ISSN : 2527-5836
e-ISSN : 2528-0074

Vol. 9 No. 1, Januari 2024

JISKa

Jurnal Informatika Sunan Kalijaga

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta

Tim Pengelola JISKa (Jurnal Informatika Sunan Kalijaga)
Edisi Januari 2024

Mitra Bestari (*Reviewer*)

Reviewer Internal:

1. Agus Mulyanto, S.Si., M.Kom., ASEAN Eng. (UIN Sunan Kalijaga Yogyakarta, Indonesia)
2. Mandahadi Kusuma, M.Eng. (UIN Sunan Kalijaga Yogyakarta, Indonesia)
3. Maria Ulfa Siregar, Ph.D. (UIN Sunan Kalijaga Yogyakarta, Indonesia)

Reviewer Eksternal (Mitra Bestari):

1. Ahmad Fathan Hidayatullah, M.Cs. (Universitas Islam Indonesia Yogyakarta, Indonesia)
2. Alam Rahmatulloh, M.T. (Universitas Siliwangi Tasikmalaya, Indonesia)
3. Alfan Farizki Wicaksono, Ph.D. (Universitas Indonesia, Indonesia)
4. Ardiansyah Musa Efendi, Ph.D. (Chonnam National University, Korea Selatan)
5. Dr. Aris Puji Widodo, M.T. (Universitas Diponegoro, Indonesia)
6. Dr. Cahyo Crysdian (UIN Maulana Malik Ibrahim Malang, Indonesia)
7. Dr. Enny Itje Sela (Universitas Teknologi Yogyakarta, Indonesia)
8. Dr.Eng. Ganjar Alfian (Universitas Gadjah Mada, Indonesia)
9. Muhammad Habibi, M.Cs. (Universitas Jenderal Achmad Yani Yogyakarta, Indonesia)
10. Muhammad Rifqi Maarif, M.Eng. (Universitas Jenderal Achmad Yani Yogyakarta, Indonesia)
11. Dr.Eng. M. Muhammad Syafrudin (Sejong University, Korea Selatan)
12. Dr.Eng. M. Alex Syaekhoni (Dongguk University Seoul, Korea Selatan)
13. Norma Latif Fitriyani, M.Sc. (Sejong University, Korea Selatan)
14. Nur Aini Rakhmawati, Ph.D. (Institut Teknologi Sepuluh November, Indonesia)
15. Prof. Dr. Hj. Okfalisa, S.T., M.Sc. (UIN Sultan Syarif Kasim Riau, Indonesia)
16. Oman Somantri, M.Kom. (Politeknik Negeri Cilacap, Indonesia)
17. Puji Winar Cahyo, M.Cs. (Universitas Jenderal Achmad Yani Yogyakarta, Indonesia)
18. Rischan Mafrur, M.Eng. (The University of Queensland Brisbane, Australia)
19. Dr.Eng. Sunu Wibirama, M.Eng. (Universitas Gadjah Mada, Indonesia)
20. Yudistira Dwi Wardhana Asnar, Ph.D. (Institut Teknologi Bandung, Indonesia)

ISSN : 2527-5836

e-ISSN: 2528-0074

JISKa (Jurnal Informatika Sunan Kalijaga)

Vol. 9, No. 1, JANUARI 2024

DAFTAR ISI

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 1, JANUARI, 2024, Pp. 1 – 9
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Segmentasi Pelanggan Penjualan *Online* Menggunakan Metode *K-means Clustering*

Candra Hafidz Ardana [(1)*], Adlian Aldita Alif Aisyah Ainur Khoyum [(2)], Muhammad Faisal [(3)]
[1,3] Teknik Informatika, Fakultas Sains dan Teknologi, UIN Maulana Malik Ibrahim, Malang
[2] Ilmu Hukum, Fakultas Hukum, Universitas Negeri Semarang, Semarang
e-mail : {candra.ardana99,adlianalditaalf}@gmail.com, mfaisal@ti.uin-malang.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 3 Juni 2023, direvisi 19 Agustus 2023, diterima 15 Desember 2023, dan dipublikasikan 25 Januari 2024.

Abstract

Customer segmentation is an essential strategy in the online selling industry to understand customer preferences and behavior. This article proposes applying the K-means clustering method in online sales customer segmentation. The method used is the descriptive method. The steps of the research method include literature studies and data processing to be analyzed using the K-means clustering method. The K-means clustering method is then applied to customer data to group it based on relevant attributes. The segmentation results are evaluated and scored using the clustering evaluation metric. The main objective is to explain the use of the K-means clustering method in online sales customer segmentation, focusing on obtaining more profound insights into customer behavior. Efficient customer segmentation allows companies to target customer groups more precisely and efficiently. This article provides practical insights and guidance for e-commerce companies in implementing customer segmentation using K-means clustering to increase efficiency in targeting segmented customers.

Keywords: Customer Segmentation, Online Sales, E-Commerce, K-means Clustering, Clustering

Abstrak

Segmentasi pelanggan merupakan strategi penting dalam industri penjualan *online* untuk memahami preferensi dan perilaku pelanggan. Artikel ini mengusulkan penerapan metode *K-means clustering* dalam segmentasi pelanggan penjualan *online*. Metode penelitian yang digunakan adalah metode deskriptif. Langkah-langkah metode penelitian yang dilakukan meliputi studi literatur, pemrosesan data untuk dianalisis dengan metode *K-means clustering.* Metode *K-means clustering* kemudian diterapkan pada data pelanggan untuk mengelompokkannya berdasarkan atribut yang relevan. Hasil segmentasi dievaluasi dan dinilai menggunakan metrik evaluasi *clustering*. Tujuan utamanya adalah menjelaskan penggunaan metode *K-means clustering* dalam segmentasi pelanggan penjualan *online* dengan fokus pada memperoleh wawasan yang lebih mendalam tentang perilaku pelanggan. Segmentasi pelanggan yang efisien memungkinkan perusahaan untuk menargetkan kelompok pelanggan dengan lebih tepat dan efisien. Artikel ini memberikan wawasan dan panduan praktis bagi perusahaan *e-commerce* dalam mengimplementasikan segmentasi pelanggan menggunakan *K-means clustering* untuk meningkatkan efisiensi dalam menargetkan pelanggan segmen.

Kata Kunci: Segmentasi Pelanggan, Penjualan Online, *E-Commerce*, *K-Means Clustering*, Pengklasteran

1. PENDAHULUAN

Dalam industri penjualan *online*, memperkuat hubungan antara pelanggan adalah tujuan utama perusahaan untuk memperoleh keuntungan yang signifikan dalam persaingan pasar. Perusahaan distribusi perlu mengidentifikasi pelanggan terbaik dan meningkatkan pemahaman mereka tentang kebutuhan pelanggan agar dapat mempertahankan loyalitas pelanggan. Era yang serba digital ini, penjualan *online* telah mengubah cara berbelanja, hingga perusahaan *e-commerce* perlu memahami pelanggan mereka secara mendalam untuk menyediakan pengalaman yang relevan dan memuaskan. Selain itu, dalam persaingan bisnis yang semakin

ketat, perusahaan harus menjaga hubungan yang baik dengan pelanggan. Salah satu strategi yang efektif dalam menjaga hubungan tersebut adalah melalui segmentasi pelanggan, yang melibatkan pengelompokan pelanggan dengan kesamaan tertentu. Dalam beberapa tahun terakhir, ranah bisnis *online* telah mengalami peningkatan pesat dalam skala global, dan atmosfer pasar secara perlahan tapi pasti mencapai tingkat kematangan. Untuk meningkatkan kehadiran bisnis *online* dalam ranah pasar ini, perusahaan bisnis *online* harus menemukan cara untuk meningkatkan kemampuan mereka sendiri. Dalam ranah ini, revolusi model pemasaran telah berkembang menjadi fokus utama bagi perusahaan bisnis *online* (Deng, 2023). Kepuasan dan kepercayaan pelanggan menjadi salah satu hal yang harus dijaga untuk mempertahankan pelanggan yang loyal (Febrianti & Beni, 2023).

Segmentasi pelanggan adalah pendekatan yang membantu perusahaan membagi basis pelanggan mereka menjadi kelompok yang berbeda, berdasarkan karakteristik dan perilaku yang serupa. Tujuan utama dari segmentasi pelanggan adalah untuk memahami perilaku pelanggan, menganalisis preferensi mereka, dan mengidentifikasi kelompok pelanggan yang memiliki kebutuhan serupa. Segmentasi pelanggan merupakan langkah awal yang memungkinkan perusahaan mengembangkan strategi yang lebih nyaman dan personalisasi. Setelah perusahaan berhasil mengidentifikasi kelompok pelanggan yang berbeda, langkah berikutnya adalah mengembangkan rencana aksi yang sesuai untuk masing-masing kelompok tersebut. Misalnya, perusahaan dapat mengadopsi strategi pemasaran yang berbeda untuk setiap kelompok pelanggan. Dengan perkembangan teknologi dan ketersediaan data yang semakin besar, perusahaan juga dapat menggabungkan metode segmentasi pelanggan dengan pendekatan analitik yang lebih canggih, seperti analisis prediktif dan mesin pembelajaran. Dengan memanfaatkan teknik-teknik ini, perusahaan dapat menghasilkan wawasan yang lebih mendalam tentang perilaku pelanggan, memprediksi kebutuhan dan preferensi pelanggan di masa mendatang (Angelie, 2017).

Salah satu metode yang efektif dalam segmentasi pelanggan adalah *metode K-means clustering*. Metode ini telah banyak digunakan dalam berbagai penelitian dan aplikasi bisnis. Misalnya, dalam penelitian Zhang dkk, mereka mengaplikasikan metode *K-means clustering* untuk mengelompokkan pelanggan penjualan secara *online* berdasarkan atribut seperti perilaku pembelian, preferensi harga, dan demografi. Hasilnya menunjukkan bahwa segmentasi pelanggan dapat membantu perusahaan *e-commerce* dalam menyusun strategi pemasaran yang lebih efektif (Zhang et al., 2020). Algoritma *K-means* akan mengelompokkan satuan data dalam suatu kumpulan data ke dalam berbagai *cluster* berdasarkan jarak terdekat dengan nilai centroid awal yang dipilih secara acak, yang merupakan titik pusat awal (Bangoria et al., 2013). Semua data ini akan digunakan untuk menghitung jarak menggunakan rumus Euclidean Distance. Data yang jaraknya dekat dengan centroid akan membuat *cluster* dan proses ini terus berlanjut sampai tidak ada perubahan pada setiap kelompok (Gupta et al., 2013).

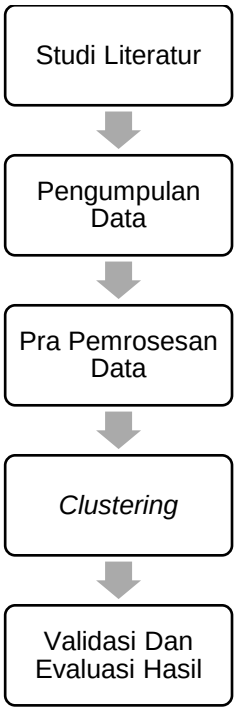
Algoritma *K-means clustering* bekerja dengan cara membagi data menjadi kelompok-kelompok yang berbeda berdasarkan kemiripan atribut yang dimiliki. Dalam konteks penjualan *online*, *K-means clustering* dapat digunakan untuk mengidentifikasi kelompok pelanggan dengan preferensi produk, perilaku pembelian, preferensi harga, dan faktor-faktor lain yang relevan. Metode ini memungkinkan perusahaan untuk memahami kelompok-kelompok pelanggan dengan lebih baik dan menyediakan layanan yang sesuai dengan kebutuhan dan preferensi masing-masing kelompok (Istiana, 2013). Penerapan algoritma pengelompokan *K-means* berbasis vektor fitur subjek, sementara *centre class* dihitung dengan rata-rata semua titik dalam *cluster* yang mudah dipengaruhi oleh *noise* dan fitur *noise* (Duo et al., 2021). Algoritma K-means, sebuah metode pengelompokan non-hierarkis, menunjukkan waktu komputasi yang relatif cepat. Analisis perbandingan yang dilakukan oleh Ghosh & Kumar (2013) antara K-Means dan Fuzzy C-Means (FCM) (Brahmana et al., 2020) menunjukkan bahwa algoritma yang pertama terbukti lebih cepat, dengan waktu yang telah berlalu selama 0,433755 detik, berbeda dengan algoritma yang terakhir, FCM, yang membutuhkan waktu 0,781679 detik untuk menyelesaikannya (Chandra et al., 2021). Meskipun metode *K-means clustering* memberikan manfaat yang signifikan dalam segmentasi pelanggan, implementasinya juga melibatkan beberapa tantangan. Salah satunya adalah

pemilihan jumlah kelompok yang optimal untuk membagi data pelanggan. Pemilihan jumlah kelompok yang tidak tepat dapat menghasilkan segmentasi yang tidak informatif atau tidak relevan. Selain itu, perusahaan juga perlu mempertimbangkan atribut yang digunakan dalam segmentasi, serta memastikan bahwa atribut tersebut dapat diandalkan dan mewakili perbedaan yang signifikan antar kelompok pelanggan.

Dengan memahami dan mengimplementasikan metode segmentasi pelanggan menggunakan *K-means clustering*, perusahaan dapat memperoleh wawasan yang lebih dalam tentang perilaku pelanggan mereka. Hal ini akan membantu perusahaan dapat menargetkan segmentasi *customer* secara lebih efisien. Segmentasi dapat dilakukan dengan menerapkan metode *K-means clustering* yang melibatkan beberapa kriteria yaitu total transaksi, frekuensi transaksi, dan total jumlah pembelian barang pada periode tertentu. Berdasarkan uraian-uraian di atas mengenai permalasahan pada penjualan *online*, penulis memilih judul "Segmentasi Pelanggan Penjualan Online Menggunakan Metode *K-means Clustering*".

