BAB II

LANDASAN TEORI

2.1 *Customer Churn*

Customer churn didefinisikan sebagai kecenderungan pelanggan untuk berhenti melakukan bisnis dengan sebuah perusahaan dalam periode waktu tertentu (Santosa & Yuliantara, 2017).

Pelanggan yang *churn* dapat dibagi menjadi dua kelompok utama (Baizal, Bijaksana, & Sastrawan, 2009), yaitu:

1. *Voluntary churners*, *churner* ini lebih sukar untuk ditentukan, sebab pada pelanggan jenis ini *churn* terjadi ketika seorang pelanggan membuat keputusan.
2. *Involuntary churners*, *churner* ini lebih mudah untuk diidentifikasi, seperti pelanggan yang menggunakan jasa ditarik atau dicabut dengan sengaja oleh perusahaan tersebut dikarenakan adanya beberapa alasan.

2.2 Teknik Klasifikasi

Teknik klasifikasi merupakan bagaimana mempelajari sekumpulan data sehingga dihasilkan aturan yang bisa mengklasifikasi atau mengenali data-data baru yang belum pernah dipelajari (Suyanto, 2017). Menurut Zaki et al (dalam Suyanto, 2017) klasifikasi adalah proses untuk menyatakan suatu objek data sebagai salah satu kategori (kelas) yang telah didefinisikan sebelumnya. Klasifikasi banyak

digunakan dalam berbagai aplikasi, diantaranya adalah deteksi kecurangan, pengelolaan pelanggan, diagnosis medis, prediksi penjualan, dan sebagainya (Suyanto, 2017).

Secara umum dalam proses klasifikasi memiliki dua proses (Jayanti, Novianti, & Sumalya, 2017), yaitu:

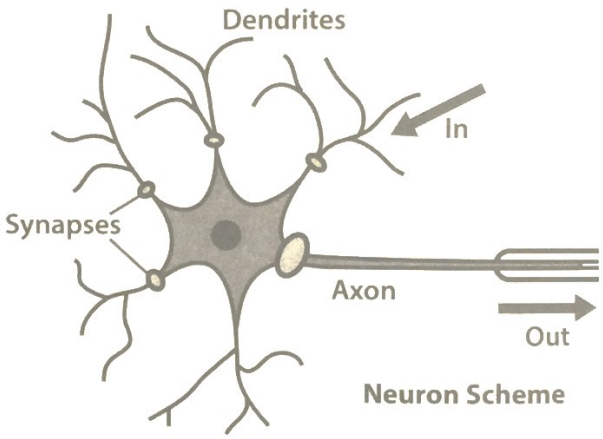
1. Proses *training*: pada proses *training* digunakan *training set* yang telah diketahui label-labelnya untuk membangun model atau fungsi.
2. Proses *testing*: untuk mengetahui keakuratan model atau fungsi yang akan dibangun pada proses *training*, maka digunakan data yang disebut dengan *testing set* untuk memprediksi label-labelnya.

2.3 Jaringan Saraf Tiruan

Artificial Neural Network (Jaringan Saraf Tiruan) adalah suatu jaringan yang memodelkan sistem saraf otak manusia yang disebut *neuron* dalam melaksanakan tugas pengenalan pola, khususnya klasifikasi (Suyanto, 2017). Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisir *neuron* sehingga mampu mengenali pola secara efektif (Suyanto, 2017).

Ide dasar *neural network* dimulai dari otak manusia, dimana otak memuat sekitar 10^{11} *neuron*. *Neuron* ini berfungsi memproses setiap informasi yang masuk. Satu *neuron* memiliki 1 akson, dan minimal 1 dendrit (Budiharto & Suhartono, Artificial Intelligence Konsep dan Penerapannya, 2014). Setiap sel saraf terhubung dengan saraf lain, jumlahnya mencapai sekitar 10^4 sinapsis. Masing-masing sel itu

saling berinteraksi satu sama lain yang menghasilkan kemampuan tertentu pada kerja otak (Budiharto & Suhartono, Artificial Intelligence Konsep dan Penerapannya, 2014). Gambar struktur neuron pada otak manusia dapat dilihat pada Gambar 2.1.

