BAB II
TINJAUAN PUSTAKA

2.1 Kajian Terkait

Penilitian terkait yang menggunakan algoritma K-Nearest Neighbor untuk melakukan prediksi pada SNMPTN pernah dilakukan sebelumnya oleh (Wibowo & Fitrianah, 2018) dengan judul "A K-Nearest Algorithm Based Application to Predict SNMPTN Acceptance for High School Students In Indonesia". Penelitian tersebut melakukan prediksi menggunakan teknik klasifikasi menggunakan algoritma K-Nearest Neighbor (KNN) berdasarkan nilai rata-rata rapor semester 1, 2, 3, 4, dan 5 untuk menentukan kelulusan pada SNMPTN. Klasifikasi yang dilakukan berdasarkan pada data alumni dengan menggunakan hasil kelulusan pada SNMPTN yaitu "Diterima" dan "Tidak diterima". Penelitian menggunakan WEKA yang menghasilkan nilai k=3 memiliki akurasi terbaik pada evaluasi data training, yaitu sebesar 80% untuk jurusan IPA dan 89% untuk jurusan IPS. Penelitian menghasilkan aplikasi berbasis web menggunakan CodeIgniter sebagai Framework PHP.

Penelitian lain yang memprediksi kelulusan pada SNMPTN dilakukan oleh (Dewi & Nursikuwagus, 2018) dengan judul "Analisis Prediksi Kelulusan Siswa SMK pada SNMPTN Menggunakan Metode Fuzzy Mamdani (Studi Kasus : SMK Negeri 4 Bandung). Metode Fuzzy Mamdani digunakan untuk proses analisa dengan variabel input Fuzzy yaitu rata-rata rapor semester 1, 2, 3, 4, dan 5, ditambah variabel output Fuzzy yaitu kelulusan. Sistem ini dikembangkan dengan perangkat lunak pendukung seperti MySQL dan PHP. Berdasarkan hasil pengujian terhadap data nilai rapor siswa SMK, tingkat akurasinya mencapai 82% yang dapat diklasifikasikan sebagai klasifikasi terbaik, dengan nilai precision (kedekatan perbedaan nilai) 79,55% dan nilai recall (pemanggilan kembali) 100%. Sistem ini ditujukan membantu pengambilan keputusan untuk keikutsertaan siswa SMK menjadi peserta SNMPTN.

Selain itu (Utomo et al., 2019) juga melakukan penelitian serupa dengan judul "Sistem Prediksi Penerimaan SNMPTN menggunakan Algoritma Decision Tree C4.5" yang memprediksi kemungkinan siswa diterima melalui jalur SNMPTN sehingga dapat meringankan beban guru bimbingan konseling. Algoritma prediksi yang digunakan adalah Decision Tree C4.5 yang membuat pohon keputusan untuk menggambarkan rule. Data yang digunakan berasal dari nilai rapor alumni yang pernah mengikuti SNMPTN dari tahun 2016-2018 dengan jumlah 681 data untuk jurusan IPA dan 90 data untuk jurusan IPS beserta daftar siswa lulus SNMPTN di tahun yang sama. Dari data nilai dan daftar siswa lulus SNMPTN tersebut atribut yang digunakan hanya atribut nilai mata pelajaran yang digunakan dalam SNMPTN 2019 beserta status lulus atau tidak siswa dalam mengikuti SNMPTN. Sistem dibangun dalam bentuk website yang memanfaatkan WEKA CLI untuk proses prediksi.

Berikut ditampilkan rangkuman kajian terkait dalam penelitian ini pada Tabel II.1 dibawah ini.

Tabel II.1 Rangkuman Kajian Terkait

No	Penulis	Judul	Keterangan
1	(Wibowo & Fitrianah, 2018)	"A K-Nearest Algorithm Based Application to Predict SNMPTN Acceptance for High School Students In Indonesia"	Melakukan prediksi kelulusan SNMPTN dengan algoritma K-Nearest Neighbor (KNN) menggunakan nilai rata-rata rapor semester 1 sampai semester 5. Penelitian menghasilkan nilai k=3 memiliki akurasi terbaik yaitu sebesar 80% untuk jurusan IPA dan 89% untuk jurusan IPS.
2	(Dewi & Nursikuwagus, 2018)	"Analisis Prediksi Kelulusan Siswa SMK pada SNMPTN Menggunakan	Melakukan prediksi kelulusan SNMPTN dengan menggunakan nilai rata-rata rapor semester 1 sampai dengan semester 5.

