BAB II
STUDI LITERATUR

2.1. Tinjauan Pustaka

Tidak terlepas dari penelitian – penelitian sebelumny. Penulisan penelitian ini meliputi penggunaan metode yang ada pada *machine learning* dalam penyelesaiannya, serta hasil yang didapat dari penelitian sebelumnya.

Penelitian pernah dilakukan oleh Putri dkk (2009), tentang Klasifikasi Penyakit Kulit Pada Manusia Menggunakan Metode Binary *Decision Tree* Support Vector Machine (BDTSVM)(Studi Kasus: Puskesmas Dinoyo Kota Malang). Dalam penelitiannya, mereka menggunakan metode Binary *Decision Tree* yang tergabung dengan Support Vector Machine untuk melakukan klasifikasi terhadap penyakit kulit pada manusia. Hasil yang didapatkan dari penelitian mereka adalah nilai akurasi yang dihasilkan sebesar 97,14% dari data yang digunakan sebanyak 150 data [7]. Sedangkan pada penelitian lainnya, yang dilakukan oleh Novianti dan Purnami (2012), tentang Analisis Diagnosis Pasien Kanker Payudara menggunakan Regresi Logistik dan *Support Vector Machine* (SVM) berdasarkan Mamografi. Pada penelitian yang dilakukan, mereka melakukan perbandingan dua algoritma, yaitu regresi logistik dan *support vector machine* (SVM) untuk mengklasifikasi kanker payudara berdasarkan data hasil mamografi. Pada algoritma SVM yang digunakan, mereka menggunakan tiga kernel yaitu linear, rbf, dan polynomial untuk mendapatkan nilai parameter yang dibutuhkan. Dengan melakukan partisi 80:80, hasil akhir yang didapatkan adalah nilai akurasi dari algoritma SVM sebesar 94,34%, sedangkan untuk nilai akurasi algoritma regresi logistik hanya sebesar 84,90% [8].

Pada penelitian Shouman dkk. (2010) tentang Penggunaan Decision Tree dalam Mendiagnosa Penyakit Jantung pada seorang pasien. Dalam penelitian yang mereka lakukan, mereka membandingkan hasil pendekatan dalam pemilihan tipe decision tree yang mereka gunakan. Untuk tipe decision tree yang mereka gunakan antara lain Information Gain, GINI Index, dan Gain Ratio. Dalam perbandingan yang dilakukan, didapatkan hasil yang menunjukkan rata-rata nilai

akurasi lebih besar pada pendekatan Gain Ratio, namun untuk nilai entropy, nilai GINI index menunjukkan akurasi yang lebih besar. Pada penelitian ini juga, mereka membandingkan hasil yang didapatkan dengan penelitian milik orang lain yang menggunakan algoritma J4.8 Decision Tree dan Bagging Algorithm. Dimana nilai akurasi yang didapatkan pada penelitian mereka sebesar 84%, dan untuk penelitian lain sebesar 78,9% dan 81,41% [9]. Sedangkan pada penelitian lainnya, yang dilakukan oleh Prasad dkk. (2013), mereka melakukan penelitian yang berjudul "*A Gini Index Based Elegant Decision Tree Classifier to Predict Precipitation*". Pada penelitian yang mereka lakukan, bentuk prediksi hasil endapan hujan yang terjadi diambil dari dataset curah hujan dan dilakukan pendekatan decision tree menggunakan GINI Index. Untuk melihat keakuratan hasil yang didapatkan, mereka melakukan perbandingan dengan SLIQ decision tree. Hasil yang didapatkan merupakan akurasi pada penelitian yang dilakukan sebesar 72,98% sedangkan untuk SLIQ decision tree menghasilkan akurasi sebesar 72,01% [10].

