

Peramalan Utilisasi Perangkat Jaringan dan *Bandwidth* dengan Metode *Holt-Winters* dan *Multilayer Perceptron*

Muhammad Taufiq¹, Dedy Sugiarto², Abdul Rochman³

^{1,3} Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Trisakti, Jakarta, Indonesia
m.taufiq064001800502@std.trisakti.ac.id

rochman@trisakti.ac.id

² Jurusan Sistem Informasi, Fakultas Teknologi Industri, Universitas Trisakti, Jakarta, Indonesia
dedy@trisakti.ac.id

Diterima 22 April 2020

Disetujui 17 Juni 2020

Abstract—Network devices become an important medium for transferring data from one node to another node in the form of switches, routers or network security devices. The reliability of network devices must be maintained both in terms of device resources and bandwidth. The study was conducted by applying the Holt-Winters and Multilayer Perceptron (MLP) method to network device and bandwidth data utilization. The two methods are compared to assess which accuracy is better when applied to network device and bandwidth utilization data by calculating Root Mean Squared Error (RMSE) and Mean Absolute Percentage (MAPE). The results of the measurement of accuracy in the network device testing data, MLP produces a value of RMSE of 5,67 and MAPE of 2.34, and Holt-Winters produces a value of RMSE of 14.56 and MAPE of 2.95. For the results of the measurement of accuracy in the bandwidth testing data with MLP produces a value of RMSE of 0.13 and MAPE of 7.27, and Holt-Winters produces RMSE values of 2.59 and MAPE of 134.31. Based on the results of these measurements it is concluded that the MLP method has a smaller error value compared to the Holt-Winters method applied to network device and bandwidth utilization data with a span of 3 years historical data. Build web application design using Mysql as a Database Management System, Laravel as a web application framework, Pentaho for data integration and R programming for data analysis.

Index Terms—Framework Laravel, Holt-Winters Method, Multi Layer Perceptron, Network Devices and Bandwidth, Pentaho Data Integration, R

I. PENDAHULUAN

Perangkat – perangkat jaringan seperti *switch*, *router* atau perangkat keamanan jaringan harus dikelola dengan baik dan benar. Dalam melakukan tugasnya perangkat jaringan harus memiliki kehandalan dan ketersediaan sumber daya untuk mengirim dan menerima data. Pemanfaatan terhadap data tersebut tentunya akan memudahkan perusahaan untuk membuat perencanaan dan pengalokasian dana untuk kebutuhan teknologi *Information Technology*

(IT) seperti perangkat jaringan dan *bandwidth*. Dibutuhkan analisis data terhadap data-data historis utilisasi perangkat dan *bandwidth* dengan menerapkan metode-metode tertentu, sehingga perencanaan menjadi tepat. Pada perusahaan untuk menentukan kebutuhan kedepannya, belum banyak memaksimalkan penggunaan data *mining* untuk menganalisa dan meramalkan data-data utilisasi perangkat jaringan dan *bandwidth*. Untuk itu perlu dilakukan analisis data-data historis pada tahun-tahun sebelumnya lalu meramalkan data yang akan muncul ke depannya.

Penelitian terkait jaringan komputer sebelumnya “Peramalan Kebutuhan *Bandwidth* Iub Jaringan UMTS dan HSDPA Menggunakan *Fuzzy Inference System* dan *Time Series*” dengan hasil *Mean Absolute Percentage Error* (MAPE) sebesar 20,95 % sedangkan metode FTS memiliki rata-rata MAPE sebesar 11,14% [1]. Penerapan algoritma *Holt-Winters* sudah diteliti dengan judul penelitian “Prediksi Kekeringan Menggunakan Metode *Holt-Winters* Studi Kasus di Wilayah Kabupaten Boyolali” dengan hasil ramalan dengan nilai kesalahan ramalan terkecil bernilai 4,27 [2]. Pendekatan selanjutnya dengan menggunakan *Neural Network*, *Learning* merupakan kemampuan utama yang mendasar dari *Neural Network*. Salah satu jenis *Neural Network* adalah *Multilayer Perceptron* (MLP), Metode *Multilayer Perceptron* pernah diteliti dengan judul penelitian “Peramalan harga beras IR64 kualitas III menggunakan metode *Multilayer Perceptron*, *Holt-Winters* dan *Auto Regressive Integrated Moving Average*” yang menghasilkan nilai akurasi RMSE 5.67 pada metode MLP [3].

Kedua metode tersebut nantinya akan digunakan untuk meramalkan data-data terkait utilisasi jaringan, dari hasil ramalan kedua metode tersebut, Peneliti membandingkan penggunaan metode *Holt-Winters* dan *Multilayer Perceptron*, dengan menilai tingkat

kesalahan pada kedua metode tersebut, kemudian data-data utilisasi jaringan dan *bandwidth* dilakukan integrasi data dengan *Pentaho Data Integration* dari format *spreadsheet* ke bentuk ekstensi *sql* pada tabel *Database Management System MySQL*. Untuk memudahkan dalam pengguna dalam meramalkan data, pengoptimalan model dan melihat hasil ramalan, penulis membangun aplikasi berbasis web dengan menerapkan konsep *Model View Controller* dengan menggunakan *Framework Laravel*.

