk,* }.

Meskipun Euclidean (*L2*) dan jarak blok kota (*L1*) adalah pilihan tipikal

untuk ukuran jarak, jarak lain dapat digunakan tergantung pada aplikasinya. Tetangga terdekat (NN atau 1-NN) menghasilkan terlalu banyak kelas, sedangkan *K-NN* memberikan hasil klasifikasi yang lebih andal. Hal ini dikarenakan nilai k memiliki efek smoothing yang membuat classifier lebih tahan terhadap outlier. Namun, kinerja pengklasifikasi *K-NN* tergantung pada pilihan k yang biasanya ditentukan secara empiris. Gambar 2.2 menunjukkan perbandingan antara *classifier NN* dan *classifier K-NN*) [10]. Hal ini dapat dilihat dari dua hasil klasifikasi, pada kasus pengklasifikasi NN (setelah penggabungan), titik data outlier membuat pulau-pulau kecil dalam suatu kelas (misalnya, titik merah dalam kelas hijau) dan sudut tajam pada batas kelas, pulau-pulau, dan sudut-sudut tajam kemungkinan mengarah pada prediksi yang salah untuk lebih jelasnya bisa melihat gambar 2.1.



Gambar 2.1 Perbandingan antara NN dan K-NN

Pengklasifikasi menghaluskan outlier ini, yang mengarah pada klasifikasi yang lebih baik pada data. Namun, pengklasifikasi 5-NN juga menyebabkan kesalahan klasifikasi yang ditandai dengan titik biru di wilayah merah dan titik merah di wilayah hijau. Bisa juga terjadi kebingungan dengan perolehan suara yang sama di antara lima tetangga terdekat (misalnya, dua tetangga berwarna merah, dua tetangga berikutnya berwarna biru, dan tetangga terakhir berwarna hijau). Kesalahan klasifikasi semacam ini dapat diatasi sampai batas tertentu dengan

menggunakan K-NN berbobot. Idenya adalah untuk memberi bobot lebih pada tetangga dengan jarak yang lebih pendek ke data uji daripada ke tetangga yang jauh. K-NN berbobot yang umum digunakan adalah K-NN berbobot Gaussian. Tidak seperti pengklasifikasi lain yang independen dari data pelatihan asli setelah dilatih, pengklasifikasi K-NN tidak memiliki memori. Jika kita menganalogikan pengklasifikasi dengan seorang ahli yang berkeliling dunia untuk menilai (mengklasifikasikan) berbagai jenis barang antik untuk orang-orang. Sementara jenis penikmat lain hanya perlu mengambil perangkat yang merangkum karakteristik utama barang antik, penikmat K-NN harus membawa setiap jenis barang antik asli dalam koleksinya untuk membuat penilaian baru. Ini mungkin terdengar terlalu rumit, namun, salah satu keuntungan utama pengklasifikasi K-NN adalah dapat mengklasifikasikan data yang tidak dapat dipisahkan secara nonlinier. Ini adalah ide kunci di balik mesin vektor dukungan berbasis kernel (SVM) [10].

2.2.5. Bagging

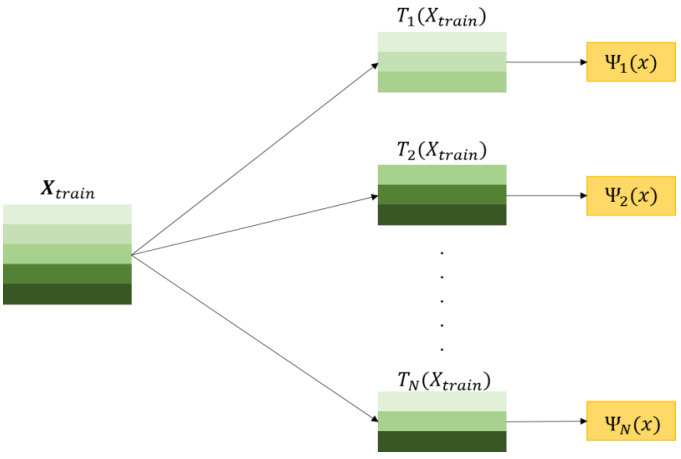
Bagging adalah gabungan dari *bootstrap* dan *aggregating*, digunakan dengan tujuan untuk mengurangi varians dari *classifier decision tree*. Metode *bagging* membagi *dataset* menjadi berbagai subset untuk *training* yang dipilih secara acak dengan substitusi. Selanjutnya subset data ini dilatih menggunakan *decision tree*. Rata-rata hasil yang diperoleh dari setiap subset data diambil yang memberikan hasil terbaik dibandingkan dengan *classifier* tunggal [7]. *Bagging* harus digunakan dengan algoritma *machine learning* yang tidak stabil seperti, *decision tree* atau *neural networks. Bagging* membangun *classification trees* menggunakan *bootstrap sampling* dari data *training* dan kemudianmenggabungkan

prediksi untuk menghasilkan meta-prediksi akhir. Model *bagging* yang digunakan pada penelitian ini adalah *random forest.*

