

# Multilayer Perceptron untuk Prediksi Sessions pada Sebuah Website Journal Elektronik

Aji Prasetya Wibawa<sup>a,1</sup>, Widya Lestari<sup>a,2</sup>, Agung Bella Putra Utama<sup>a,3</sup>, Irzan Tri Saputra<sup>a,4</sup>, Zahra Nabila Izdihar<sup>a,5</sup>

<sup>a</sup> Universitas Negeri Malang, Jl. Semarang No. 5, Malang 65145, Indonesia

<sup>1</sup> aji.prasetya.ft@um.ac.id; <sup>2</sup> lestariwidya792@gmail.com; <sup>3</sup> agungbpu02@gmail.com; <sup>4</sup> saputraitran@gmail.com; <sup>5</sup> zahra.informatics@gmail.com;

INFORMASI ARTIKEL	ABSTRAK
Diterima : 04 – 10 – 2020 Direvisi : 01 – 11 – 2020 Diterbitkan : 31 – 12 – 2020	Peramalan <i>session website journal</i> dilakukan untuk pendukung pengambilan keputusan dalam rangka meningkatkan kualitas dan nilai akreditasi pada <i>website journal</i> . Data <i>sessions</i> dianalisis berdasarkan pergerakan pola data <i>time series</i> menggunakan metode <i>multilayer perceptron</i> . Karakteristik yang dimiliki oleh <i>multilayer perceptron</i> yaitu keunggulan dalam penentuan nilai bobot yang lebih baik daripada metode lain, <i>multilayer perceptron</i> dapat digunakan tanpa pengetahuan sebelumnya dan algoritma dapat diimplementasikan dengan mudah serta mampu menyelesaikan masalah linear dan nonlinear sehingga nilai peramalan menjadi lebih baik. Penelitian menggunakan berbagai persentase data <i>train</i> dan <i>test</i> . Perbandingan data <i>train</i> dan <i>test</i> yang memiliki nilai terbaik adalah 80% data <i>train</i> dan 20% data <i>test</i> dengan <i>learning rate</i> 0.4 dan arsitektur 2-1-1. Hasil evaluasi model diperoleh nilai MSE dan RMSE, 0.015357 dan 0.123999 untuk <i>training set</i> serta, 0.018996 dan 0.137826 untuk MSE dan RMSE dari <i>test set</i> . Waktu eksekusi yang dibutuhkan untuk melakukan peramalan adalah 580.0651 <i>second</i> atau 9.667751 menit.
<b>Kata Kunci:</b> Multilayer Perceptron Prediksi Sessions Website Journal	
	

## I. Pendahuluan

*Forecasting* adalah salah satu metode *data mining* yang digunakan untuk memperkirakan sesuatu yang belum terjadi [1]. Peramalan diperlukan dalam berbagai situasi seperti membuat perencanaan terkait kegiatan usaha dalam beberapa periode tertentu [2]. Fungsi dari peramalan adalah penunjang dalam pengambilan keputusan. Keputusan yang baik didasarkan atas pertimbangan terhadap sesuatu yang terjadi pada saat keputusan itu diambil [3].

Ada banyak penelitian menggunakan metode *forecasting*. Berbagai penelitian yang menggunakan metode *forecasting* antara lain, metode *ARIMA* untuk peramalan bahan baku produk plastik berbasis data pendapatan [4], metode *ARMA* untuk peramalan angin [5], metode *neural network* untuk *power system short-term load forecasting* [6],[7], metode *exponential smoothing* untuk kombinasi konsumsi suku cadang [8], [9], metode *radial base function* (RBF) untuk peramalan kecepatan angin [10], *multilayer perceptron* untuk analisis properti *Jordan Oil Shale* [11], dan lain-lain. Setiap metode *forecasting* memiliki kelebihan dan kekurangan sendiri apabila diimplementasikan pada suatu kasus tertentu.

Metode *forecasting* yang sering digunakan dalam penelitian adalah *multilayer perceptron neural network* (MLP) [12]. Karakteristik yang dimiliki oleh MLP yaitu keunggulan dalam penentuan nilai bobot yang lebih baik daripada metode lain [13], MLP dapat digunakan tanpa pengetahuan sebelumnya dan algoritma dapat diimplementasikan dengan mudah [12] serta mampu menyelesaikan masalah linear dan nonlinear [14]. Berdasarkan dengan karakteristik yang dimiliki oleh MLP membuat nilai peramalan menjadi lebih baik.

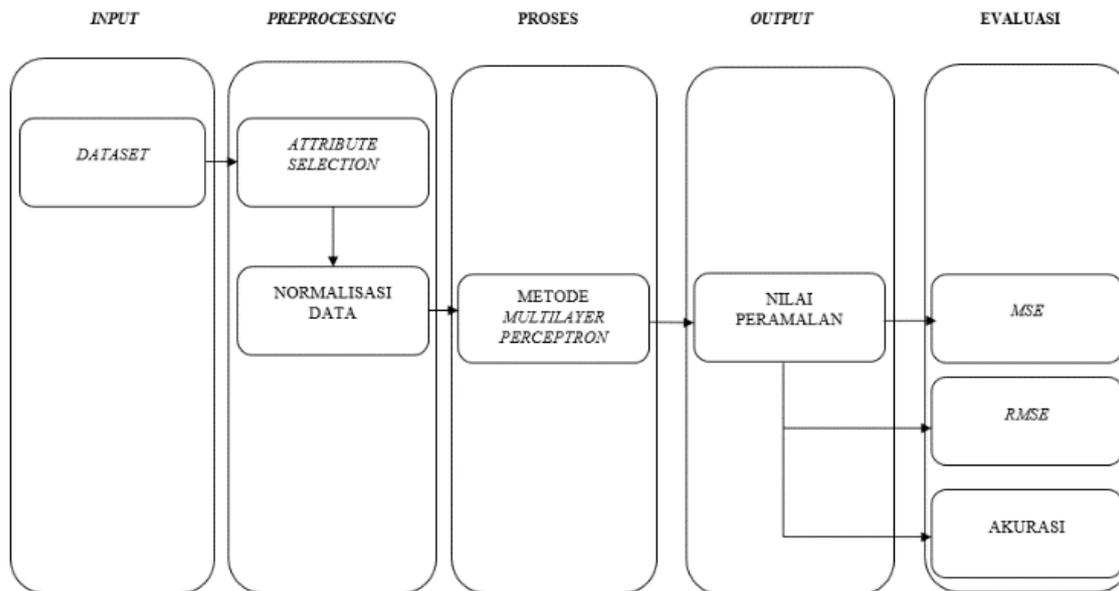
Metode MLP akan digunakan untuk melakukan peramalan terhadap *sessions* dari suatu *website journal*. *Sessions* adalah serangkaian halaman *web* yang ditelusuri pengguna dalam *single access*. Identifikasi *sessions* dapat didefinisikan sebagai satu set halaman yang dikunjungi oleh pengguna yang sama dalam durasi satu kunjungan tertentu ke situs *web* [15]. Tujuan identifikasi *sessions* adalah membagi akses halaman setiap pengguna sekaligus menjadi *sessions* individual [16]. Seorang pengguna dapat memiliki *sessions* tunggal atau ganda selama suatu periode tertentu.