		Metode Fuzzy Mamdani”	Menggunakan metode Fuzzy Mamdani yang menunjukan tingkat akurasi (accuracy) sebesar 82%, precision 79,55%, dan recall 100%.
3	(Utomo et al., 2019)	“Sistem Prediksi Penerimaan SNMPTN menggunakan Algoritma Decision Tree C4.5”	Memanfaatkan WEKA CLI untuk proses prediksi status lulus atau tidak siswa dalam mengikuti SNMPTN. Atribut yang digunakan yaitu atribut nilai mata pelajaran yang digunakan dalam SNMPTN 2019 dari semester 1 sampai dengan semester 5 beserta status lulus atau tidak siswa dalam mengikuti SNMPTN.

2.2 LTMPT

Lembaga Tes Masuk Perguruan Tinggi (LTMPT) adalah lembaga penyelenggara tes masuk perguruan tinggi bagi calon mahasiswa baru. Berdasarkan informasi yang didapatkan pada laman resmi (LTMPT, 2021), lembaga ini berada di bawah naungan Kementerian Pendidikan dan Kebudayaan Republik Indonesia sebagai satu-satunya lembaga penyelenggara tes perguruan tinggi terstandar di Indonesia. LTMPT melakukan peningkatan kualitas proses seleksi penerimaan mahasiswa baru di Perguruan Tinggi (PT) di mana terdapat tiga jalur seleksi yaitu SNMPTN, SBMPTN, dan seleksi Mandiri. Pengembangan model proses seleksi dilakukan sesuai perkembangan teknologi informasi, dan era digitalisasi melalui Ujian Tulis Berbasis Komputer (UTBK).

LTMPT memiliki fungsi untuk mengelola dan mengolah data calon mahasiswa untuk bahan seleksi jalur SNMPTN dan SBMPTN oleh rektor/direktur PTN, melaksanakan Ujian Tulis Berbasis Komputer UTBK, dan menyampaikan

hasil UTBK kepada peserta dan perguruan tinggi tujuan. Adapun tujuan dari LTMPT adalah sebagai berikut yaitu :

1. Melaksanakan tes masuk perguruan tinggi yang kredibel, adil, transparan, fleksibel, efisien, dan akuntabel.
2. Membantu perguruan tinggi memperoleh calon mahasiswa berdasarkan nilai akademik atau nilai akademik dan prestasi lainnya, melalui jalur SNMPTN.
3. Membantu memperoleh calon mahasiswa berdasarkan hasil UTBK saja atau UTBK dan kriteria lain yang ditetapkan bersama oleh PTN, melalui jalur SBMPTN.

LTMPT memposisikan diri sebagai lembaga penyelenggara tes masuk perguruan tinggi yang terbaik dan terdepan di Indonesia. Dengan demikian, keberadaan LTMPT diharapkan bisa yang benar-benar mendapatkan calon mahasiswa baru yang diperkirakan mempunyai keberhasilan studi di Perguruan Tinggi. Selain itu diharapkan masyarakat akan mendapat kenyamanan dan kemanfaatan yang lebih.

2.3 SNMPTN

SNMPTN atau Seleksi Nasional Masuk Perguruan Tinggi merupakan jalur penerimaan mahasiswa baru program sarjana pada PTN yang berdasarkan nilai akademik saja atau nilai akademik dan prestasi lainnya (yang ditetapkan oleh PTN). Biaya SNMPTN sepenuhnya ditanggung oleh pemerintah. Adapun tahapan pendaftaran SNMPTN berdasarkan informasi resmi dari laman LTMPT adalah sebagai berikut (LTMPT, 2021):

1. Pengumuman kuota oleh LTMPT bagi sekolah sesuai dengan akreditasi dan jumlah siswa.
2. Registrasi akun LTMPT bagi sekolah yang wajib dilakukan oleh sekolah yang belum mempunyai akun LTMPT.
3. Registrasi akun LTMPT bagi siswa yang wajib dilakukan oleh semua siswa kelas 12.