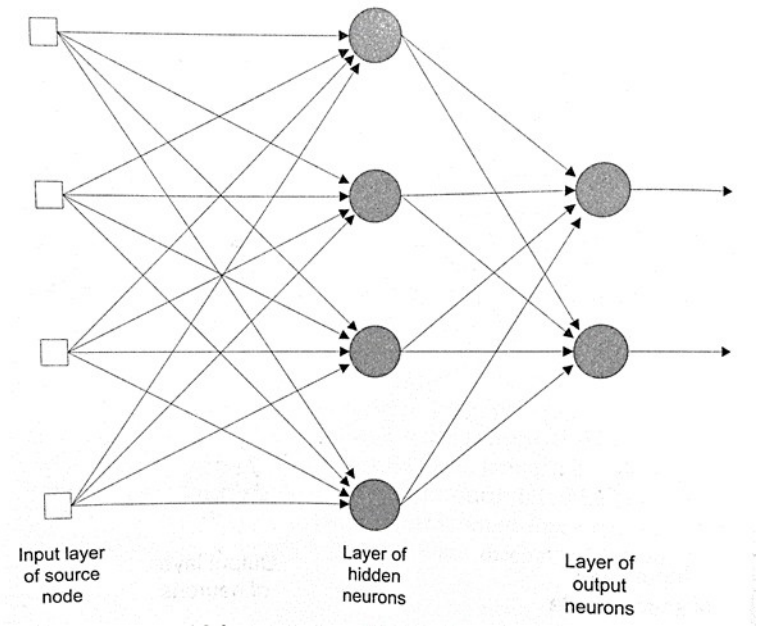


Gambar 2. 1 Struktur Neuron pada Otak Manusia

Sumber: Budiharto & Suhartono (2014)

2.3.1 *Multi Layer Perceptron*

Multi layer perceptron (MLP) adalah arsitektur jaringan yang memiliki banyak lapisan. *Multi layer perceptron* memiliki satu atau lebih *hidden layer*, dengan *computation nodes* yang berhubungan disebut *hidden neurons* atau *hidden units* (Suyanto, 2017). *Multilayer* mampu menyelesaikan lebih banyak permasalahan yang rumit dibandingkan dengan *single layer* (Budiharto & Suhartono, Artificial Intelligence Konsep dan Penerapannya, 2014). Jaringan jenis ini diilustrasikan pada Gambar 2.2 untuk kasus empat *input nodes*, empat neuron pada satu *hidden layer*, dan dua neuron pada *output layer*.



Gambar 2. 2 Arsitektur *Multi Layer Perceptron*

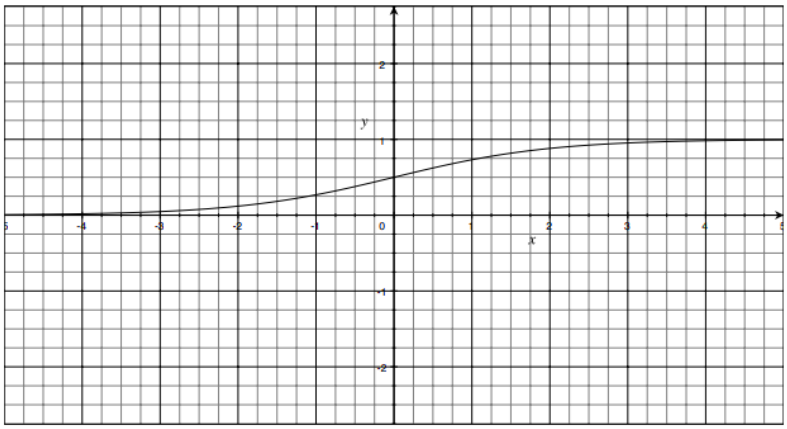
Sumber: Suyanto (2017)