Langkah-langkah dalam penerapan *bagging classifier :*

1. Misalkan terdapat N observasi dan M ciri dalam *dataset training*.Sampel dari kumpulan *data training* diambil secara acak dengan bergantian.
2. Subset ciri M dipilih secara acak dan ciri manapun yang memberikan pemisahan terbaik digunakan untuk membagi node secara iteratif.
3. Langkah-langkah diatas diulangi sebanyak n kali dan prediksi diberikan berdasarkan agregasi prediksi dari n jumlah *trees.*

Gambar 2.2 menunjukkan proses pengulangan N, dengan N yang memproduksi data *training* baru dan N model yang berpotensi sebagai independen (Narassiguin, 2019).



Gambar 2.2 Tahap *training bagging*

N bootstrap $(T_n(Xtrain))\ 1 \leq N$ dipilih dari data pelatihan yaitu *Xtrain* dan digunakan untuk menghasilkan *ensemble model*. *Bagging* menggunakan *majority voting* atau perhitungan rata-rata, dengan menyesuaikan apakah klasifikasi atau regresi). *Bagging classifier* merupakan *ensemble meta-estimator* yang sesuai dengan pengklasifikasi dasar masing-masing pada *random subset* dari kumpulan data asli dan kemudian menggabungkan prediksi masing-masing pada untuk menentukan prediksi akhir. *Meta estimator* biasanya digunakan sebagai cara untuk mengurangi varian dari *black-box estimator* (contohnya *decision tree*), dengan memperkenalkan pengacakan ke dalam prosedur konstruksi dan kemudian membuat *ensemble*.

Pada *jupyter notebook* untuk mengimplementasikan metode *bagging classifier* menggunakan pustaka *sklearn.ensemble.BaggingClassifier* dengan penulisan *code*sebagai berikut :

```
class sklearn.ensemble.BaggingClassifier(estimator=None, n_estimators=10,
*, max_samples=1.0, max_features=1.0, bootstrap=True,
bootstrap_features=False, oob_score=False, warm_start=False,
n_jobs=None, random_state=No ne, verbose=0,
```

Parameter :

a. *Estimator : object, default=None*

   Estimator dasar digunakan agar sesuai dengan *random subset* dari *dataset.*

   Jika tidak ada, maka *estimator* dasarnya adalah *DecisionTreeClassifier.*

b. *n_estimators : int, default=10*

Jumlah *estimator* dasar dalam *ensemble*.

c. *max_samples : int or float, default=1.0*

Jumlah sampel yang diambil dari X untuk melatih setiap *estimator* dasar.

d. *max_features : int or float, default=1.0*

Jumlah ciri yang akan diambil dari X untuk *training* setiap *estimator* dasar.

e. *bootstrap : bool, default=True*

Apakah sampel diambil dengan *replacement.*

f. *boostrap_features : bool, default=False*

Apakah ciri diambil dengan *replacement.*

g. *oob_score : bool, default=False*

Apakah akan menggunakan sampel *out-of-bag* untuk memperkirakan kesalahan generalisasi, dapat dijalankan jika *bootstrap=True*.

h. *warm_start : bool, default=False*

Saat bernilai *True*, solusi sebelumnya akan dipanggil untuk menyesuaikan dan menambahkan lebih banyak *estimator* ke *ensemble*, jika tidak digunakan *ensemble* yang baru.

i. *n_jobs : int, default=None*

Jumlah tugas yang dijalankan secara paralel untuk kesesuaian dan prediksi.

j. *Random_state : int, RandomState instance or None, default=None*

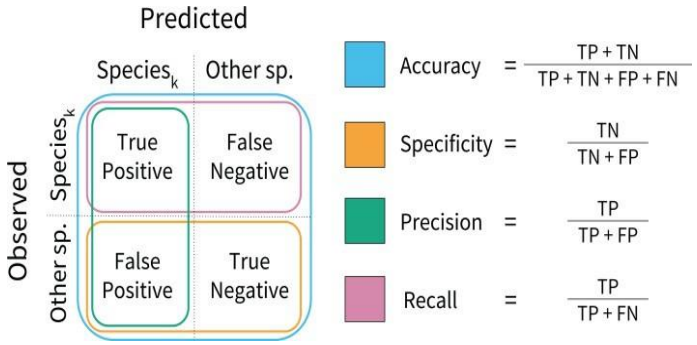
Mengontrol resampling acak dari *dataset* asli.