Penelitian lain pernah juga dilakukan oleh Khafiizh (2012) yaitu tentang Analisis Komparasi Algoritma Klasifikasi Data Mining untuk Prediksi Mahasiswa Non Aktif. Dalam penelitian yang dilakukannya, ia membandingkan beberapa algoritma seperti *logistic regression, Decision Tree, naïve bayes,* dan *neural network* untuk melakukan prediksi jumlah mahasiswa non-aktif yang ada di Universitas Dian Nuswantoro. Perbandingan algoritma yang dijalankan berguna untuk mengetahui algoritma mana yang memiliki nilai paling akurat dalam memprediksi mahasiswa non-aktif. Hasil akhir yang diperoleh dari penelitiannya tersebut menjelaskan bahwa algoritma *Decision Tree* menjadi algoritma yang memiliki nilai yang paling akurat dari algoritma yang lainnya. Namun, memiliki kelemahan dimana algoritma *Decision Tree* tidak dominan terhadap algoritma lain dalam segi T-test (pengujian hipotesis satu individu). Berbanding terbalik dengan *Decision Tree*, pada nilai T-*test* yang dilakukan, algoritma *logistic regression* merupakan algoritma yang paling dominan terhadap algoritma lainnya. Namun, algoritma ini memiliki nilai akurasi yang lebih rendah dibanding algoritma lainnya [11]. Sedangkan pada penelitian lainnya, yang pernah dilakukan oleh Setiawan dkk. (2015) tentang Ketepatan Klasifikasi Keikutsertaan Keluarga

Berencana Menggunakan Regresi Logistik Biner dan Regresi Probit Biner (Studi Kasus di Kabupaten Semarang Tahun 2014). Dalam penelitian yang dilakukan, mereka membandingkan algoritma regresi logistic biner dan algoritma regresi probit biner dalam kasus ketepatan klasifikasi keikutsertaan keluarga berencana. Hasil yang didapatkan dalam perbandingan yang dilakukan adalah algoritma yang digunakan dinilai sama baiknya dalam melakukan klasifikasi, dan untuk nilai akurasi pada regresi logistik biner sedikit lebih besar dibanding dengan regresi probit biner. Selisih perbandingannya adalah 69% : 68,4%, dimana nilai *error* yang didapatkan 31% : 31.6% [12].

Pada penelitian lain, Andriani (2013) melakukan penelitian tentang Sistem Prediksi Penyakit Diabetes berbasis *Decision Tree*. Dalam penelitian tersebut, andirani menggunakan algoritma *Decision Tree* C4.5 untuk klasifikasi dan *Microsoft Visual Basic* 6.0 beserta dataset*base MySQL* untuk pembuatan aplikasi sistem. Hasil akurasi yang didapatkan berdasarkan evaluasi menggunakan *confusion matrix* sebesar 73,33% dengan penggunaan data terbaru sebanyak 50 *record* data [13]. Sedangkan pada penelitian lainnya, penelitian lain pernah dilakukan oleh Rumaenda (2016) yaitu tentang Perbandingan Klasifikasi Penyakit Hipertensi Menggunakan Regresi Logistik Biner Dan Algoritma C4. 5 (Studi Kasus Upt Puskesmas Ponjong I, Gunungkidul). Dalam penelitian yang dilakukannya, ia membandingkan kedua metode yang dia angkat yaitu regresi logistik biner dan algoritma C4.5. Ia menggunakan regresi logistik biner dalam menjelaskan keterkaitan antara variabel respon dengan variabel prediktor yang digunakan dalam mengklasifikasi penderita hipertensi. algoritma C4.5 digunakan sebagai salah satu metode untuk mengklasifikasi data yang ia gunakan untuk membentuk pohon keputusan (*Decision Tree*). Hasil yang ia dapatkan dalam mengklasifikasi penyakit hipertensi memiliki nilai ketepatan sebesar 72.5%, sedangkan pada algoritma C4.5 nilai ketepatan yang dihasilkan sebesar 64% [14].

Penelitian lain juga dilakukan oleh Febriana dkk. (2017), tentang Klasifikasi Penyakit *Typhoid Fever* (TF) Dan *Dengue Haemorhagic Fever* (DHF) Dengan Menerapkan Algoritma *Decision Tree* C4. 5 (Studi Kasus: Rumah Sakit Wilujeng Kediri). Dalam penelitian tersebut, mereka melakukan klasifikasi terhadap

