JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika

[E-ISSN 2656-0321]
[Vol 16 No 1]
[Januari 2022]

PERBANDINGAN WAKTU EKSEKUSI PERAMALAN HARGA KOMODITAS PANGAN MENGGUNAKAN SPARKR DAN R STUDIO

Dedy Sugiarto [a,1,*], Dimmas Mulya [b,2], Abdul Rochman [b,3], Is Mardianto [b,4]

[a] Program Studi Sistem Informasi, Fakultas Teknologi Industri, Universitas Trisakti, Jarkata Indonesia
[b] Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Trisakti, Jarkata Indonesia
[1] dedy@trisakti.ac.id*; [2] dimmas.mulya@trisakti.ac.id; [3] abdul.rochman@trisakti.ac.id; [4] mardianto@trisakti.ac.id
* corresponding author

ARTICLE INFO

Peramalan Big Data
Harga Komoditas Pangan
SparkR
R studio
Multilayer Perceptron

ABSTRACT

The arrival of the big data era with characteristics such as large volumes of data makes the calculation of execution time a concern when carrying out data analytics processes, such as forecasting food commodity prices. This study aims to examine the effect of the big data framework through the use of sparkR. The test is carried out by varying several deep learning forecasting models, namely the multi-layer perceptron model and by using the price of one food commodity from 2018 to 2020. The results show that sparkR is significantly shorter its execution time when compared to R studio. The results of testing the influence of the MLP model also show that a model with two hidden layers with a maximum node of 13 nodes in hidden layers 1 and 2 produces the longest execution time compared to only using 1 hidden layer with 5 nodes or using two hidden layers with a number of nodes of 5 and 3.

1. Pendahuluan

Data harga komoditas pangan mempunyai data yang skala besar (*large-scale*) dan mempunyai keragaman data baik pada level pasar induk maupun pada level retail serta keragaman jenis atau varian komoditasnya yang dapat berubah setiap waktu. Data harga pangan di DKI dapat dilihat dari berbagai situs antara lain https://infopangan.jakarta.go.id/, https://hargapangan.id/, http://www.foodstation.co.id/ dan dapat berubah dari hari ke hari.

Proses pembentukan harga dipengaruhi baik oleh faktor internal seperti biaya produksi dan biaya transportasi maupun faktor eksternal seperti harga alternatif untuk lingkungan pasar [1]. Di sektor pertanian, setiap kegiatan dalam proses produksi selalu dihadapkan pada situasi risiko dan ketidakpastian. Sumber ketidakpastian yang penting di sektor pertanian adalah ketidakpastian produk dan harga pertanian. Ketidakpastian harga produk pertanian akan mengakibatkan turunnya dan naiknya pendapatan yang diterima petani dari hasil produksi pertaniannya [2]. Salah satu upaya untuk mengantisipasi fluktuasi harga adalah dengan melakukan peramalan harga. Peramalan harga dimaksudkan untuk meramalkan harga-harga di masa yang akan datang dalam jangka waktu tertentu, dengan output berupa harga di masa yang akan datang.

Metode pembelajaran atau metode cerdas melalui penggunaan jaringan syaraf tiruan dan *deep learning* telah menjadi metode prediksi harga yang semakin banyak digunakan [3,4]. Dengan datangnya era data besar dan kekuatan komputasi baru-baru ini, pembelajaran mesin dan pembelajaran mendalam telah menjadi bagian penting dari model peramalan deret waktu generasi berikutnya. Hal ini semakin didukung dengan tersedianya *framework back propagation open source* [3].

Penelitian terdahulu telah dapat menghasilkan beberapa model peramalan harga beras dengan jenis IR 64-III [5,6,7] namun hanya mempertimbangkan aspek akurasi hasil peramalan dan belum melibatkan aspej waktu eksekusi terkait penggunaan ukuran data yang lebih besar. Oleh karena itu muncul permasalahan apakah penggunakan salah satu *framework big data* yaitu Apache Spark baik yang

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 16 No 1]
[Januari 2022]

diintegrasikan dengan R (sparkR) maupun dengan python (pyspark) dapat mempercepat waktu eksekusi. Sasarannya adalah tidak hanya menghasilkan satu hasil ramalan berdasarkan komoditas tertentu tetapi juga tidak membutuhkan waktu lama dalam eksekusi. Tujuan penelitian ini adalah menguji perbedaan waktu eksekusi peramalan antara sparkR dan R studio dengan menggunakan metode *multilayer perceptron* dengan kondisi layar tersembunyi yang berbeda-beda jumlahnya.