Parameter tersebut dapat berubah disesuaikan dengan versi *Python* yang digunakan. Jika sampel diambil dengan *replacement*, maka metode tersebut dikenal dengan *bagging*. Ketika *random subset* diambil dari *dataset* maka metode ini dikenal sebagai *random subspace*. Namun, ketika *estimator* dasar dibangun di atas himpunan bagian dari sampel dan ciri, maka metode tersebut dikenal sebagai *random patch* (Breiman, 2001).

2.2.6. Confusion Matrik

Matriks konfigurasi adalah tabel yang terdiri dari jumlah baris data uji yang diprediksi benar dan salah dengan model klasifikasi yang digunakan. Tabel Confusion Matrix diperlukan untuk memilih kinerja terbaik dari sebuah model klasifikasi [11]. Selama pelatihan model, kinerja dinilai berdasarkan per-sampel menggunakan akurasi model dan skor kerugian log. Akurasi model menghitung proporsi sampel yang diklasifikasikan dengan benar dalam data uji (Gambar.2.3), dan nilai akurasi model yang tinggi diinginkan. Log loss menilai apakah probabilitas prediksi dikalibrasi dengan baik, menghukum prediksi yang salah dan tidak pasti. Skor kehilangan log yang rendah menunjukkan bahwa kesalahan klasifikasi terjadi pada tingkat yang mendekati tingkat probabilitas yang diprediksi. Selama pengujian model, kinerja dinilai menggunakan akurasi peringkat-1 dan biaya lintas-entropi [11]. Akurasi peringkat-1 dihitung berdasarkan ID spesies mana yang diprediksi dengan probabilitas tertinggi. Skor lintas-entropi mirip dengan fungsi kehilangan log, tetapi diskalakan menggunakan fungsi indikator. Ini dapat ditafsirkan dengan cara yang mirip dengan akurasi dan kehilangan log; akurasi

peringkat-1 yang tinggi dan skor entropi silang yang rendah diinginkan. Metrik pengujian model sekunder dihitung untuk setiap spesies menggunakan data uji. Ini termasuk model spesifisitas, presisi, dan recall yang perumusanya dapat dilihat pada Gambar 2.3.



Gambar 2.3 Metrik kinerja model

Gambar 2.3 merupakan metrik kinerja model yang mengungkapkan perilaku model yang skor akurasinya mungkin tidak jelas. Spesifisitas menilai kinerja model pada spesies non-target, menghukum overprediksi spesies target (yaitu, sejumlah besar positif palsu). Presisi juga menghukum overprediksi, tetapi menilai tingkat overprediksi relatif terhadap tingkat prediksi positif yang benar. Recall menghitung proporsi prediksi positif sejati dengan jumlah total pengamatan positif per spesies. Nilai yang lebih tinggi diinginkan untuk masing-masing. Metrik ini dihitung untuk membantu interpretasi, tetapi tidak digunakan untuk memeringkat kinerja model secara formal. Akurasi adalah salah satu metrik untuk mengevaluasi model klasifikasi. Secara informal, akurasi adalah sebagian kecil dari prediksi model kami yang benar. Secara formal, akurasi memiliki definisi sebagai berikut:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Untuk klasifikasi biner, akurasi juga dapat dihitung dalam hal positif dan negatif sebagai berikut:

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

Dimana TP = Positif Benar, TN = Negatif Benar, FP = Positif Palsu, dan FN = Negatif Palsu.