penyakit TF yang disebabkan oleh bakteri *Salmonella Typhi* dan penyakit DHF yang disebabkan oleh gigitan nyamuk *Aedes Aegypti*. Mereka menggunakan algoritma *Decision Tree* C4.5 dengan pengujian menggunakan *k-fold cross validation*. Hasil yang didapatkan dari penelitian tersebut adalah nilai rata-rata akurasi yang dihasilkan berada pada 5-*fold* dengan nilai akurasi sebesar 97% dinyatakan sebagai nilai akurasi terbaik, dikarenakan memiliki nilai akurasi rata - rata terbaik dari k-fold lainnya.. Meski terdapat nilai akurasi sebesar 100% di 16-fold [15]. Selanjutnya pada penelitian lain yang dilakukan oleh Widiastiwi dan Ernawati (2020), tentang Klasifikasi Penyakit Batu Ginjal menggunakan Algoritma *Decision Tree* C4.5 dengan Membandingkan Hasil Uji Akurasi. Pada penelitian yang dilakukan, mereka, penggunaan algoritma *Decision Tree* C4.5 digunakan untuk mengklasifikasi dan membandingkan hasil uji akurasi dari data rekam medis tentang penyakit batu ginjal. Mereka membagi proses pelatihan data (dataset *training*) menjadi tiga bagian yaitu 70%. 80% dan 90%. Hasil akhir yang didapat dari penelitian tersebut adalah akurasi terbesar diperoleh dari data training 70% dengan nilai akurasi sebesar 95,71% [16].

Penelitian lain juga dilakukan oleh Young (2017) di *Rochester Insititute of Technology*, dengan judul penelitian *Predicting Cholera Positive Cases in* Haiti. Pada penelitiannya, ia membandingkan lima metode diantaranya: *Decision Tree* (DT), *Support vector machine* (SVM), *Novelty Detection, Model Averaging*, dan *Random Forest* (RF). Dataset yang digunakan diambil melalui survey yang dilakukan oleh peneliti, dimana dataset tersebut dibagi menjadi dua bagian yaitu *Baseline* Dataset dan *Augmented* Dataset. Berdasarkan dataset yang tersedia, akan diambil nilai data berupa *Area Under the ROC Curve* (AUC), *Sensitivity*, *Specificity, F-Measure, dan Geometric Mean* yang nantinya akan dijadikan sebagai data uji performa. Dari hasil penelitian yang didapatkan, ia menyatakan bahwa *Random Forest* dapat dijadikan sebagai metode yang dapat dikatakan paling baik dari yang lainnya. Hal ini dikarenakan pada metode *Random Forest* memberikan nilai terbaik pada komponen F-*measure* dan G-*means* dibanding dengan metode lainnya [17]. Sedangkan melalui penelitian yang pernah dilakukan oleh Leo dkk. (2019), yang berjudul *Machine Learning Model for Imbalanced Cholera* Dataset *in Tanzania*. Mereka menggunakan beberapa metode dalam

penelitiannya, diantaranya: *Decision Tree* (DT), *K – Nearest Neighbor* (KNN), *Adaptive Boosting* (AdaBoost), *Linear Discriminant Analysis* (LDA), *XGBoost*, *ExtraTree*, dan *Random Forest* (RF). Dataset yang digunakan selama penelitian diambil dari tahun 1989 – 2017 melalui situs resmi WHO, dengan mengambil studi kasus dari wilayah Tanzania. Dalam penelitiannya, terdapat 3 kategori yang dijadikan tumpuan dalam menentukan metode mana yang memiliki nilai evaluasi terbaik dalam pemrosesan dataset. Ketiga kategori tersebut meliputi *Sensitivity score*, *Specificity score* dan *Balanced accuracy score*. Berdasarkan evaluasi yang dilakukan selama penelitian, terpilih dua metode pendekatan yang memiliki hasil terbaik dalam pemrosesannya, yaitu metode *XGBoost* dan KNN. Namun untuk mengetahui metode yang lebih baik diantara kedua metode tersebut, maka dilakukan kembali pengecekan performa menggunakan *Wilcoxon sign-rank test*. Setelah melalui proses evaluasi performa, *XGBoost* terpilih sebagai metode paling baik dalam melakukan prediksi kolera dari segi dataset yang digunakan [18]. Penelitian terkait dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Penelitian Terkait

No.	Judul	Penulis	Metode	Hasil
1	Klasifikasi Penyakit Kulit Pada Manusia Menggunakan Metode Binary *Decision Tree* Support Vector Machine (BDTSVM) (Studi Kasus: Puskesmas Dinoyo Kota Malang).	Putri, dkk. (2009)	*Binary Decision Tree Support Vector Machine* (BDTSVM).	Akurasi: 97.14%
2	*Using Decision Tree for Diagnosing Heart Disease Patients*	Mai Shouman, dkk. (2011)	*Decision Tree*	Akurasi: 84.1%