Pada bagian II, akan menjelaskan metode penelitian untuk menyelesaikan permasalahan penelitian. Pada bagian III, mendeskripsikan tentang hasil penelitian dan pembahasan. Pada bagian terakhir kesimpulan dari penelitian.

## II. Metode

### A. Desain Penelitian

Pengolahan data menggunakan metode MLP dilakukan dengan tahapan sesuai dengan Gambar 1.



Gambar 1. Desain Penelitian

### B. Dataset

Data yang digunakan dalam penelitian adalah *sessions*, terhitung mulai tanggal 01 Januari 2018 sampai 31 Desember 2018. Data penelitian diperoleh melalui situs *web* terpercaya yaitu <https://statcounter.com/> yang bertugas untuk mengolah data aktifitas *website journal* dari salah satu Universitas yang ada di Indonesia. Aktifitas pengguna *website journal* akan dijadikan suatu laporan statistik. Penelitian ini menggunakan 3 atribut yaitu data  $t$  (data sekarang),  $t - 1$  (data 1 hari sebelumnya pada periode  $t$ ) dan  $t - 7$  (data 7 hari sebelumnya pada periode  $t$ ). Total *instance* yang digunakan terdiri dari 358 *instances*.

### C. Attribute Selection

*Attribute selection* atau *feature selection* bertujuan untuk memilih fitur yang berpengaruh (fitur optimal) dan mengesampingkan fitur yang tidak berpengaruh [17]. *Attribute selection* tidak memiliki wewenang yang terkait dengan pemberian atau pembuatan model. *Attribute selection* sering ditempatkan pada tahap *preprocessing data* [18]. *Attribute selection* dalam penelitian ini dilakukan dengan uji autokorelasi.

Uji autokorelasi dilakukan dengan mengguna fungsi =  $CORREL(array1, array 2)$  yang terdapat pada *Microsoft Excel*. *array 1* untuk data  $t$  dan *array 2* untuk data yang akan dilakukan uji autokorelasi. Uji autokorelasi dilakukan pada  $t - 1$  hingga  $t - 7$  terhadap  $t$ . Hubungan autokorelasi akan ditentukan berdasarkan nilai dari variabel  $\rho$ . Bila nilai  $\rho = 0$  maka, tidak ada korelasi sedangkan, bila nilai  $\rho$  mendekati 1 maka, dikatakan memiliki korelasi yang tinggi [19]. Nilai  $\rho$  memiliki kriteria seperti yang ditampilkan pada Tabel 1.

Tabel 1. Kriteria Uji Autokorelasi

$\rho$	Keterangan
0,00 – 0,199	Memiliki hubungan yang sangat lemah
0,20 – 0,399	Memiliki hubungan yang lemah
0,40 – 0,599	Memiliki hubungan yang sedang
0,60 – 0,799	Memiliki hubungan kuat
0,80 – 1,000	Memiliki hubungan yang sangat kuat

#### D. Normalisasi Data

Normalisasi data dilakukan untuk mempertahankan variasi prediksi dan peramalan yang besar [20]. Konsep dasar normalisasi data diperlukan untuk memberikan *range* yang lebih dekat. Teknik normalisasi data Min-Max merupakan teknik yang populer dibandingkan dengan teknik yang lain [12].

Normalisasi *min-max* dipilih karena menyediakan *linear transformation* pada data *original range* dan memiliki tingkat kesalahan yang kecil dibandingkan dengan metode yang lain [21]. *Range value* berada antara 0 dan 1. Proses penyelesaian normalisasi data dilakukan sesuai dengan [22].

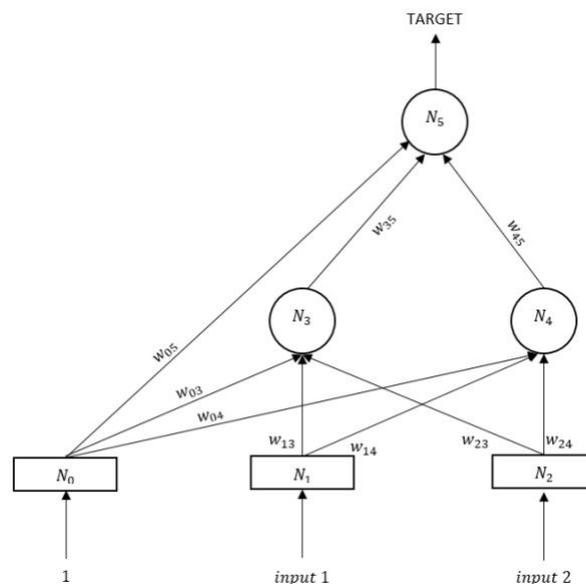
$$X_{i(norm)} = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad (1)$$

$X_{i(norm)}$  adalah hasil normalisasi,  $X_i$  adalah data yang akan dinormalisasi,  $X_{min}$  adalah nilai minimal dari keseluruhan data, dan  $X_{max}$  adalah nilai maksimal dari keseluruhan data.