4. Sekolah menentukan calon peserta berdasarkan data LTMPT dari jumlah siswa dan akreditasi di PUSDATINKEMDIKBUD atau EMIS PENDIS KEMENAG.
5. Pengisian PDSS (pangkalan data sekolah dan siswa) yang dilakukan oleh sekolah. PDSS merupakan basis data yang berisikan rekam jejak kinerja sekolah dan nilai rapor siswa yang *eligible* mendaftar. Kebenaran data yang diisikan menjadi tanggung jawab kepala sekolah.
6. Pendaftaran SNMPTN.
7. Pilihan PTN & program studi.
8. Pengunggahan portofolio yang wajib bagi peserta yang memilih program studi Bidang Seni dan Olahraga.
9. Seleksi jalur SNMPTN berdasarkan kriteria yang ditetapkan oleh masing-masing PTN.
10. Pengumuman kelulusan hasil SNMPTN.
11. Daftar ulang sesuai dengan PTN tempat calon mahasiswa dinyatakan diterima.

Pemeringkatan siswa dilakukan oleh sekolah yang pada dasarnya memperhitungkan nilai mata pelajaran sebagai berikut :

1. Jurusan IPA: Matematika, Bahasa Indonesia, Bahasa Inggris, Kimia, Fisika, dan Biologi.
2. Jurusan IPS: Matematika, Bahasa Indonesia, Bahasa Inggris, Sosiologi, Ekonomi, dan Geografi.
3. Jurusan Bahasa: Matematika, Bahasa Indonesia, Bahasa Inggris, Sastra Indonesia, Antropologi, dan salah satu Bahasa Asing.
4. SMK: Matematika, Bahasa Indonesia, Bahasa Inggris, dan Kompetensi Keahlian.

Sekolah dapat menambahkan kriteria lain berupa prestasi akademik dalam menentukan peringkat siswa apabila ada nilai yang sama. Jumlah siswa yang masuk dalam pemeringkatan sesuai dengan ketentuan kuota akreditasi sekolah.

2.4 *Data Mining*

Data Mining adalah proses penemuan pola dan pengetahuan yang didapatkan dari data berukuran besar. Data Mining bertujuan untuk memanfaatkan data dalam basis data dengan mengolahnya sehingga menghasilkan informasi baru yang berguna (Pradnyana & Agustini, 2022). Secara umum terdapat 5 peranan dalam data mining, yaitu estimasi, prediksi, klasifikasi, klustering, dan asosiasi.

Data mining merupakan nama populer yang digunakan untuk penyebutan istilah *Knowledge in Data Discovery* (KDD). Adapun pendapat lain beranggapan bahwa data mining merupakan salah satu tahap penting dari KDD. Tahapan dari KDD dapat diuraikan sebagai berikut (Han et al., 2012) :

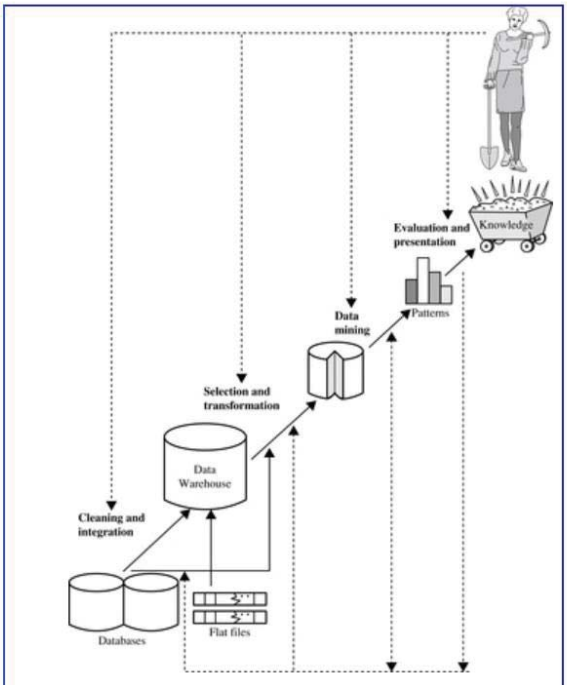


Figure 1.4

Data mining as a step in the process of knowledge discovery.