No.	Judul	Penulis	Metode	Hasil
3	Analisis Diagnosis Pasien Kanker Payudara menggunakan Regresi Logistik dan *Support Vector Machine* (SVM) berdasarkan Mamografi.	Novianti & Purnami (2012)	SVM & Regresi Logistik	SVM: 94.34%, Regresi Logistik:84.90%
4	Analisis Komparasi Algoritma Klasifikasi Data Mining untuk Prediksi Mahasiswa Non Aktif.	Khafiizh (2012)	*Logistic regression, Decision Tree, naïve bayes,* dan *neural network*	DT: 95.29% , LR: 81.64%, NB: 93.47%, NN: 94.56%.
5	Sistem Prediksi Penyakit Diabetes berbasis *Decision Tree.*	Andriani (2013)	*Decision Tree*	Akurasi: 73.33%
6	*A Gini Index Based Elegant Decision Tree Classifier to Predict Precipitation*	K.R. Patro, dkk. (2013)	*Decision Tree*	SLIQ *DecisionTree:* 72.01% *Elegant Decision Tree*: 72.98%
7	Ketepatan Klasifikasi Keikutsertaan Keluarga Berencana Menggunakan Regresi Logistik Biner dan Regresi Probit Biner (Studi Kasus di Kabupaten	Setiawan dkk. (2015)	Regresi Logistik Biner & Regresi Probit Biner	Regresi Logistik Biner 69% Regresi Probit Biner 68,4%

No.	Judul	Penulis	Metode	Hasil
	Semarang Tahun 2014).			
8	Perbandingan Klasifikasi Penyakit Hipertensi Menggunakan Regresi Logistik Biner Dan Algoritma C4. 5 (Studi Kasus Upt Puskesmas Ponjong I, Gunungkidul).	Rumaenda (2016)	Regresi Logisitik Biner & *Decision Tree* C4.5	Regresi Logistik Biner: 72.53%, DT C4.5: 64.08%
9	Klasifikasi Penyakit *Typhoid Fever* (TF) Dan *Dengue Haemorhagic Fever* (DHF) Dengan Menerapkan Algoritma *Decision Tree* C4. 5 (Studi Kasus: Rumah Sakit Wilujeng Kediri).	Febriana dkk. (2017)	DT C4.5	5-fold Akurasi 97%, dan 16-fold akurasi 100%
10	*Predicting Cholera Positive Cases in* Haiti	Young (2017)	*Decision Tree* (DT), *Support vector machine* (SVM), *Novelty Detection*, *Model Averaging*, dan *Random Forest* (RF).	Akurasi *F-Measure* dan *G-Mean* pada Random Forest berada diatas 75% dibandingkan dengan algoritma lainnya.

No.	Judul	Penulis	Metode	Hasil
11	*Machine Learning Model for Imbalanced Cholera* Dataset *in Tanzania*.	Leo (2019)	*Decision Tree* (DT), *K – Nearest Neighbor* (KNN), *Adaptive Boosting* (AdaBoost), *Linear Discriminant Analysis* (LDA), *XGBoost*, *ExtraTree*, dan *Random Forest* (RF)	*Balanced Accuracy* XGB 76%, lebih akurat dibanding algoritma lain.
12	Klasifikasi Penyakit Batu Ginjal menggunakan Algoritma *Decision Tree* C4.5 dengan Membandingkan Hasil Uji Akurasi.	Widiastiwi dan Ernawati (2020)	*Decision Tree* C4.5	Akurasi 95.71%

2.2. Landasan Teori

Pada sub-bab ini akan membahas landasan teori yang digunakan dalam penelitian.

1.2.1. *Machine Learning*

Dalam buku Pengenalan Pembelajaran Mesin dan *Deep Learning* edisi 1.4 karya Jan Wira Gotama Putra menyebutkan, istilah pembelajaran mesin (*Machine Learning*) merupakan bagian dari kecerdasan buatan (*Artificial Intelligence*) [19].

Istilah kecerdasan buatan mengartikan sebuah program yang memiliki bentuk matematis (instruksi), yang bertujuan untuk menciptakan program yang mampu mem-program (*output* program adalah sebuah program). Secara teori, program adalah automaton yang menjalankan suatu instruksi, yang menjadikan kecerdasan buatan berbeda dengan program biasa yaitu kecerdasan buatan merupakan program yang memiliki kemampuan untuk belajar.