#### E. Peramalan dengan Menggunakan Multilayer Perceptron

*Multilayer perceptron* tergolong dalam *feed-forward network* [23]. Model *multilayer perceptron* tersusun dari beberapa lapisan *node* yang terdiri dari *input layer*, *hidden layer* dan *output layer*. Penentuan elemen *multilayer perceptron* sangat mempengaruhi kinerja jaringan sehingga, harus dipertimbangkan dengan hati-hati [24]. Elemen dari *multilayer perceptron* terdiri dari, arsitektur jaringan, algoritma pembelajaran dan fungsi aktivasi. Penentuan arsitektur jaringan akan bergantung pada data. Banyak literatur menyatakan bahwa tidak ada aturan umum yang mengatur terkait penentuan arsitektur yang baik, banyak arsitektur harus diperiksa untuk hasil yang benar dengan teknik *trial and error* [24], [25].

*Node* dalam arsitektur jaringan akan terhubung lurus mulai dari *node* pada *input layer* terhubung ke *node* pada *hidden layer* dan *node* pada *hidden layer* akan terhubung ke *node* pada *output layer*. Setiap *node* yang terhubung memiliki bobot masing-masing dengan nilai yang sama atau bisa berbeda karena bobot awal ditentukan secara *random*.



Gambar 2. Salah Satu Arsitektur *Multilayer Perceptron*

Gambar 2 menjelaskan salah satu arsitektur yang digunakan dalam penelitian. Simbol  $N$  menyatakan *node* untuk setiap *layer*. Simbol  $N_1$  dan  $N_2$  adalah *node* untuk *input layer*.  $N_1$  berperan sebagai *input 1* yang diisi oleh nilai  $t - 1$ .  $N_2$  berperan sebagai *input 2* yang diisi oleh nilai  $t - 7$ . Nilai  $N_0$  tidak menggunakan data dari *dataset*, karena *node* tersebut adalah *node* untuk *bias* dengan nilai awal 1. Nilai 1 yang ada di arsitektur *bias* adalah konstanta [11]. Simbol  $N_3$  dan  $N_4$  adalah *node* untuk *hidden layer* dan  $N_5$ , *node* untuk *output layer*. Bobot pada setiap *node* yang terhubung disimbolkan dengan  $w$ . Penomoran simbol untuk  $w$  yaitu, nomor pertama adalah nomor *node* awal sedangkan, nomor kedua adalah nomor dari *node* tujuan.

Algoritma pembelajaran pada *multilayer perceptron* dilakukan sesuai dengan [26][27] sebagai berikut :

Langkah 0 (*input*):

- *Input* data pelatihan ke jaringan.

Langkah 1 (inisialisasi):

- Melakukan inisialisasi bobot dengan nilai acak yang kecil.

- Mengatur parameter seperti, *learning rate* dan *momentum coefficient*.

Langkah 2 (*training loop*):

- Menerapkan pola *input network* ke *input layer*.

Langkah 3 (*forward propagation*):

- Menyebarkan *sinyal forward* melalui jaringan.
- Menghitung *network output vector*.

Langkah 4 (Menghitung *output error*):

- Menghitung *error* pada setiap *output*, perbedaan antara target yang diinginkan dan *network output*.

Langkah 5 (*error backpropagation*):

- Menyebarkan *error backward* untuk menyesuaikan bobot dengan cara meminimalkan *error*.

Langkah 6 (satu *loop iteration*):

- Memeriksa apakah seluruh data telah digunakan.
- Mengulang langkah ke dua dengan menggunakan langkah 5 untuk seluruh *training dataset*.

Penelitian terhadap peramalan *sessions* menggunakan MLP dilakukan sebanyak 324 kali percobaan untuk berbagai perbandingan *train* dan *test*, arsitektur jaringan dan *learning rate*. Pengujian dilakukan dengan mengubah jumlah data *train* dan *test*, *learning rate* dan arsitektur jaringan.

Persentase data *train* dan *test* yang dilakukan dalam penelitian terdiri dari 10%, 20%, 30%, 40% 50% 60%, 70%, 80%, dan 90% data *train*. Penggunaan berbagai variasi persentase data *train* dan *test* bertujuan untuk menemukan jumlah data terbaik untuk peramalan *session*. *Epoch* maksimum yang digunakan adalah 50000 untuk masing-masing data *input*. *Learning rate* yang akan digunakan adalah 0,1 sampai 0,9 dengan selisih untuk setiap *learning rate* adalah 0.1 [28].

Peramalan *sessions* mengimplementasikan 4 jenis arsitektur yang memiliki 1 *input layer*-1 *hidden layer*-1 *output layer* [29] terdiri dari 2 *node input layer*-1 *node hidden layer*- 1 *node output layer* (2-1-1), 2 *node input layer*-2 *node hidden layer*- 1 *node output layer* (2-2-1), 2 *node input layer*-3 *node hidden layer*- 1 *node output layer* (2-3-1) [26],[30], dan 2 *node input layer*-4 *node hidden layer*- 1 *node output layer* (2-4-1)[31].

#### F. Evaluasi

Evaluasi hasil peramalan dilakukan dengan menggunakan MSE dan RMSE untuk setiap pengujian data *training* dan *test*. Evaluasi dilakukan untuk memverifikasi keefektifan dan keakuratan model yang diusulkan. Nilai akurasi dari MSE dan RMSE akan dibandingkan dan dipilih metode evaluasi terbaik. Proses perhitungan dilakukan sesuai dengan [22], [28].

$$\text{Mean Square Error (MSE)} = \frac{1}{N} \sum_{t=1}^N e_t^2 \quad (2)$$

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\frac{\sum_{t=1}^N e_t^2}{N}} \quad (3)$$

Nilai *error* yang diperoleh menggunakan MSE dan RMSE akan digunakan untuk menghitung akurasi. Akurasi yang diperoleh dari MSE dan RMSE akan dibandingkan. Tujuan Pembandingan guna mengetahui akurasi terbaik yang dihasilkan dari kedua metode tersebut. Persentase data *train* dan data *test* pada grafik akurasi akan dinyatakan sebagai nilai 1 hingga 9. Nilai yang terdapat pada grafik nilai akurasi memiliki keterangan sesuai dengan Tabel 2.