Gambar II.1 Tahapan KDD

1. *Data cleaning* (yaitu proses penghapusan data yang tidak sesuai dan tidak konsisten).
2. *Data integration* (dimana data-data dari beberapa sumber data digabungkan). Banyak pula yang menyatukan tahap data cleaning dan data integration menjadi satu tahapan yaitu preprocessing data dimana data yang dihasilkan kemudian disimpan.
3. *Data selection* (dimana data yang diperlukan untuk proses analisis diseleksi dan diambil dari database).
4. *Data transformation* (dimana data diubah dan disatukan ke dalam bentuk yang sesuai untuk proses mining dengan melakukan peringkasan atau pengumpulan) Proses data transformation dan penyatuan data terkadang dapat dilakukan sebelum proses data selection, terutama pada kasus data yang berukuran sangat besar. Data reduction juga dapat dilakukan untuk mengurangi data dengan tetap menggambarkan data asli tanpa mengurangi integritas data.
5. *Data mining* (merupakan proses penting dengan melibatkan berbagai metode yang digunakan untuk menemukan pola data).
6. *Pattern evaluation* (proses untuk penilaian hasil yang menunjukan pengetahuan berdasarkan ragam menarik).
7. *Knowledge presentation* (dimana visualisasi dan teknik penyajian digunakan untuk menyampaikan hasil pengetahuan).

Metode pelatihan data mining dapat dibedakan menjadi 2; yaitu *supervised learning* dan *unsupervised learning*. Menurut (Larose, 2005), sebagian besar metode data mining termasuk ke dalam *supervised learning*, yaitu metode dimana algoritma diberikan data latih yang banyak dengan variabel target (label) yang telah ditentukan sebelumnya sebagai training sehingga kemudian algoritma dapat menentukan variabel target untuk dikaitkan pada variabel yang belum diketahui labelnya. Sementara itu, unsupervised learning merupakan metode yang mencari pola dan struktur di antara seluruh variabel tanpa diketahui variabel target yang diidentifikasikan sebelumnya. Dengan kata lain metode ini diterapkan tanpa adanya latihan dan tanpa ada pelatih yaitu label dari data.

Contoh klasik dari teknik supervised learning diwakili oleh proses klasifikasi (metode prediktif) yang dilakukan dengan menggunakan sebagian variabel untuk memprediksi satu atau lebih variabel lain; sedangkan contoh klasik dari teknik *unsupervised learning* diwakili oleh proses *clustering* (metode deskriptif) yang dilakukan dengan identifikasi pola yang menggambarkan atau mewakilkan data agar dapat dengan mudah dipahami oleh pengguna (Gorunescu, 2011).

2.5 Klasifikasi dan Regresi untuk Analisis Prediktif

Pada model *supervised learning* dapat lebih lanjut dibedakan menjadi klasifikasi dan regresi. Secara singkat model klasifikasi menentukan objek kedalam kategori sedangkan regresi memprediksi hasil dalam bentuk nilai kontinu. Walau demikian, batasan antara klasifikasi dan regresi terkadang masih ambigu dan klasifikasi hanyalah model regresi dengan ambang batas yang diterapkan. Misal apabila bernilai lebih tinggi dari ambang batas, akan diklasifikasikan sebagai benar sementara apabila lebih rendah akan diklasifikasikan sebagai salah. Banyak algoritma yang dapat digunakan baik untuk klasifikasi maupun regresi seperti *Decision Tree*, *Random Forrest*, *Support Vector Machine* dan *K-Nearest Neighbor*.

Klasifikasi adalah proses menemukan model (atau fungsi) yang menggambarkan dan membedakan kelas-kelas data. Model dihasilkan berdasarkan analisis himpunan data training (yaitu, objek data yang label kelasnya diketahui). Model ini kemudian digunakan untuk memprediksi label kelas objek yang label kelasnya tidak diketahui. Terdapat berbagai algoritma yang dapat digunakan untuk proses klasifikasi antara lain: *Logistic Regression*, *Naive Bayes*, C 4.5, dan *Support Vector Machine*.

Berbeda dengan klasifikasi yang memprediksi label kategorikal (diskrit), model regresi digunakan pada fungsi yang bernilai kontinu. Artinya, regresi digunakan untuk memprediksi nilai data numerik yang hilang atau tidak tersedia label kelasnya. Istilah prediksi mengacu pada prediksi numerik dan prediksi label kelas. Analisis regresi adalah metodologi statistik yang paling sering digunakan untuk prediksi numerik, meskipun metode lain juga dapat digunakan. Regresi juga mencakup identifikasi tren distribusi berdasarkan data yang tersedia.

