

# Penggunaan *Deep Learning* untuk Prediksi *Churn* pada Jaringan Telekomunikasi *Mobile*

Fikrieabdillah

Ilmu Komputasi Universitas  
Telkom Bandung 40257,  
Indonesia

[fikrieabdillah@gmail.com](mailto:fikrieabdillah@gmail.com)

---

## Abstrak

Turun naiknya jumlah pelanggan merupakan salah satu masalah besar pada perusahaan jaringan telekomunikasi selular. Pada tugas akhir ini, akan dibangun sebuah sistem prediksi terhadap turun naiknya pelanggan pada layanan jaringan telekomunikasi selular menggunakan arsitektur jaringan *Multilayer Perceptron*. Selain menggunakan arsitektur jaringan *Multilayer Perceptron* dilakukan juga implementasi sistem dengan menggunakan *Autoencoder* untuk mendapatkan bobot yang optimal. Dari metode dan sistem yang digunakan dilakukan perhitungan menggunakan *F-Measure* didapatkan nilai *precision* sebesar 70,2 % dan *Recall* sebesar 70,27 %. Untuk nilai akurasi sebesar 81,35 % pada data *training*. Didapatkan nilai *Precision* sebesar 80,4 % dan *Recall* sebesar 63,07 %. Pada nilai akurasi sebesar 83,12 % untuk data *testing*. Dengan nilai akurasi yang cukup besar maka metode *Deep Learning* layak untuk digunakan dalam prediksi *Churn*.

**Kata Kunci** : Prediksi *Churn*, *Deep Learning*, *Deep Neural Network*, *Multilayer Perceptron*, *Autoencoders*, *F-Measure*

## Abstract

The fluctuation of the amount of subscribers is one of the biggest problem in Telecommunication based companies. In this final project, will be built a prediction system for the fluctuation of the amount of subscribers of mobile telecommunication service by using *Churn* prediction that is implemented by using *Multilayer Perceptron* architecture. Apart from using *Multilayer Perceptron* architecture, also done a system implementation by using *Autoencoder* to gain an optimal weight Of the methods and systems used to do the calculation using the *F-Measure* precision values obtained 70.2 % and amounted to 70.27 % Recall . For a value of 81.35 % accuracy on training data . Precision value obtained by 80.4 % and amounted to 63.07 % Recall . At a value of 83.12 % accuracy for data testing. With the value of accuracy is big enough then *Deep Learning* method viable for use in the prediction of *Churn* .

**Keywords**: *Churn* Prediction, *Deep Learning*, *Deep Neural Network*, *Multilayer perceptron*, *autoencoders*, *F-Measure*

## I. PENDAHULUAN

*Customer Churn* atau turun naiknya pelanggan adalah masalah mendasar bagi perusahaan jaringan telekomunikasi selular, yang didefinisikan sebagai hilangnya pelanggan karena pindah ke pesaing atau layanan lain. Untuk meningkatkan penjualan suatu perusahaan, dapat dilakukan dengan mempertahankan pelanggan lama agar tidak beralih menggunakan produk perusahaan lain. Untuk itu perusahaan membutuhkan strategi agar pelanggan tetap menggunakan produknya. Strategi tersebut dapat dilakukan dengan

memprediksi jumlah pelanggan yang akan berpindah ke perusahaan lain. Dengan prediksi *Churn* memberikan suatu pengetahuan yang tinggi kepada perusahaan tentang bagaimana cara mempertahankan dan meningkatkan basis pelanggan. Dengan menggunakan prediksi *Churn* dapat menampung kapasitas data dalam jumlah besar untuk memprediksi pelanggan[1]. Untuk prediksi *Churn* digunakan metode dan beberapa algoritma. Metode yang digunakan pada model ini adalah metode *Deep Learning* dan arsitektur

jaringan *Multilayer Perceptron*[3,4]. Metode *Deep Learning* termasuk dalam bagian pengklasifikasian *Machine Learning* yang di dalamnya memiliki performa sistem yang baik dan menghasilkan sistem yang optimal untuk melakukan pengolahan data. Metode *Deep Learning* yang digunakan untuk menguji kelayakan dari prediksi *Churn* menggunakan arsitektur jaringan *Multilayer Perceptron* [3-4]. Arsitektur jaringan *Multilayer Perceptron* mengolah inputan atribut – atribut data yang besar dengan ditambah inputan bobot dan bias perlayer.