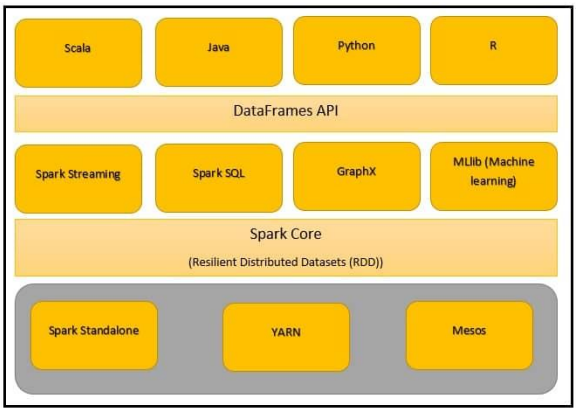
2. Landasan Teori

2.1. RStudio

Rstudio adalah *Integrated Development Environment (IDE)* untuk bahasa pemrograman R dan memiliki *user interface* yang lebih baik daripada RGui. Rstudio memiliki edisi *open source* dan edisi *commercial*. Rstudio dapat mempermudah pengguna dalam menggunakan bahasa pemrograman R dengan *user interface* yang lebih mudah dipahami. [8,9] Bahasa pemrograman R sendiri adalah bahasa pemrograman yang dikembangkan secara khusus untuk menangani permasalahan statistik. [10]

2.2. Apache Spark

Apache Spark adalah sebuah mesin analitik terpadu bersifat open-source yang ditujukan untuk memproses data yang berskala besar. Apache Spark telah muncul sebagai kerangka kerja *de facto* untuk analitik data besar dengan model pemrograman dalam memori yang canggih dan pustaka tingkat atas untuk pembelajaran mesin yang dapat diskalakan, analisis grafik, *streaming*, dan pemrosesan data terstruktur. Apache Spark juga menyediakan kerangka kerja *cluster computing* secara umum dengan API terintegrasi untuk bahasa pemrograman Scala, Java, Python dan R, sehingga memungkinkan untuk menjalankan beberapa komputer sebagai satu komputer. [11,12]



Gambar 1. Arsitektur Apache Spark

2.3. *Artificial Neural Network*

Artificial Neural Network atau jaringan syaraf tiruan adalah paradigma yang berusaha meniru struktur mikro otak dan digunakan secara luas dalam masalah kecerdasan buatan mulai dari tugas pengenalan pola sederhana, hingga manipulasi simbolik tingkat lanjut. *Multilayer Perceptron* adalah salah satu contoh dari bentuk jaringan syaraf tiruan yang sering digunakan untuk menemukan solusi dari beberapa masalah yang berbeda, termasuk pengenalan pola dan interpolasi. Jaringan Syaraf Tiruan terdiri dari beberapa bagian, yaitu lapisan masukkan (*Input Layer*) yang berfungsi untuk menerima nilai masukkan sistem, lapisan tersembunyi (*Hidden Layer*) berfungsi mengelolah nilai masukkan dari lapisan masukkan dan memberikan ke lapisan keluaran, kemudian lapisan keluaran (*Output Layer*) berfungsi untuk memberikan nilai keluaran hasil perhitungan. [13]