2. METODE PENELITIAN

Dalam artikel ini, metode penelitian yang digunakan adalah metode deskriptif. Pendekatan deskriptif digunakan untuk menjelaskan dan menganalisis penggunaan metode *K-means clustering* (Nawangsih, 2023) dalam segmentasi pelanggan penjualan *online*. Gambar 1 adalah gambar alur metode penelitian yang dilakukan.



Gambar 1 Metode Penelitian

2.1 Studi Literatur

Tahapan awal yang dilakukan dalam penelitian ini yaitu tahapan studi literatur. Studi literatur dilakukan dengan melakukan kajian pustaka segmentasi pelanggan, metode *K-means clustering,* dan penerapannya dalam penjualan *online*. Studi literatur membantu membangun landasan teori serta memahami konsep dan langkah-langkah yang terlibat dalam metode pengelompokan *K-means*.

2.2 Pengumpulan Data

Data diperoleh dari platform penjualan *online*. Pada penelitian ini, *dataset* yang digunakan berupa data transaksi dari toko-toko *online*/retail di UK yang terdaftar di suatu perusahaan retail *online*, dengan rentang waktu periode 1 Desember 2010 sampai dengan tanggal 9 Desember 2011 sebanyak 541.909 pelanggan retail *online* (Chen, 2015). Data memiliki 8 atribut, yaitu: InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID dan Country.

2.3 Pra Pemrosesan Data

Tahap *pre-process* dilalui sebelum dilakukan proses *cluster*ing data untuk menyiapkan data pelanggan sebelum dianalisis menggunakan metode *K-means clustering*. Data pra-pemrosesan dapat mencakup pembersihan data, transformasi variabel, normalisasi data, atau pengurangan dimensi jika diperlukan. Tujuan dari *pre-process* data untuk mentransformasikan rentang tiap nilai variavel menjadi lebih kecil dan normalisasi *min-max* untuk mengubah rentang tiap nilai variabel menjadi 0 sampai dengan 1 (Kurniawati, 2018).

2.4 *Clustering*

Proses *cluster*ing dilakukan melalui tiga tahapan yaitu penentuan nilai k dengan menggunakan metode *Elbow* (Dewa & Jatipaningrum, 2019). Selanjutnya, untuk mendapatkan jumlah *cluster* yang optimal dilakukan perhitungan menggunakan metode SSE (*Sum of Square Error*) dari setiap nilai *cluster*. Semakin besar jumlah *cluster*, nilai SSE akan semakin kecil (Perdana et al., 2022).

Tahapan algoritma Elbow dalam penentuan nilai k pada K-Means *clustering* adalah:
1) Tentukan nilai awal k, misal k=1.
2) Tambahkan nilai k dengan 1.
3) Menghitung nilai SSE dari nilai k yang mengalami penurunan drastis, dengan rumus perhitungan seperti pada Pers. (1).

$$SSE = \sum_{i=1}^{n} (d)^2 \quad (1)$$

Setelah nilai k ditentukan, tahapan berikutnya adalah proses *K-means clustering*. Hasil pengelompokan yang dilakukan metode K-Means selanjutnya dilakukan pengujian kinerja *cluster* dengan metode perhitungan *silhouette index*, *Dunn Index*, lebar *silhouette*, dan konektivitas. Input dari proses ini adalah data pada variabel RFM (*recency*, *frequency*, dan *monetary value*). Analisis FM memudahkan pemilik bisnis untuk mengetahui peningkatan pendapatan dengan menargetkan kelompok tertentu dari pelanggan. Dengan analisis ini, maka segmentasi pelanggan akan lebih personal, menyesuaikan dengan perilaku pelanggan waktu sebelum transaksi. Tahap selanjutnya dilakukan analisis terhadap hasil *cluster* dengan melakukan denormalisasi data yaitu data yang telah dinormalisasi dan ditransformasikan kembali ke nilai aslinya. Tahapan ini dilakukan untuk memudahkan analisis *cluster* dengan membandingkan pada layanan variabel (Savitri et al., 2018).

2.5 Validasi dan Evaluasi Hasil

Validasi dan evaluasi hasil segmentasi pelanggan bertujuan untuk memastikan penjelasan dan interpretasi yang tepat. Pada penelitian ini, menggunakan metrik evaluasi pengelompokan *silhouette index* untuk evaluasi kualitas segmentasi.

3. HASIL DAN PEMBAHASAN

3.1 Analisis RFM

Analisis RFM merupakan salah satu langkah yang dapat digunakan untuk segmentasi pelanggan penjualan *online*. Dalam analisis ini, faktor kunci yang diperhatikan adalah seberapa baru

pelanggan melakukan pembelian (*recency*), seberapa sering melakukan berbelanja (*frequency*), dan berapa total nilai pembelian yang dilakukan (*monetary*) (Brahmana et al., 2020).

Tabel 1 *Monetary*

	CustomerID	Monetary
0	12346.0	0.00
1	12347.0	4310.00
2	12348.0	1797.24
3	12349.0	1757.55
4	12350.0	334.40

Tabel 2 *Frequency*

	CustomerID	Frequency
0	12346.0	2
1	12347.0	182
2	12348.0	31
3	12349.0	73
4	12350.0	17

Tabel 3 *Recency*

	CustomerID	Diff
0	12346.0	325 days 02:33:00
1	12347.0	1 days 20:58:00
2	12348.0	74 days 23:37:00
3	12349.0	18 days 02:59:00
4	12350.0	309 days 20:49:00

Data pada Tabel 1 berisi tentang pelanggan dengan total pembelian yang telah dilakukan. Selanjutnya, pada Tabel 2 menunjukan besar frekuensi pembelian atau seberapa sering pembelian yang telah dilakukan. Sedangkan pada Tabel 3 berisi tentang seberapa baru pembelian yang telah dilakukan. Ketiga nilai tersebut diperlukan untuk menentukan RFM.

3.2 Pra Pemrosesan Data

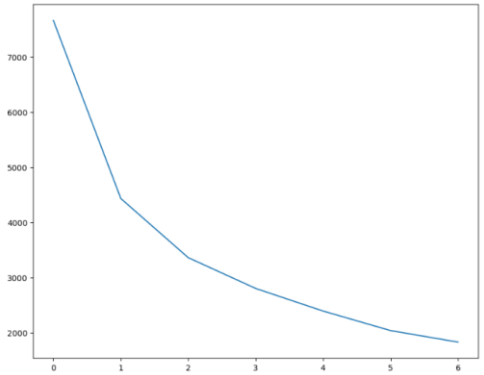
Terdapat 3 tahapan proses untuk pra proses data yaitu transformasi data*, data cleansing*, dan normalisasi menggunakan metode *min-max*. Hasil dari pra proses, yaitu gabungan jumlah total pembelian, frekuensi pembelian, dan keterbaruan pembelian oleh pelanggan ditunjukan pada Tabel 4.

Tabel 4 Hasil Pra Proses *Recency*, *Monetary*, dan *Frequency*

	Amount	Frequency	Recency
0	-0.723738	-0.752888	2.301611
1	1.731617	1.042467	-0.906466
2	0.300128	-0.463636	-0.183658
3	0.277517	-0.044720	-0.738141
4	-0.533235	-0.603275	2.143188

3.3 Penentuan Jumlah *Cluster*

Dalam menentukan jumlah *cluster* yang optimal diperlukan metode tertentu. Metode *Elbow* merupakan salah satu metode yang bisa menentukan jumlah *cluster* yang optimal. Namun nilai k yang diperoleh dari metode *Elbow Curve* tidak pasti menghasilkan jumlah *cluster* (k) yang optimal. Maka diperlukan juga *Silhouette Analysis* guna mencari nilai k optimal. Berikut adalah *Elbow Curve* yang diilustrasikan pada Gambar 2.



Gambar 2 *Elbow Curve*

Setelah tahap analisis *elbow*, langkah selanjutnya menggunakan *silhouette analysis* untuk mendapatkan nilai k yang optimal. Algoritma *silhouette analysis* digunakan untuk mengukur seberapa dekat setiap titik pusat pada sebuah *cluster* dengan titik pusat data lain di *cluster*. Bila semakin tinggi nilai rata-rata dari *silhouette analysis*, maka nilai *cluster* semakin optimal.

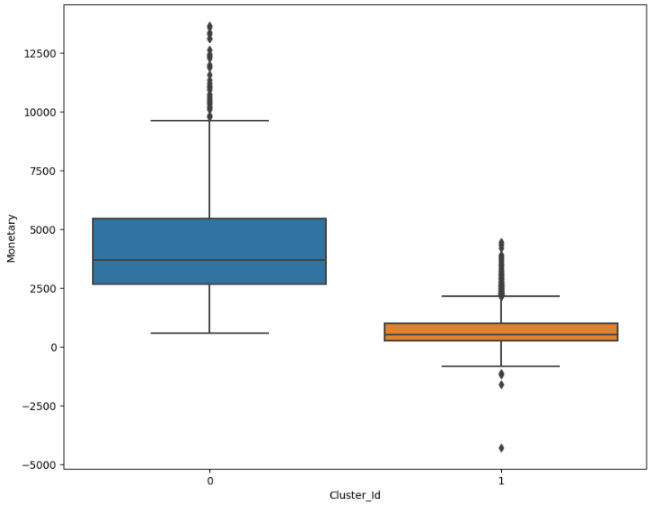
Nilai *silhouette* berada pada rentang nilai -1 sampai dengan 1. Jika nilainya mendekati angka 1, maka titik pusat data akan sangat mirip dengan titik pusat data lainnya di *cluster* yang sama. Jika mendekati -1 maka titik pusat data tersebut menjadi tidak mirip dengan titik pusat data di klusternya. Dari *silhouette analysis* diperoleh hasil jumlah *cluster* yang optimal adalah 2 *cluster* karena memiliki nilai yang paling tinggi. Gambar 3 merupakan hasil dari *silhouette analysis*.

```
For n_clusters=2, the silhouette score is 0.5415858652525395
For n_clusters=3, the silhouette score is 0.5084896296141937
For n_clusters=4, the silhouette score is 0.4809515569009417
For n_clusters=5, the silhouette score is 0.4642236188382193
For n_clusters=6, the silhouette score is 0.41701094135102007
For n_clusters=7, the silhouette score is 0.4173587427187674
For n_clusters=8, the silhouette score is 0.40679173812416936
```

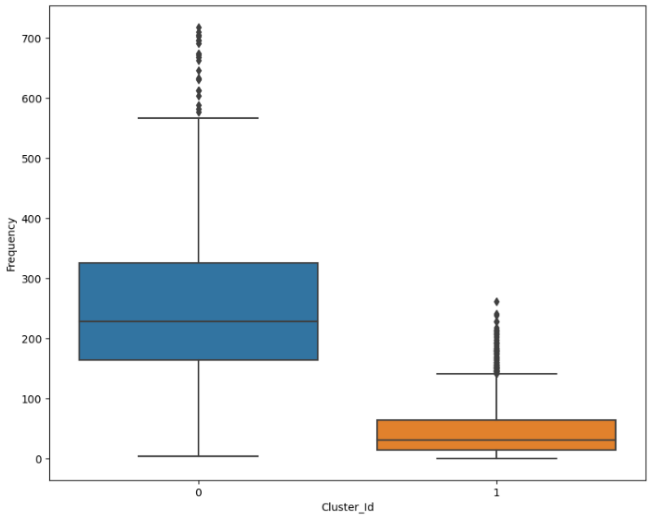
Gambar 3 Hasil *Silhouette Analysis* dengan Dua *Cluster* yang Tertinggi

3.4 *K-means Clustering*

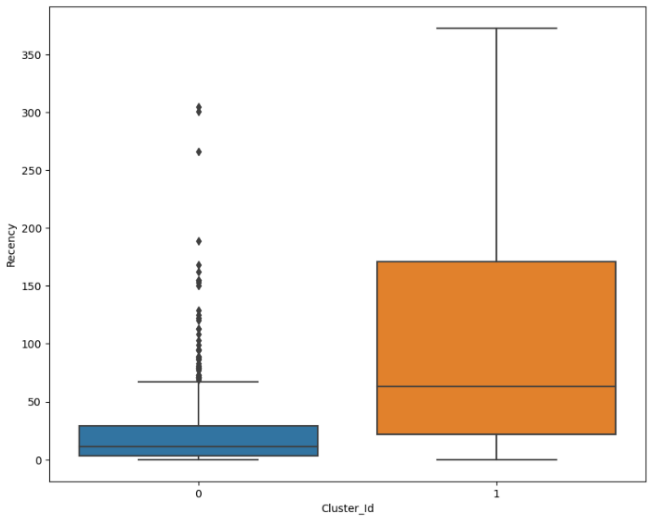
Setelah dilakukan *K-means clustering* maka mendapatkan hasil berupa informasi yang menunjukkan kelompok dari anggota setiap *cluster*, *center point* atau centroid, dan nilai *performance* dari *cluster*. Pada tahap ini menggunakan boxplot untuk memvisualisasikan *clustering*. Terdapat beberapa perbandingan untuk *clustering* yaitu *cluster_id* dan *monetary* yang terlihat pada Gambar 4, *cluster_id* dan *frequency* yang terlihat pada Gambar 5, dan *cluster_id* dan *recency* yang terlihat pada Gambar 6.



Gambar 4 *Cluster_Id* dan *Monetary*



Gambar 5 *Cluster_Id* dan *Frequency*



Gambar 6 *Cluster_Id* dan *Recency*

Dengan demikian bisa dideskripsikan bahwa:

1) Pelanggan dengan jumlah total transaksi tinggi dibandingkan dengan pelanggan lain terdapat *Cluster* 0.
2) Pelanggan yang paling sering melakukan transaksi dibandingkan dengan pelanggan di *cluster* lain terdapat pada *Cluster* 1.