Inputan bobot dan bias tidak didapat secara acak melainkan dihitung dengan *autoencoders*. Dari *autoencoders* akan didapat nilai bobot dan bias yang optimal berdasarkan data yang digunakan. Oleh karena itu dalam pelaksanaan tugas akhir ini, akan dilakukan analisis prediksi *Churn* menggunakan metode *Deep Learning* dan arsitektur jaringan *Multilayer Perceptron*. Tujuan dari analisis tugas akhir ini diharapkan dapat membantu perusahaan untuk mengambil tindakan setelah mengetahui hasil prediksi *Churn*.

**II. LANDASAN TEORI**

**A. Prediksi Churn**

Prediksi *Churn* adalah salah satu kasus penggunaan Data Besar yang paling populer dalam memprediksi turun naiknya pelanggan. Yang berisikan mengenai cara mendeteksi suatu perusahaan yang kehilangan pelanggan atau pindah ke pesaing[3]. Dengan menggunakan prediksi churn perusahaan dapat mengetahui perkembangan pada beberapa tahun ke depan. Perusahaan dapat mengantisipasi banyaknya pelanggan yang berpindah ke layanan lain.

**B. Customer Churn**

*Churn* adalah kata yang berasal dari *change* yang artinya perubahan dan *turn* yang artinya pergantian. Itu artinya penghentian kontrak. Ada tiga jenis churn:

1. **Aktif** - pelanggan memutuskan untuk berhenti nya kontrak dan beralih ke layanan lain. Alasan untuk ini mungkin termasuk ketidakpuasan dengan kualitas layanan (misalnya tidak sesuai perjanjian yang ditawarkan),
2. **rotasi / insidental** - pelanggan berhenti kontrak tanpa tujuan beralih ke layanan lain. Alasan

untuk hal ini adalah perubahan keadaan yang mencegah pelanggan dari lebih membutuhkan layanan.

3. **discontinues perusahaan** - pasif / non-sukarela kontrak itu sendiri.

**C. Deep Learning**

*Deep Learning* (*deep machine learning, or deep structured learning, or hierarchical learning, or sometimes DL*) adalah cabang dari *machine learning* berdasarkan satu set algoritma yang digunakan untuk model abstraksi tingkat tinggi pada data dengan menggunakan beberapa lapisan implementasi dan menggunakan struktur yang kompleks atau sebaliknya, terdiri dari beberapa transformasi non-linear[4].

**1. Autoencoders**

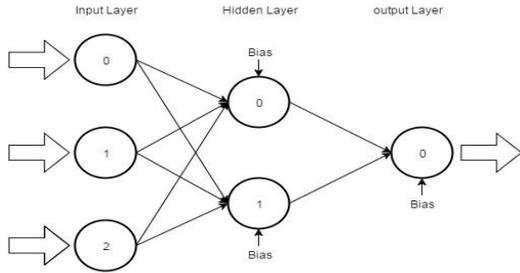
Salah satu penemuan penting dalam *Deep Learning* sejak sekitar tahun 2000 adalah menggunakan *Deep Belief Network* untuk melakukan *pretraining* pada arsitektur jaringan[11]. Pendekatan ini didasarkan dari pengamatan bahwa melakukan inisialisasi secara acak memberikan hasil yang kurang baik. Dengan melakukan *pretraining* menggunakan algoritma *unsupervised learning* dapat memberikan inisialisasi nilai bobot awal yang lebih baik. Contoh algoritma *unsupervised learning* salah satunya adalah *Deep Belief Network*, yang didasarkan pada *Autoencoders*. Pada tugas akhir ini, *Autoencoders* digunakan untuk mencari nilai bobot awal yang lebih baik dibanding dengan nilai bobot awal yang didapat secara acak. Untuk mencari nilai bobot awal digunakan fungsi objektif sebagai berikut :

$$J = \sum_{i=1}^n \left( \frac{1}{2} \|x_i - \hat{x}_i\|^2 + \lambda \left( \sum_{l=1}^{L-1} \|W^l\|^2 + \sum_{l=1}^{L-1} \|b^l\|^2 \right) \right)$$

**2. Multilayer Perceptron**

*Multilayer Perceptron* adalah jaringan syaraf tiruan *feed-forward* yang terdiri dari sejumlah neuron yang dihubungkan oleh bobot-bobot penghubung. Neuron-neuron tersebut disusun dalam lapisan-lapisan yang terdiri dari satu lapisan input (*input layer*), satu atau lebih lapisan tersembunyi (*hidden layer*), dan satu lapisan output (*output layer*). Lapisan input menerima sinyal dari luar, kemudian melewatkannya ke

lapisan tersembunyi pertama, yang akan diteruskan sehingga akhirnya mencapai lapisan output[7].