Menurut Ian Goodfellow, dalam bukunya yang berjudul *Deep Learning* (2016) menyebutkan, pembelajaran mesin (*machine learning*) adalah suatu bentuk statistik terapan dengan peningkatan penekanan pada penggunaan komputer untuk memperkirakan fungsi rumit secara statistik, dimana algoritmanya mampu belajar dari data [20]. Kemampuan pembelajaran mesin didapatkan dari proses/pengalaman yang dilalui dari membaca sebuah dataset. Dataset sendiri merupakan kumpulan dari banyak contoh data.

Istilah pembelajaran mesin (*machine learning*) mengartikan suatu teknik untuk melakukan inferensi (hubungan variabel) terhadap data dengan pendekatan matematis. Kemampuan untuk belajar pada kecerdasan buatan terdiri dari bagian inti pembelajaran mesin untuk membuat model (matematis) yang merefleksikan pola-pola data. Pada dasarnya, ada 2 tujuan utama dari pembelajaran mesin, yaitu: memprediksi masa depan (*unobserved event*); dan/atau memperoleh ilmu pengetahuan (*knowledge discovery/discovering unknown structure*) [19].

Penggunaan algoritma yang digunakan dalam proses pembelajaran mesin pada dataset akan menentukan hasil yang didapatkan. Pada pembelajaran mesin, terdapat beberapa algoritma yang sering kali digunakan, contohnya seperti: pembelajaran terarah/diawasi (*Supervised learning*), dan pembelajaran tak diawasi (*Unsupervised learning*). Pada pembelajaran terarah/diawasi, mengartikan proses pembelajaran mesin yang dilakukan memiliki target atau fitur-fitur tertentu yang berasosasi dengan kejadian kemunculan target atau label. Pembelajaran terarah/diawasi dapat juga disebutkan bahwa hasil dari proses pembelajaran telah diketahui [20]. Pada pembelajaran tak diawasi, mengartikan proses pembelajaran mesin yang dilakukan memiliki banyak fitur, dan struktur yang berbeda untuk setiap kolom pada dataset. Yang menjadikan pembeda antara pembelajaran

diawasi dan pembelajaran tak diawasi, adalah pada pembelajaran tak diawasi, hasil dari proses pembelajaran tidak bergantuk pada label atau target.

Statistical learning theory pada *machine learning* adalah sebutan teknik untuk memprediksi masa depan dan/atau menyimpulkan/mendapatkan pengetahuan secara rasional dan nonparanormal. Dalam penerapannya, pemilihan sampel (*training* dataset) adalah hal yang sangat penting. Perlu diketahui, *training* adalah proses konstruksi model dan *testing* adalah proses meguji kinerja model pembelajaran. Apabila *training* dataset tidak mampu merepresentasikan suatu populasi, maka model yang dihasilkan dari proses pembelajaran (*training*) tidak bagus. Untuk mengetahui bagus atau tidaknya suatu pembelajaran mesin yang dilakukan, biasanya akan terdapat juga hal yang dinamakan *validation* data dan *test* data. Mesin akan dilatih menggunakan *training* dataset, kemudian hasil dari *training* akan diuji kinerjanya menggunakan *validation* data dan *test* data. Pada umumnya, rasio pembagian dataset pada proses pembelajaran mesin (*training*, *validation* dan *testing*) adalah 80% : 20%, dimana 80% data *training* :10% data *validation* :10% data *testing* dan 90% : 10%, dimana 90% data *training* : 5% data *validation* : 5% data *testing*. Namun berbeda halnya jika dataset yang digunakan berukuran kecil, maka pembagian proses pembelajaran mesin hanya dibagi menjadi data latih (*training)*, dan data uji (*testing)* saja. Pada umumnya, apabila *validation* data tidak didefinisikan, rasio pembagian data latih dan data uji adalah (80% : 20%), (70% : 30%), atau (50% : 50%) [19].