Dengan memanfaatkan metode *K-means clustering* dalam segmentasi pelanggan penjualan *online*, perusahaan dapat memperoleh pemahaman yang lebih baik tentang perilaku pelanggan mereka. Hal ini dapat mendukung pengembangan strategi pemasaran yang lebih efisien, peningkatan personalisasi produk, serta peningkatan kepuasan pelanggan secara keseluruhan.

4. KESIMPULAN

Berdasarkan dari hasil penelitian, pelanggan dengan *cluster* 0 adalah pelanggan dengan jumlah total transaksi yang paling tinggi jika dibandingkan dengan pelanggan di *cluster* yang lain. Sedangkan pelanggan dengan *cluster* 1 adalah pelanggan yang paling sering melakukan transaksi jika dibandingkan dengan pelanggan di *cluster* yang lain. Dengan menggunakan metode *K-means clustering* dalam segmentasi pelanggan penjualan *online*, perusahaan dapat memahami perilaku dari masing-masing pelanggan mereka. Hal ini dapat mendukung perusahaan untuk mengembangkan strategi pemasaran mereka agar menjadi lebih efisien. Saran untuk penelitian selanjutnya adalah segmentasi penjualan *online* menggunakan lebih dari satu *dataset* agar memiliki perbandingan akurasi melalui pemilihan algoritma tertentu.

DAFTAR PUSTAKA

Angelie, A. V. (2017). *Segmentasi Pelanggan Menggunakan Clustering K-Means dan Model RFM (Studi Kasus: PT. Bina Adidaya Surabaya)* [Institut Teknologi Sepuluh Nopember]. https://repository.its.ac.id/42240/

Chandra, M. D., Irawan, E., Saragih, I. S., Windarto, A. P., & Suhendro, D. (2021). Penerapan Algoritma K-Means dalam Mengelompokkan Balita yang Mengalami Gizi Buruk Menurut Provinsi. *BIOS : Jurnal Teknologi Informasi Dan Rekayasa Komputer*, *2*(1), 30–38. https://doi.org/10.37148/bios.v2i1.19

Chen, D. (2015). *Online Retail*. UCI Machine Learning Repository.

Deng, Y. (2023). Specific Strategies for Innovating Marketing Models of E-commerce Enterprises in the Internet Era. *Academic Journal of Business & Management*, *5*(13), 22–26. https://doi.org/10.25236/AJBM.2023.051304

Dewa, F. A., & Jatipaningrum, M. T. (2019). Segmentasi E-Commerce dengan Cluster K-Means dan Fuzzy C-Means. *Jurnal Statistika Industri Dan Komputasi*, *4*(01), 53–67. https://doi.org/10.34151/STATISTIKA.V4I01.1054

Duo, J., Zhang, P., & Hao, L. (2021). A K-means Text Clustering Algorithm Based on Subject Feature Vector. *Journal of Web Engineering*, *20*(6), 1935-1946–1935–1946. https://doi.org/10.13052/jwe1540-9589.20612

Febrianti, F., & Beni, S. (2023). Strategi Mempertahankan Loyalitas Pelanggan pada Usaha Kuliner di Kecamatan Bengkayang. *Inovasi Pembangunan : Jurnal Kelitbangan*, *11*(02), 189–210. https://doi.org/10.35450/jip.v11i02.384

Ghosh, S., & Kumar, S. (2013). Comparative Analysis of K-Means and Fuzzy C-Means Algorithms. *International Journal of Advanced Computer Science and Applications*, *4*(4). https://doi.org/10.14569/IJACSA.2013.040406

Gupta, G. K., Agrawal, D., Singh, R. K., & Arya, R. K. (2013). Prevalence, Risk Factors and Socio Demographic Co-Relates of Adolescent Hypertension in District Ghaziabad. *Indian Journal of Community Health*, *25*(3), 293–298. https://www.iapsmupuk.org/journal/index.php/IJCH/article/view/331

Istiana, M. I. (2013). *Segmentasi Pelanggan Menggunakan Algoritma K-Means Sebagai Dasar Strategi Pemasaran pada LAROIBA Seluler Oleh: Mike Indra Istiana* [Universitas Dian Nuswantoro Semarang]. http://eprints.dinus.ac.id/12733/2/abstrak_12903.pdf

Kurniawati, I. Y. (2018). *Segmentasi Pelanggan Menggunakan Clustering K-Means* [Universitas 17 Agustus 1945]. http://repository.untag-sby.ac.id/868/

Bangoria, B., Mankad, N., & Pambhar, V. (2013). A survey on Efficient Enhanced K-Means Clustering Algorithm. *International Journal for Scientific Research and Development*, *1*(9), 1756–1758. https://doi.org/10.2/JQUERY.MIN.JS

Nawangsih, I. (2023). Analisa Penjualan Produk Kosmetik Dengan Metode Algoritma K-Means Di Toko Erremy. *Bulletin of Information Technology (BIT)*, *4*(1), 140–145. https://doi.org/10.47065/bit.v4i1.468

Perdana, S. A., Florentin, S. F., & Santoso, A. (2022). Analisis Segmentasi Pelanggan Menggunakan K-Means Clustering Studi Kasus Aplikasi Alfagift. *Sebatik*, *26*(2), 446–457. https://doi.org/10.46984/sebatik.v26i2.1991

Savitri, A. D., Bachtiar, F. A., & Setyawan, N. Y. (2018). Segmentasi Pelanggan Menggunakan Metode K-Means Clustering Berdasarkan Model RFM Pada Klinik Kecantikan (Studi Kasus: Belle Crown Malang). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(9), 2957–2966. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/2489

Brahmana, R. W. S., Mohammed, F. A., & Chairuang, K. (2020). Customer Segmentation Based on RFM Model Using K-Means, K-Medoids, and DBSCAN Methods. *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, *11*(1), 32. https://doi.org/10.24843/LKJITI.2020.v11.i01.p04

Zhang, Z., Ni, G., & Xu, Y. (2020). Comparison of Trajectory Clustering Methods based on K-means and DBSCAN. *Proceedings of 2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence, ICIBA 2020*, 557–561. https://doi.org/10.1109/ICIBA50161.2020.9277214

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 1, JANUARI, 2024, Pp. 10 – 26
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Improving Stock Price Prediction Accuracy with StacBi LSTM

Mohammad Diqi [(1)*], Hamzah [(2)]
Informatika, Fakultas Sains dan Teknologi, Universitas Respati Yogyakarta, Yogyakarta
e-mail : {diqi,hamzah}@respati.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 10 Agustus 2023, direvisi 31 Oktober 2023, diterima 1 November 2023, dan dipublikasikan 25 Januari 2024.

Abstract

This research aimed to enhance stock price prediction accuracy using the Stacked Bidirectional Long Short-Term Memory (StacBi LSTM) model. The study addressed the challenge of capturing long-term dependencies and temporal patterns inherent in stock price data. The research objectives were to evaluate the model's performance across different input sequence lengths and identify the optimal length for prediction. Leveraging a dataset from the Indonesian Stock Exchange, the model's predictions were evaluated using key metrics such as RMSE, MAE, MAPE, and R2. Results indicated that the StacBi LSTM model excelled in capturing stock price trends and demonstrated strengths over traditional methods. The optimal input sequence length was identified, balancing computational efficiency and prediction accuracy. This research contributes valuable insights into improving stock price prediction techniques and offers practical implications for traders and investors. Future research directions encompass hybrid models and integrating external factors to enhance predictive capabilities further.

Keywords: Stock Price Prediction, Stacked Bidirectional LSTM, Time Series Analysis, Indonesian Stock Exchange, Input Sequence Length

Abstrak

Penelitian ini bertujuan untuk meningkatkan akurasi prediksi harga saham menggunakan model *Stacked Bidirectional Long Short-Term Memory* (StacBi LSTM). Penelitian ini mengatasi tantangan dalam menangkap ketergantungan jangka panjang dan pola temporal yang inheren dalam data harga saham. Tujuan penelitian adalah mengevaluasi kinerja model dalam berbagai panjang urutan masukan dan mengidentifikasi panjang masukan yang optimal untuk prediksi. Dengan menggunakan *dataset* dari Bursa Efek Indonesia, prediksi model dievaluasi menggunakan metrik kunci seperti RMSE, MAE, MAPE, dan R2. Hasil penelitian menunjukkan bahwa model StacBi LSTM mampu dengan baik dalam menangkap tren harga saham dan memiliki keunggulan dibandingkan metode tradisional. Panjang masukan optimal diidentifikasi, menciptakan keseimbangan antara efisiensi komputasi dan akurasi prediksi. Penelitian ini memberikan wawasan berharga dalam meningkatkan teknik prediksi harga saham dan memberikan implikasi praktis bagi para trader dan investor. Arah penelitian di masa depan meliputi model hibrida dan integrasi faktor eksternal untuk meningkatkan kemampuan prediksi lebih lanjut.

Kata Kunci: Prediksi harga saham, Stacked Bidirectional LSTM, Analisis runtun waktu, Bursa Efek Indonesia, Panjang urutan masukan

1. INTRODUCTION

Stock price prediction is a crucial domain within financial research, holding substantial implications for global financial markets in an increasingly interconnected and dynamic economy (Miftahurrohmah et al., 2021). Accurate predictions of stock prices are imperative for enabling informed decision-making among investors, traders, and financial institutions, leading to enhanced portfolio management, effective risk mitigation, and optimal resource allocation (Aydin et al., 2022). Furthermore, such predictions are instrumental in uncovering potential profit avenues and formulating efficacious trading strategies. In this regard, advanced machine learning and deep learning techniques have been transformative, providing novel means to enhance

prediction precision and unravel complex patterns in historical stock data, thereby empowering market participants to make more lucrative investment decisions (Rajamoorthy et al., 2022).

In evaluating predictive methodologies, fundamental analysis emerges as a prominent approach, anchored in examining a company's financial and economic data to gauge its intrinsic value and determine the stock's market standing (Williams et al., 2020). Despite its merits, this method is not without its challenges; it is inherently time-intensive, subjective in nature, and often neglects short-term market sentiments, rendering it less effective for tracking swift market dynamics (Naumoski et al., 2022; Wang et al., 2021). Conversely, technical indicators offer real-time analyses rooted in historical price and volume data, shedding light on market trends and potential trading positions (Jamous et al., 2021). However, they have limitations, including a tendency to rely heavily on past data, produce delayed signals, and generate false signals in volatile markets. These challenges necessitate expertise in selecting and adapting indicators and parameters to individual stocks and resolving discrepancies when multiple indicators are employed simultaneously (Htun et al., 2023).

From a modeling perspective, AutoRegressive Integrated Moving Average (ARIMA) models and their seasonal variant, SARIMA, are widely recognized for their efficacy in time series forecasting, especially for data with linear dependencies and periodic fluctuations respectively (Brahma & Wadhvani, 2020; Jiang et al., 2019). Nevertheless, they exhibit limitations in addressing non-stationary data and capturing non-linear patterns, with performance challenges arising when dealing with long-term and noisy data (Musarat et al., 2021; Shuai et al., 2021). Support Vector Machine (SVM) and Decision Tree algorithms. However, capturing complex relationships and providing intuitive insights also present challenges regarding dataset balance, feature scaling, and model stability (Jamous et al., 2021; Sekiguchi et al., 2019).

Ensemble methods like Random Forest and XGBoost attempt to mitigate these issues by aggregating multiple models, enhancing predictive performance and robustness to overfitting (Campbell et al., 2020; Pamir et al., 2022). Nevertheless, they still grapple with challenges related to computational efficiency, hyperparameter tuning, and interpretability (Kim et al., 2021; Lind & Anderson, 2019). On the other hand, K-Nearest Neighbors (KNN) stands out for its simplicity and robustness, although it requires careful parameter selection and is computationally demanding for large datasets (Lokanan, 2022; Ma et al., 2020).

Delving into deep learning, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) represent robust architectures for handling sequential data, each with its own set of challenges and areas of applicability (Gao et al., 2018; Succetti et al., 2022; Tang & Mahmoud, 2022). LSTM and GRU, as variants of RNNs, offer solutions to the vanishing gradient problem, facilitating the capture of long-term dependencies, albeit with considerations for computational cost and overfitting (Ahmad et al., 2022; Choi & Shin, 2019; Suleman & Shridevi, 2022). Autoencoders and Generative Adversarial Networks (GANs) further extend the deep learning repertoire, providing capabilities in feature extraction, dimensionality reduction, and data augmentation, but necessitate careful model design and training (Erizal & Diqi, 2023; Fathy et al., 2021; Mo et al., 2022; Pan et al., 2020; Zhang et al., 2022).

The research presented here revolves around the StacBi LSTM model, chosen for its proficiency in capturing long-term dependencies and temporal patterns in stock prices, integrating bidirectional processing and multiple LSTM layers to discern complex relationships in time series data (Suleman & Shridevi, 2022). The investigation is structured to predict stock prices using this model with varied input sequences, aiming to discern the impact of sequence length on predictive performance and employing a comprehensive suite of evaluation metrics to benchmark against conventional forecasting methods. This study contributes to the domain of stock price prediction by elucidating the potentials of the StacBi LSTM model, exploring the influence of input sequence variations, and advancing the evaluation methodology within this context.

2. METHODS

2.1 Long Short-Term Memory (LSTM)

LSTM is an RNN that addresses the vanishing gradient problem, allowing it to capture long-term dependencies in sequential data (Qaddoura et al., 2021). It achieves this by introducing a memory cell and three gating mechanisms: the input gate, forget gate, and output gate. Table 1 summarizes the mathematical notation used in the context of LSTM.

Table 1 Mathematical Notation for LSTM

Symbol	Description
x_t	The input at time step t.
h_t	The hidden state at time step t.
c_t	The cell state (memory) at time step t.
i_t	The input gate at time step t.
f_t	The forget gate at time step t.
o_t	The output gate at time step t.
σ	The sigmoid activation functions.
$tanh$	The hyperbolic tangent activation function.