II. METODOLOGI PENELITIAN

A. Teknik Pengumpulan dan Analisis Data

Analisis data dapat digambarkan sebagai suatu proses yang terdiri dari beberapa langkah di mana data mentah ditransformasikan dan diproses untuk menghasilkan visualisasi data dan dapat membuat prediksi dengan model yang dipilih berdasarkan data yang dikumpulkan [4]. Langkah-langkah yang dilakukan dalam analisis data perangkat jaringan dan *bandwidth*:

1. Data Set dan Ekstraksi Data

Penulis memperoleh data histori utilisasi perangkat jaringan dan *bandwidth* perusahaan yang ekstraksi dengan bantuan *network monitoring server* baik yang diekstrak melalui aplikasi maupun diekstrak dengan manual. Data perangkat jaringan yang telah diekstrak merupakan data rata-rata *memory* setiap bulannya dari beberapa titik perangkat jaringan di Kantor Pusat Jakarta, mulai dari Januari 2017 hingga Desember 2019. Untuk data *bandwidth* yang telah diekstrak merupakan data penggunaan rata-rata *bandwidth* yang diterima setiap bulannya pada titik-titik perangkat *router* pada kantor di daerah, mulai dari Januari 2017 hingga November 2019.

2. Data Preparation

Data-data utilisasi perangkat jaringan dan *bandwidth* memiliki variabel yang bervariasi. Pada Perangkat Jaringan terdapat variabel *Average Unreachable*, *Average CPU usage*, *Average Response time*, *Average Memory usage* sedangkan pada *bandwidth* terdapat variabel *Average Transmit*, *Average Receive*, *Max Transmit*, *Max Received*. Penulis memilih *Average Memory usage* dan *Average Receive* untuk diteliti. Dilakukan penormalan terhadap data yang menjadi objek penelitian seperti satuan dan rentang bilangan. Pengoptimalan data sesuai format data *time series* yang dapat dieksekusi oleh metode peramalan. Setiap data utilisasi perangkat jaringan dan *bandwidth* lalu dibagi menjadi dua bagian yaitu data *training* dan data uji, dimana persentase data *training* adalah 85% dan data uji 15%. Data *training* akan digunakan

untuk pembelajaran pola dengan menggunakan *Holt-Winters* dan *Multilayer Perceptron* dan data uji akan dilakukan pengukuran akurasi. Berikut data set utilisasi perangkat jaringan pada Tabel 1 dan data set *bandwidth* pada Tabel 2, dengan penjelasan warna kolom sebagai berikut:

	Data Training
	Data Uji

Tabel 1. Data set utilisasi perangkat jaringan

No.	Bulan	Tahun		
		2017	2018	2019
1	Januari	401.79	413.10	414.81
2	Februari	400.29	413.10	410.75
3	Maret	406.06	416.31	421.43
4	April	410.97	417.37	417.80
5	Mei	411.40	419.29	417.37
6	Juni	404.56	416.52	418.65
7	Juli	410.97	416.95	423.35
8	Agustus	411.61	418.87	431.25
9	September	410.97	418.01	436.59
10	Oktober	415.02	417.16	444.70
11	November	413.96	417.16	447.26
12	Desember	414.38	418.23	

Tabel 2. Data set bandwidth

No.	Bulan	Tahun		
		2017	2018	2019
1	Januari	0.45	1.02	1.04
2	Februari	0.54	0.76	0.92
3	Maret	0.41	0.51	1.32
4	April	0.29	0.69	1.19
5	Mei	0.17	0.51	1.07
6	Juni	0.15	0.37	1.16
7	Juli	0.15	0.64	1.27
8	Agustus	0.33	0.64	1.31
9	September	0.65	0.91	1.50
10	Oktober	0.73	0.93	1.61
11	November	0.76	1.05	
12	Desember	0.84	1.16	

3. Metode Peramalan Holt-Winters

Metode *Holt-Winters* menggunakan tiga parameter pemulusan yaitu konstanta pemulusan, parameter untuk tren, dan parameter untuk musiman. Metode *Holt-Winters* memiliki keunggulan pada *Exponential smoothing* bersifat sederhana, intuitif dan mudah dipahami. Sederhana tetapi sangat bermanfaat untuk peramalan pendek dari data *time series* yang panjang.

Dilihat dari fluktuasi musiman pada data utilisasi perangkat jaringan dan *bandwidth* yang bersifat variasi penulis menggunakan model *Multiplicative* jika besarnya variasi musiman meningkat dengan peningkatan tingkat rata-rata deret waktu[5].

Persamaan dasar untuk model *Holt-Winters Multiplicative*:

Nilai Pemulusan keseluruhan

$$L_t = \alpha \frac{Y_t}{S_{t-s}} + (1 - \alpha)(L_{t-1} + b_{t-1})$$

Nilai Pemulusan komponen *trend*

$$b_t = \beta(L_t - L_{t-1}) + (1 - \beta)b_{t-1}$$

Nilai Pemulusan komponen musiman

$$S_t = \gamma \frac{Y_t}{L_t} + (1 - \gamma)S_{t-s}$$

Nilai Ramalan

$$F_{t+m} = (L_t + b_t m)S_{t-s+m}$$

Data utilisasi perangkat jaringan dan *bandwidth* setiap bulannya akan diproses pada persamaan model *Holt-Winters Multiplicative* dengan 3 parameter yang bisa disesuaikan yaitu (α) konstanta pemulusan, (β) parameter *trend*, dan (γ) parameter musiman. Untuk mengetahui pola pada data *training* baik utilisasi perangkat jaringan dan *bandwidth*. Model yang terbentuk dari penyesuaian parameter konstanta pemulusan, parameter *trend* dan parameter musiman. Menghasilkan nilai parameter pemulusan $\alpha = 0.471$, $\beta = 0$ dan $\gamma = 1$ pada utilisasi perangkat jaringan dan dihasilkan nilai parameter pemulusan $\alpha = 0.881$, $\beta = 0$ dan $\gamma = 1$ pada *bandwidth*.