Misalkan apabila manajer toko elektronik ingin mengklasifikasikan sekumpulan besar barang di toko berdasarkan tiga jenis respons pelanggan terhadap promosi penjualan yaitu: respons baik, respons ringan, dan tidak ada respons. Model untuk ketiga kelas ini didapatkan berdasarkan fitur deskriptif pada barang seperti harga, merek, jenis, dan kategori. Klasifikasi yang dihasilkan harus secara maksimal membedakan setiap kelas antara satu dari yang lain. Misalkan model klasifikasi dapat mengidentifikasi harga sebagai faktor tunggal yang paling membedakan ketiga kelas tersebut. Model dapat mengungkapkan bahwa selain harga, terdapat fitur lain yang membantu membedakan lebih lanjut objek antar kelas satu sama lain. Model tersebut dapat membantu memahami dampak dari promosi penjualan yang dilakukan, sehingga kemudian dapat digunakan untuk merancang strategi promosi yang lebih efektif pada masa mendatang.

Apabila tidak ingin memprediksi label respons kategoris untuk setiap item toko, melainkan ingin memprediksi jumlah pendapatan yang akan dihasilkan setiap item selama penjualan mendatang berdasarkan data penjualan sebelumnya. Dalam hal ini, dapat dilakukan analisis menggunakan regresi karena model regresi yang dibangun dapat memprediksi fungsi kontinu. Sebelum melakukan proses klasifikasi dan regresi mungkin perlu didahului dengan analisis relevansi, yaitu mencoba mengidentifikasi atribut yang secara signifikan berhubungan dengan hasil klasifikasi dan regresi tersebut. Atribut tersebut akan dipilih untuk proses klasifikasi dan regresi selanjutnya. Atribut lain yang tidak relevan, kemudian dapat dikeluarkan dari pertimbangan (Han et al., 2009).

2.6 Algoritma K-Nearest Neighbor

Algoritma K-Nearest Neighbor (KNN) merupakan salah satu algoritma yang termasuk ke dalam *supervised learning* dan juga merupakan contoh dari *instance-based learning*. Pada *instance-based learning,* data yang dijadikan pembelajaran yang sudah diketahui kelasnya disimpan dan secara langsung dikaitkan/dibandingkan pada data baru yang belum diketahui kelasnya. Pekerjaan dilakukan saat data baru diberikan, tanpa membuat kesimpulan terlebih dahulu melalui data pembelajaran (Witten et al., 2011).

Algoritma K-Nearest Neighbor melakukan prediksi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Pada algoritma KNN perlu untuk menentukan nilai k, yaitu jumlah objek tetangga (neighbor) yang terdekat. Hasil akhir yaitu label kelas dengan jumlah terbanyak diantara k objek. KNN juga dapat digunakan pada prediksi numerik, yang memberikan prediksi berbentuk nilai untuk kelas dengan label tidak diketahui. Dalam hal ini, model memberikan hasil berdasarkan rata-rata dari label nilai sejumlah k objek (Han et al., 2009).

Untuk menentukan jarak antara data training dan data testing maka digunakan rumus *Euclidean distance*. Rumus yang digunakan untuk mengukur nilai *Euclidean distance* dapat ditunjukkan pada persamaan berikut (Suntoro, 2019) :

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

dimana :

d = Jarak kedekatan antara data training dan data testing

x = Data Training

y = Data Testing

i = Record (baris) ke-i dari tabel

n = Jumlah data training

Langkah-langkah algoritma k-NN adalah sebagai berikut :

1. Tentukan nilai parameter k (nilai k dipilih secara manual).
2. Hitung jarak antara data training dan data testing (menggunakan metode *Euclidean distance*).
3. Urutkan data training berdasarkan jarak terkecil.
4. Menetapkan kelas, yang digunakan sebagai hasil prediksi.

Dataset dibagi menjadi dua bagian, yaitu data training dan data testing. Data training adalah data yang sudah mempunyai kelas, sedangkan data testing adalah data yang akan dicari kelasnya. Data training akan membentuk suatu

model/pola/pengetahuan, sedangkan data testing digunakan untuk pengukuran evaluasi algoritma.