The LSTM computation consists of four main steps for each time step t:

Input Gate. The input gate determines how much of the new input information is stored in the cell state, as calculated in Equation 1.

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \quad (1)$$

Forget Gate. The forget gate determines how much of the previous cell state to retain for the current time step, as calculated in Equation 2.

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \quad (2)$$

Cell State. The cell state is updated by combining the previous cell state with the new input information using the input and forget gates, as calculated in Equation 3.

$$\begin{gathered}\tilde{c}_t = \tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \\ c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t\end{gathered} \quad (3)$$

Output Gate. The output gate determines how much of the updated cell state to output as the hidden state for the current time step, as calculated in Equation 4.

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \quad (4)$$

Hidden State. The hidden state is obtained by applying the output gate to the updated cell state, as calculated in Equation 5.

$$h_t = o_t \cdot \tanh(c_t) \quad (5)$$

Here, W_{ix}, W_{ih}, W_{fx}, W_{fh}, W_{cx}, W_{ch}, W_{ox}, W_{oh} are weight matrices and b_i, b_f, b_c, b_o are bias vectors.

The LSTM memory cell's ability to control the flow of information through the input, forget, and output gates enables it to retain essential information over long sequences, making it effective in capturing long-term dependencies in time series data. The vanishing gradient problem is

mitigated by the constant error flow through the cell state, allowing for more stable and efficient training.

2.2 Stacked Bidirectional (StacBi) LSTM

The StacBi LSTM consists of N LSTM layers, $N/2$ in the forward direction and $N/2$ in the backward direction. Table 2 summarizes the mathematical notation used in the StacBi LSTM model, distinguishing between the forward and backward LSTM layers and their respective hidden states, cell states, and gates.

Table 2 Mathematical Notation for StacBi LSTM

Symbol	Description
x_t	The input at time step t.
$h_t^{(n,f)}$	The hidden state of the n-th forward LSTM layer at time step t.
$c_t^{(n,f)}$	The cell state (memory) of the n -th forward LSTM layer at time step t.
$h_t^{(n,b)}$	The hidden state of the n-th backward LSTM layer at time step t.
$c_t^{(n,b)}$	The cell state (memory) of the n-th backward LSTM layer at time step t.
$i_t^{(n,f)}$	The input gate of the n-th forward LSTM layer at time step t.
$f_t^{(n,f)}$	The forget gate of the n-th forward LSTM layer at time step t.
$o_t^{(n,f)}$	The output gate of the n-th forward LSTM layer at time step t.
$i_t^{(n,b)}$	The input gate of the n-th backward LSTM layer at time step t.
$f_t^{(n,b)}$	The forget gate of the n-th backward LSTM layer at time step t.
$o_t^{(n,b)}$	The output gate of the n-th backward LSTM layer at time step t.
$\tilde{c}_t^{(n,f)}$	The candidate cell state of the n-th forward LSTM layer at time step t.
$\tilde{c}_t^{(n,b)}$	The candidate cell state of the n-th backward LSTM layer at time step t.

Forward LSTM. The forward LSTM computes the hidden state $h_t^{(n,f)}$ and the cell state $c_t^{(n,f)}$ for the n-th forward layer at time step t, as shown in Equation 6.

$$\begin{aligned} i_t^{(n,f)} &= \sigma\left(W_{ix}^{(n,f)} x_t + W_{ih}^{(n,f)} h_{t-1}^{(n,f)} + b_i^{(n,f)}\right) \\ f_t^{(n,f)} &= \sigma\left(W_{fx}^{(n,f)} x_t + W_{fh}^{(n,f)} h_{t-1}^{(n,f)} + b_f^{(n,f)}\right) \\ \tilde{c}_t^{(n,f)} &= \tanh\left(W_{cx}^{(n,f)} x_t + W_{ch}^{(n,f)} h_{t-1}^{(n,f)} + b_c^{(n,f)}\right) \\ c_t^{(n,f)} &= f_t^{(n,f)} \cdot c_{t-1}^{(n,f)} + i_t^{(n,f)} \cdot \tilde{c}_t^{(n,f)} \\ o_t^{(n,f)} &= \sigma\left(W_{ox}^{(n,f)} x_t + W_{oh}^{(n,f)} h_{t-1}^{(n,f)} + b_o^{(n,f)}\right) \\ h_t^{(n,f)} &= o_t^{(n,f)} \cdot \tanh\left(c_t^{(n,f)}\right) \end{aligned} \tag{6}$$

Backward LSTM. The backward LSTM computes the hidden state $h_t^{(n,b)}$ and the cell state $c_t^{(n,b)}$ for the n-th backward layer at time step t, as shown in Equation 7.

$$\begin{aligned} i_t^{(n,b)} &= \sigma\left(W_{ix}^{(n,b)} x_t + W_{ih}^{(n,b)} h_{t+1}^{(n,b)} + b_i^{(n,b)}\right) \\ f_t^{(n,b)} &= \sigma\left(W_{fx}^{(n,b)} x_t + W_{fh}^{(n,b)} h_{t+1}^{(n,b)} + b_f^{(n,b)}\right) \\ \tilde{c}_t^{(n,b)} &= \tanh\left(W_{cx}^{(n,b)} x_t + W_{ch}^{(n,b)} h_{t+1}^{(n,b)} + b_c^{(n,b)}\right) \\ c_t^{(n,b)} &= f_t^{(n,b)} \cdot c_{t+1}^{(n,b)} + i_t^{(n,b)} \cdot \tilde{c}_t^{(n,b)} \\ o_t^{(n,b)} &= \sigma\left(W_{ox}^{(n,b)} x_t + W_{oh}^{(n,b)} h_{t+1}^{(n,b)} + b_o^{(n,b)}\right) \\ h_t^{(n,b)} &= o_t^{(n,b)} \cdot \tanh\left(c_t^{(n,b)}\right) \end{aligned} \tag{7}$$

StacBi LSTM. The StacBi LSTM is formed by stacking N LSTM layers, where each forward layer's output serves as the input to the next forward layer, and each backward layer's output is the input to the next backward layer, as shown in Equations 8-9.

Forward Pass. For the n-th forward layer:

$$x_t^{(n,f)} = h_t^{(n-1,f)}$$
$$h_t^{(n,f)} = \text{Forward LSTM computation using } x_t^{(n,f)} \text{ as input} \quad (8)$$

Backward Pass. For the n-th backward layer:

$$x_t^{(n,b)} = h_t^{(n+1,b)}$$
$$h_t^{(n,b)} = \text{Backward LSTM computation using } x_t^{(n,b)} \text{ as input} \quad (9)$$

The final hidden state h_t of the StacBi LSTM is obtained by concatenating the outputs from the last forward layer $h_t^{(N/2,f)}$ and the first backward layer $h_t^{(1,b)}$, as shown in Equation 10.

$$h_t = \left[h_t^{(N/2,f)}; h_t^{(1,b)}\right] \quad (10)$$

In this way, the StacBi LSTM captures past and future context, enabling it to model long-term dependencies and complex temporal patterns more effectively than a single LSTM layer.

2.3 Dataset

The dataset utilized in this research is sourced from Yahoo Finance (Erizal & Diqi, 2023). It encompasses the top 10 stocks listed in the Indonesia Stock Exchange within the period spanning from July 6, 2015, to October 14, 2021. The stocks and corresponding symbols and sectors are presented in Table 3.

Table 3 List of the Observed Stocks

Symbol	Company	Sector
ACES.JK	Ace Hardware Indonesia Tbk.	Consumer Non-Cyclicals
ADRO.JK	Adaro Energy Tbk.	Energy
EXCL.JK	XL Axiata Tbk.	Infrastructures
KLBF.JK	Kalbe Farma Tbk.	Healthcare
PGAS.JK	Perusahaan Gas Negara (Persero) Tbk.	Energy
PTBA.JK	Tambang Batubara Bukit Asam (Persero) Tbk.	Energy
PTPP.JK	PP (Persero) Tbk.	Infrastructures
PWON.JK	Pakuwon Jati Tbk.	Properties & Real Estate
SMRA.JK	Summarecon Agung Tbk.	Properties & Real Estate
TPIA.JK	Chandra Asri Petrochemical Tbk.	Basic Materials

The dataset includes essential features such as Date, Open, High, Low, Close, and Volume, with records collected daily. Notably, this study exclusively focuses on the Close Price variable, as depicted in Figures 1 to 10.