Tabel 2. Keterangan Persentase Data *Train* dan *Test* untuk Grafik Akurasi

Nilai	Keterangan
1	10% Data <i>Train</i> , 90% Data <i>Test</i>
2	20% Data <i>Train</i> , 80% Data <i>Test</i>
3	30% Data <i>Train</i> , 70% Data <i>Test</i>
4	40% Data <i>Train</i> , 60% Data <i>Test</i>
5	50% Data <i>Train</i> , 50% Data <i>Test</i>
6	60% Data <i>Train</i> , 40% Data <i>Test</i>
7	70% Data <i>Train</i> , 30% Data <i>Test</i>
8	80% Data <i>Train</i> , 20% Data <i>Test</i>
9	90% Data <i>Train</i> , 10% Data <i>Test</i>

### III. Hasil dan Pembahasan

Hasil uji autokorelasi sangat menuntukan dalam peramalan *sessions* dengan menggunakan MLP. Pemilihan atribut yang kurang tepat untuk masukan model akan mempengaruhi hasil peramalan. Pada penelitian ini nilai  $\rho$  tertinggi yang dihasilkan adalah t-1 dan t-7 dengan nilai 0.670103 dan 0.630831. Nilai  $\rho$  yang dihasilkan dalam penelitian memiliki hubungan yang kuat terhadap t yang dapat dilihat pada Tabel 3.

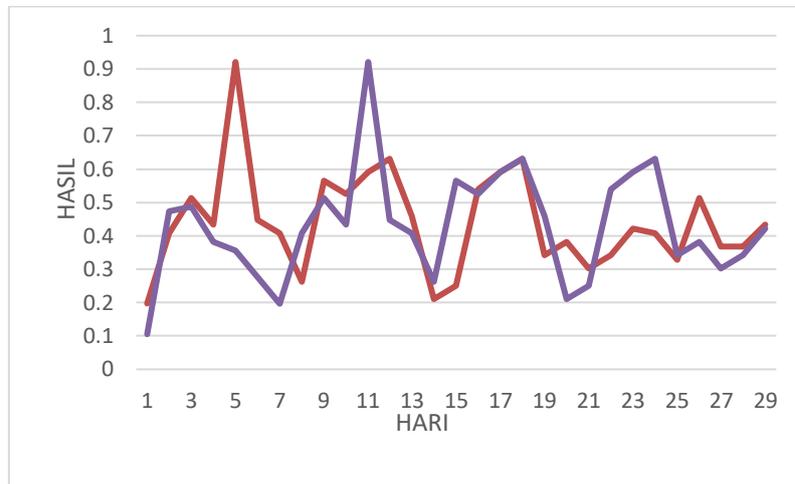
Tabel 3. Hasil Uji Autokorelasi

Variabel Uji Autokorelasi terhadap $t$	$\rho$
$(t - 1)$	0.670103
$(t - 2)$	0.520902
$(t - 3)$	0.440526
$(t - 4)$	0.434863
$(t - 5)$	0.454625
$(t - 6)$	0.568469
$(t - 7)$	0.630831

Tabel 4 merupakan tabel hasil normalisasi data target dan data *input* sebanyak 4 *instance* data awal pada *dataset*. Gambar 3 merupakan sampel data *input* untuk waktu 1 bulan sedangkan, Gambar 4 adalah sampel data target.

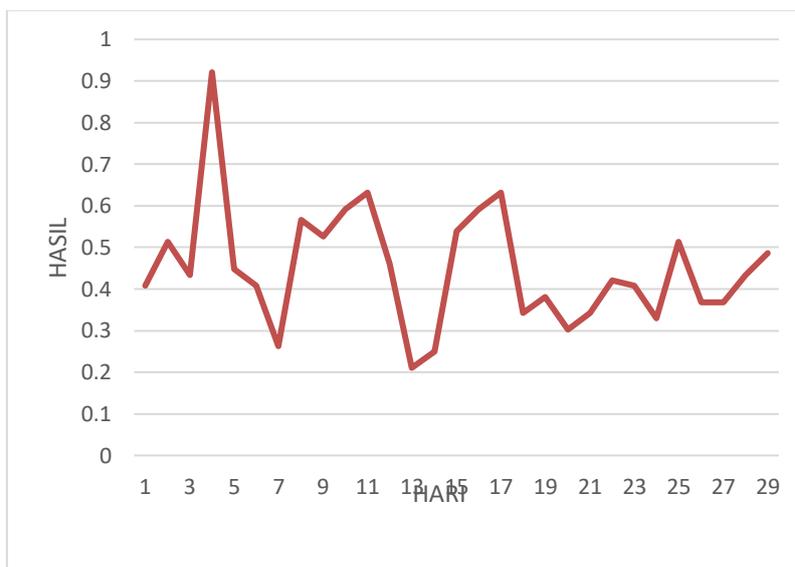
Tabel 4. Data *Input* dan Target

Input Jaringan		Target Jaringan
$t - 1$	$t - 7$	$t$
0	0.7857142857	1
1	0.9642857143	0.8214285714
0.9333333333	0.7857142857	0.6785714286
0.6	0.6428571429	0.0357142857



Keterangan : — Data  $t - 1$  — Data  $t - 7$

Gambar 3. Grafik Data Input



Gambar 4. Grafik Data Target

Proses memperoleh nilai peramalan terbaik membutuhkan waktu yang cukup lama. Penyelesaian peramalan membutuhkan waktu sekitar 2 bulan. Selama proses penelitian ditemukan faktor-faktor yang dapat mempengaruhi nilai peramalan. Salah satu faktor yang mempengaruhi nilai peramalan adalah kondisi *hardware* yang digunakan untuk mengeksekusi model. Apabila *hardware* yang digunakan menjalankan berbagai program ketika melakukan pengekseskuan peramalan maka, waktu eksekusi akan lama. Jadi, proses peramalan harus dilakukan dengan kondisi *hardware* yang hanya menjalankan model peramalan MLP. Waktu pengekseskuan model peramalan selain dipengaruhi oleh *hardware* juga dipengaruhi oleh jumlah dari data *train* dan data *test*.