Untuk menentukan nilai terbaik dari k tergantung berdasarkan pada data. Pada nilai k yang lebih besar akan mengurangi efek *noise* (*outlier*) pada klasifikasi, tetapi membuat batas antar kelas menjadi kurang jelas. Pemilihan parameter k yang baik dapat dilakukan dengan menggunakan metode *cross-validation*. Secara teknis, dataset yang diberikan dibagi menjadi sejumlah r data yang diambil secara acak ke dalam suatu subset. Untuk nilai k yang telah ditentukan, algoritma k-nn diterapkan untuk membuat prediksi pada subset ke-r, untuk kemudian dievaluasi kesalahan pada satu siklus. Pada akhir siklus r, kesalahan yang dihitung kemudian dirata-ratakan untuk menghasilkan ukuran seberapa baik algoritma dalam memprediksi objek baru. Langkah-langkah di atas kemudian diulangi untuk berbagai k dan nilai yang mencapai hasil terbaik kemudian dipilih sebagai nilai optimal untuk k. Akurasi algoritma K-NN dapat sangat menurun apabila terdapat fitur yang tidak relevan (atau *noisy*), atau jika skala fitur tidak konsisten dengan kepentingannya. Oleh karena itu, fitur mungkin harus diatur sebelumnya untuk mencegah kebingungan karena 'dominasi' fitur tertentu.

Algoritma K-NN mudah diimplementasikan. Tetapi disisi lain K-NN merupakan *lazy learner* yang terutama pada set pelatihan berukuran besar dapat membutuhkan waktu proses yang lama dan relatif 'mahal' secara komputasi. Tidak seperti *eager learner*, di mana model mencoba untuk membangun model berdasarkan aturan baru. Prediksi didasarkan pada informasi lokal, sehingga kemungkinan akan dipengaruhi oleh nilai ekstrim/*outlier* (Gorunescu, 2011).

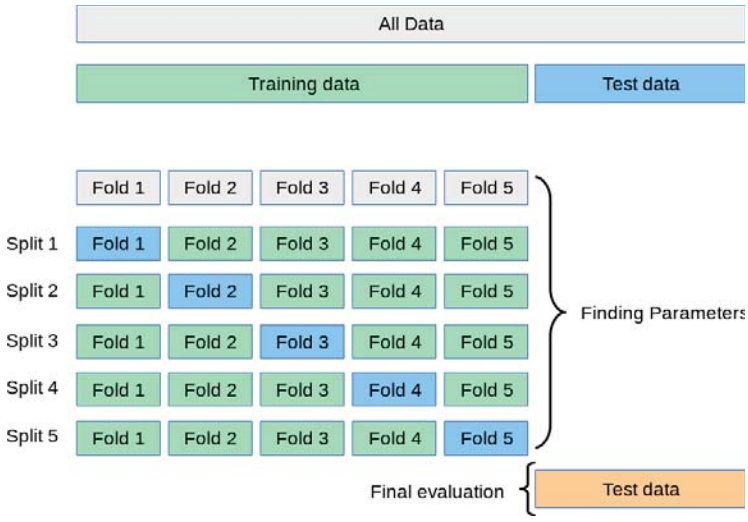
2.7 *Cross Validation*

Mencari parameter pada suatu fungsi prediksi dan mengujikannya pada data yang sama dapat menghasilkan model dengan skor performa yang sangat baik, namun dapat gagal dalam memprediksi data yang belum diketahui. Situasi ini disebut sebagai *overfitting*. Untuk menghindari *overfitting*, umumnya pada *supervised machine learning* dapat dilakukan dengan membagi sebagian dari data yang tersedia sebagai dataset testing. Pemilihan parameter terbaik dapat ditentukan dengan teknik *grid search*.

Saat mengevaluasi untuk pengaturan yang berbeda ("*hyperparameter*") pada estimator, masih ada risiko *overfitting* pada dataset testing karena parameter dapat diatur hingga estimator bekerja secara optimal. Untuk mengatasi masalah ini, beberapa bagian dari dataset dapat dibagi menjadi "dataset validasi" untuk proses t*raining* yang dilakukan pada dataset *training*. Setelah evaluasi dilakukan pada dataset validasi, kemudian evaluasi terakhir dapat dilakukan pada dataset test.

Namun, dengan membagi data menjadi tiga bagian dapat mengurangi jumlah sampel yang dapat digunakan sebagai pembelajaran pada model dan hasil yang didapatkan dapat bergantung pada set data acak yang terpilih. Solusi dari permasalahan ini dapat dilakukan metode *cross-validation* (CV). Dataset test diperlukan untuk evaluasi terakhir, namun dataset validasi tidak lagi diperlukan pada CV.