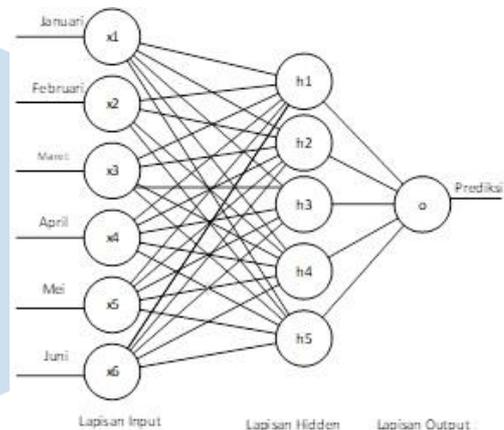
Selanjutnya dengan nilai parameter yang sudah ditetapkan, penulis melakukan peramalan 5 bulan kedepannya yaitu bulan juli, agustus, september, oktober, november pada data utilisasi perangkat jaringan dan bulan juli, agustus, september, oktober pada data *bandwidth*.

4. Metode Peramalan *Multilayer Perceptron*

Cara kerja MLP ini mengadopsi jaringan saraf mahluk hidup, sehingga terkenal bagus karena proses pembelajaran yang mampu dilakukan secara terarah. MLP dikembangkan dari *Perceptron* dimana untuk memecahkan

permasalahan yang tidak bisa dipecahkan oleh *perceptron*. Arsitektur MLP:

- Lapisan *Input* merupakan lapisan dimana setiap unit nya menyimpan data, pada lapisan *input* ini akan disimpan data utilisasi perangkat jaringan dan data *bandwidth*.
- Lapisan *Hidden* merupakan lapisan diantara lapisan *input* dan lapisan *output*, lapisan *hidden* dapat terdiri dari 1 lapisan atau lebih dari 1, dilakukan pembobotan dari sejumlah *input* dan bias dimasukkan ke fungsi aktivasi melalui fungsi *transfer* untuk menghasilkan *output*.
- Lapisan *Output* merupakan hasil kalkulasi atau pembelajaran dari lapisan sebelumnya.



Gambar 1. Arsitektur MLP

Backpropagation merupakan suatu algoritma pembelajaran yang dapat memperkecil tingkat sebuah *error* dengan bobotnya yang disesuaikan berdasarkan perbedaan dari *output* dan atau target. Di dalam *backpropagation* terdapat fungsi aktivasi [6].

Fungsi aktivasi digunakan agar nilai *threshold* membatasi nilai keluaran agar selalu dalam batas nilai yg ditetapkan. Fungsi yang digunakan adalah *sigmoid biner* memiliki nilai pada *range* 0 sampai 1, didefinisikan:

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

5. Algoritma *Backpropagation*

Inisialisasi semua bobot, lapisan *input* akan diisi bobot nilai data sesuai bulan secara sekuensial, lapisan tersembunyi akan diisi dengan nilai acak.

a. Tahap Propagasi Maju

Tiap unit *input* menerima sinyal dan meneruskannya ke unit *hidden* setelahnya. Tiap unit *hidden* mendapatkan nilai dengan mengalikan nilai dari tiap unit *input* dengan bobot menuju pada unit *hidden* lalu menghitung keluaran di unit *hidden* dengan fungsi aktivasi. Selanjutnya unit *output* akan mendapatkan nilai dengan mengalikan nilai dari tiap unit *hidden* dengan bobot menuju unit *output*, lalu menghitung keluaran unit di unit *output* dengan fungsi aktivasi.

b. Tahap Propagasi Mundur

Mengitung faktor δ unit keluaran berdasarkan *error* di setiap unit keluaran. Hitung *factor* δ unit tersembunyi berdasarkan eror di setiap lapisan tersembunyi.

c. Tahap Perubahan bobot

Menghitung semua perubahan bobot. Perubahan bobot garis yang menuju ke lapisan keluaran dan perubahan bobot garis yang menuju ke lapisan *hidden*.

B. Pengujian dan Pengukuran Akurasi Peramalan

Pada tahapan pengujian dilakukan pengukuran akurasi terhadap ramalan dengan:

- *Root Mean Squared Error (RMSE)*

Pengukuran dengan menggunakan RMSE dengan menghitung nilai akar rata-rata kuadrat kesalahan dari suatu peramalan, dimana X_i adalah nilai target aktual, F_i adalah nilai keluaran prediksi, dan n adalah jumlah baris data.

$$RMSE = \sqrt{\left(\frac{\sum_{i=1}^n (X_i - F_i)^2}{n}\right)}$$

- *Mean Absolute Percentage Error (MAPE)*

Metode ini dihitung dengan menggunakan kesalahan yang absolut pada setiap periode lalu dibagi dengan nilai penelitian atau observasi yang nyata untuk periode itu. Kemudian, merata-ratakan kesalahan persentase absolut tersebut.

$$MAPE = \frac{\sum \frac{|e_i|}{X_i} \times 100\%}{n} = \frac{\sum \frac{|X_i - F_i|}{X_i} \times 100\%}{n}$$