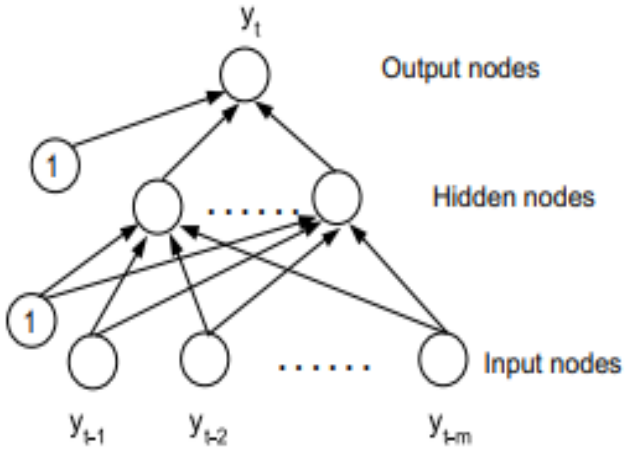
https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 16 No 1]
[Januari 2022]

3. Metode Penelitian

Metode penelitian yang digunakan adalah metode penelitian eksperimen dengan melibat beberapa faktor dan beberapa respon. Faktor atau variable *predictor* yang digunakan adalah kondisi infrastuktur yang digunakan yaitu menggunakan *framework* sparkR sebagai salah satu *framework cluster computing* untuk pengolahan big data serta alternatifnya adalah kondisi R studio berjalan tanpa integrasi dengan apache spark. Faktor kedua adalah beberapa model peramalan menggunakan multilayer perceptron yang divariasikan berdasarkan jumlah *hidden layer* dan jumlah *node* dalam *hidden layer*. Respon yang diukur adalah akurasi hasil peramalan dalam sebuah ukuran numerik yang *mean square error* (MSE) dan waktu eksekusi. Tahapan pada penelitian terbagi menjadi beberapa tahapan yaitu pertama adalah analisis kebutuhan dan penyusunan aplikasi analitik *big data* untuk peramalan pasokan dan harga berbagai komoditas di situs info pangan Jakarta. Akuisisi data harga harian mulai dari tahun 2018 sampai dengan 2020 ke dalam *database* menggunakan teknik *extract-transform-load* dengan bantuan *software* Pentaho sehingga dapat membentuk *Data Warehouse* yang siap dilakukan penambangan data. Pemodelan peramalan menggunakan metode *Multilayer Perceptron*, namun divariasikan menggunakan satu dan beberapa *hidden layer* serta dilakukan dalam ekosistem *big data* pada klaster tervirtualisasi yaitu Apache Spark yang terintegrasi dengan R. Penelitian dilakukan rumah dengan menggunakan fasilitas komputer dengan spesifikasi *RAM* 16 GB dan *processor* Intel Core i7-6700 3,40 GHz.

Metode MLP merupakan salah satu metode dalam jaringan syaraf tiruan yang digunakan untuk kasus peramalan data deret waktu dengan struktur jaringan seperti dapat dilihat pada Gambar 2. Peramalan dilakukan dengan bantuan perangkat lunak R serta dengan bantuan library “sparkR” dan “nnfor” seperti dapat dilihat pada Gambar 3.



Gambar 2. Model jaringan syaraf tiruan untuk peramalan data deret waktu [15]

```
library(nnfor)
library(dplyr)
library(SparkR,lib.loc = c(file.path(Sys.getenv("SPARK_HOME"),"R","lib")))
library(Metrics)
library(ggplot2)
library(ggpubr)
library(forecast)
data_kino = read.delim("clipboard")
data_kino = subset(data_kino, id_pasar == 70 & id_komoditi == 66)
sparkR.session()
data_train = subset(data_kino, tanggal < as.Date("2020-07-01"))
data_test = subset(data_kino, tanggal >= as.Date("2020-07-01"))
day = nrow(data_test)
getForecastArima <- function(x) {
```


https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 16 No 1]
[Januari 2022]