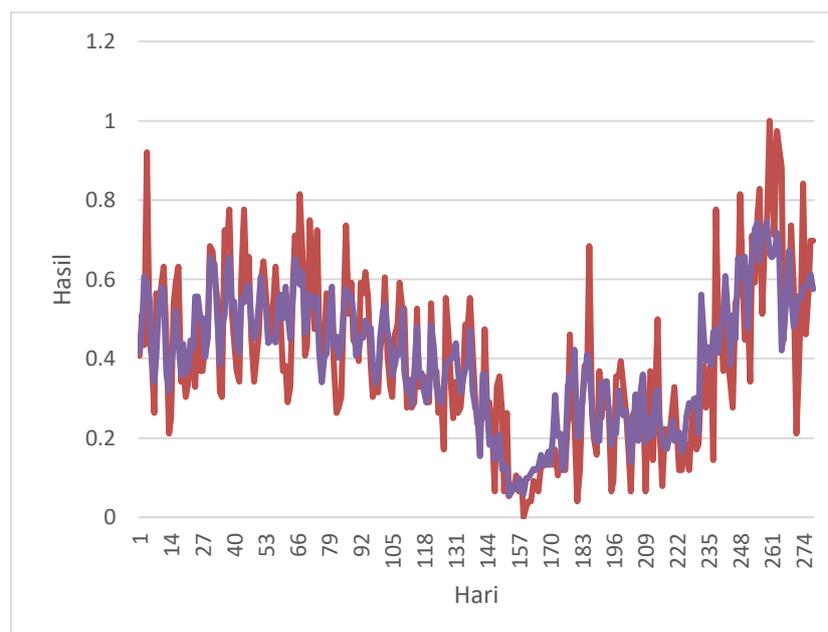
Persentase data *train* yang kecil di awal-awal percobaan model menghasilkan waktu eksekusi yang kecil yaitu 91.20220971 *seconds* atau 1.52 menit. Semakin lama proses pengekseskuan model berlangsung waktu yang dihasilkan juga semakin besar. Waktu terlama yang dibutuhkan untuk melakukan pengekseskuan model adalah 851.28556947 *seconds* atau 14.188 menit. Teori awal yang menyatakan bahwa nilai *learning rate* kecil akan membutuhkan waktu eksekusi yang lama dengan hasil peramalan terbaik tidak terbukti dalam penelitian ini. Nilai *learning rate* kecil tidak selalu menghasilkan nilai peramalan yang baik dan waktu eksekusi yang rendah begitu juga sebaliknya.

Tabel 5 dan Tabel 6 menampilkan *error* dari hasil peramalan data *train* dan data *test* yang diperoleh dari selisih antara data target dan data hasil peramalan. Rata-rata *error* yang dihasilkan antara peramalan data *train* dan data *test* menunjukkan bahwa *error* pada data *train* lebih kecil dari data *test*. Hasil tersebut dipengaruhi adanya proses pelatihan terlebih dahulu pada data *train* sedangkan, data *test* tidak melewati proses tersebut.

Dalam kasus ini *error tolerance* tidak diberikan pada model sehingga penentuan hasil peramalan ditentukan dari jumlah *epoch*. Selama penelitian, tiga nilai *epoch* telah diberikan pada model. Nilai *epoch* terdiri dari 10000, 20000 dan 50000. Pada nilai *epoch* 10000 nilai *error* peramalan tergolong besar, sehingga digunakan *epoch* 20000. Penggunaan nilai *epoch* 20000 dalam percobaan menghasilkan nilai peramalan yang cenderung lebih baik daripada *epoch* 10000. *Epoch* 50000 yang diberikan pada model mengalami perubahan nilai peramalan yang cukup besar. Nilai peramalan yang dihasilkan pada *epoch* 50000 jauh lebih baik daripada *epoch* 10000 dan *epoch* 20000. Sedangkan nilai *error* yang dihasilkan *epoch* 50000 stabil dan tidak mengalami perubahan nilai. Selain ketiga nilai yang disebutkan diatas *epoch* 100000 juga pernah digunakan pada penelitian. Hasil peramalan menggunakan *epoch* 100000 menghasilkan nilai *error* yang kecil seperti pada *epoch* 50000. Nilai *error* berhenti mengalami perubahan di nilai *epoch* kurang dari 50000 dan lebih dari 20000.

Tabel 5. Hasil Peramalan Data Train

Data Target	Ramalan Data Train	Error
0.40789473	0.4165182	-0.00862347
0.5131579	0.48795533	0.02520257
0.4342105	0.6066792	-0.1724687
0.92105263	0.58910716	0.33194547

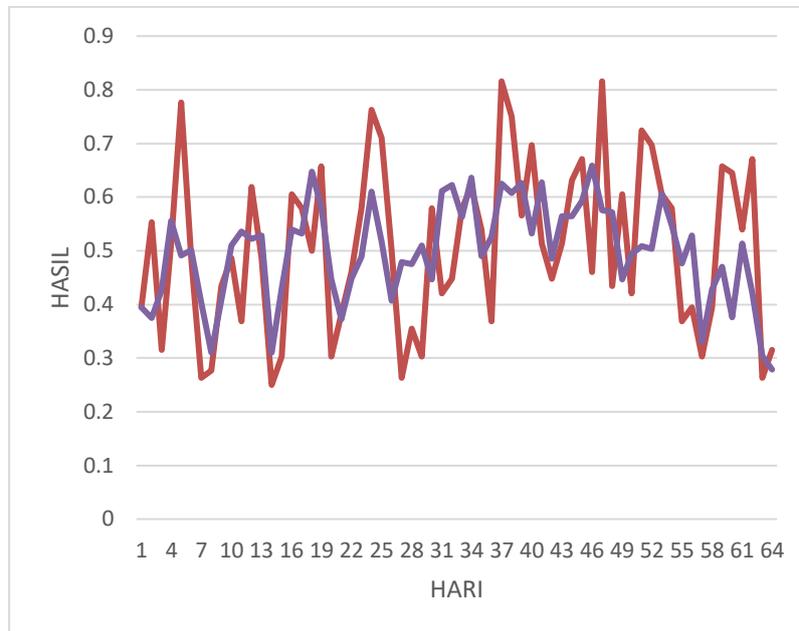


Keterangan : — Data Target — Prediksi Data Train

Gambar 5. Grafik Data Target dan Ramalan Data Train

Tabel 6. Hasil Peramalan Data Test

Data Target	Ramalan Data Train	Error
0.39473683	0.39485264	-0.000115881
0.55263156	0.3746122	0.17801936
0.31578946	0.42650282	-0.11071336
0.5131579	0.55589926	-0.04274136