Gambar 2.1 Contoh *Multilayer Perceptron*

3. Fungsi Aktivasi

Fungsi Aktivasi mendefinisikan nilai output dari sebuah neuron pada level aktivasi tertentu berdasarkan nilai output dari pengombinasi linier. Fungsi ini dipakai untuk menentukan keluaran suatu neuron. Secara khusus, fungsi aktivasi yang sesuai digunakan pada semua model neuron dalam beberapa lapisan tertentu dari jaringan syaraf. Berikut beberapa jenis fungsi aktivasi yaitu : fungsi linear(identitas), fungsi biner dengan threshold, fungsi Hard Limit, fungsi biner sigmoid, fungsi bipolar sigmoid, fungsi bipolar.

$$Y = \begin{cases} 0, & \text{jika } X \leq 0 \\ 1, & \text{jika } X > 0 \end{cases}$$

Gambar 2.2 Fungsi aktivasi

Apabila nilai x yang dihasilkan kurang dari atau sama dengan 0 maka output yang akan jadi 0 dan jika nilai x yang dihasilkan lebih dari 0 maka output yang dikeluarkan akan jadi 1

4. Performansi

Untuk mengevaluasi model dari keseluruhan akan dihitung menggunakan *F-Mearsure*. *F-measure* merupakan salah satu perhitungan evaluasi dalam menemukan kembali informasi yang mengkombinasikan recall dan precision. Nilai *recall* dan *precision* pada suatu keadaan dapat memiliki bobot yang berbeda. Ukuran yang menampilkan timbal balik antara recall dan precision adalah *F-measure* yang merupakan bobot harmonic mean dari

recall dan precision. Berikut adalah cara menentukan akurasi dengan menggunakan *F-Measure*.

Tabel 2.1 *F-Measure*

		Aktual	
		Tidak Churn	Churn
Prediksi	Churn	a	b
	Tidak Churn	c	d

1. ) — )
2. ) — )
3. ) — )
4. ) — )
5. — ) )
6. — )
7. — )

III. ANALISIS DAN PERANCANGAN SISTEM

A. Deskripsi dan Analisis Data

Pada penelitian ini, akan dibangun sistem yang mendukung prediksi *churn* dengan menggunakan metode *Deep Learning* dan arsitektur jaringan *Multilayer Perceptron* pada data pelanggan jaringan telekomunikasi. Data ini didapat dari sumber yang sudah dikaji sebelumnya.

Table 3.1 atribut

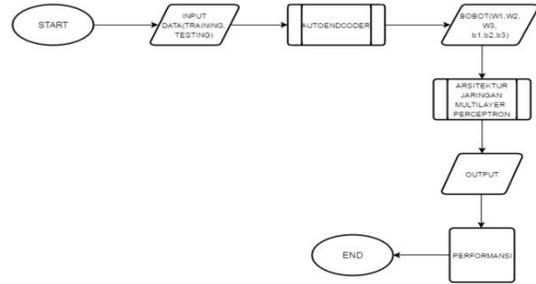
No	Nama Atribut
1	Account Length
2	VMail Message
3	Day Mins
4	Eve Mins
5	Night Mins

6	Intl Mins
7	CustServ Calls
8	Int'l Plan
9	VMail Plan
10	Day Calls
11	Day Charge
12	Eve Calls
13	Eve Charge
14	Night Calls
15	Night Charge
16	Intl Calls
17	Intl Charge

B. Perancangan Sistem

Alur dari *Deep Learning* dijelaskan sebagai berikut :