2.3.2 *Activation Function*

Pada jaringan saraf tiruan terdapat fungsi aktivasi. Fungsi aktivasi ini digunakan untuk menentukan *output* suatu *neuron* berdasarkan proses yang dilakukan terhadap *input* yang dimasukkan (Budiharto & Suhartono, 2014). Salah satu fungsi aktivasi adalah sigmoid logistik. Fungsi aktivasi sigmoid menggunakan fungsi sigmoid untuk menentukan aktivasi. Istilah sigmoid berarti melengkung ke dua arah, seperti huruf "S" yang dapat dilihat pada Gambar 2.3. Fungsi sigmoid dapat didefinisikan sebagai berikut (Heaton, 2008):

$$f(x) = \frac{1}{1 + e^{-x}}$$

Rumus 2. 1 Rumus Fungsi Aktivasi Sigmoid



Gambar 2. 3 Fungsi Sigmoid

Sumber: Heaton (2008)

2.3.3 Algoritma *Backpropagation*

Algoritma pelatihan yang dapat digunakan untuk melatih *multi layer perceptron* adalah *backpropagation*. Algoritma ini melakukan pelatihan *multi layer perceptron* dalam dua tahap, yaitu perhitungan maju untuk menghitung galat antara keluaran aktual dan target, dan perhitungan mundur yang mempropagasikan balik galat tersebut untuk memperbaiki bobot-bobot sipnatik pada semua neuron yang ada (Suyanto, 2017). Pada kasus pengidentifikasian algoritma *backpropagation* dapat mewujudkan sistem yang tahan akan kerusakan (*robustness*), yang artinya sistem tidak akan terpengaruh oleh gangguan yang akan mengacaukan sistem (Pristanti & Windana, 2015).

Menurut Han et al, algoritma *backpropagation* bekerja melalui proses secara iteratif menggunakan data *training*, membandingkan nilai prediksi dari jaringan dengan setiap data yang terdapat pada data *training* (Irawan & Wahono,

2015). Langkah pembelajaran dalam algoritma *backpropagation* adalah sebagai berikut (Irawan & Wahono, 2015):

1. Menginisialisasi bobot jaringan secara acak. Bobot jaringan yang umumnya digunakan berkisar antara -0.1 sampai 1.0.
2. Menghitung input untuk simpul berdasarkan nilai input dan bobot jaringan tersebut untuk setiap data *training*, berdasarkan rumus:

$$Input\ j = \sum_{i=1}^{n} O_i W_{ij} + \theta_i$$

Rumus 2. 2 Nilai Input Simpul

Keterangan:

O_i = *Output* simpul I dari *layer* sebelumnya

W_{ij} = Bobot relasi dari simpul I pada *layer* sebelumnya ke simpul j

θ_i = Bias (sebagai pembatas)

3. Berdasarkan langkah sebelumnya kemudian membangkitkan *output* untuk simpul menggunakan fungsi aktivasi sigmoid:

$$F = \frac{1}{1 + e^{-input}}$$

Rumus 2. 3 Fungsi Aktivasi *Backpropagation*

4. Menghitung nilai *error* antara nilai sesungguhnya dengan nilai prediksi menggunakan rumus:

$$Error\ j = Output_j \times (1 - Output_j) \times (Target_j - Output_j)$$

Rumus 2. 4 Nilai *Error* Sesungguhnya

5. Setelah nilai *error* dihitung, selanjutnya dibalik ke *layer* sebelumnya (*backpropagation*). Untuk menghitung nilai *error* pada *hidden layer*, menggunakan rumus:

$$Error\ j = Output_j \times (1 - Output_j) \sum_{k=1}^{n} Error_k\, W_{jk}$$

Rumus 2. 5 Nilai *Error Hidden Layer*

Keterangan:

$Output_j$ = *Output* aktual dari simpul j

$Error_k$ = *Error* Simpul k

W_{jk} = Bobot relasi dari simpul j ke simpul k pada *layer* berikutnya

6. Setelah nilai *error* dihitung kemudian bobot relasi diperbaharui menggunakan rumus:

$$W_{ij} = W_{ij} + I \times Error_j \times Output_i$$

Rumus 2. 6 Perbaharui Bobot

Keterangan:

W_{ij} = Bobot relasi dari unit I pada *layer* sebelumnya ke unit j

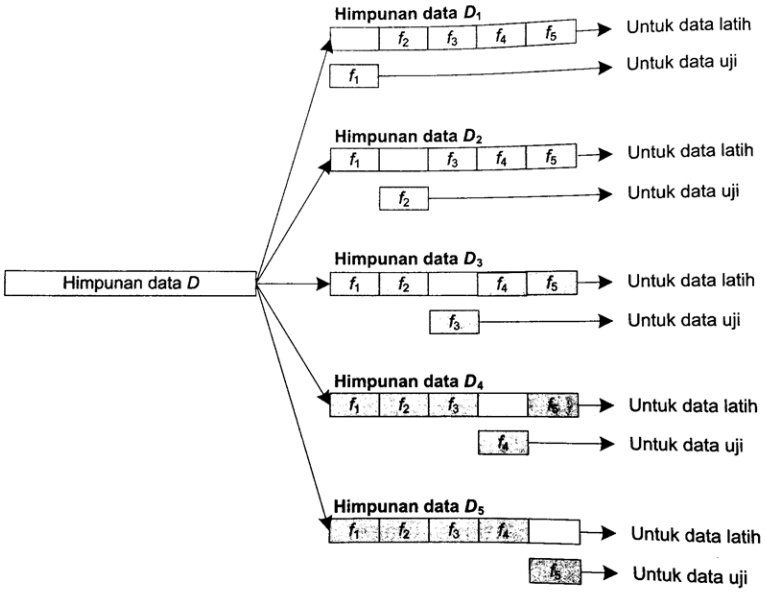
I = *Learning Rate*

$Error_j$ = *Error* pada *output layer* simpul j

$Output_i$ = *Output* dari simpul i

2.4 *K-Fold Cross Validation*

K-Fold Cross Validation merupakan salah satu cara untuk mengukur kualitas dari *classifier* dengan membagi data ke dalam *fold* (Suyanto, 2017). Metode *k-Fold Cross Validation* mempartisi himpunan data D secara acak menjadi *k fold* (subhimpunan) yang saling bebas: $f_1, f_2, \dots, f_k$, sehingga masing-masing *fold* berisi 1/k bagian data (Suyanto, 2017). Misalnya, dengan k = 5, maka himpunan data D_1 berisi empat *fold*, yaitu f_2, f_3, f_4, dan f_5 untuk data latih serta satu *fold* f_1 untuk data uji. Himpunan data D_2 berisi empat *fold*, yaitu f_1, f_3, f_4, dan f_5 untuk data latih serta satu *fold* f_2 untuk data uji. Demikian seterusnya untuk himpunan data D_3, D_4, D_5 (Suyanto, 2017). Ilustrasi metode *k-fold cross validation* dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Ilustrasi *K-Fold Cross Validation*

Sumber: Suyanto (2017)

2.5 *Confusion Matrix*

Confusion matrix dapat digunakan untuk mengevaluasi kualitas dari *classifier* (Han, Kamber, & Pei, 2012). *Confusion matrix* menurut Han dan Kamber (dalam Fibrianda & Bhawiyuga (2018) dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah *classifier* tersebut baik dalam mengenali *tuple* dari kelas yang berbeda. Nilai dari *true positive* dan *true negative* memberikan informasi ketika *classifier* dalam melakukan klasifikasi data bernilai benar sedangkan *false positive* dan *false negative* memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data (Fibrianda & Bhawiyuga, 2018).

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P'	N'	$P + N$

Gambar 2. 5 *Confusion Matrix*

Sumber: Fibrianda & Bhawiyuga (2018)

Berdasarkan *confusion matrix* yang dapat dilihat pada Gambar 2.5 di atas dapat disimpulkan (Fibrianda & Bhawiyuga, 2018):

a. *True Positives (TP)* adalah jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif.
b. *False Positives (FP)* adalah jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif.

c. *False Negatives (FN)* adalah jumlah data dengan nilai sebenarnya positif dan nilai prediksi negatif.
d. *True Negatives (TN)* adalah jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif.