Keterangan : Data Target Prediksi Data Test

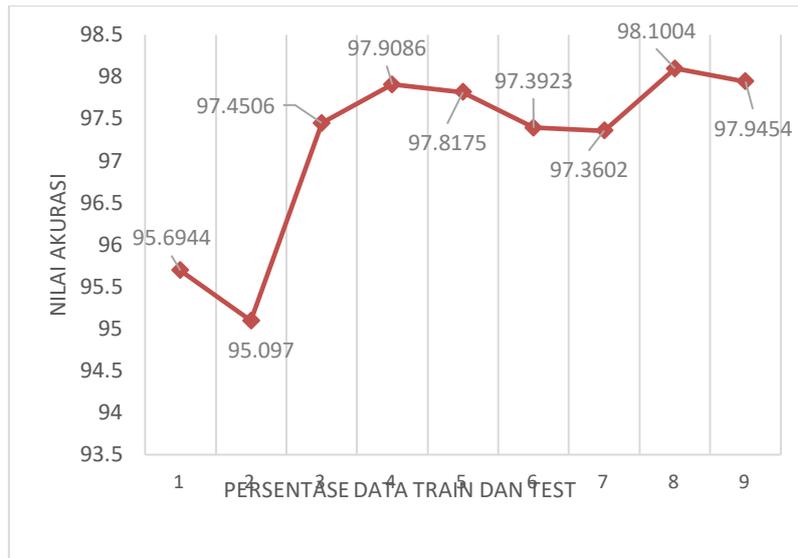
Gambar 6. Grafik Data Target dan Ramalan Data Test

Perbandingan Train Test	Arsitektur	Learning Rate	Training Set		Test Set		Waktu Eksekusi
			MSE	RMSE	MSE	RMSE	
10% 90%	2-3-1	0.1	0.016952	0.130203	0.043056	0.207501	136.5531 s
20% 80%	2-3-1	0.7	0.015867	0.125965	0.049030	0.221427	153.7222 s
30% 70%	2-4-1	0.9	0.011481	0.107151	0.025494	0.159670	263.7689 s
40% 60%	2-1-1	0.1	0.010978	0.104776	0.020914	0.144618	344.5562 s
50% 50%	2-1-1	0.2	0.012861	0.113409	0.021825	0.147735	629.9537 s
60% 40%	2-1-1	0.5	0.011087	0.105298	0.026077	0.161486	434.7651 s
70% 30%	2-1-1	0.5	0.012274	0.110791	0.026398	0.162474	502.1310 s
80% 20%	2-1-1	0.4	0.015375	0.123999	0.018996	0.137826	580.0651 s
90% 10%	2-1-1	0.8	0.015697	0.125289	0.020546	0.143341	698.2841 s

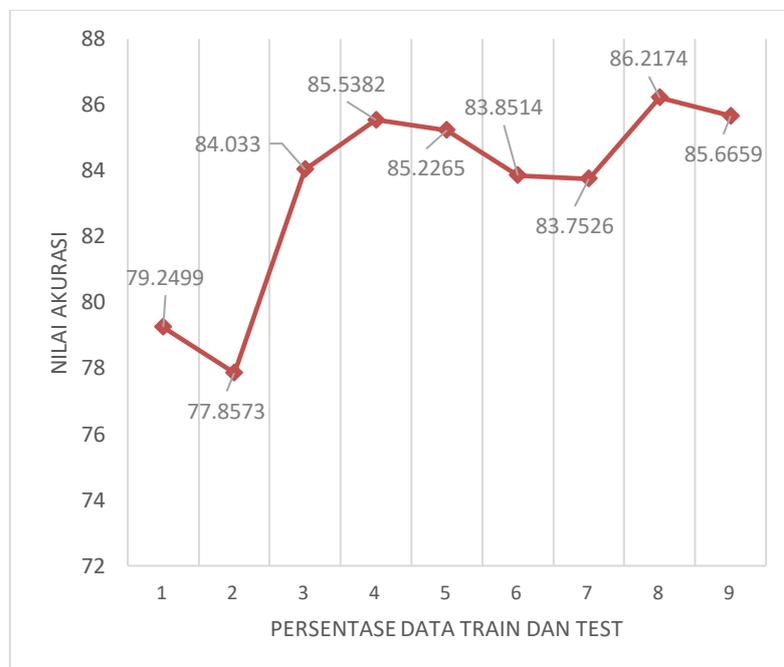
Gambar 7. Hasil Penelitian dengan Nilai Error Tertinggi

Gambar 7 menampilkan nilai peramalan terbaik dari setiap persentase data train dan data test. Perbandingan data train dan test yang memiliki nilai terbaik dari keseluruhan percobaan adalah 80% data train. Jumlah data yang digunakan adalah 287 data untuk data train dan 71 data untuk data test. Berdasarkan persentase nilai train dan test, nilai peramalan terbaik diperoleh dengan learning rate 0.4 dan arsitektur 2-1-1. Hasil evaluasi model diperoleh nilai MSE dan RMSE, 0.015357 dan 0.123999 untuk training set serta, 0.018996 dan 0.137826 untuk MSE dan RMSE dari data test. Akurasi yang dihasilkan dari nilai MSE jauh lebih tinggi dibandingkan dengan RMSE. Model peramalan terbaik menghasilkan 98.1004% untuk MSE dan 86.2174% untuk RMSE. Waktu eksekusi yang dibutuhkan untuk melakukan peramalan adalah 580.0651 second atau 9.667751 menit.