2.2.7. Klasifikasi Kurva *ROC* dan *AUC*

a) Kurva ROC

Sebuah ROC kurva (*Receiver Operating Characteristic Curve*) adalah grafik yang menunjukkan kinerja model klasifikasi sama sekali ambang klasifikasi. Kurva ini memplot dua parameter: [12]

1) Tingkat Positif BenarTingkat Positif Palsu

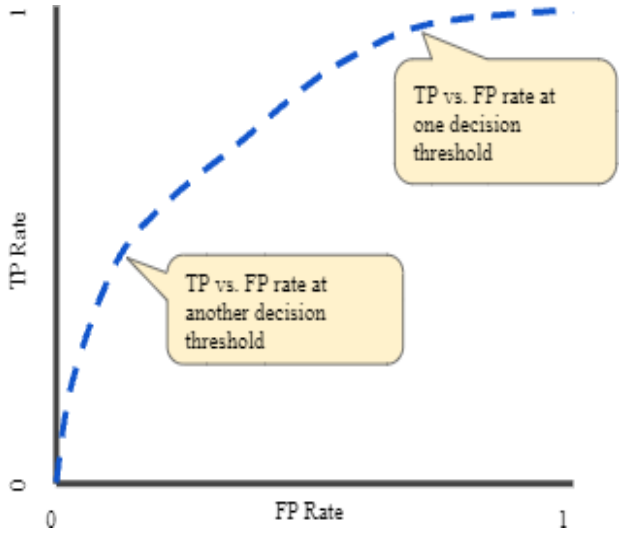
   True Positive Rate (TPR) adalah sinonim untuk mengingat dan oleh karena itu didefinisikan sebagai berikut:

   $$TPR = \frac{TP}{TP + FN}$$

   False Positive Rate (FPR) didefinisikan sebagai berikut:

   $$TPR = \frac{TP}{TP + FN}$$

Kurva ROC memplot TPR vs. FPR pada ambang klasifikasi yang berbeda. Menurunkan ambang klasifikasi mengklasifikasikan lebih banyak item sebagai positif, sehingga meningkatkan Positif Palsu dan Positif Benar. Gambar 2.4 menunjukkan kurva ROC yang khas.



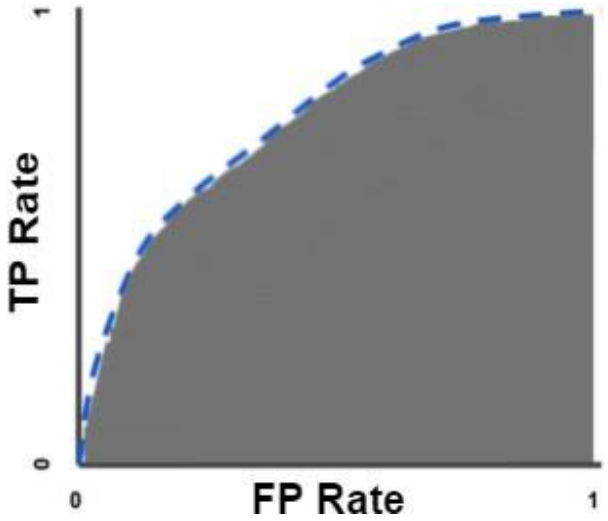
Gambar 2.4 Tingkat TP vs. FP pada ambang klasifikasi yang berbeda

Gambar 2.4 Tingkat TP vs. FP pada ambang klasifikasi untuk menghitung titik-titik dalam kurva ROC, kita dapat mengevaluasi model regresi logistik berkali-kali dengan ambang klasifikasi yang berbeda, tetapi ini akan menjadi tidak efisien. Untungnya, ada algoritma berbasis penyortiran yang efisien yang dapat memberikan informasi ini untuk kita, yang disebut AUC.

2.2.8. AUC : Area Under the Kurva ROC

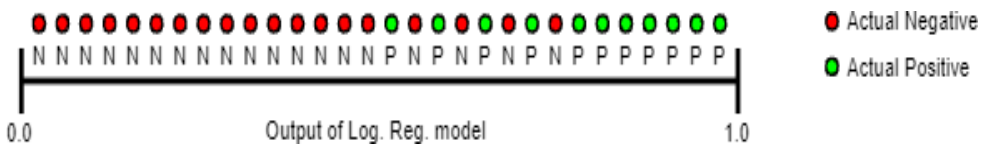
AUC adalah singkatan dari "Area di bawah Kurva ROC." Artinya, AUC mengukur seluruh area dua dimensi di bawah seluruh kurva ROC (pikirkan

kalkulus integral) dari (0,0) hingga (1,1) untuk lebih jelasnya bisa dilihat pada gambar 2.5.



Gambar 2.5 AUC (Area di bawah Kurva ROC)

AUC memberikan ukuran kinerja agregat di semua ambang klasifikasi yang mungkin. Salah satu cara untuk menginterpretasikan AUC adalah sebagai probabilitas bahwa model memberi peringkat contoh positif acak lebih tinggi daripada contoh negatif acak. Sebagai contoh, diberikan contoh berikut, yang disusun dari kiri ke kanan dalam urutan menaik dari prediksi regresi logistic seperti pada gambar 2.6.