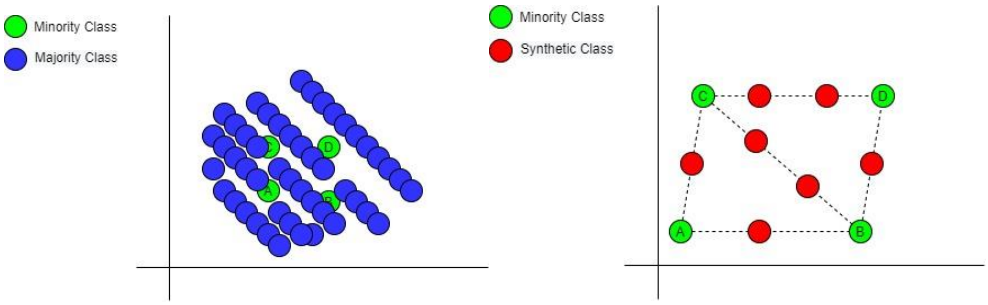
1.2.2. *Imbalanced Dataset*

Konteks dataset tidak seimbang (*imbalanced* dataset) dalam permasalahan klasifikasi pada pembelajaran mesin, adalah sebuah masalah dimana sistem melakukan pembelajaran dari sejumlah data sampel yang memiliki jumlah data yang tidak seimbang di kedua sisinya. Meski dikatakan data sampel tak seimbang, tidak ada standar kesepakatan yang tertulis secara pasti, yang menentukan bahwa suatu dataset dapat benar-benar dikatakan tidak seimbang (*imbalanced*). Dalam praktik pembelajaran mesin yang sering dilakukan, sebagian besar praktisi setuju bahwa dataset yang memiliki kelas umum kurang dari dua kali lebih umum dari kelas lainnya dengan rasio 10:1 dapat dikatakan sebagai dataset tidak seimbang

(*imbalanced*), dan dataset yang memiliki rasio kelas umum sebesar 1000:1 dapat dikatakan sebagai dataset sangat tidak seimbang (*extremely unbalanced*) [21].

1.2.3. *Synthetic Minority Oversampling Technique* (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) merupakan salah satu metode *oversampling* dimana data diciptakan melalui teknik pemilihan kelas minoritas secara acak dari titik k-*nearest neighbors* [21]. Teknik SMOTE dilakukan dengan membentuk keseimbangan antar dua kelas, dimana jumlah kelas minoritas menyamai jumlah kelas mayoritas yang ada pada dataset yang digunakan [22].



Gambar 2. 1 (kiri) *majority class* dan *minority class*, (kanan) *minority class* dan *synthtetic class.*

Pada gambar 2.1 sebelah kiri, menunjukkan contoh pembagian distribusi kelas mayoritas dan kelas minoritas pada dataset yang tidak seimbang. Sedangkan pada gambar sebelah kanan, menunjukkan distribusi dan pembentuk kelas sintetik dalam penggunaan SMOTE. Proses pembentukan kelas sintetik terdiri dari beberapa langkah dimana untuk membuat titik kelas sintetis minoritas baru, SMOTE pertama memilih titik kelas minoritas A secara acak dan menemukan nilai K ketetanggaan dari kelas minoritas terdekat. Misalnya titik sintetis kemudian dibuat dengan memilih salah satu dari tetangga terdekat B secara acak dan menghubungkan A dan B untuk membentuk segmen garis di ruang fitur. Titik sintetis dihasilkan sebagai kombinasi dari dua contoh yang dipilih A dan B [21]. Berikut merupakan persamaan yang digunakan dalam pembentukan SMOTE:

$$X_{Synth} = X_i + \delta \times (X_{knn} - Xi) \tag{2.1}$$

Keterangan:

X_{synth} : nilai sintetis yang dihasilkan.

X_i : nilai yang akan direplikasi.

X_{knn} : nilai ketetanggan terdekat dengan data yang akan direplikasi.

δ : nilai acak antara 0 dan 1.