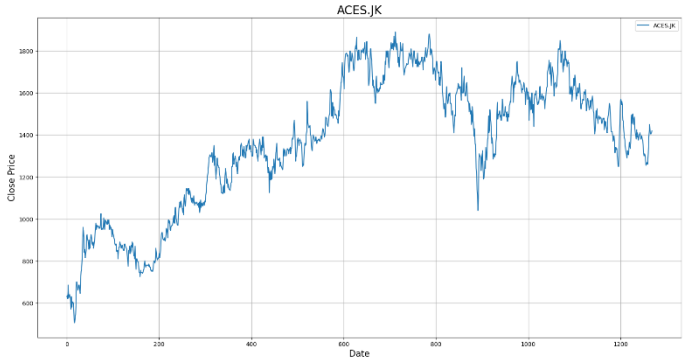


Figure 1 ACES.JK

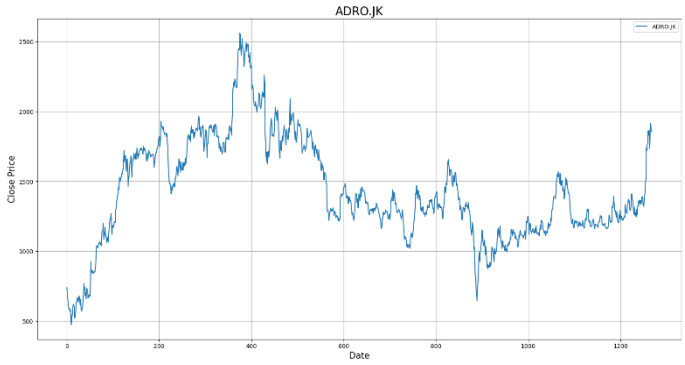


Figure 2 ADRO.JK

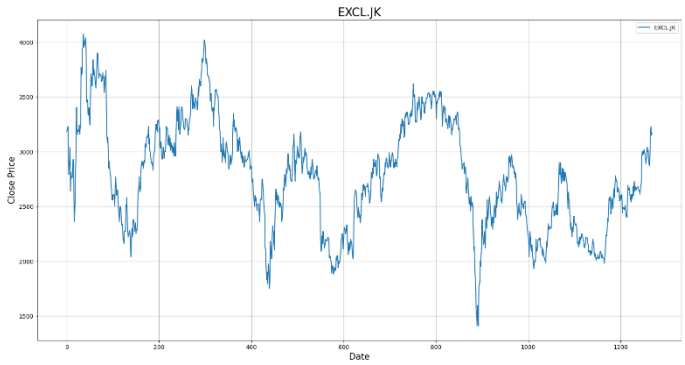


Figure 3 EXCL.JK

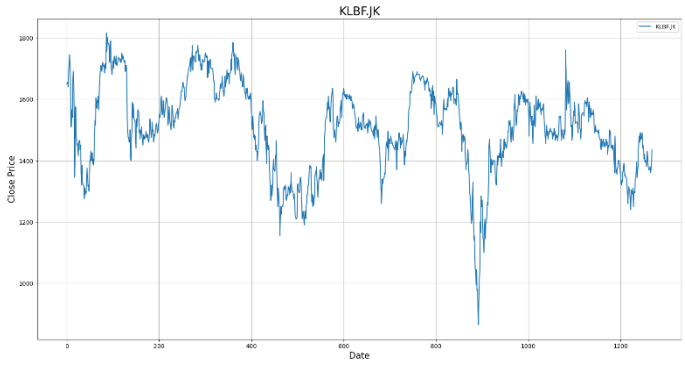


Figure 4 KLBF.JK

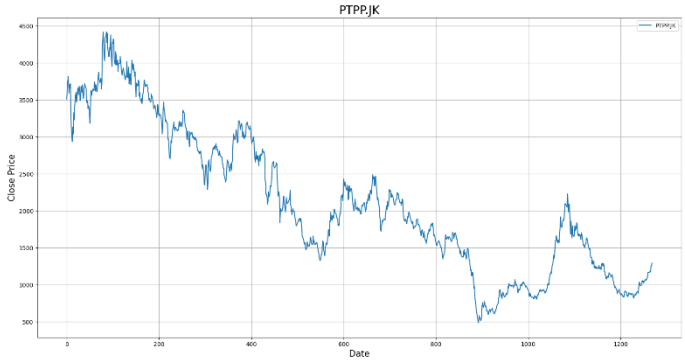


Figure 5 PGAS.JK

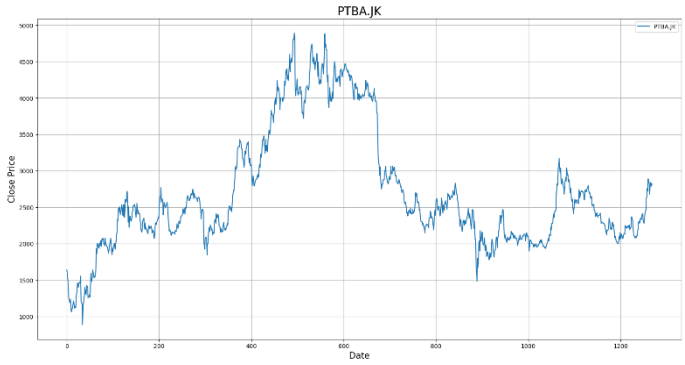


Figure 6 PTBA.JK

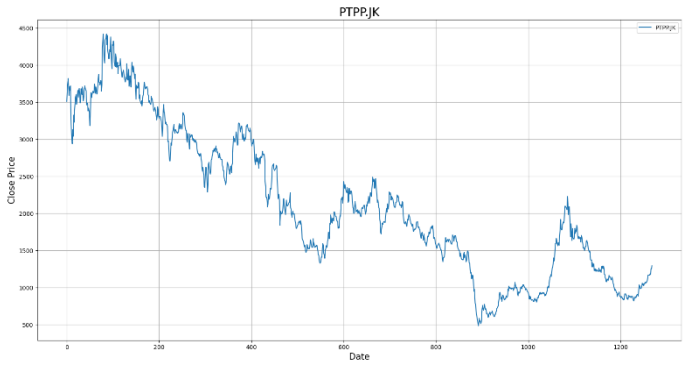


Figure 7 PTPP.JK

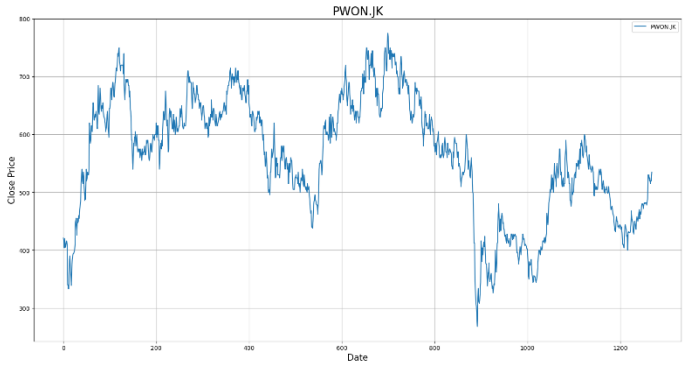


Figure 8 PWON.JK

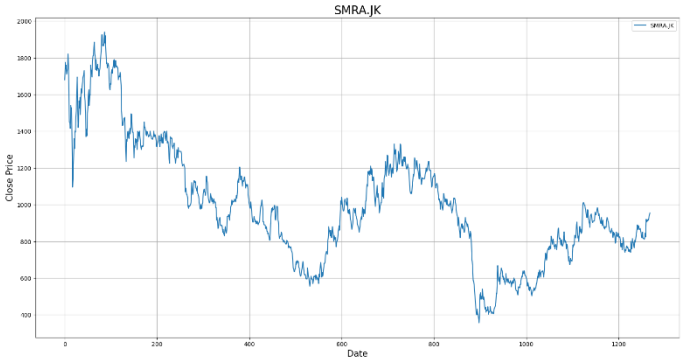


Figure 9 SMRA.JK

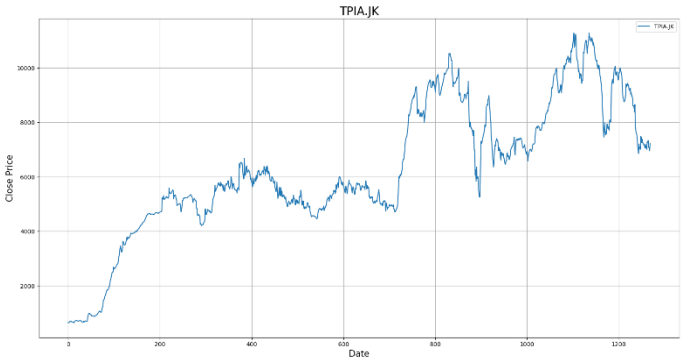


Figure 10 TPIA.JK

2.4 Data Processing

In the preparatory phase of data preprocessing, meticulous measures were implemented to refine the dataset for predicting stock prices, ensuring its quality and relevance. Instances with a trading volume of zero were deliberately removed to prevent potential distortions in the model's learning trajectory. The min-max scaler technique was applied in a normalization procedure to foster model convergence and maintain numerical stability. This method recalibrates the range of feature values to a standardized scale between 0 and 1, accommodating various data magnitudes and contributing to a more uniform and manageable dataset for the model. The mathematical formulation of the min-max scaler is provided in Equation 11, detailing the procedure of this normalization process.

$$X_{\text{normalized}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{11}$$

Where X is the original feature value, X_{min} and X_{max} are the minimum and maximum values of the feature, respectively. This preprocessing strategy ensures that the dataset lacks missing values and zero volume instances and is normalized for effective utilization in the subsequent analysis.

2.5 Data Splitting

The dataset, consisting of 1269 data points, was split into distinct subsets to facilitate practical model training, validation, and testing. Specifically, 40 data points were reserved for testing the model's generalization performance on unseen data. Most of the data, totaling 983 instances, was allocated for training the model to learn patterns and relationships within the dataset. An additional subset comprising 246 data points was designated for validation, enabling the fine-tuning of model hyperparameters and preventing overfitting. This data-splitting strategy ensured

that the model's performance was rigorously assessed across different training and testing phases, enhancing its predictive capabilities for stock price forecasting.

2.6 Model Training Process

The model training process involves a StacBi LSTM architecture for stock price prediction. The model is structured as follows:

1) **Input Configuration**: The input features are defined by $n\_features = 1$, indicating that the model utilizes one feature (Close Price) for prediction. Three different input sequence lengths, $n\_steps = [30,50,70]$, are considered to capture varying historical information.
2) **Model Construction**: A Sequential model is established. The first layer is a Bidirectional LSTM with 256 units, employing the 'relu' activation function and 'return_sequences=True' to pass sequences to the subsequent layer. A dropout layer (dropout rate = 0.2) follows, aiding in regularization.
3) **Second Bidirectional LSTM**: Another Bidirectional LSTM with 128 units and "relu' activation is added, capturing complex temporal patterns. Another dropout layer (dropout rate = 0.2) helps prevent overfitting.
4) **Output Layer**: A Dense layer with 1 unit is employed for the final prediction.
5) **Compilation**: The model is compiled with the Adam optimizer (learning rate = 0.001) and mean squared error (MSE) loss function.
6) **Training**: The model is trained using the provided training data (X and y) for 100 epochs, with a batch size 32. Training progress is run in a quiet mode (verbose=0).

This training process enables the model to learn and capture intricate patterns in the input data, ultimately enhancing its ability to forecast stock prices accurately.

2.7 Evaluation Metrics

In this research, several evaluation metrics were employed to assess the performance of the StacBi LSTM model in predicting stock prices. The following metrics were utilized and formulated in Equations 12-15.

1) Root Mean Squared Error (RMSE) (Gutmann et al., 2021):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{12}$$

2) Mean Absolute Error (MAE) (Gutmann et al., 2021):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{13}$$

3) Mean Absolute Percentage Error (MAPE) (Patel et al., 2022):

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{14}$$

4) Coefficient of Determination (R2) (Baek & Chung, 2023):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{15}$$

Here, y_i represents the actual stock price, $\hat{y}_i$ represents the predicted stock price, $\bar{y}$ is the mean of the actual stock prices, and n is the number of data points. These evaluation metrics collectively quantify the model's accuracy, precision, and goodness of fit in predicting stock prices.

3. RESULTS AND DISCUSSION

The StacBi LSTM model was employed to forecast stock prices for 40 days. The model's predictive accuracy was assessed using fundamental metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Coefficient of Determination (R2), as illustrated in Table 4.

Table 4 Performance of the StacBi LSTM Model

Symbol	n_steps	RMSE	MAE	MAPE	R2
ACES.JK	30	0.01049	0.00769	0.01221	0.94227
	50	0.01014	0.00820	0.01319	0.94608
	70	0.01473	0.01131	0.01775	0.88621
ADRO.JK	30	0.01047	0.00758	0.01451	0.99064
	50	0.01123	0.00829	0.01594	0.98923
	70	0.00980	0.00763	0.01596	0.99180
EXCL.JK	30	0.01254	0.01131	0.02076	0.96542
	50	0.00901	0.00708	0.01328	0.98214
	70	0.01489	0.01397	0.02653	0.95126
KLBF.JK	30	0.01656	0.01449	0.02599	0.92209
	50	0.01141	0.00946	0.01763	0.96302
	70	0.01573	0.01366	0.02475	0.92972
PGAS.JK	30	0.01003	0.00641	0.03496	0.95224
	50	0.00888	0.00797	0.05998	0.96260
	70	0.00915	0.00858	0.06007	0.96025
PTBA.JK	30	0.01072	0.00897	0.02264	0.97277
	50	0.00690	0.00534	0.01334	0.98873
	70	0.01755	0.01644	0.04525	0.92700
PTPP.JK	30	0.00632	0.00513	0.03823	0.95554
	50	0.00507	0.00423	0.03271	0.97131
	70	0.00645	0.00557	0.04492	0.95361
PWON.JK	30	0.01153	0.01005	0.02550	0.95741
	50	0.01699	0.01434	0.03395	0.90751
	70	0.00710	0.00556	0.01407	0.98383
SMRA.JK	30	0.00631	0.00522	0.01633	0.95927
	50	0.00949	0.00911	0.02990	0.90789
	70	0.00397	0.00317	0.01030	0.98392
TPIA.JK	30	0.00875	0.00644	0.00982	0.97799
	50	0.01171	0.01045	0.01630	0.96063
	70	0.01474	0.01304	0.02052	0.93765

To visually depict the model's performance, Figures 11 to 20 illustrate the actual stock prices over the next 40 days (indicated by the red line) alongside the predicted prices for the same period using input sequence lengths of $n\_steps = 30$ (represented by the blue line), $n\_steps = 50$ (depicted by the yellow line), and $n\_steps = 70$ (illustrated by the green line). These figures provide a comprehensive insight into the model's predictive capabilities under varying input conditions, allowing for a comprehensive assessment of its effectiveness in capturing short-term stock price trends.