Gambar 2.6 Prediksi peringkat dalam urutan skor regresi logistik

AUC mewakili probabilitas bahwa contoh acak positif (hijau) diposisikan di sebelah kanan contoh acak negatif (merah). Rentang nilai AUC dari 0 hingga 1. Model yang prediksinya 100% salah memiliki AUC 0,0; yang prediksinya 100% benar memiliki AUC 1,0.

AUC diinginkan karena dua alasan berikut:

a. AUC adalah skala-invarian . Ini mengukur seberapa baik prediksi diberi peringkat, bukan nilai absolutnya.
b. AUC adalah klasifikasi-ambang-invarian . Ini mengukur kualitas prediksi model terlepas dari ambang klasifikasi apa yang dipilih.

   Namun, kedua alasan ini disertai dengan peringatan, yang dapat membatasi kegunaan AUC dalam kasus penggunaan tertentu:

   a) Skala invarians tidak selalu diinginkan. Misalnya, terkadang kami benar-benar membutuhkan keluaran probabilitas yang terkalibrasi dengan baik, dan AUC tidak akan memberi tahu kami tentang itu.
   b) Klasifikasi-ambang invarians tidak selalu diinginkan. Dalam kasus di mana ada perbedaan besar dalam biaya negatif palsu vs positif palsu, mungkin penting untuk meminimalkan satu jenis kesalahan klasifikasi. Misalnya, saat melakukan deteksi spam email, Anda mungkin ingin memprioritaskan meminimalkan positif palsu (bahkan jika itu menghasilkan peningkatan negatif palsu yang signifikan). AUC bukan metrik yang berguna untuk jenis pengoptimalan ini.
   c) Performance keakurasian AUC dapat diklasifikasikan menjadi beberapa kelompok yaitu:

1. 0.90 – 1.00 = *Exellent Classification*
2. 0.80 – 0.90 = *Good Classification*
3. 0.70 – 0.80 = *Fair Classification*
4. 0.60 – 0.70 = *Poor Classification*
5. 0.50 – 0.60 = *Failure Classification*

2.2.8.1. Feature Selection

Proses Pemilihan Fitur sangat penting dalam pembelajaran mesin yang sangat memengaruhi kinerja model Anda. Fitur data yang Anda gunakan untuk melatih model atau algoritme pembelajaran mesin memiliki pengaruh besar pada kinerja yang dapat Anda capai. Fitur yang tidak relevan atau hilang dapat berdampak negatif pada kinerja sistem Pemilihan fitur adalah salah satu langkah pertama dan penting saat melakukan tugas pembelajaran mesin apa pun. Fitur dalam kasus kumpulan data berarti kolom. Ketika kita mendapatkan dataset apa pun, belum tentu setiap kolom (fitur) akan berdampak pada variabel output. Jika kami menambahkan fitur yang tidak relevan ini ke dalam model, itu hanya akan memperburuk model dan dapat mengurangi keakuratan model dan membuat model Anda belajar berdasarkan fitur yang tidak relevan. Hal ini menimbulkan perlunya melakukan seleksi fitur. Ketika datang ke implementasi pemilihan fitur dalam fitur Numerik dan Kategoris harus diperlakukan secara berbeda. Oleh karena itu sebelum menerapkan metode berikut, kita perlu memastikan bahwa Data Frame hanya berisi fitur Numerik.

Manfaat melakukan pemilihan fitur sebelum memodelkan data Anda adalah sebagai berikut:

a. Mengurangi Overfitting: Data yang lebih sedikit berarti lebih sedikit kesempatan untuk membuat keputusan berdasarkan noise.
b. Meningkatkan Akurasi: Data yang kurang menyesatkan berarti akurasi pemodelan meningkat.
c. Mengurangi Kompleksitas: lebih sedikit titik data mengurangi kompleksitas algoritme dan membuatnya lebih mudah dipahami.
d. Pelatihan Lebih Cepat: Ini memungkinkan algoritme pembelajaran mesin untuk berlatih lebih cepat.