- a. Input data (training/testing), yaitu melakukan input data ke dalam algoritma. Data yang diinputkan dibagi menjadi dua. Data pertama yang diinput adalah training dan data kedua adalah testing. Dari 3333 record akan digunakan sebagai data training sebanyak 2000 record dan digunakan data testing sebanyak 1333 record.
  - b. Kemudian masuk ke implementasi *autoencoders*.
  - c. Setelah masuk pada *autoencoder* maka akan didapatkan nilai dari bobot atribut dan bias ).
  - d. Lalu masuk ke dalam arsitektur jaringan *Multilayer Perceptron* dengan inputan data dan inputan dari nilai bobot yang sudah didapatkan dari *autoencoders*.
  - e. Nilai output dari hasil arsitektur jaringan *Multilayer Perceptron* 0 dan 1. Jika nilai yang dikeluarkan dari algoritma *Multilayer Perceptron* kurang dari 0 maka output akan menjadi 0. Dan jika nilai yang dikeluarkan lebih dari 0 maka output yang dikeluarkan 1. Contoh : nilai yang dikeluarkan -1.254 maka output akan menjadi 0, jika nilai yang dikeluarkan 0.326 maka output akan menjadi 1.
  - f. Semua nilai yang dikeluarkan sudah menjadi 1 dan 0 kemudian akan dibandingkan dengan nilai *churn* yang sudah didapatkan dari data. Maka akan dihitung nilai performansinya/akurasinya dengan menggunakan *confusion matrix*.
- Berikut Flowchart dari *Deep Learning* :



Gambar 3.1 Flowchart *Deep Learning*

IV. IMPLENTASI SISTEM

A. Data yang digunakan

Data yang digunakan dalam tugas akhir ini adalah data pengguna jaringan mobile telekomunikasi didapatkan dari website [www.knime.org](http://www.knime.org). Data ini menunjukkan berapa banyak penggunaan jaringan mobile telekomunikasi di siang hari dan malam hari pada setiap nomer nya. Data yang digunakan sebanyak 3333 record dan menggunakan 17 atribut. Penggunaan data dibagi menjadi 2 bagian yaitu training dan testing. Untuk data training digunakan 2000 record dan data testing digunakan 1333 record.

B. Hasil Deep Learning

1. Hasil Autoencoders

Pada *autoencoders* untuk mendapat nilai bobot atribut dan bobot bias. Pada layer pertama bobot atribut berjumlah 68, di layer kedua atribut berjumlah 8 dan ketiga berjumlah 2 bobot. Dan bias mengikuti jumlah neuron yang terhung disetiap layer. pada layer pertama terhubung dengan 4 neuron, layer kedua berjumlah 2 neuron dan layer berneuron 1. Bobot yang didapatkan sebagai berikut :

Tabel 4.1 sample bobot atribut pada layer 1

neuron	Bobot
1	1.749186
	1.16867
	7.19525
	1.679279
·	...
·	...
·	...
·	...
17	1.348792
	0.94771
	2.313582
	0.254805

2.

Tabel 4.2 bobot atribut pada layer 2

neuron	Bobot
1	0.696326
	0.69608
2	0.959101
	0.220846
3	0.13438
	0.123781
4	0.177541
	0.178182

Table 4.3 bobot atribut pada layer 3

neuron	Bobot
1	-5.0183
	2.5725

3.

Table 4.4 bobot bias

layer	Bobot
Bobot bias 1	5.75211542
Bobot bias 2	2.9386535
Bobot bias 3	-6.3588486

2. Hasil Multilayer Perceptron

Pada algoritma ini akan dihitung data dari atribut dan bobot yang sudah didapat dari *autoencoders*. Dari perhitungan akan didapatkan keluaran nilai kontinu yang sesuai dengan data yang dimasukkan di awal. Untuk data training digunakan 2000 record dan data testing digunakan 1333 record. Total dari data yang digunakan adalah sebanyak 3333 record dan memiliki 17 atribut. Data yang dikeluarkan akan masuk fungsi aktivasi agar memiliki nilai keluaran yang sama dengan nilai churn yang sudah didapatkan pada data. Jadi hasil dari perhitungan atau disebut prediksi dibandingkan dengan data yang sebenarnya. Kemudian dihitung performansinya menggunakan *F-Measure*. Berikut data hasil prediksi dari arsitektur jaringan *Multilayer Perceptron* yang sudah dibuat :