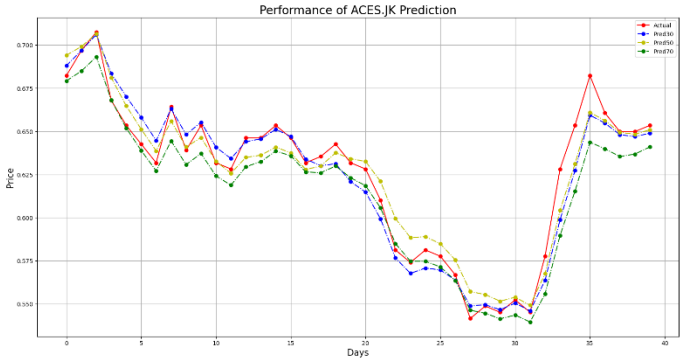


Figure 11 Performance of ACES.JK

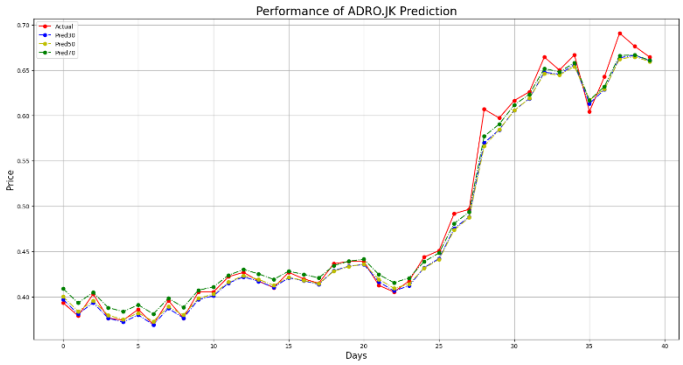


Figure 12 Performance of ADRO.JK

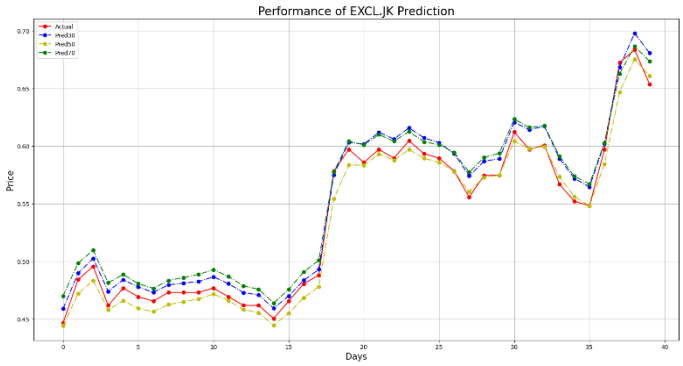


Figure 13 Performance of EXCL.JK

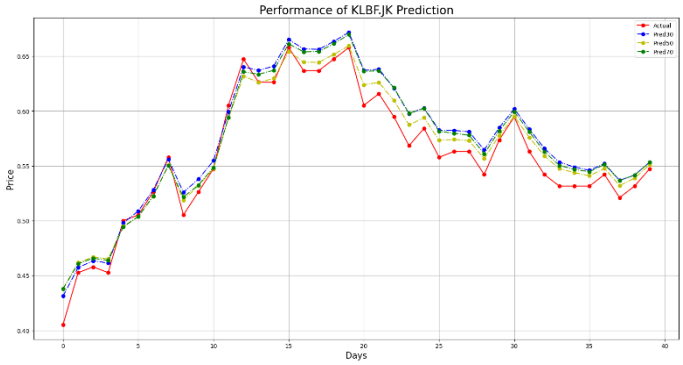


Figure 14 Performance of KLBF.JK

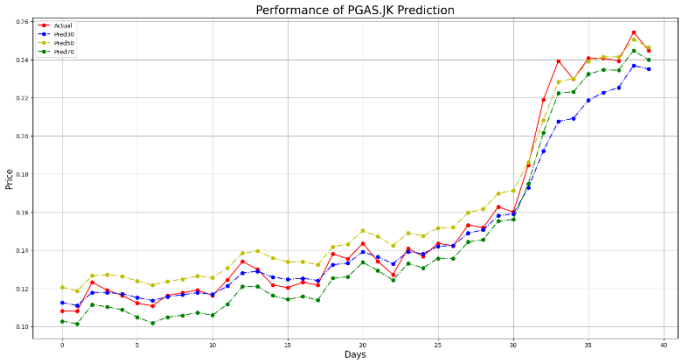


Figure 15 Performance of PGAS.JK

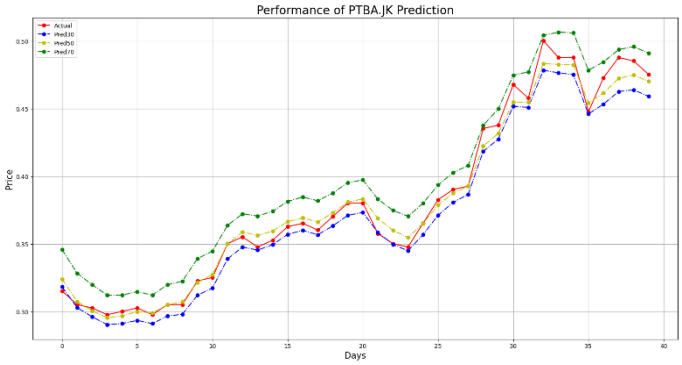


Figure 16 Performance of PTBA.JK

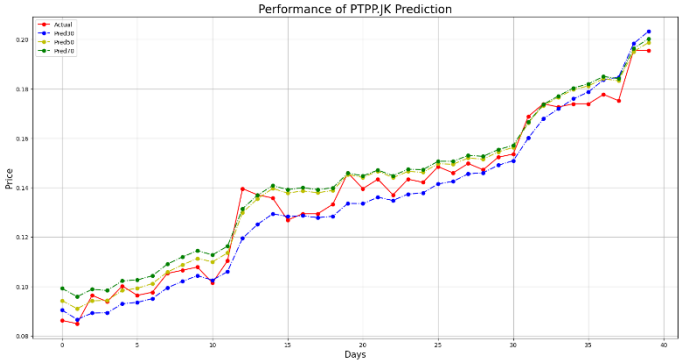


Figure 17 Performance of PTPP.JK

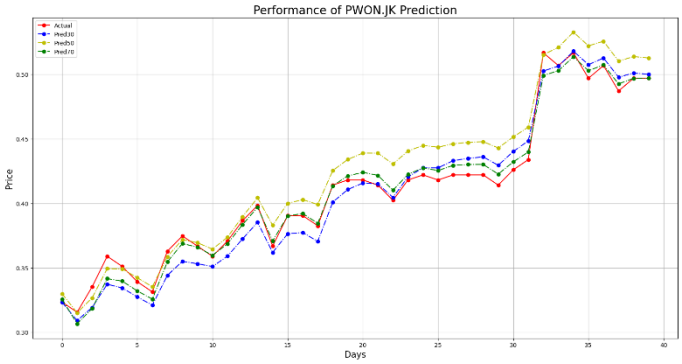


Figure 18 Performance of PWON.JK

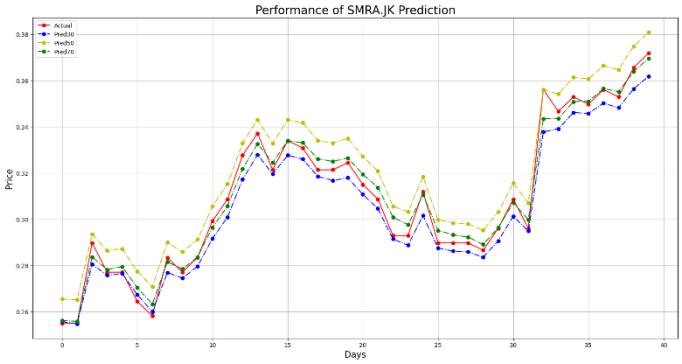


Figure 19 Performance of SMRA.JK

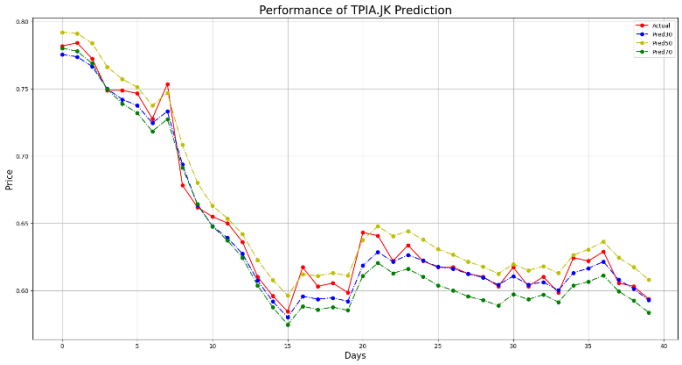


Figure 20 Performance of TPIA.JK

3.1 Impact of Input Sequence Lengths

The investigation into the StacBi LSTM model's performance across varied input sequence lengths (30, 50, and 70 days) elucidates the nuanced relationship between sequence length and predictive accuracy, assessed through key performance indicators including RMSE, MAE, MAPE, and R2. A general trend of enhanced predictive accuracy with increased input sequence length is discernible, as longer sequences give the model a more comprehensive historical context, reducing prediction errors. Nevertheless, this enhancement is not consistent across all stocks. For some, there may be a threshold beyond which extending the sequence length yields diminishing returns, as evidenced by fluctuations in RMSE and MAE values. This implies that overly extensive sequences may inadvertently introduce noise or foster overfitting, compromising prediction quality. The MAPE metric, denoting percentage errors, corroborates these findings, indicating that while longer sequences typically correlate with reduced percentage errors, the model may still grapple with predicting extreme market fluctuations, occasionally resulting in elevated MAPE scores.

Additionally, R2 values, reflecting the model's capacity to capture stock price variations, generally increase with longer sequences, signifying enhanced explanatory power. However, this trend may plateau or reverse beyond a certain sequence length. In summary, the relationship between input sequence length and model performance is complex and non-linear; optimizing sequence length necessitates a delicate balance between harnessing more historical information and mitigating the risks of noise introduction and overfitting.

3.2 Optimal Input Sequence Length

After analyzing the performance metrics of the StacBi LSTM model for predicting stock prices using input sequence lengths of 30, 50, and 70 days, an optimal range for prediction was observed. The 50-day input sequence length often results in lower RMSE, MAE, and MAPE

values than shorter or longer alternatives, making it a balanced choice. This implies that it effectively assimilates an adequate amount of historical data for precise predictions while concurrently circumventing the perils of noise and overfitting associated with unduly lengthy sequences. Shorter sequences like the 30-day option may enhance computational efficiency due to their reduced data and computational demands. Still, they risk neglecting longer-term trends vital for accurate forecasting, a drawback particularly pronounced in turbulent market conditions. Conversely, the 70-day sequence length holds the potential to discern more complex stock price patterns but at the expense of heightened computational requirements and elevated overfitting risk, especially in volatile and rapidly changing markets. Thus, the 50-day input sequence length balances efficiency with predictive accuracy, capturing a comprehensive range of short- to medium-term patterns while averting the extremes of sequence length. Nonetheless, it is crucial to acknowledge that the ideal sequence length may vary, contingent upon the individual characteristics of each stock and the broader market context.

3.3 Comparative Analysis

The comparative analysis of performance metrics across varying input sequence lengths (30, 50, and 70 days) for the StacBi LSTM model elucidates the inherent trade-offs between prediction precision and explanatory power and the challenges in optimizing both concurrently. Shorter sequences, such as the 30 days, exhibit higher precision in forecasting imminent stock price movements for certain stocks, as evidenced by lower RMSE, MAE, and MAPE values. However, these stocks tend to display slightly reduced R2 values, indicating a compromise in the model's ability to account for overall price variability due to the restricted historical context. Conversely, a 70-day sequence length often results in enhanced explanatory power, capturing a broader spectrum of price fluctuations as reflected in higher R2 values. Yet, it can also lead to increased prediction errors and potential overfitting, as denoted by elevated RMSE, MAE, and MAPE scores. Instances where the model excels in either precision or explanatory power, but not both, underscore the complexity of achieving an optimal balance and highlight the strategic nature of selecting an input sequence length tailored to specific stocks and market conditions. Shorter sequences may be preferable for day traders prioritizing immediate accuracy, while longer sequences could benefit investors seeking a comprehensive understanding of overall price trends. This analysis ultimately emphasizes the criticality of a nuanced understanding of each performance metric and the necessity of a strategic and informed approach to balance precision with explanatory power, aligning with market participants' specific objectives and strategies.

The StacBi LSTM model demonstrates notable strengths in stock price prediction, capitalizing on its unique architecture to capture long-term dependencies and temporal patterns within financial data. Its dynamic adaptability to changing market conditions, combined with a deep architecture skilled at handling non-linearities and noisy data, offers a comprehensive framework for precise predictions and insightful market analysis despite sudden market shifts and volatility challenges. In the Indonesian Stock Exchange context, the model showcases its versatility, effectively interpreting local market patterns while navigating periods of volatility characteristic of emerging markets. Nonetheless, the research process highlighted data-related challenges, necessitating meticulous preprocessing to maintain data integrity and prevent biases. Balancing model complexity with computational demands is crucial, as excessive complexity can lead to overfitting and inefficient resource use. The StacBi LSTM model is a valuable tool for traders and investors, with optimal input sequence length selection enhancing predictive accuracy and informing robust trading strategies. Integrating the model with other predictive techniques, external factors, and novel architectures presents promising avenues for advancing stock price prediction capabilities.

4. KESIMPULAN

This research paper investigated the application of the StacBi LSTM model for stock price prediction. The key findings highlight that the choice of input sequence length significantly impacts the model's performance. An optimal input sequence length was identified through rigorous evaluation, offering a balance between computational efficiency and predictive accuracy. Notably, the StacBi LSTM model demonstrated a remarkable ability to capture stock price trends by

effectively incorporating long-term dependencies and temporal patterns. The model's strengths surpassed traditional methods, enabling traders and investors to make more informed decisions.

This study opens avenues for practical applications and future enhancements in stock price prediction. The insights gained from the optimal input sequence length can guide decision-making for predictive models. At the same time, the StacBi LSTM's adeptness in capturing stock price trends underscores its potential in real-world financial forecasting scenarios. Future directions could involve hybrid approaches, integrating external factors, and addressing market shifts. In conclusion, this research underscores the significance of leveraging deep learning techniques like the StacBi LSTM for stock price prediction, presenting an impactful tool that bridges the gap between data-driven insights and effective financial strategies.

REFERENCES

Ahmad, I., Wang, X., Zhu, M., Wang, C., Pi, Y., Khan, J. A., Khan, S., Samuel, O. W., Chen, S., & Li, G. (2022). EEG-Based Epileptic Seizure Detection via Machine/Deep Learning Approaches: A Systematic Review. *Computational Intelligence and Neuroscience*, *2022*, 1–20. https://doi.org/10.1155/2022/6486570

Aydin, M., Pata, U. K., & Inal, V. (2022). Economic policy uncertainty and stock prices in BRIC countries: evidence from asymmetric frequency domain causality approach. *Applied Economic Analysis*, *30*(89), 114–129. https://doi.org/10.1108/AEA-12-2020-0172/FULL/PDF

Baek, J.-W., & Chung, K. (2023). Multi-Context Mining-Based Graph Neural Network for Predicting Emerging Health Risks. *IEEE Access*, *11*, 15153–15163. https://doi.org/10.1109/ACCESS.2023.3243722

Brahma, B., & Wadhvani, R. (2020). Solar Irradiance Forecasting Based on Deep Learning Methodologies and Multi-Site Data. *Symmetry*, *12*(11), 1830. https://doi.org/10.3390/sym12111830

Campbell, T., Dixon, K. W., Dods, K., Fearns, P., & Handcock, R. (2020). Machine Learning Regression Model for Predicting Honey Harvests. *Agriculture*, *10*(4), 118. https://doi.org/10.3390/agriculture10040118

Choi, J.-E., & Shin, D. W. (2019). The roles of differencing and dimension reduction in machine learning forecasting of employment level using the FRED big data. *Communications for Statistical Applications and Methods*, *26*(5), 497–506. https://doi.org/10.29220/CSAM.2019.26.5.497

Erizal, E., & Diqi, M. (2023). Performance Evaluation of Stock Prediction Models using EMAGRU. *Applied Computer Science*, *19*(3), 160–173. https://doi.org/10.35784/acs-2023-30

Fathy, Y., Jaber, M., & Brintrup, A. (2021). Learning With Imbalanced Data in Smart Manufacturing: A Comparative Analysis. *IEEE Access*, *9*, 2734–2757. https://doi.org/10.1109/ACCESS.2020.3047838

Gao, Y., Lian, J., & Gong, B. (2018). Automatic classification of refrigerator using doubly convolutional neural network with jointly optimized classification loss and similarity loss. *Eurasip Journal on Image and Video Processing*, *2018*(1), 1–11. https://doi.org/10.1186/S13640-018-0329-Z/FIGURES/9

Gutmann, S., Maget, C., Spangler, M., & Bogenberger, K. (2021). Truck Parking Occupancy Prediction: XGBoost-LSTM Model Fusion. *Frontiers in Future Transportation*, *2*, 693708. https://doi.org/10.3389/ffutr.2021.693708

Htun, H. H., Biehl, M., & Petkov, N. (2023). Survey of feature selection and extraction techniques for stock market prediction. *Financial Innovation*, *9*(1), 1–25. https://doi.org/10.1186/S40854-022-00441-7/FIGURES/3

Jamous, R., ALRahhal, H., & El-Darieby, M. (2021). A New ANN-Particle Swarm Optimization with Center of Gravity (ANN-PSOCoG) Prediction Model for the Stock Market under the Effect of COVID-19. *Scientific Programming*, *2021*, 1–17. https://doi.org/10.1155/2021/6656150

Jiang, H., Fang, D., Spicher, K., Cheng, F., & Li, B. (2019). A New Period-Sequential Index Forecasting Algorithm for Time Series Data. *Applied Sciences*, *9*(20), 4386. https://doi.org/10.3390/app9204386