Nilai yang dihasilkan melalui metode *Confusion Matrix* adalah berupa evaluasi sebagai berikut (Singh, Tiwari, & Singh, 2018):

1. *Accuracy*, persentase keakuratan *classifier* dalam mengklasifikasi *test set* dengan benar. *Accuracy* (akurasi) dapat dihitung dengan rumus berikut:

$$Accuracy = \frac{TP+TN}{P+N} \times 100$$

Rumus 2. 7 Rumus *Accuracy* Pada *Confusion Matrix*

2. *Error rate (misclassficiation rate)* dari *classifier* adalah 1-*accuracy*. Dapat juga dihitung dengan menggunakan rumus berikut:

$$Error\ Rate = \frac{FP+FN}{P+N}$$

Rumus 2. 8 Rumus Error Rate Pada *Confusion Matrix*

3. *Precision* merupakan ukuran ketepatan untuk mengetahui persentase *tuple* yang diberi label positif dan benar-benar positif.

$$Precision = \frac{TP}{TP+FP}$$

Rumus 2. 9 Rumus *Precision* Pada *Confusion Matrix*

4. *Recall* atau *true positive rate* merupakan porsi dari *tuple* positif yang diklasifikasi dengan benar.

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

Rumus 2. 10 Rumus *Recall* Pada *Confusion Matrix*

5. Alternatif untuk menggabungkan *precision* dan *recall* menjadi satu ukuran adalah dengan pendekatan *F-Measure* yang juga dikenal sebagai *F1 score* atau *F-score* (Suyanto, 2017). *F-Measure* menilai ketepatan dan kekokohan sebuah *classifier* dengan rata-rata harmonik dari nilai *precision* dan *recall*-nya (Singh, Tiwari, & Singh, 2018). *F-Measure* dihitung dengan rumus berikut:

$$F\text{-}Measure = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

Rumus 2. 11 Rumus *F-Measure* Pada *Confusion Matrix*

2.6 R

R adalah bahasa dan *environtment* (lingkungan) untuk komputasi statistik dan grafik yang dibuat oleh Ross Ihaka dan Robert Gentleman (Primartha, 2018). R sudah dikompilasi untuk berbagi sistem operasi, seperti UNIX, Linux, Windows, dan MacOS. R merupakan aplikasi sistem statistik yang dapat mendukung *linear* dan *nonlinear modelling*, *classical statistical tests*, *time-series analysis*, *classification*, dan *clustering* (Primartha, 2018).

R merupakan rangkaian *software* terintegrasi yang dapat digunakan untuk manipulasi data, perhitungan, dan tampilan grafis. R memiliki beberapa fasilitas, yaitu penanganan dan penyimpanan data yang efektif, *intermediate tools* untuk analisis data yang memiliki koleksi yang terintegrasi, menyediakan fasilitas grafis untuk analisis data baik secara langsung di komputer ataupun *hardcopy*, dan bahasa pemrograman yang dikembangkan dengan sederhana dan efektif yang mencakup *conditionals*, *loop*, fungsi rekursif dan fasilitas *input* dan *output* (Venables, Smith, & RCoreTeam, 2019). *Software* yang bersifat *open source* ini memiliki satu fitur terbaik, yaitu memiliki *collaborative repository* yang sangat besar dan disebut dengan CRAN yang memiliki lebih dari 7.300 *packages* untuk berbagai tujuan (Pacheco, 2015).

2.7 Microsoft Excel

Microsoft Excel adalah program perangkat lunak yang dirancang untuk membantu mengevaluasi dan menyajikan informasi dalam format *spreadsheet* (Berk & Carey, 2010). *Spreadsheet* paling sering digunakan oleh bisnis untuk analisis arus kas, laporan keuangan, dan manajemen inventaris. Microsoft Excel merupakan aplikasi yang sangat fleksibel sehingga dapat melampaui *spreadsheet* tradisional ke bidang analisis data. Excel dapat digunakan untuk memasukkan data, menganalisis data dengan tes statistik dasar dan bagan, dan juga membuat laporan yang merangkum temuan (Berk & Carey, 2010).