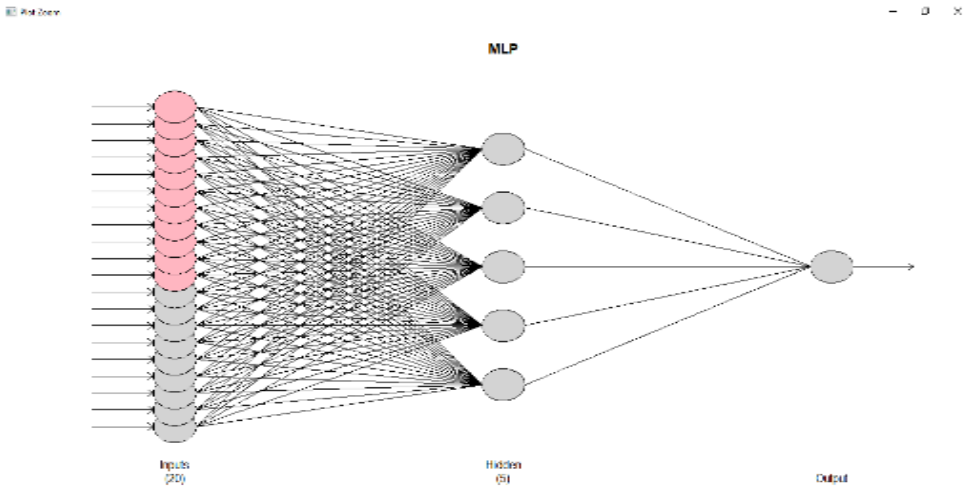
```
  require(forecast)
  x <- ts(x, frequency = 365)
  fit <- auto.arima(x)
  f <- forecast(fit,h=day)
  return(f)
}
getForecastMLP <- function(x) {
  require(forecast)
  require(nnfor)
  x <- ts(x, frequency = 365)
  fit <- mlp(x)
  print(fit$MSE)
  f <- forecast(fit,h=day)
  return(f)
}
getForecastMLP2 <- function(x) {
  require(forecast)
  require(nnfor)
  x <- ts(x, frequency = 365)
  fit <- mlp(x, hd=c(5,3))
  f <- forecast(fit,h=day)
  return(f)
}
getForecastMLP3 <- function(x) {
  require(forecast)
  require(nnfor)
  x <- ts(x, frequency = 365)
  fit <- mlp(x, hd=c(5,3,2))
  f <- forecast(fit,h=day)
  return(f)
}

system.time({
fcastArima <- spark.lapply(list(data_train$harga),getForecastArima)
})
fcastArima = as.data.frame(fcastArima)
data_arima = data.frame(Tanggal = data_test$tanggal, Harga= fcastArima$Point.Forecast)
rmse_arima = rmse(data_test$harga, data_arima$Harga)

system.time({
fcastMLP <- spark.lapply(list(data_train$harga),getForecastMLP)
})
```

Gambar 3. Potongan Script R yang digunakan untuk pengujian waktu eksekusi

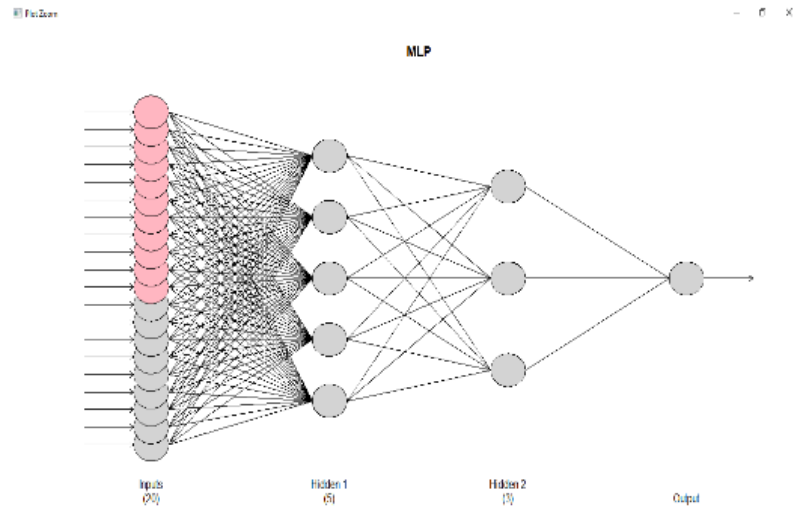
Model jaringan MLP yang digunakan dibangkitkan dengan menggunakan perintah di R Studio dengan model seperti dapat dilihat pada Gambar 4 sampai dengan Gambar 6.