C. Pengkodean Program

C.1 Pengkodean pada R

Pada tahapan ini dilakukan pengkodean program untuk implementasi dari metode algoritma peramalan dan mendapat akurasi dari tingkat peramalan. Untuk metode peramalan dengan *Holt-Winters* dapat dilihat pada *Listing 1* dan untuk metode peramalan dengan *Multilayer Perceptron* dapat dilihat pada *Listing 2*. Data set akan dimasukan ke dalam program (1)berisi tentang data-data utilisasi perangkat jaringan dan *bandwidth*. Kemudian akan dibuat pemisahan data untuk data *training* sebanyak 85% dari data set(3). Pendefinisian data *training* yaitu dimulai dari data awal hingga data batas *training* (4). Didefinisikan juga data uji yang dimulai dari data batas *training* hingga jumlah data yang diinput(5). Pendefinisian variabel ramal dengan nilai kosong dan juga variabel *history* yang diisi dengan data *training* (6,7). Dilakukan peramalan dengan fungsi pengulangan dengan batas banyaknya data pada data *testing* (8). Masing-masing data *training* disesuaikan dengan model peramalannya, penyesuaian parameter *alfa* (α), *beta* (β) dan *gamma* (γ) pada *Holt-Winters*, dan penyesuaian lapisan *input*, *hidden*, dan *output* pada MLP (9). Dilakukan pemodelan untuk menemukan model yang paling optimal pada data *training* dengan menyesuaikan parameter yang ada pada model (10). Dilakukan peramalan ke x dengan menggunakan model yang paling optimal (11), kemudian nilai ramalan ke x diappend ke dalam variabel ramal (12). Data *testing* ke x lalu dimasukkan kedalam variabel observasi(13). kemudian nilai observasi di masukan kedalam variabel *history* untuk (14). Nilai yang ada pada variabel observasi di *append* kedalam variabel *history*, sebagai penanda batas terakhir permalan pada pengulangan *for* selanjutnya. Dilakukan pengulangan (9...14) sampai nilai x sama dengan nilai banyaknya data pada data *testing*. Setelah data-data ramalan didapat, lalu dilakukan pengukuran akurasi peramalan dengan menggunakan *Mean Squared Error* dengan menggunakan data *testing* dan data ramal (16), setelah itu dilakukan akar untuk menghasilkan nilai RMSE (17). Dilakukan perhitungan akurasi juga menggunakan *Mean Absolute Percent Error* (MAPE) (18).

Listing 1. Pseudo code peramalan *Holt-Winters*

```
#Proses Data
(1)Inputs: data
(2)Output: RMSE dan MAPE dari peramalan data

#Data Preparation
(3)batas <- length(data) * 0.85
(4)training <- series[0...batas]
(5)testing <- series[batas...length(data)]
(6)ramal<- empty
(7)history <- training
#Model dan Peramalan
(8) For each x in range(length(testing)) do
(9)  model <- Holt-Winters(training, alfa, beta,
    gamma, Multiplicative)
(10) model_fit<- model.fit()
(11) nramal <- model_fit.forecast()
```

```

(12) ramal.append(nramal)
(13) observasi <- testing[x]
(14) history.append(observasi)
(15) end for
      #Pengukuran Akurasi
(16) MSE = mean_squared_error(testing,ramal)
(17) RMSE = sqrt(MSE)
(18)MAPE=mean_absolute_percent_error(testing,
ramal)
    
```

Listing 2. Pseudo code peramalan MLP

```

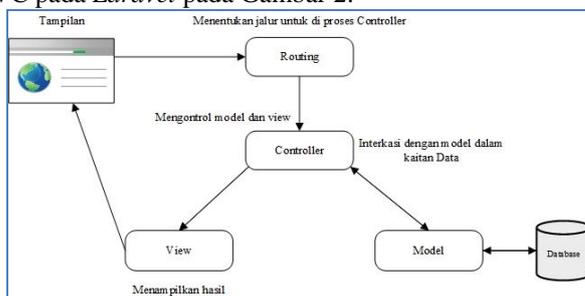
#Proses Data
(1)Inputs: data
(2)Output: RMSE dan MAPE dari peramalan
data
      #Data Preperation
(3)batas <- length(data) * 0.85
(4)training <- series[0..size]
(5)testing <- series[batas...length(data)
(6)ramal<- empty
(7)history <- training

      #Model dan Peramalan
(8)For each x in range(length(testing)) do
(9)model <- mlp(training)
(10)model fit<- model.fit()
(11) nramal <- model_fit.forecast()
(12) ramal.append(nramal)
(13) observasi <- test[t]
(14) history.append(observasi)
(15) end for

      #Pengukuran Akurasi
(16)MSE = mean_squared_error(test,ramal)
(17)RMSE = sqrt(MSE)
(18)MAPE = mean_absolute_percent_error
    
```

C.2 Laravel dan Model View Controller (MVC)

Framework Laravel menggunakan kerangka kerja pengembangan web MVC yang ditulis dalam bahasa pemrograman PHP. Dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya perawatan yang berkelanjutan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi kita dengan menyediakan sintaksis mudah yang jelas dan serangkaian fungsionalitas inti yang akan menghemat waktu dalam implementasi [7]. Berikut konsep arsitektur MVC pada Laravel pada Gambar 2.