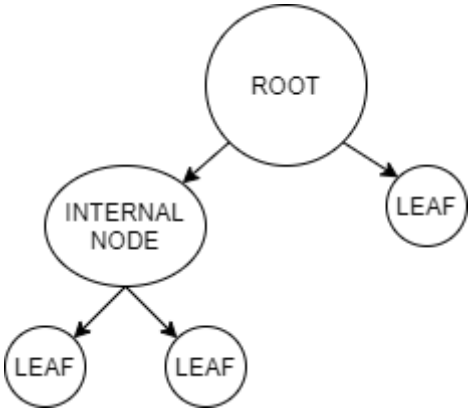
1.2.4. *Random Undersampling*

Random undersampling merupakan salah satu metode sampling dalam mengubah dataset yang tak seimbang (*imbalanced*) menjadi dataset yang seimbang (*balanced*). Proses yang dilakukan dalam random undersampling adalah dengan cara menghilangkan contoh kelas mayoritas dari data sampel pelatihan (data *training*), dimana proses yang dilakukan tersebut dapat menyebabkan kehilangan informasi pada model yang dilakukan pelatihan [21].

Dalam contoh penerapannya, pada penelitian yang dilakukan kali ini terdapat sebanyak 451 data sampel untuk kelas diare akut, dan sebanyak 31 data sampel untuk kelas diare kronis. Dengan rasio *imbalanced* 14:1, *random undersampling* akan dilakukan dengan mengambil kumpulan data kelas mayoritas terkecil yaitu 31. Untuk kelas mayoritas lainnya akan dibuang secara acak hingga distribusi seimbang tercapai. Hasil akhir yang didapat dari *random undersampling* adalah jumlah data yang digunakan berakhir sama yaitu 31 data sampel (acak) untuk diare akut dan 31 data sampel untuk diare kronis.

1.2.5. *Decision Tree*

Salah satu algoritma yang sering kali ditemukan dalam menemukan sebuah pengetahuan dan dalam pengenalan pola adalah algoritma *Decision Tree*. Algoritma *decision tree* adalah sebuah metode non-parametrik yang efisiensinya digunakan untuk melakukan tugas klasifikasi ataupun tugas regresi. Pada algoritma pohon keputusan dalam pembelajaran mesin, struktur data hierarki yang digunakan pada pohon keputusan merupakan struktur data yang sering digunakan dalam pembelajaran terarah (*supervised learning*), dimana ruang *input* dibagi menjadi beberapa wilayah lokal untuk memberikan prediksi pada variabel dependen [23]. Contoh pohon keputusan dapat dilihat di Gambar 2.2.



Gambar 2. 2 Struktur *decision tree*

Pada Gambar 2.2, merupakan contoh dari sebuah pohon keputusan. Sebuah pohon keputusan pada dasarnya dapat dilihat sebagai sebuah graf. Graf yang membangun sebuah pohon keputusan terdiri dari tiga jenis *node*, antara lain:

1. Akar (*Root*), adalah *node* pertama atau *node* yang memiliki derajat tertinggi dari sebuah pohon keputusan. *Node root* tidak memiliki *input* dan dapat mempunyai *output* satu atau lebih dari satu, baik itu *internal node* atau *leaf*.
2. Cabang (*Internal Node*), adalah *node* percabangan, dimana *node* ini hanya memiliki satu *input* atau tidak ada *input* sama sekali.
3. Daun (*Leaf*), adalah *node* akhir atau terminal *node*. *Node* ini hanya memiliki satu *input* dari *internal node* atau dari *root*, dan tidak memiliki *output* sendiri (simpul terminal).

1.2.6. *Decision Rule*

Pada algoritma pohon keputusan, dalam setiap pemilihan keputusan terdapat alur atau syarat aturan yang harus dipenuhi, dalam pengambilan keputusan baik itu dalam kasus klasifikasi atau regresi. Aturan keputusan (*decision rule*) merupakan sebutan untuk aturan yang menetapkan suatu nilai yang tercipta dari pohon keputusan. Menurut Myles (2004), sebuah pohon keputusan sendiri merupakan model hierarki yang terdiri dari fungsi diskriminan atau aturan keputusan, yang diterapkan secara rekursif untuk mempartisi bentuk ruang fitur dari kumpulan data menjadi sub-ruang kelas tunggal yang murni. Aturan keputusan adalah ekspresi ketidaksamaan yang menggambarkan sebuah batasan keputusan [24].