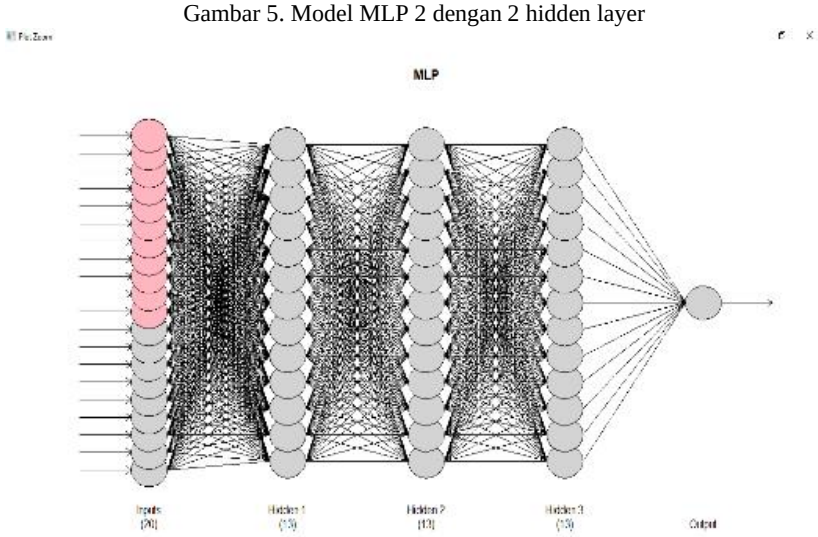
Gambar 4. Model MLP 1 dengan 1 hidden layer

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 16 No 1]
[Januari 2022]



Gambar 5. Model MLP 2 dengan 2 hidden layer



Gambar 6. Model MLP 3 dengan 3 hidden layer.

4. Hasil dan Analisis

Uji coba untuk mengukur *running time* berhasil dilakukan dengan menggunakan *script* pada Gambar 3 yang dijalankan pada komputer yang sama. Kemudian berdasarkan hasil uji coba yang telah berhasil dilakukan yang dapat ditunjukkan pada Tabel 1. dimana menghasilkan nilai *running time* yang beragam dalam satuan sekon. Hasil uji coba terhadap waktu mendapatkan informasi bahwa pengujian untuk mengolah data yang berskala besar dengan menggunakan sparkR menghasilkan waktu yang relatif lebih singkat secara signifikan. Hal ini diperkuat dengan melakukan Uji Anova yang ditunjukkan pada Gambar 7. dan Uji Tukey yang ditunjukkan pada Gambar 8. dimana menghasilkan rata-rata selisih waktu sekitar 16 sekon. Kemudian, selain itu, hasil uji coba mengenai pengaruh model MLP terhadap durasi waktu eksekusi atau *running time* juga menunjukkan bahwa model MLP dengan tiga *hidden layer* dengan *node* maksimum yakni sebesar 13 *node* pada masing - masing *hidden layer* tersebut menghasilkan waktu eksekusi yang lebih lama daripada model lainnya yang memiliki jumlah *hidden layer* dan *node* yang lebih sedikit.

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 16 No 1]
[Januari 2022]

Tabel 1. Hasil Pengujian Waktu Eksekusi

MLP Model	*Software*	*Running Time (Second)*	*Rata-rata Running Time per Model (Second)*	*Rata-Rata Running Time per software (Second)*
1	sparkR	60.58		
1	sparkR	60.71	64.49	
1	sparkR	72.19		
2	sparkR	67.3		
2	sparkR	72.74	68.01	72.6456
2	sparkR	64		
3	sparkR	89.71		
3	sparkR	82.25	85.43	
3	sparkR	84.33		
1	R studio	81.42		
1	R studio	83.22	83.77	
1	R studio	86.67		
2	R studio	94.87		
2	R studio	90.3	91.66	89.2411
2	R studio	89.82		
3	R studio	96.11		
3	R studio	89.17	92.29	
3	R studio	91.59		

```
> summary(aov_rt)
                  Df Sum Sq Mean Sq F value   Pr(>F)
dataku2$Software   1 1239.4  1239.4   39.47 2.02e-05 ***
dataku2$MLP_Model  2  661.8   330.9   10.54  0.00161 **
Residuals         14  439.7    31.4
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> aov_rt2=aov(dataku2$Running_time~dataku2$Software+dataku2$MLP_Model+dataku2$Software:data
ku2$MLP_Model)
> summary(aov_rt2)
                                 Df Sum Sq Mean Sq F value   Pr(>F)
dataku2$Software                  1 1239.4  1239.4   70.13 2.35e-06 ***
dataku2$MLP_Model                 2  661.8   330.9   18.72 0.000204 ***
dataku2$Software:dataku2$MLP_Model  2  227.6   113.8    6.44 0.012590 *
Residuals                        12  212.1    17.7
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Gambar 7. Hasil uji ANOVA untuk respon *running time*

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 16 No 1]
[Januari 2022]