Gambar 2. Konsep arsitektur MVC Laravel [8]

C.3.1 Model

Model biasanya berhubungan dengan data dan interaksi ke database atau *web service*. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk *file* teks, file XML maupun *web service*. Dengan adanya model, dengan mudah melakukan *create*, *read*, *update* dan *delete* pada tabel *cdevice*, namun pada penelitian ini hanya digunakan untuk menampilkan/*Read* pada tabel *cdevice* yang digunakan untuk menampilkan tabel perangkat yang di hubungkan nantinya ke dalam *ip address* tabel utilisasi harian, kemudian dilakukan visualisasi.

- Pada jendela *command* dilakukan pembuatan model pada *Laravel* yaitu:


```
php artisan make:model cdevice -m
```
- Kemudian dilakukan pengeditan pada direktori `htdocs\NetworksControllerv5(database\migration s\2019__131221_create_cdevices_table.php` seperti yang terlihat pada Gambar 3.

```

<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateCdevicesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('cdevices', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('address');
            $table->string('nama');
            $table->string('tipe');
            $table->string('manufacturer');
            $table->string('mac');
            $table->string('status');
            $table->string('kategori');
            $table->integer('jumlahport');
            $table->integer('rateport');
            $table->integer('portup');
            $table->integer('portuppenasional');
            $table->integer('adminportup');
            $table->integer('jumlahdevice');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('cdevices');
    }
}
    
```

Gambar 3. Kode program *create_cdevices_table.php*

- Selanjutnya digunakan fitur *migration* pada *Laravel* sehingga dimungkinkan untuk membuat dan memodifikasi tabel yang ada pada basisdata sesuai dengan kerangka model yang telah dibuat pada proses sebelumnya.

```
php artisan migrate
```

C.3.2 View

Bagian *presentation logic* merupakan bagian penting dalam aplikasi. Pada suatu aplikasi web,

bagian ini biasanya berupa file template *HyperText Markup Language*, yang diatur oleh *controller.view* bertugas untuk menerima dan merepresentasikan data hasil dari model dan *controller* kepada pengguna aplikasi. Pada aplikasi web disediakan halaman untuk melakukan peramalan dan menampilkan hasil ramalan data.

C.3.3 Controller

Ketika pengguna aplikasi melakukan *request* data dengan mengklik *link*, maka *controller* akan melakukan pemanggilan ke model untuk mendapatkan data yang sesuai, dan kemudian mempersiapkan data tersebut untuk ditampilkan. *File controller* di letakkan pada *folder /app/Http/Backend/AnalisisController.php* dengan kode program yang dapat dilihat pada Gambar 4.

```
public function forecastingview($address)
{
    return view("backend.analisis.forecastingview", [$address]);
}
public function forecastingviewmlp($address)
{
    return view("backend.analisis.forecastingviewmlp", [$address]);
}
public function forecastingviewmlpht($address)
{
    return view("backend.analisis.forecastingviewmlpht", [$address]);
}
```

Gambar 4. Kode program pada AnalisisController

C.3.4 Routing

Routing merupakan suatu proses untuk mengarahkan suatu permintaan atau item agar sampai pada tujuan. Fungsi *routing* pada *Laravel* terletak pada direktori *web.php* dengan kode program pada aplikasi *web* yang dapat dilihat pada Gambar 5.

```
Route::get('/backend/analisis/forecastingview/{id}', 'Backend\AnalisisController@forecastingview');
Route::get('/backend/analisis/forecastingviewmlp/{id}', 'Backend\AnalisisController@forecastingviewmlp');
Route::get('/backend/analisis/forecastingviewmlpht/{id}', 'Backend\AnalisisController@forecastingviewmlpht');
```

Gambar 5. Kode pada web.php

C.3 Integrasi Data

Setelah data di-*cleansing*, tahap selanjutnya melakukan transformasi format data *spreadsheet*, penyesuaian terhadap struktur nama kolom dan pengisian data kosong pada *data set* serta bagaimana mengubahnya dalam format *structured query language* sehingga data dapat di load kedalam MySQL bersama dengan *records*nya. Berikut skenario *transform* data yang terlihat pada Gambar 6.



Gambar 6. Skenario *transform* data

C.3.1 Microsoft Excel Input

Pada tahap ini dilakukan penarikan data utilisasi jaringan dan bandwidth ke dalam Lembar kerja *Pentho Data Integration*.

C.3.2 Select Values

Pada tahap ini dilakukan perubahan *header* kolom pada tabel dengan nama sesuai dengan *field* nantinya pada MySQL, sehingga diperlukan transformasi terhadap *header* datanya. Penyesuaian dilakukan agar membuat penyeragaman struktur *field*, sehingga nantinya memudahkan dalam pengkodean dengan *Framework Laravel*.

C.3.3 Replace In String

Pada tahap ini dilakukan perubahan nilai *record* yang tidak memiliki nilai atau memiliki string “-“ dengan mengganti nilai tersebut dengan “0”. Hal ini dilakukan agar tidak ada data yang kosong, untuk selanjutnya diberikan nilai 0.

C.3.4 Table Output

Pada tahap ini dilakukan pembuatan *field* sesuai struktur tabel yang telah di proses sebelumnya dan ekstraksi *records* ke dalam bentuk sql. Berikut hasil integrasi data pada Gambar 7 dan *records* pada MySQL pada Gambar 8.