Kim, B., Yuvaraj, N., Sri Preethaa, K. R., Hu, G., & Lee, D.-E. (2021). Wind-Induced Pressure Prediction on Tall Buildings Using Generative Adversarial Imputation Network. *Sensors*, *21*(7), 2515. https://doi.org/10.3390/s21072515

Lind, A. P., & Anderson, P. C. (2019). Predicting drug activity against cancer cells by random forest models based on minimal genomic information and chemical properties. *PLOS ONE*, *14*(7), e0219774. https://doi.org/10.1371/journal.pone.0219774

Lokanan, M. (2022). The determinants of investment fraud: A machine learning and artificial intelligence approach. *Frontiers in Big Data*, *5*, 961039. https://doi.org/10.3389/FDATA.2022.961039/BIBTEX

Ma, S.-C., Chou, W., Chien, T.-W., Chow, J. C., Yeh, Y.-T., Chou, P.-H., & Lee, H.-F. (2020). An App for Detecting Bullying of Nurses Using Convolutional Neural Networks and Web-Based Computerized Adaptive Testing: Development and Usability Study. *JMIR MHealth and UHealth*, *8*(5), e16747. https://doi.org/10.2196/16747

Miftahurrohmah, B., Wulandari, C., & Dharmawan, Y. S. (2021). Investment Modelling Using Value at Risk Bayesian Mixture Modelling Approach and Backtesting to Assess Stock Risk. *Journal of Information Systems Engineering and Business Intelligence*, *7*(1), 11. https://doi.org/10.20473/jisebi.7.1.11-21

Mo, S., Lu, P., & Liu, X. (2022). AI-Generated Face Image Identification with Different Color Space Channel Combinations. *Sensors*, *22*(21), 8228. https://doi.org/10.3390/s22218228

Musarat, M. A., Alaloul, W. S., Rabbani, M. B. A., Ali, M., Altaf, M., Fediuk, R., Vatin, N., Klyuev, S., Bukhari, H., Sadiq, A., Rafiq, W., & Farooq, W. (2021). Kabul River Flow Prediction Using Automated ARIMA Forecasting: A Machine Learning Approach. *Sustainability*, *13*(19), 10720. https://doi.org/10.3390/su131910720

Naumoski, A., Arsov, S., & Cvetkoska, V. (2022). Asymmetric Information and Agency Cost of Financial Leverage and Corporate Investments: Evidence from Emerging South-East European Countries. *Scientific Annals of Economics and Business*, *69*(2), 317–342. https://doi.org/10.47743/saeb-2022-0010

Pamir, Javaid, N., Akbar, M., Aldegheishem, A., Alrajeh, N., & Mohammed, E. A. (2022). Employing a Machine Learning Boosting Classifiers Based Stacking Ensemble Model for Detecting Non Technical Losses in Smart Grids. *IEEE Access*, *10*, 121886–121899. https://doi.org/10.1109/ACCESS.2022.3222883

Pan, D., Zeng, A., Jia, L., Huang, Y., Frizzell, T., & Song, X. (2020). Early Detection of Alzheimer's Disease Using Magnetic Resonance Imaging: A Novel Approach Combining Convolutional Neural Networks and Ensemble Learning. *Frontiers in Neuroscience*, *14*, 501050. https://doi.org/10.3389/FNINS.2020.00259/BIBTEX

Patel, R. K., Kumari, A., Tanwar, S., Hong, W.-C., & Sharma, R. (2022). AI-Empowered Recommender System for Renewable Energy Harvesting in Smart Grid System. *IEEE Access*, *10*, 24316–24326. https://doi.org/10.1109/ACCESS.2022.3152528

Qaddoura, R., M. Al-Zoubi, A., Faris, H., & Almomani, I. (2021). A Multi-Layer Classification Approach for Intrusion Detection in IoT Networks Based on Deep Learning. *Sensors*, *21*(9), 2987. https://doi.org/10.3390/s21092987

Rajamoorthy, R., Saraswathi, H. V., Devaraj, J., Kasinathan, P., Elavarasan, R. M., Arunachalam, G., Mostafa, T. M., & Mihet-Popa, L. (2022). A Hybrid Sailfish Whale Optimization and Deep Long Short-Term Memory (SWO-DLSTM) Model for Energy Efficient Autonomy in India by 2048. *Sustainability*, *14*(3), 1355. https://doi.org/10.3390/su14031355

Sekiguchi, Hayashi, Sugino, & Terada. (2019). The Effects of Differences in Individual Characteristics and Regional Living Environments on the Motivation to Immigrate to Hometowns: A Decision Tree Analysis. *Applied Sciences*, *9*(13), 2748. https://doi.org/10.3390/app9132748

Shuai, C., Pan, Z., Gao, L., & Zuo, H. (2021). Short-Term Traffic Flow Prediction of Expressway: A Hybrid Method Based on Singular Spectrum Analysis Decomposition. *Advances in Civil Engineering*, *2021*, 1–10. https://doi.org/10.1155/2021/4313970

Succetti, F., Rosato, A., Di Luzio, F., Ceschini, A., & Panella, M. (2022). A Fast Deep Learning Technique for Wi-Fi-Based Human Activity Recognition. *Progress In Electromagnetics Research*, *174*, 127–141. https://doi.org/10.2528/PIER22042605

Suleman, M. A. R., & Shridevi, S. (2022). Short-Term Weather Forecasting Using Spatial Feature Attention Based LSTM Model. *IEEE Access*, *10*, 82456–82468. https://doi.org/10.1109/ACCESS.2022.3196381

Tang, L., & Mahmoud, Q. H. (2022). A Deep Learning-Based Framework for Phishing Website Detection. *IEEE Access*, *10*, 1509–1521. https://doi.org/10.1109/ACCESS.2021.3137636

Wang, G., Cao, L., Zhao, H., Liu, Q., & Chen, E. (2021). Coupling Macro-Sector-Micro Financial Indicators for Learning Stock Representations with Less Uncertainty. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(5), 4418–4426. https://doi.org/10.1609/aaai.v35i5.16568

Williams, R. I., Smith, A., Aaron, J. R., Manley, S. C., & McDowell, W. C. (2020). Small business strategic management practices and performance: A configurational approach. *Economic Research-Ekonomska Istraživanja*, *33*(1), 2378–2396. https://doi.org/10.1080/1331677X.2019.1677488

Zhang, J., Olatosi, B., Yang, X., Weissman, S., Li, Z., Hu, J., & Li, X. (2022). Studying patterns and predictors of HIV viral suppression using A Big Data approach: a research protocol. *BMC Infectious Diseases*, *22*(1), 122. https://doi.org/10.1186/s12879-022-07047-5

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 1, JANUARI, 2024, Pp. 27 – 38
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Klasifikasi Buah dan Sayuran Segar atau Busuk Menggunakan *Convolutional Neural Network*

Eka Aenun Nisa Munfaati [(1)*], Arita Witanti [(2)]
Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana, Yogyakarta
e-mail : aenunnisaeka@gmail.com, arita@mercubuana-yogya.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 25 Juli 2023, direvisi 19 Desember 2023, diterima 19 Desember 2023, dan dipublikasikan 25 Januari 2024.

Abstract

Fresh fruits and vegetables contain many nutrients, such as minerals, vitamins, antioxidants, and beneficial fiber, superior to those found in rotten or almost rotten produce. On the other hand, fruits and vegetables that are nearly spoiled or already rotten have significantly lost their nutritional value. Rotten produce also harbors bacteria and fungi that can lead to infections and food poisoning when consumed. Convolutional Neural Network (CNN) offers a programmable solution for classifying fresh and rotten fruits and vegetables. Image processing using the TensorFlow library is employed in this classification process. During testing on the training data, the CNN achieved an accuracy of 90.42%. In comparison, the validation accuracy reached 94.21% when using the SGD optimizer, 20 epochs, a batch size 16, and a learning rate of 0.01. For the testing data, the accuracy obtained was 80.83%.

Keywords: Convolutional Neural Network, Tensorflow, Classification, Fruits, Vegetables

Abstrak

Buah dan sayuran segar mengandung banyak nutrisi seperti mineral, vitamin, antioksidan, serta serat yang baik bagi tubuh dibandingkan buah dan sayuran busuk. Di sisi lain, buah dan sayuran yang hampir busuk atau yang sudah busuk telah kehilangan sebagian besar nilai gizi yang dimiliki. Buah dan sayuran yang busuk mengandung bakteri dan jamur yang dapat menyebabkan infeksi dan keracunan makanan apabila dikonsumsi oleh tubuh. *Convolutional Neural Network* (CNN) dapat menjadi solusi untuk mengklasifikasikan buah dan sayuran yang segar dengan busuk secara terprogram. Proses pengolahan citra dengan CNN juga menggunakan *library* Tensorflow. Dalam proses klasifikasi ini menggunakan data citra berjumlah 12.220, hasil pengujian pada data *training* menghasilkan akurasi sebesar 90,42% dan akurasi validasi sebesar 94,21% dengan menggunakan optimizer SGD (*Stochastic Gradient Descent*), *epochs* 20, 16 *batch size*, dan *learning rate* 0,01. Sedangkan untuk data *testing* memperoleh akurasi sebesar 80,83%.

Kata Kunci: *Convolutional Neural Network*, Tensorflow, Klasifikasi, Buah, Sayuran

1. PENDAHULUAN

Buah dan sayuran dapat menjadi sumber nutrisi yang berguna dalam menjaga kesehatan tubuh manusia dikarenakan kandungan yang dimiliki oleh buah dan sayuran sangat banyak seperti mineral, antioksidan, vitamin, dan serat yang tinggi. Buah dan sayuran sangat bermanfaat bagi tubuh dalam menjaga kepadatan tulang dan jantung. Selain itu, buah dan sayuran juga dapat meminimalisir risiko penyakit *stroke*, jantung koroner, kanker, hipertensi, dan diabetes. *Phytoingredients* yang terkandung dalam buah dan sayuran berfungsi untuk mencegah terjadinya *stress* oksidatif (Rarastiti, 2022). Dalam penelitian Jafaruddin (2023), mengonsumsi buah dan sayuran segar dapat meningkatkan daya ingat. Hal ini dikarenakan buah dan sayuran segar mengandung zat antioksidan yang berguna untuk melindungi sel-sel pada otak. Kandungan antioksidan dalam buah dan sayuran segar juga dapat mengobati penyakit kanker dan jenis penyakit lainnya.

Dalam *The World Health Organization* (WHO) oleh Magalhães et al. (2022) menganjurkan untuk mengonsumsi setidaknya 400 gram buah dan sayuran segar. Mengonsumsi buah dan sayuran segar mengandung gizi yang sangat tinggi sehingga dalam menjaga kesehatan tubuh, sedangkan mengonsumsi buah dan sayuran busuk sangat berbahaya karena dapat menyebabkan keracunan. Meskipun demikian, dalam penelitian oleh Gregori et al. (2019) mengatakan bahwa minat akan buah dan sayuran pada tingkat dunia masih terbilang rendah jika didasarkan pada anjuran WHO. Pada tahun 2007, rata-rata konsumsi harian per kapita buah dan sayuran di Amerika Serikat (USA) adalah 325 g dan 505 g (830 g); Jerman 624 g dan 106 g (730 g); Finlandia 429 g dan 130 g (559 g). Angka-angka ini menunjukan adanya kesenjangan antara anjuran WHO dengan minat konsumsi buah dan sayuran di berbagai negara.

Upaya dalam meningkatkan konsumsi buah dan sayuran setiap harinya dapat dimulai dengan meningkatkan kualitas buah dan sayuran yang segar. Salah satu upaya dalam meningkatkan kualitas buah dan sayuran segar adalah dengan memisahkan antara buah dan sayuran segar dengan buah dan sayuran busuk. Buah dan sayuran segar yang tercampur dengan buah dan sayuran busuk dapat mengakibatkan buah dan sayuran segar menjadi busuk karena terkontaminasi oleh buah dan sayuran busuk. Untuk memisahkan buah dan sayuran segar dengan busuk dapat dilakukan secara otomatis dengan memanfaatkan teknologi masa kini. Perkembangan teknologi memberikan kemudahan dalam melakukan pemisahan terhadap buah dan sayuran segar atau busuk dengan lebih cepat dibandingkan dilakukan secara manual.

Convolutional Neural Network (CNN) termasuk algoritma *deep learning* yang dapat digunakan dalam klasifikasi objek Iswantoro & UN (2022). Dengan menggunakan teknologi masa kini maka dapat melakukan proses pengolahan gambar untuk diklasifikasikan dengan hanya menggunakan fitur penangkap gambar berupa kamera atau galeri dari sistem. Menurut Sya'ban et al. (2022) dalam penelitiannya mengatakan bahwa CNN bekerja sama halnya dengan cara kerja *neuron* yang terdapat pada otak manusia berdasarkan persepsi visual.

Pengembangan model CNN untuk klasifikasi buah segar atau busuk pernah dilakukan oleh Sya'ban et al. (2022) dengan cara melakukan perancangan terhadap arsitektur CNN dengan tujuan agar model mampu mengklasifikasikan jenis buah beserta tingkat kesegarannya. Penelitian ini terdiri dari buah apel, pisang, dan jeruk mandarin dari yang segar hingga busuk di mana *dataset* dalam penelitian ini diperoleh dari Kaggle. *Dataset* dalam penelitian ini terdiri dari 10,900 citra buah yang terbagi menjadi 3 data di antaranya 8.720 pada data *train*, 2.180 pada data validasi, dan 1.090 pada data *testing*. Hasil riset memperoleh akurasi *training* mencapai 98,20%, 99,22% pada akurasi validasi, dan 91,65% pada akurasi *testing* dengan menggunakan 50 *epochs*.