Tabel 4.5 Sample nilai keluaran data training

Data ke -	Output(prediksi)	Data churn(aktual)
1	0	0
2	0	0

3	0	0
4	1	1
5	0	0
6	0	0
7	0	0
8	1	0
9	0	0
10	0	0
.	.	.
.	.	.
.	.	.
2000	0	0

Tabel 4.6 Sample nilai keluaran data testing

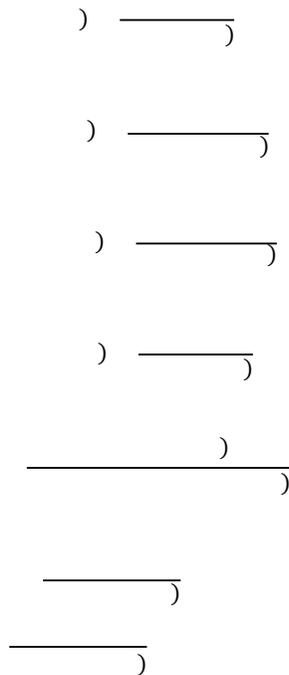
Data ke -	Output(prediksi)	Data churn(aktual)
1	0	0
2	0	1
3	1	0
4	0	0
5	0	1
6	0	0
7	1	0
9	0	0
10	0	1
.	.	.
.	.	.
.	.	.
1333	0	0

3. Performansi

Untuk mengevaluasi arsitektur jaringan *Multilayer Perceptron* secara keseluruhan menggunakan *F-Measure*. *F-measure* merupakan salah satu perhitungan evaluasi dalam menemukan kembali informasi yang mengkombinasikan recall dan precision. Nilai *recall* dan *precision* pada suatu keadaan dapat memiliki bobot yang berbeda. Ukuran yang menampilkan timbal balik antara recall dan precision adalah *F-measure* yang merupakan bobot harmonic mean dari recall dan precision. Berikut adalah cara menentukan akurasi dengan menggunakan *F-Measure*.

Tabel 4.7 *F-Measure data training*

		Aktual	
		Tidak churn	Churn
Prediksi	Tidak churn	1490	315
	Churn	58	137

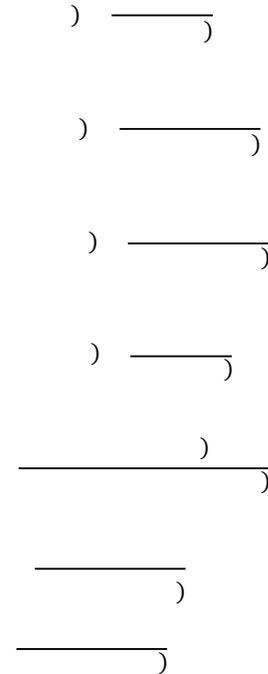


Dilihat dari nilai prediksi churn yang benar menghasilkan nilai yang cukup besar dibandingkan dengan prediksi churn yang salah, maka dapat disimpulkan perhitungan dengan arsitektur jaringan *Multilayer Perceptron* dapat dikategorikan baik. Mendapatkan nilai *Precision* sebesar 70,2 % dan *Recall* sebesar 70,27 %. Untuk nilai akurasi sebesar 81,35 %.

Tabel 4.8 *F-Measure data testing*

		Aktual	
		Tidak churn	Churn
Prediksi	Tidak	1050	191

		Aktual	
		Tidak churn	Churn
Prediksi	Tidak churn	34	58



Pada data testing dapat dilihat dari nilai prediksi *churn* yang benar menghasilkan nilai yang cukup besar dibandingkan dengan prediksi *churn* yang salah, maka dapat disimpulkan perhitungan dengan jaringan arsitektur jaringan *Multilayer Perceptron* dapat dikategorikan baik. Mendapatkan nilai *Precision* sebesar 80,4 % dan *Recall* sebesar 63,07 %. Pada nilai akurasi sebesar 83,12 %.

**V. KESIMPULAN**

Berdasarkan analisis terhadap pengujian yang dilakukan pada sistem prediksi Churn menggunakan metode *Deep Learning* dengan mengimplementasikan *Autoencoders* dan *Multilayer Perceptron*, maka dapat ditarik kesimpulan sebagai berikut:

1. Pada *Autoencoders* memiliki peranan penting untuk mendapatkan bobot atribut dan bias yang memiliki nilai optimal.
2. Dengan menggunakan arsitektur jaringan *Multilayer Perceptron*

menghasilkan nilai keluaran yang memiliki nilai prediksi yang baik.

3. Pada hasil akhir prediksi *churn* dengan menggunakan *F-Measure* pada data training mendapatkan nilai *Precision* sebesar 70,2 %, nilai *Recall* sebesar 70,27 % dan nilai akurasi sebesar 81,35 %. Pada data testing didapatkan nilai *Precision* sebesar 80,4 %, nilai *Recall* sebesar 63,07 % dan nilai akurasi sebesar 83,12 %.