```
> TukeyHSD(aov_rt, which = "dataku2$Software")
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = dataku2$Running_time ~ dataku2$Software + dataku2$MLP_Model)

$`dataku2$Software`
                     diff       lwr       upr    p adj
sparkR-R studio -16.59556 -22.26148 -10.92963 2.02e-05

> TukeyHSD(aov_rt, which = "dataku2$MLP_Model")
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = dataku2$Running_time ~ dataku2$Software + dataku2$MLP_Model)

$`dataku2$MLP_Model`
         diff        lwr      upr     p adj
2-1  5.706667 -2.7613705 14.17470 0.2173811
3-1 14.728333  6.2602961 23.19637 0.0012292
3-2  9.021667  0.5536295 17.48970 0.0363247
```

Gambar 8. Hasil uji TUKEY untuk respon *running time*

Dari uji coba tersebut, dapat ditarik informasi yang penting bahwa pentingnya penggunaan Apache Spark sebagai mesin analitik terpadu dalam menangani data berskala besar dan juga penentuan dalam penggunaan arsitektur dari jaringan syaraf tiruan. Meskipun Program dijalankan menggunakan tahapan yang sama, dengan membedakan *platform* yang menangani dapat memberikan tingkat signifikan yang berbeda dalam hal waktu eksekusi (*running time*) dengan perbedaan rata-rata 16.59556 detik. Hal ini dapat terjadi dikarenakan kemampuan manajemen memori dari Apache Spark yang dapat menggunakan *resource* komputer dengan secara menyeluruh dan efisien, sehingga dapat mempercepat proses dalam pengelolahan data yang berskala besar (*Big Data*). Hal lain yang juga menjadi pertimbangan adalah data yang digunakan untuk menganalisa suatu permasalahan pada zaman teknologi ini sudah jarang menggunakan data yang berskala kecil. Kemudian, mengenai perbedaan kecepatan waktu eksekusi (*running time*) dari ketiga model MLP, menunjukkan bahwa penentuan tahapan pemrograman, struktur dan parameter yang digunakan juga sangat berpengaruh dalam waktu eksekusi program (*running time*). Hal ini ditunjukkan bahwa komputer cenderung lebih cepat menyelesaikan model MLP 1 ketimbang dengan MLP 2 dan 3. Sehingga, meskipun sudah menggunakan platform untuk menangani data yang berskala besar, tetap bisa melambat jika tingkat kompleksitas dari tahapan tidak diringankan. Dengan demikian, dapat ditarik kesimpulan analisa bahwa dengan bantuan platform untuk menangani dan mengelolah data yang berskala besar dan mengoptimalkan tingkat kompleksitas dari tahapan pemrograman, struktur dan parameter yang digunakan dapat mempermudah dan mempercepat dalam proses menganalisa data yang berskala besar.

5. Kesimpulan

Pengujian pengaruh *platform* atau *framework big data* melalui penggunakan sparkR (integrasi apache spark dan R studio) menunjukkan bahwa hasil yang secara signifikan lebih singkat waktu eksekusinya bila dibandingkan dengan R studio. Hal ini ditunjukkan dengan pembuktian *running time* yang dihasilkan dari eksperimen dan memiliki nilai perbedaan rata-rata, yakni sebesar 16.5956 detik.

Hasil pengujian pengaruh model MLP juga menunjukkan model dengan dua hidden layer dengan node maksimum yaitu sebesar 13 node pada ketiga hidden layer menghasilkan waktu eksekusi yang paling lama (rata-rata *running time* 85.43 detik pada sparkR dan 92.29 detik pada R Studio) dibandingkan hanya menggunakan 1 hidden layer dengan 5 node (rata-rata *running time* 63.49 detik