#	Stepname	Copied	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (1/s)	input/output	
1	Microsoft Excel Input	0	0	781	781	0	0	0	0	0	Finished	0.5s	1.616	-
2	Select values	0	781	781	0	0	0	0	0	0	Finished	0.5s	1.658	-
3	Replace in string	0	781	781	0	0	0	0	0	0	Finished	0.5s	1.637	-
4	Table output	0	781	781	0	781	0	0	0	0	Finished	0.6s	1.276	-

Gambar 7. Log hasil *transformation*

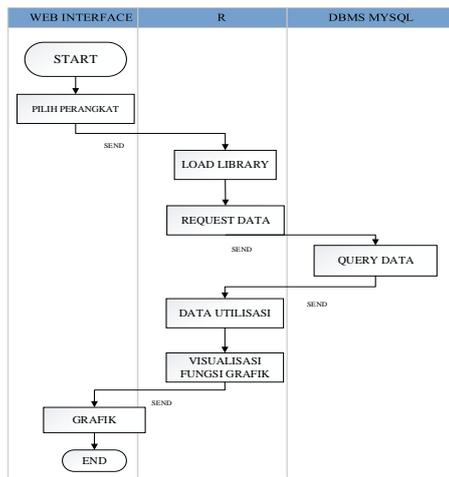
Year	Month	Bandwidth	Name	avgthroughput	maxthroughput	avgcpu	avgmemory	maxmemory	avgconnection	maxconnection	total	
2017	10	10,242,228,108	Interconnection Switch	0	0	10.0	11	24	11	0.343	43	0
2017	10	172,251,115	Git Squashed (Desktop)	0	0	10.0	10.00	24	10.0	0	0	0

Gambar 7. Records pada MySQL

C.4 Visualisasi Data pada Antarmuka Web

Dalam melakukan visualisasi data diperlukan fungsi untuk membuat grafik, fungsi tersebut ada pada program *r* yang biasa digunakan adalah *plot*.

Untuk melakukan hal tersebut diperlukan skenario pada Gambar 9 untuk dapat menampilkan grafik data pada antarmuka *web*.



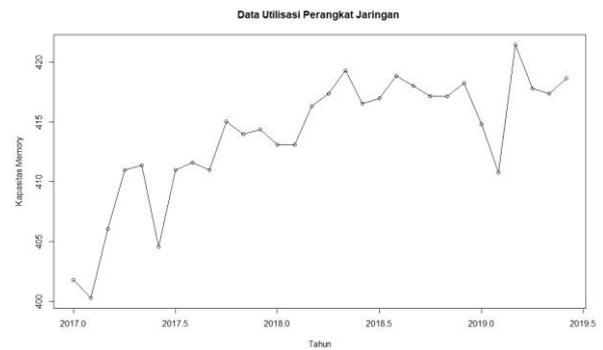
Gambar 9. Skenario visualisasi *plot*

Untuk dapat menampilkan output pada R pada antarmuka web dilakukan *editing* kode program seperti pada sintaks dibawah ini dengan bantuan fungsi *sink()*:

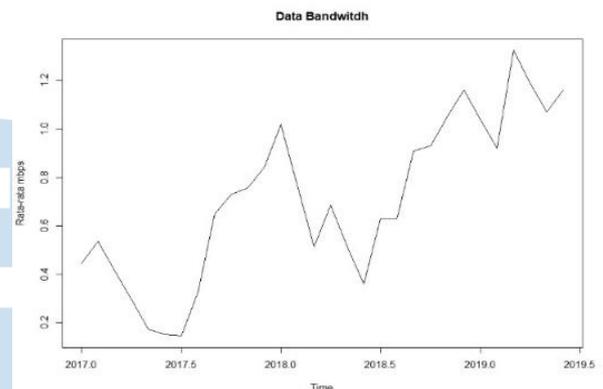
```
sink("htdocs/NetworksControllerv5/public/backend/ht
data.txt")
show(timeseries.all)
show(timeseries.training)
show(timeseries.uji)
fihht.timeseries
fihht.timeseries$fitted
htcast
sink() #stop sinking, =sink(NULL)
```

III. HASIL DAN ANALISIS

Pada plot data *Average Memory* dan data *bandwidth* seperti yang terlihat pada Gambar 10 dan Gambar 11. Utilisasi perangkat jaringan dan *bandwidth* cenderung naik di bulan Juni-Desember. Hal tersebut dipengaruhi dengan pengaksesan konten-konten yang semakin banyak pada bulan tersebut. Konten-konten yang dimaksud adalah konten terkait kegiatan dalam implementasi Indikator Kinerja Utama (IKU) pada unit, yang semakin banyak menggunakan aplikasi-aplikasi elektronik *office* dan aplikasi proses bisnis. Hal yang paling rawan adalah interval oktober hingga desember, karena pada bulan tersebut penutupan anggaran tahunan.



Gambar 10. Data *training* utilisasi perangkat jaringan



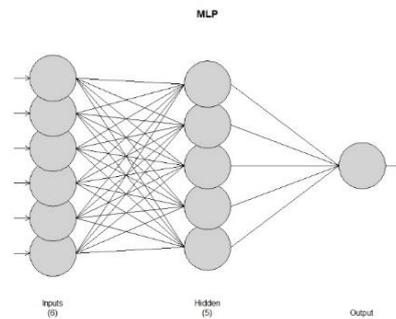
Gambar 11. Data *training* utilisasi perangkat *bandwidth*

Peramalan dengan menggunakan metode *Holt-Winters* pada data utilisasi perangkat jaringan menghasilkan nilai parameter pemulusan $\alpha = 0,47$ $\beta = 0$, dan $\gamma = 1$. Pada Gambar 12 terlihat model yang dihasilkan menyesuaikan pola pada data *training* utilisasi perangkat jaringan.