Pada *k-fold* CV, dataset *training* dibagi kedalam set-set kecil sejumlah k. Model dilatih pada k-1 fold sebagai *training*, kemudian divalidasi menggunakan set data yang tersisa, yang digunakan sebagai data test untuk mengukur performa model. Tahapan ini dilakukan untuk setiap k"*folds*". Pengukuran performa menggunakan k-*fold* CV merupakan hasil rata-rata yang diukur pada loop. Pendekatan ini menjadi mahal secara komputasi, namun tidak mubazir data yang menjadi sebuah keuntungan pada jumlah sampel yang kecil. Ilustrasi *k-fold* CV dapat dilihat pada Gambar II.15 dibawah ini (Pedregosa et al., 2011).



Gambar II.2 k-*fold cross validation*

K-Fold membagi semua sampel ke dalam grup-grup sampel sejumlah k yang dinamakan fold dengan sama rata. Prediksi dilatih menggunakan k-1 fold dan sisanya digunakan sebagai testing. Apabila jumlah fold k = n (jumlah sampel), cara ini dapat disamakan dengan *Leave One Out cross-validation* . *LeaveOneOut* (LOO) adalah contoh *cross-validation* sederhana dimana setiap set training dibuat dengan menggunakan semua sampel kecuali satu. Satu sampel yang dikeluarkan ini dijadikan sebagai test set. Sehingga pada sampel akan terdapat training set dan test set yang berbeda-beda. *Cross-validation* dengan cara ini tidak menyianyiakan banyak data karena hanya satu sampel yang dikeluarkan dari set training. Dengan demikian LOO dapat digunakan untuk jumlah sampel yang sedikit.

Ketika dibandingkan dengan k-fold CV, LOO membangun n model dari n sampel dan dilatih pada n-1 sampel. K-fold membangun k model dari n sampel, dimana n > k dan dilatih pada (k-1)n/k. Dari keduanya dengan asumsi k tidak terlalu besar dan k < n, LOO lebih mahal secara komputasi dari pada k-fold CV. Dalam hal akurasi, LOO sering menghasilkan variasi tinggi pada estimator dalam test error. Secara intuitif, karena n-1 dari n sampel digunakan untuk membangun setiap model, model yang terbentuk dari fold hampir identik satu sama lain dan terhadap model yang dibangun menggunakan seluruh set training. Namun apabila kurva pembelajaran curam pada ukuran training yang digunakan, maka 5- atau 10- fold *cross-validation* dapat melebih-lebihkan sebagian besar kesalahan. Dalam aturan umum, sebagian besar penulis, dan bukti empiris, menyarankan 5 atau 10-fold lebih disukai daripada LOO. (Pedregosa et al., 2011).

Dalam penelitian ini, *Leave One Out cross-validation* dan 10-*fold cross-validation* digunakan sebagai evaluasi terhadap data training pada model yang sedang dibangun. *Cross-validation* dibangun dalam bahasa pemrograman Python dengan memanfaatkan *library*, yaitu *Pandas*, *Scikit-Learn*, dan lainnya.

2.8 *Root Mean Square Error* (RMSE)

Untuk melakukan evaluasi terhadap suatu peramalan atau prediksi dapat dilakukan dengan beberapa cara pengukuran yang salah satunya yaitu dengan menggunakan *Root Mean Square Error* (RMSE). RMSE adalah aturan penilaian kuadrat yang mengukur besarnya rata-rata kesalahan (Sutoyo & Almaarif, 2020).

RMSE merupakan besarnya tingkat kesalahan hasil prediksi, dimana semakin kecil (mendekati 0) nilai RMSE maka hasil prediksi akan semakin akurat (Sulaiman & Juarna, 2021). RMSE digunakan untuk membandingkan nilai yang diprediksi oleh model hipotetis dengan nilai dari hasil pengamatan. Dengan kata lain, RMSE mengukur kualitas kesesuaian antara data aktual dan model prediksi (Prasetyo et al., 2021).

Berikut persamaan *root means square error* (RMSE) (Budiman, 2016):

$$RMSE = \sqrt{\frac{\sum(prediksi - aktual)^2}{n}} \quad \text{(II.2)}$$

Keterangan :

n = jumlah data