https://doi.org/10.47111/JTI

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 16 No 1]
[Januari 2022]

pada sparkR dan 83.37 detik pada R Studio) atau menggunakan 2 hidden layer dengan jumlah node sebesar 5 dan 3 (rata-rata *running time* 68.01 detik pada sparkR dan 91.66 detik pada R Studio).

Dengan eksperimen ini dapat menunjukkan bahwa dengan menggunakan bantuan SparkR dapat memberikan *running time* yang lebih rendah dibandingkan tanpa menggunakan SparkR pada semua model MLP. Hal ini akan sangat mempengaruhi jika arsitektur model MLP yang digunakan sudah memiliki banyak jumlah hidden layer dan node, maka akan semakin kompleks model tersebut, sehingga memerlukan waktu eksekusi yang lebih tinggi.

Dari kesimpulan diatas, SparkR dapat menjadi salah satu solusi dalam mempercepat waktu *running time* pada RStudio.

Daftar Pustaka

[1] L. Wang, J. Feng, X. Sui, X. Chu, and W. Mu,"Agriculture product price forecasting method : research advances and trend," British Food Journal, vol. 122, no. 7, pp. 2121-2138. Januari. 2020

[2] M. E. Lasulika,"Prediction of corn commodities using k-NN and particle swarm optimization as a selection feature,"ILKOM:Jurnal Ilmiah, vol. 9, no. 3, December 2017

[3] H. Wu, H. Wu, M. Zhu. W. Chen, W. Chen,' A new method of large-scale short-term forecasting of agricultural commodity prices: illustrated by the case of agricultural markets in Beijing,"Journal of Big Data, vol. 4, no. 1, pp. 1-22, 2017.

[4] B. Lim and S. Zohren,"Time-series forecasting with deep learning : a survey." Philosophical Transactions, Juli 2020

[5] S. Sen, D. Sugiarto, A. Rochman, Rice Price Prediction Using Multilayer Perceptron (MLP) and Long Short Term Memory (LSTM). Ultimatics : Jurnal Teknik Informatika, Vol 12 No 1 (2020)

[6] A. B. Ariwibowo, D. Sugiarto, I.A.Marie, J.A.Siahaan, Peramalan harga beras IR64 kualitas III menggunakan metode Multi Layer Perceptron, Holt-Winters dan Auto Regressive Integrated Moving Average, Ultimatics : Jurnal Teknik Informatika, Vol 11 No 2 (2019)

[7] I. Mardianto, M.I. Gunawan, D. Sugiarto, A. Rochman, Perbandingan Peramalan Harga Beras Menggunakan Metode ARIMA pada Amazon Forecast dan Sagemaker, Jurnal Resti, Vol 4 No 3 (2020)

[8] M. R. Faisal, D. T. Nugrahadi, "Belajar Data Science: Klasifikasi dengan Bahasa Pemrograman R", Vol. 1, Google: M. R. Faizal, 2017. Access on: Aug. 1, 2021. [Online]. Available: https://books.google.co.id/books?id=svXUDQAAQBAJ

[9] MPJ. Van der Loo, "Learning RStudio for R statistical computing", Packt Publishing Ltd. 2012

[10] P. U. Gio, A. R. Effendie, "Belajar Bahasa Pemrograman R", 2018.

[11] S. Salloum, R. Dautov, X. Chen et al., "Big data analytics on Apache Spark", International Journal of Data Science and Analytics, Vol. 1, No. 3, pp. 145-164, 2016.

[12] https://spark.apache.org/

[13] L. Noriega, "Multilayer perceptron tutorial." School of Computing. Staffordshire University. 2005.

[14] C. Krome, V. Sander, Time series analysis with apache spark and its applications to energy informatics, Proceeding of Conference on Energy Informatics , Oldenburg, Germany. 11-12 October 2018

[15] 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.

https://doi.org/10.47111/JTI