Gambar 8 dan Gambar 9 menampilkan grafik nilai akurasi tertinggi untuk setiap persentase data train dan data test. Nilai akurasi berdasarkan MSE cenderung memiliki nilai akurasi diatas 95%. Nilai akurasi yang dibentuk dari nilai RMSE memiliki rentang nilai diantara 77% hingga 86%. Berdasarkan dengan percobaan yang telah dilakukan akurasi dari MSE memiliki nilai lebih tinggi dari RMSE. Kondisi demikian dilatarbelakangi oleh penggunaan akar ( $\sqrt{\phantom{x}}$ ) untuk menghitung nilai error pada RMSE.

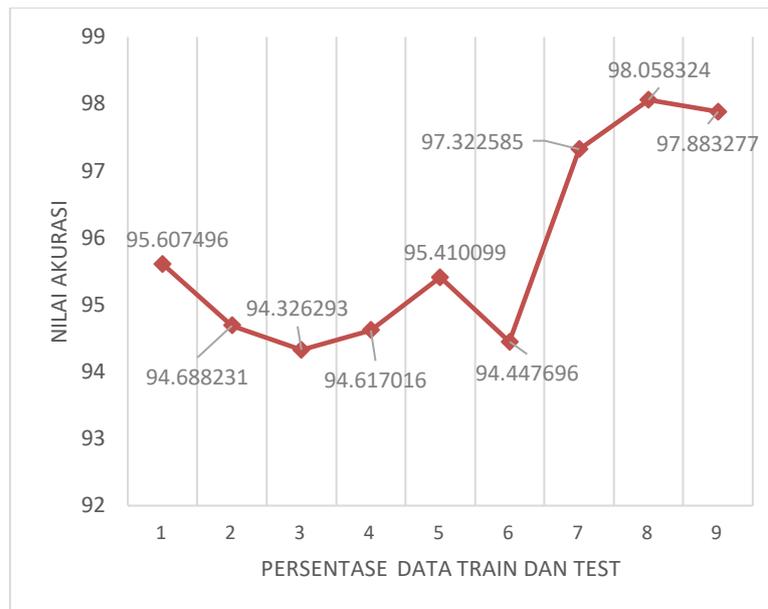


Gambar 8. Grafik Nilai Akurasi Tertinggi dengan MSE untuk Setiap Persentase Data *Train* dan *Test*



Gambar 9. Grafik Nilai Akurasi Tertinggi dengan RMSE untuk Setiap Persentase Data *Train* dan *Test*

Kondisi peramalan dengan nilai terendah yang dihasilkan dalam penelitian adalah persentase data *train* 30% dengan *learning rate* 0.9 dan arsitektur 2-1-1. Hasil evaluasi model diperoleh nilai MSE dan RMSE, 0.01919521 dan 0.10991967 untuk *training set* serta, 0.03502402 dan 0.18714707 untuk MSE dan RMSE dari *test set*. Berdasarkan dengan nilai MSE dan RMSE diperoleh nilai akurasi 94.326293% dan 76.180455%. Waktu eksekusi yang dibutuhkan selama 403.71355259 *second* atau 6.728559 menit. Gambar 10 dan Gambar 11 menampilkan nilai akurasi terendah dari setiap persentase data *train* dan data *test*.



Gambar 10. Grafik Nilai Akurasi Terendah dengan MSE untuk Setiap Persentase Data *Train* dan *Test*



Gambar 11. Nilai Akurasi Terendah dengan MSE untuk Setiap Persentase Data *Train* dan *Test*

#### IV. Kesimpulan

Berdasarkan hasil penelitian, metode MLP berhasil melakukan peramalan terhadap sessions dari website journal. Nilai akurasi terbaik yang berhasil diperoleh adalah 98.1004% untuk MSE dan 86.2174% untuk RMSE. Akurasi terbaik diperoleh dari persentase data train 80% dengan learning rate 0.4 dan arsitektur 2-1-1. Waktu eksekusi yang dibutuhkan adalah 580.0651 second atau 9.667751 menit. Untuk penelitian lebih lanjut, disarankan menggunakan data penelitian yang berbeda.

#### Daftar Pustaka

- [1] V. R. Sutrisno, "Analisis Forecasting untuk Data Penjualan Menggunakan Metode Simple Moving Average dan Single Exponential Smoothing: Studi Kasus PT Guna Kemas Indah."
- [2] R. J. Hyndman, "Forecasting: An Overview," *Int. Encycl. Stat. Sci.*, pp. 536–539, 2011, doi: 10.1007/978-3-642-04898-2\_256.
- [3] R. Ginting *et al.*, "Peramalan dan faktor faktor yang mempengaruhi harga bawang merah di sumatera