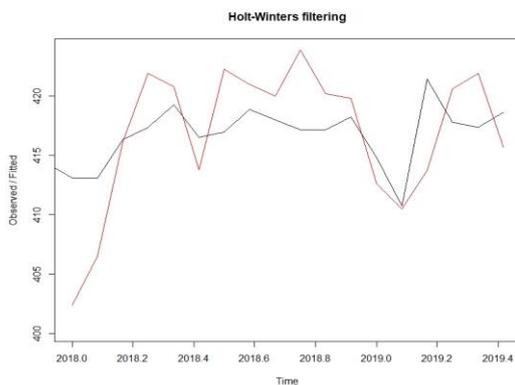
Peramalan dengan menggunakan metode *Holt-Winters* pada data *bandwidth* menghasilkan nilai parameter pemulusan $\alpha = 0,88$, $\beta = 0$, dan $\gamma = 1$. Pada Gambar 13 terlihat model yang dihasilkan menyesuaikan pola pada data *training bandwidth*.



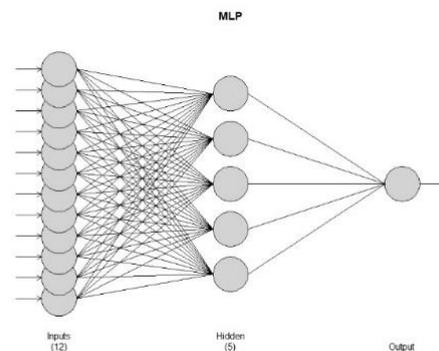
Gambar 12. Penyesuaian parameter data utilisasi perangkat jaringan dengan *Holt-Winters*



Gambar 14. Struktur model MLP pada data utilisasi perangkat jaringan



Gambar 13. Penyesuaian parameter data *bandwith* dengan *Holt-Winters*



Gambar 15. Struktur model MLP pada data *bandwidth*

Peramalan dengan menggunakan metode MLP pada data utilisasi perangkat jaringan menghasilkan model lapisan 6-5-1 pada Gambar 13 dimana 6 unit pada lapisan *input*, 5 unit lapisan *hidden*, dan 1 unit pada lapisan *output*. Data *training* utilisasi perangkat jaringan akan diformulasikan dengan model tersebut untuk mendapat nilai peramalan. Peramalan dengan menggunakan metode MLP pada data utilisasi perangkat jaringan menghasilkan model lapisan 12-5-1 pada Gambar 15 dimana 12 unit pada lapisan *input*, 5 unit lapisan *hidden*, dan 1 unit pada lapisan *output*. Data *training* utilisasi perangkat jaringan akan diformulasikan dengan model tersebut untuk mendapat nilai peramalan.

Hasil peramalan dengan menggunakan *Holt-Winters* dan MLP pada data utilisasi perangkat jaringan dan data *bandwidth* pada Tabel 3.

Tabel 3. Tabel hasil peramalan

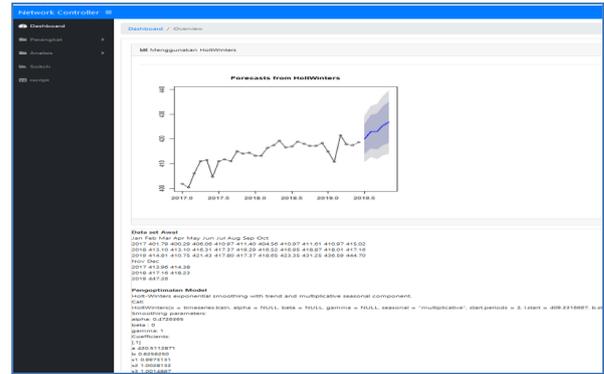
Data	Bulan	Data Aktual	Ramalan MLP	Ramalan Holt-Winters
Data Utilisasi Perangkat Jaringan	July	423.35	426.27	420.00
	Agust	431.25	420.96	422.95
	Sep.	436.59	421.85	423.01
	Okt	444.7	429.92	425.32
	Nov	447.26	438.76	426.77
Data Bandwidth	Juli	1.27	1.30	1.18
	Agust	1.31	1.25	2.28
	Sept.	1.50	1.35	4.71
	Okt	1.61	1.40	5.58

Dilakukan pengukuran akurasi peramalan pada data uji dengan menggunakan RMSE dan MAPE pada Tabel 4 akurasi peramalan.