Pada penelitian ini, *GINI Index* digunakan dalam hal pengambilan aturan keputusan yang digunakan. *GINI index* merepresentasikan ukuran dari seberapa seringnya suatu objek yang dipilih secara acak dari sebuah proses pelatihan data [24]. Persamaan *GINI index* diberikan pada persamaan 2.2:

$$I_G(i) = 1 - \sum_{j=1}^{m} f^2(i,j) \quad (2.2)$$

Keterangan:

I_G : *GINI index*

f(i,j) : Proporsi frekuensi objek milik kelas *j* pada node ke-*i*

m : Objek kelas berbeda

Sedangkan pada percabangan yang terbentuk dari percabangan node '*parent'* menjadi partisi p '*children'*, *GINI splitting index* digunakan sebagai aturan keputusan dalam menentukan kualitas percabangan yang diberikan. Kualitas yang dimaksud merupakan nilai optimal percabangan pada sebuah node, dimana nilai optimal yang tercipta merupakan nilai terkecil, atau dapat juga dikatakan sebagai nilai yang mendekati nilai 0 (nol). Persamaan GINI splitting index diberikan pada pada persamaan 2.3:

$$GINI_{split} = \sum_{i=1}^{p} \frac{n_i}{n} GINI(i) \quad (2.3)$$

Keterangan:

p : Partisi kelas

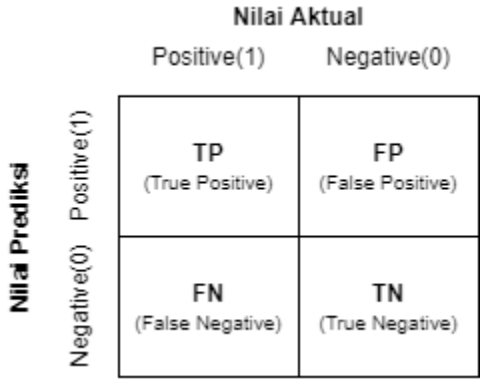
n_i : Jumlah objek per kelas

n : Total objek

i : node ke-i

1.2.7. *Confusion Matrix*

Confusion matrix adalah bentuk metode ukur atau bentuk evaluasi yang dilakukan untuk menemukan performa terbaik dari nilai akurasi yang didapatkan setelah data selesai melalui proses pelatihan (*training*) dan pengujian (*testing*) [18]. Pada *confusion matrix*, nilai akurasi akan dibandingkan dengan nilai aktual yang ada. Bentuk *confusion matrix* dapat dilihat pada Gambar 2.3.



Gambar 2. 3 *Confusion matrix*

Pada Gambar 2.3, nilai *True Positive* (TP) merupakan nilai yang didapat apabila nilai aktual terprediksi dengan nilai benar positif dan *True Negative* (TN) merupakan nilai yang didapat apabila nilai aktual terprediksi dengan nilai benar negatif. Kedua nilai TP dan TN mengindikasikan tingkat ketepatan klasifikasi yang didapatkan. Semakin tinggi nilai TP dan TN, maka akan semakin baik pula nilai ketepatan klasifikasi yang didapatkan. Nilai *False Positive* (FP) didapatkan apabila nilai aktual bernilai negatif namun terprediksi dengan nilai positif. Sedangkan *False Negative* (FN) didapatkan apabila nilai aktual bernilai positif terprediksi dengan nilai negatif.. Terdapat beberapa indikasi nilai pada *confusion matrix*, seperti nilai akurasi, presisi dan *recall*. Akurasi merupakan nilai yang dihasilkan dari tingkat kesamaan antara nilai prediksi dengan nilai aktual. Rumus yang digunakan untuk mencari nilai akurasi dapat dilihat pada persamaan 2.4:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.4}$$

Keterangan:

TP : *True Positive*

TN *: True Negative*

FP *: False Positive*

FN*: False Negative*

Nilai presisi merupakan tingkat ketelitian atau tingkat ketepatan dalam proses pengklasifikasian. Rumus yang digunakan untuk mencari nilai dari presisi dapat dilihat pada persamaan 2.5:

$$Presisi = \frac{TP}{TP+FP} \quad (2.5)$$

Keterangan:

TP : *True Positive*

TN *: True Negative*

FP *: False Positive*

FN *: False Negative*

Nilai *recall* merupakan nilai yang berfungsi untuk mengukur proporsi positif aktual yang benar diidentifikasi. Rumus yang digunakan untuk mencari nilai dari *recall* dapat dilihat pada persamaan 2.6:

$$Recall = \frac{TP}{TP+FN} \quad (2.6)$$

Keterangan:

TP : *True Positive*

TN *: True Negative*

FP *: False Positive*

FN *: False Negative*