2.8 Penelitian Terdahulu

Penelitian terdahulu merupakan penelitian telah dilakukan oleh peneliti sebelumnya untuk menjadi referensi bagi penelitian saat ini dalam membangun model klasifikasi.

2.8.1 Penelitian Pertama

Aulck, Velagapudi, Blumenstock, dan West melakukan penelitian untuk memprediksi mahasiswa yang akan *dropout* dari *University of Washington*. Data yang digunakan berisi informasi yang termasuk tapi tidak terbatas pada: ras, jenis kelamin, tanggal lahir, status penduduk, catatan transkrip, dan jurusan untuk semua mahasiswa di *University of Washington*. Fokus penelitian ini adalah mahasiswa S1 yang terdaftar antara tahun 1998 dan tahun 2006 (Aulck, Velagapudi, Blumenstock, & West, 2016).

Mereka mengambil *sample* secara acak dengan jumlah yang sama dari kelas *completion* dan *non-completion* untuk membuat *dataset* yang *balance* dan terdiri dari 32.538 mahasiswa. Dengan menggunakan 70% data yang diambil secara acak, mereka menggunakan *10-fold cross-validation* untuk menyesuaikan parameter model. 30% dari sisa data akan digunakan sebagai data *test*. Data *test* tersebut tidak digunakan dalam *10-fold cross-validation*. Metode yang digunakan untuk membangun model adalah *logistic regression*, *random forest,* dan *k-nearest neighbors.* Parameter model yang disesuaikan adalah kekuatan regularisasi untuk

logistic regression, jumlah tetangga di *k-nearest neighbors*, dan kedalaman pohon di *random forest.*

Kesimpulan penelitian di atas adalah nilai dalam mata pelajaran matematika, bahasa inggris, kimia, dan psikologi termasuk *predictor* terkuat untuk *churn*. Hasil akurasi yang didapatkan adalah sebesar 66.59% untuk *logistic regression*, 62.24% untuk *random forest,* dan 64.60% untuk *k-nearest neighbors.*

2.8.2 Penelitian Kedua

Delen melakukan penelitian tentang analisis komparatif menggunakan beberapa teknik *machine learning* untuk *student attrition*. Ada empat metode klasifikasi yang digunakan untuk penelitian ini, yaitu *artificial neural networks, decision trees, support vector machines,* dan *logistic regression* (Delen D. , 2010).

Delen melakukan dua kali eksperimen pada model prediksi tersebut. Di eksperimen yang pertama, ia menggunakan *original dataset* yang berisi 16.066 *records*, sedangkan di eksperimen kedua, ia menggunakan *well-balanced dataset* dimana kedua kelas memiliki jumlah data yang sama. Ia mengambil semua sampel dari kelas minoritas dan secara *random* memilih jumlah sampel yang sama dari kelas mayoritas. Proses pengambilan sampel ini menghasilkan kumpulan data dari 7.018 *records*. Data yang berjumlah 3.509 diberi label sebagai "Tidak" dan 3.509 data lainnya diberi label sebagai "Ya".

Hasil analisis menunjukkan bahwa *dataset* yang *balance* menghasilkan hasil prediksi yang lebih baik daripada *dataset* yang tidak *balance*. Penelitian sebelumnya juga berpendapat tentang pentingnya memiliki *dataset* yang seimbang untuk membangun model prediksi yang akurat dalam masalah klasifikasi biner (Delen D. , 2010). Berdasarkan hasil dari *10-fold cross-validation, support vector machines* menghasilkan hasil terbaik dengan akurasi prediksi secara keseluruhan sebesar 81.18%, diikuti oleh *decision trees, artificial neural networks,* dan *logistic regression* dengan akurasi prediksi keseluruhan 80,65%, 79,85% dan 74,26%. Ia juga melakukan *sensitivity analysis* dari model yang dikembangkan dan mendapatkan kesimpulan bahwa variabel pendidikan dan keuangan adalah salah satu *predictor* terpenting dari fenomena tersebut.