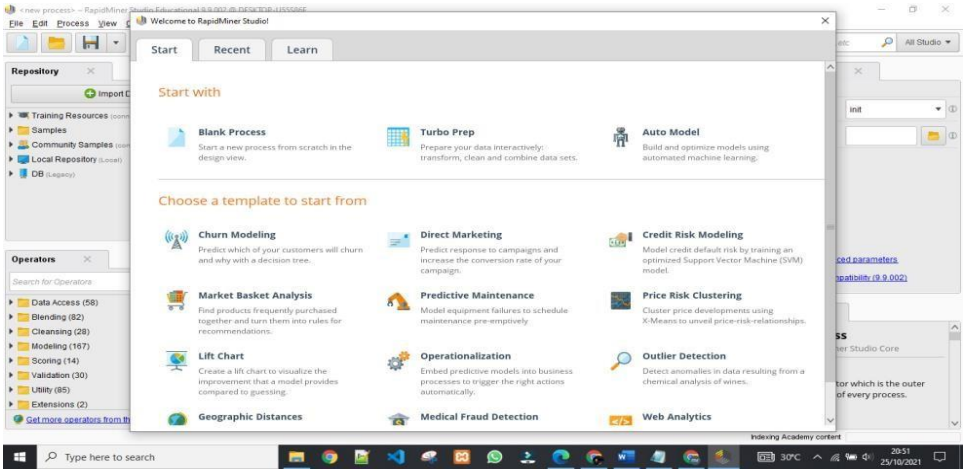
Pada sistem ini digunakan dua proses seleksi fitur yaitu proses seleksi fitur sekuensial/forward dan proses seleksi fitur mundur. Pada sistem ini digunakan dua proses seleksi fitur yaitu proses seleksi fitur sekuensial/forward dan proses seleksi fitur mundur [12]

2.2.9.Rapid Miner

Rapid miner adalah alat analisis penambangan data yang digunakan untuk menganalisis data dan mendukung berbagai teknik data mining. Digunakan untuk aplikasi industri, penelitian, pelatihan, pengembangan aplikasi dan pendidikan. Itu mengandung sekitar 100 skema pembelajaran untuk pengelompokan, klasifikasi dan regresi analisis. Ini mendukung sekitar 22 format file seperti .xls, .csv dan sebagainya. Dalam informasi ini dapat diimpor dari berbagai database untuk analisis dan tujuan prediksi [14]. RapidMiner , sebelumnya dikenal sebagai YALE

(Yet Another Learning Environment), dikembangkan mulai tahun 2001 oleh Ralf Klinkenberg, Ingo Mierswa, dan Simon Fischer di Artificial Intelligence Unit of the Technical University of Dortmund. Mulai tahun 2006, perkembangannya dimotori oleh Rapid-I, perusahaan yang didirikan oleh Ingo Mierswa dan Ralf Klinkenberg pada tahun yang sama. Pada tahun 2007, nama perangkat lunak diubah dari YALE menjadi RapidMiner. Pada tahun 2013, perusahaan berganti nama dari Rapid-I menjadi RapidMiner. RapidMiner memiliki beberapa sifat sebagai berikut:

1. Ditulis dengan bahasa pemrograman Java sehingga dapat dijalankan diberbagai sistem operasi;
2. Proses penemuan pengetahuan dimodelkan sebagai operator trees;
3. Representasi XML internal untuk memastikan format standar pertukaran data;
4. Bahasa scripting memungkinkan untuk eksperimen skala besar dan otomatisasi eksperimen;
5. Konsep multi-layer untuk menjamin tampilan data yang efisien dan menjamin penanganan data;
6. Memiliki GUI, command line mode, dan Java API yang dapat dipanggil dari program lain. RapidMiner memiliki tampilan antarmuka yang *user friendly* yang memberikan kemudahan kepada pengguna dalam memakainya dan dikenal dengan sebutan *Perspective*. Berikut adalah tampilan aplikasi RapidMiner seperti yang ditunjukan pada gambar 2.7



Gambar 2.7 Tools Rapid Miner

Beberapa fitur Rapid Miner antara lain:

1. Banyaknya algoritma data mining, seperti *K-Nears Neighbor* dan *Teknik Bagging*;
2. Bentuk grafis yang canggih, seperti tumpang tindih diagram histogram, tree chart dan 3D Scatter plots;
3. Banyaknya variasi plugin, seperti text plugin untuk melakukan analisis teks;
4. Menyediakan prosedur data mining dan machine learning termasuk: ETL(extraction, transformation, loading), data preprocessing, visualisasi,modelling dan evaluasi;
5. Proses data mining tersusun atas operator-operator yang nestable, dideskripsikan dengan XML, dan dibuat dengan GUI;
6. Mengintegrasikan proyek data mining Weka dan statistika R.