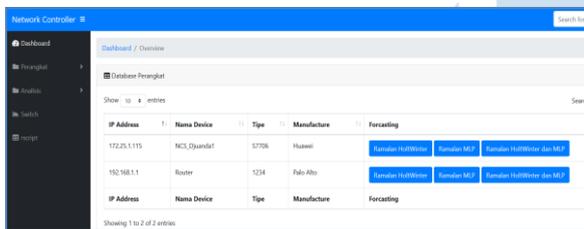
Tabel 4. Akurasi ramalan

Data	Metode Ramalan	RMSE	MAPE
Data Utilisasi Perangkat Jaringan	MLP	11.16	2.34
	HW	14.56	2.95
Data Bandwidth	MLP	0.13	7.27
	HW	2.59	134.31

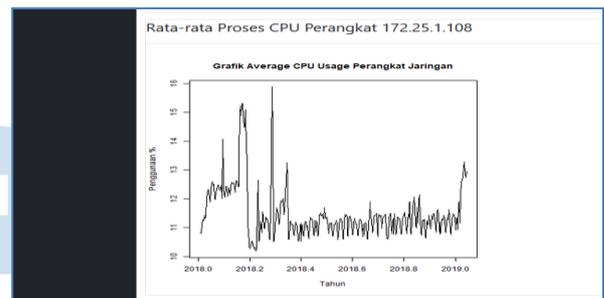
Hasil aplikasi berbasis web dengan *Framework Laravel* untuk peramalan dengan mengkoneksikan *Framework Laravel*, R dan MySQL menjadikan aplikasi dapat digunakan oleh user pengguna tanpa melakukan *editing* kode program pada R. Pada Gambar 16 dapat dilihat tampilan halaman pemilihan metode peramalan dan juga data yang ingin diramalan, pada Gambar 17 dan 18 halaman hasil ramalan *Holt-Winters* dan MLP, sedangkan pada Gambar 19 visualisasi grafik data histori pada perangkat jaringan.



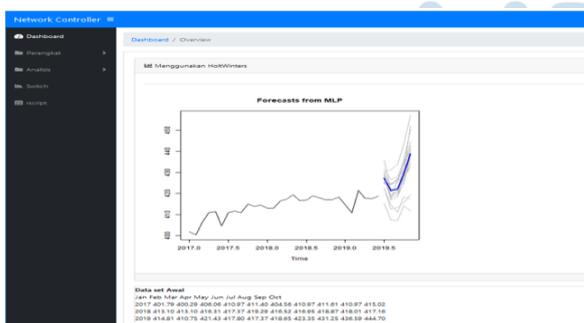
Gambar 18. Halaman hasil ramalan *Holt-Winters*



Gambar 16. Halaman pemilihan metode peramalan



Gambar 19. Halaman grafik utilisasi



Gambar 17. Halaman hasil ramalan MLP

IV. SIMPULAN

Data utilisasi perangkat jaringan dan *bandwidth* mengalami peningkatan bulan-bulan pada akhir tahun. Berdasarkan pengukuran nilai akurasi menggunakan RMSE dan MAPE terhadap utilisasi perangkat jaringan dan *bandwidth* didapatkan bahwa metode MLP memiliki nilai kesalahan (*error*) lebih kecil dibandingkan dengan metode *Holt-Winters*. Perlu dilihat nilai ambang batas *memory* pada perangkat jaringan, jika ramalan data melebihi nilai ambang batas, maka menjadi perhatian untuk dievaluasi. Untuk *bandwidth* perlu dilihat kapasitas *bandwidth* yang disewa, jikalau hasil ramalan data pada *bandwidth* melebihi perlu dilakukan evaluasi penambahan kapasitas *bandwidth* jaringan. Aplikasi berbasis web dengan konsep MVC pada *Framework Laravel* dapat di visualisasikan hasilnya dalam bentuk grafik dengan dikoneksikan pada program R, hal tersebut dapat mempermudah dalam menggambarkan keadaan dari pemanfaatan sumber daya perangkat jaringan dan peramalan data. Penggunaan *Pentaho Data Integration* untuk melakukan ekstraksi data yang *records*-nya banyak dengan format *spreadsheet* ditransformasikan dan di-load ke dalam MySQL dapat mengefesienkan waktu pekerjaan.

DAFTAR PUSTAKA

- [1] A. Wahyu, E. Yudaningsy, and S. H. Pramono, "Peramalan Kebutuhan Bandwidth Iub Jaringan UMTS dan HSDPA Menggunakan Fuzzy Inference System dan Time Series," *J. EECCIS*, 2014.
- [2] D. A. W. S. Putra, K. D. Hartomo, and R. Tanone, "Model Prediksi Kekeringan Menggunakan Metode Holt-Winters (Studi Kasus: Wilayah Kabupaten Boyolali)," *Indones. J. Comput. Model.*, 2018, doi: 10.24246/j.icm.2018.v1.i1.p36-41.
- [3] A. B. Ariwibowo, D. Sugiarto, I. A. Marie, and J. F. Agatha, *Peramalan harga beras IR64 kualitas III menggunakan metode Multilayer Perceptron, Holt-Winters dan Auto Regressive Integrated Moving Average*, vol. XI. 2019.
- [4] F. Nelli, *Python Data Analytics*. New York: Apress, 2015.
- [5] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction Time Series Analysis and Forecasting*, Second. New Jersey: John Wiley & Sons, Inc., 2016.
- [6] Y. A. Lesnussa, S. Latuconsina, and E. R. Persulesy, "Aplikasi Jaringan Saraf Tiruan Backpropagation untuk Memprediksi Prestasi Siswa SMA (Studi kasus: Prediksi Prestasi Siswa SMAN 4 Ambon)," *J. Mat. Integr.*, 2015, doi: 10.24198/jmi.v11.n2.9427.149-160.
- [7] S. McCool, *Laravel Starter*. Birmingham: Packt Publishing, 2012.
- [8] Chandra Shekhar, "General architecture of Laravel based application Chandra Shekhar Pandey." [Online]. Available: <http://www.cutehits.com/2017/06/general-architecture-of-laravel-based-application/>. [Accessed: 21-Apr-2020].