- utara.”
- [4] B. Siregar, E. B. Nababan, A. Yap, U. Andayani, and Fahmi, “Forecasting of raw material needed for plastic products based in income data using ARIMA method,” *Proceeding - 2017 5th Int. Conf. Electr. Electron. Inf. Eng. Smart Innov. Bridg. Futur. Technol. ICEEIE 2017*, vol. 2018-Janua, pp. 135–139, 2018, doi: 10.1109/ICEEIE.2017.8328777.
  - [5] D. R. Chandra, M. S. Kumari, and M. Sydulu, “A detailed literature review on wind forecasting,” *Proc. 2013 Int. Conf. Power, Energy Control. ICPEC 2013*, pp. 630–634, 2013, doi: 10.1109/ICPEC.2013.6527734.
  - [6] L. Zhuang, H. Liu, J. Zhu, S. Wang, and Y. Song, “Comparison of forecasting methods for power system short-Term load forecasting based on neural networks,” *2016 IEEE Int. Conf. Inf. Autom. IEEE ICIA 2016*, no. 61473174, pp. 114–119, 2017, doi: 10.1109/ICInfA.2016.7831806.
  - [7] G. De Luca and M. Gallo, “Artificial neural networks for forecasting user flows in transportation networks: Literature review, limits, potentialities and open challenges,” *5th IEEE Int. Conf. Model. Technol. Intell. Transp. Syst. MT-ITS 2017 - Proc.*, pp. 919–923, 2017, doi: 10.1109/MTITS.2017.8005644.
  - [8] F. Guo, C. Y. Liu, B. Zhou, and S. Q. Zhang, “Spares consumption combination forecasting based on genetic algorithm and exponential smoothing method,” *Proc. - 2012 5th Int. Symp. Comput. Intell. Des. Isc. 2012*, vol. 2, pp. 198–201, 2012, doi: 10.1109/ISCID.2012.201.
  - [9] J. Lian and L. Li, “Predictive Analysis of E-Commerce Enterprises Soft Operating Costs Based on Exponential Smoothing Technique,” no. 1, pp. 1–5.
  - [10] X. Qin, C. Jiang, and J. Wang, “Online clustering for wind speed forecasting based on combination of RBF neural network and persistence method,” *Proc. 2011 Chinese Control Decis. Conf. CCDC 2011*, pp. 2798–2802, 2011, doi: 10.1109/CCDC.2011.5968687.
  - [11] J. M. Nazzal, I. M. El-etary, S. a Najim, A. Ahliyya, P. O. Box, and K. S. Arabia, “Multilayer Perceptron Neural Network ( MLPs ) For Analyzing the Properties of Jordan Oil Shale,” *World Appl. Sci. J.*, vol. 5, no. 5, pp. 546–552, 2008.
  - [12] Y. S. Park and S. Lek, *Artificial Neural Networks: Multilayer Perceptron for Ecological Modeling*, vol. 28. Elsevier, 2016.
  - [13] A. B. HERMANIANTO, “Optimasi multi-layer perceptron pada model prediksi karakteristik musim hujan dan kemarau di kabupaten pacitan abdul basith hermanianto,” 2017.
  - [14] T. Marwala, “Multi-layer Perceptron,” *Handb. Mach. Learn.*, no. 2001, pp. 23–42, 2018, doi: 10.1142/9789813271234\_0002.
  - [15] S. C. Satapathy, A. Govardhan, K. S. Raju, and J. K. Mandal, “An Overview on Web Usage Mining,” *Adv. Intell. Syst. Comput.*, vol. 338, no. November 2014, pp. V–VI, 2015, doi: 10.1007/978-3-319-13731-5.
  - [16] P. M. Chawan, “Web Usage Mining,” no. June 2013, 2017.
  - [17] Z. M. Kesuma, “Feature Selection Data Indeks Kesehatan Masyarakat Menggunakan Algoritma Relief,” *Statistika*, vol. 11, no. 1, pp. 61–66, 2011.
  - [18] S. E. Buttrey, “Data Mining Algorithms Explained Using R,” *J. Stat. Softw.*, vol. 66, no. Book Review 2, 2015, doi: 10.18637/jss.v066.b02.
  - [19] G. M. Tinungki, “Metode Pendeteksian Autokorelasi Murni dan Autokorelasi Tidak Murni,” vol. 13, no. 1, pp. 46–54, 2016.
  - [20] S. G. K. Patro and K. K. sahu, “Normalization: A Preprocessing Stage,” *Iarjset*, no. April, pp. 20–22, 2015, doi: 10.17148/iarjset.2015.2305.
  - [21] C. Saranya and G. Manikandan, “A study on normalization techniques for privacy preserving data mining,” *Int. J. Eng. Technol.*, vol. 5, no. 3, pp. 2701–2704, 2013.
  - [22] Z. Alameer, M. A. Elaziz, A. A. Ewees, H. Ye, and Z. Jianhua, “Forecasting gold price fluctuations using improved multilayer perceptron neural network and whale optimization algorithm,” *Resour. Policy*, vol. 61, no. September 2018, pp. 250–260, 2019, doi: 10.1016/j.resourpol.2019.02.014.
  - [23] H. K. Cigizoglu, “Estimation and forecasting of daily suspended sediment data by multi-layer perceptrons,” vol. 27, pp. 185–195, 2004, doi: 10.1016/j.advwatres.2003.10.003.
  - [24] E. Eğrioğlu, Ç. H. Aladağ, and S. Günay, “A new model selection strategy in artificial neural networks,” *Appl. Math. Comput.*, vol. 195, no. 2, pp. 591–597, 2008, doi: 10.1016/j.amc.2007.05.005.
  - [25] M. Ettaouil and Y. Ghanou, “Neural architectures optimization and genetic algorithms,” *WSEAS Trans. Comput.*, vol. 8, no. 3, pp. 526–537, 2009.
  - [26] J. Gomez-avila, *Adaptive PID Controller Using a Multilayer Perceptron Trained With the Extended Kalman Filter for an Unmanned Aerial Vehicle*. Elsevier Inc., 2019.
  - [27] W. Setiawan and U. T. Madura, “Prediksi Harga Saham Menggunakan Jaringan Syaraf Tiruan Multilayer Feedforward Network dengan Algoritma,” no. January, 2015, doi: 10.13140/2.1.3467.5525.
  - [28] N. P. Sakinah, I. Cholissodin, and A. W. Widodo, “Prediksi Jumlah Permintaan Koran Menggunakan

- 
- Metode Jaringan Syaraf Tiruan Backpropagation,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 7, pp. 2612–2618, 2017.
- [29] S. Naduvil-vadukootu and R. A. Angryk, “Evaluating Preprocessing Strategies for Time Series Prediction Using Deep Learning Architectures,” pp. 520–525, 2017.
- [30] M. H. SAZLI, “A brief review of feed-forward neural networks,” *Commun. Fac. Sci. Univ. Ankara*, no. January 2006, pp. 11–17, 2006, doi: 10.1501/0003168.
- [31] G. P. Zhang and M. Qi, “Neural network forecasting for seasonal and trend time series,” *Eur. J. Oper. Res.*, vol. 160, no. 2, pp. 501–514, 2005, doi: 10.1016/j.ejor.2003.08.037.