Dalam penelitian "Klasifikasi *Chest X-Ray Images* Berdasarkan Kriteria Gejala Covid-19 Menggunakan *Convolutional Neural Network*" yang dilakukan oleh Ayumi & Nurhaida (2021) mengusulkan implementasi metode CNN dengan menggunakan 2 lapisan *layer convolution, maxpooling,* dan *fully connected layer* untuk melakukan klasifikasi terhadap citra *chest X-ray* pada pasien yang diduga terkena Covid-19. Penelitian ini menggunakan *dataset* Covid-19 *Radiography Database* yang terdiri dari 1.200 citra pada *class* kasus positif Covid-19 dan 1.341 citra pada *class* normal. Riset ini memperoleh hasil akurasi *training* mencapai 94,05% dengan *loss* 25,58% dan akurasi validasi mencapai 96,06% dengan *loss* 14,71%.

Penelitian oleh Hawari et al. (2022) membahas mengenai implementasi metode CNN dalam melakukan klasifikasi terhadap 4 jenis tanaman padi di antaranya yaitu: *Leaf Brown, Brown Spot, Daun Sehat,* dan *Hawar*. Dalam risetnya, ia melakukan *training* model dengan menggunakan 10 *epochs* sehingga memperoleh akurasi *training* sebesar 85% dengan *loss* mencapai 38% dan akurasi validasi sebesar 95% dengan *loss* 30%.

Penelitian berikutnya oleh Qotrunnada et al. (2022) tentang klasifikasi terhadap wajah bermasker atau tidak bermasker dengan *convolutional neural network*. Penelitian yang ia lakukan

menggunakan citra wajah dengan ukuran 150x150 piksel. Pada penelitiannya ia memperoleh akurasi sebesar 82,35% pada *validation* dan 98,20% pada *training*.

Berdasarkan penelitian di atas, dapat disimpulkan bahwa CNN mampu melakukan klasifikasi terhadap suatu *object* dengan tingkat akurasi yang tinggi dan mudah digunakan. Sehingga dalam penelitian ini penulis memutuskan untuk menggunakan metode CNN dalam melakukan klasifikasi terhadap buah segar dan busuk. Dengan digunakannya metode ini diharapkan dapat menghasilkan akurasi yang baik.

2. METODE PENELITIAN

2.1 Konsep Dasar

2.1.1 *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) yaitu *deep learning* untuk mengidentifikasi objek yang terdapat pada sebuah gambar. Hasil identifikasi objek tersebut dimanfaatkan untuk menganalisa pada citra yang ada (Iswantoro & UN, 2022). Dalam penelitian Mulyanto et al. (2021) mengatakan bahwasannya CNN dapat digunakan untuk mengklasifikasikan citra atau gambar dengan menggunakan pendekatan *deep learning*. *Pooling, convolution*, dan *fully connected* merupakan *layer* pada CNN. Selain itu juga terdapat *feature learning* dan *classification*.

Feature learning yaitu suatu proses yang melakukan "*encoding"* pada citra menjadi sebuah fitur yang berbentuk numerik dengan tujuan untuk menampilkan citra di dalamnya (Mahaputri et al., 2022). Menurut Wulandari et al. (2020) *feature learning* termasuk bagian dari arsitektur *Convolutional Neural Network* yang berperan dalam mengkonversi matriks input menjadi *feature maps*. *Convolutional layer* dan *pooling layer* merupakan *layer* yang dimilki *feature learning*.

Classification bertujuan untuk mengidentifikasi *neuron* yang diekstrak yang terdapat pada *feature learning*. Dalam *classification* terdapat *flatten* dan *fully connected layer*. Menurut Mahaputri et al. (2022) *flatten* dapat melakukan konversi antara *feature map* yang berbentuk *multidimensional array* menjadi vektor.

2.1.2 *Confusion Matrix*

Confusion matrix yaitu suatu proses untuk mempresentasikan dan mencocokan antara nilai sebenarnya dengan nilai perkiraan pada model untuk mengevaluasi nilai akurasi, *precision*, *recall*, dan *F1-score* (Mahaputri et al., 2022). Istilah-istilah yang digunakan dalam *confusion matrix* di antaranya yaitu: *False Negative* (FN), *True Negative* (TN), *False Positive* (FP), dan *True Positive* (TP).

2.1.3 Python dan Tensorflow

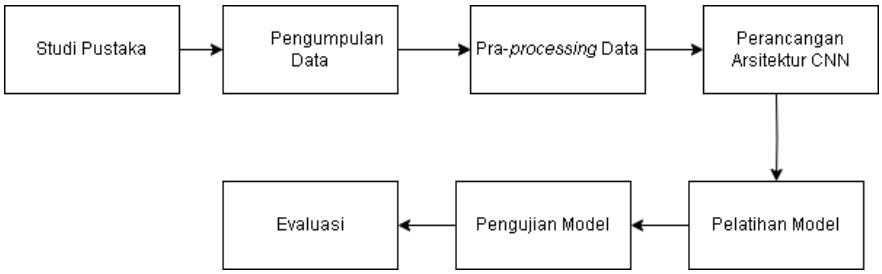
Bahasa pemrograman Python termasuk jenis bahasa pemrograman tingkat tinggi dikarenakan memiliki kemampuan yang lebih tinggi dalam abstraksi dan pemrograman yang mudah dipahami oleh manusia dibandingkan bahasa lainnya. Python memiliki *library* yang lengkap dan bersifat *open source*. Oleh sebab itu bahasa pemrograman Python sangat cocok jika digunakan dalam pengembangan *deep learning* dan juga *machine learning* (Alfarizi et al., 2023).

Tensorflow merupakan sebuah *open-source library* untuk mengembangkan dan melatih model *machine learning* atau *deep learning*. Tensorflow mengintegrasikan model algoritma *deep learning* dan jaringan saraf ke dalam satu set *library* untuk melakukan tugas-tugas di bidang pembelajaran mesin dan jaringan saraf (Apendi et al., 2023).

2.2 Tahap Penelitian

Proses pengembangan model CNN untuk klasifikasi buah dan sayuran segar atau busuk dilakukan melalui 6 tahap. Keenam tahapan tersebut adalah studi pustaka, pengumpulan data,

preprocessing data, perancangan arsitektur CNN, pelatihan model, serta pengujian dan evaluasi model. Alur penelitian ditunjukkan seperti pada Gambar 1.



Gambar 1 Tahap Penelitian

2.2.1 Studi Pustaka

Tahapan awal yang dilakukan dalam penelitian ini adalah melakukan studi pustaka. Studi pustaka dilakukan dengan mempelajari berbagai buku dan jurnal ilmiah. Hal ini dilakukan untuk mempelajari algoritma klasfikasi gambar atau objek dengan menggunakan *convolutional neural network.*

2.2.2 Pengumpulan Data

Tabel 1 Data Primer Buah dan Sayuran

No.	*Class*	Data *Testing*
1	Apel Segar	10
2	Apel Busuk	10
3	Pisang Segar	10
4	Pisang Busuk	10
5	Timun Segar	10
6	Timun Busuk	10
7	Tomat Segar	10
8	Tomat Busuk	10
9	Kentang Segar	10
10	Kentang Busuk	10
11	Okra Segar	10
12	Okra Busuk	10
	Total	**120**

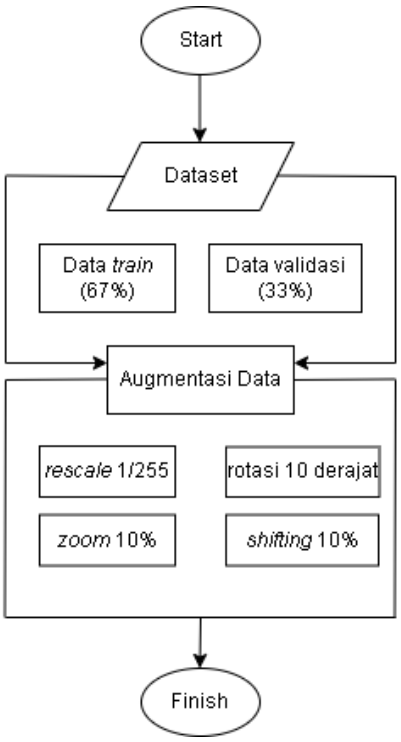
Tabel 2 Data Sekunder Buah dan Sayuran

No.	*Class*	Data *Train*	Data Validasi
1	Apel Segar	731	396
2	Apel Busuk	906	387
3	Pisang Segar	887	511
4	Pisang Busuk	708	370
5	Timun Segar	496	279
6	Timun Busuk	421	255
7	Tomat Segar	877	255
8	Tomat Busuk	843	353
9	Kentang Segar	536	270
10	Kentang Busuk	802	370
11	Okra Segar	635	370
12	Okra Busuk	338	224
	Total	**8180**	**4040**

Penelitian ini menggunakan data primer dan data sekunder. Data primer diperoleh dari kamera peneliti dan Google Photo, sedangkan data sekunder menggunakan *dataset Vegetable and Fruits Fresh and Stale* pada Kaggle (Baloch & Khalique, 2022). Data primer dan sekunder terdiri dari buah dan sayuran segar hingga busuk dengan total citra 120 citra dan 12.220 citra. Data primer dan data sekunder seperti pada Tabel 1 dan Tabel 2. Data primer digunakan dalam pengujian atau evaluasi model CNN, sedangkan data sekunder digunakan untuk tahap pelatihan model CNN.

2.2.3 *Preprocessing* Data

Pre-processing data dilakukan dengan menggunakan proses *thresholding*, yaitu mengubah nilai citra dari 0-255 menjadi 0-1. Selanjutnya dilakukan augmentasi data yang terdiri dari melakukan rotasi pada gambar sebesar 10 derajat, *zoom* pada gambar sebesar 10%, dan mengatur *shifting* sebesar 10%. Tahapan *preprocessing* seperti pada Gambar 2.

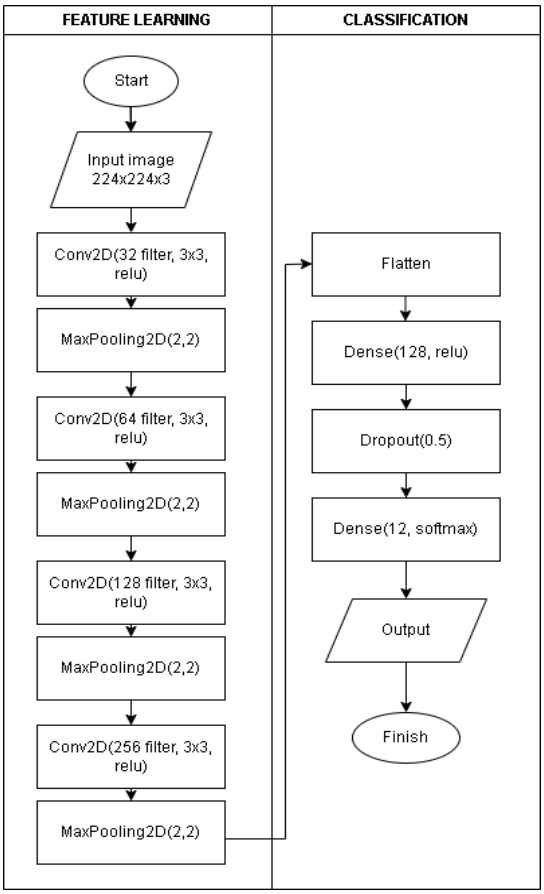


Gambar 2 Tahap *Preprocessing*

2.2.4 Perancangan Arsitektur CNN

Dalam tahap perancangan arsitektur CNN, citra pada *dataset* di-*resize* menjadi 224x224x3 piksel. Perancangan arsitektur CNN terdiri dari *feature learning* dan *classification*. Pada tahap *feature learning,* terdapat proses *convolution* dan *pooling* dalam citra *input*. *Convolution* pertama menggunakan 32 filter dan 3x3 kernel matriks dengan aktivasi ReLu. Setelah itu, terdapat proses *Max pooling2D* dengan ukuran filter 2x2. Setelah itu, dilakukan proses konvolusi kedua dengan 64 filter dan 3x3 kernel matriks dengan aktivasi ReLu. Kemudian, dilakukan kembali proses *Max pooling2D* dengan ukuran filter 2x2 dan dilanjutkan dengan proses konvolusi ketiga dan keempat, masing-masing dengan jumlah filter 128 dan 256, kernel matriks 3x3 dengan aktivasi ReLu serta *Max pooling2D* dengan ukuran filter 2x2. Setelah tahap *feature learning* selesai, kemudian akan dilakukan proses *flatten* untuk mengubah *output* dari proses *convolution* yang berupa *matrix* menjadi vektor dan kemudian dilanjutkan dengan tahap *classification*.

Pada tahap *classification*, lapisan *dense* pertama terdiri dari 128 *neuron* dengan aktivasi ReLu dan dilanjutkan dengan teknik *dropout* sebesar 0,5. Lapisan *dense* kedua terdiri dari 12 *neuron* dengan aktivasi *softmax* tanpa menggunakan teknik *dropout*. Arsitektur CNN pada penelitian ini seperti pada Gambar 3.



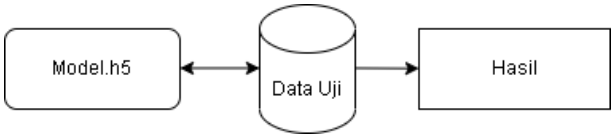
Gambar 3 Arsitektur CNN

2.2.5 Pelatihan Model

Prose pelatihan model dilakukan dengan data sekunder yang telah dilakukan teknik *pre-processing*. Pada pelatihan model dilakukan scenario uji coba terhadap 4 *hyperparameter* yaitu: *optimizer* (ADAM, SGD, dan RMSprop), *learning rate* (0,01, 0,001, dan 0,0001), *batch size* (16, 32, dan 64), dan *epochs* (10, 15, dan 20). Hal ini bertujuan untuk memperoleh hasil akhir yang optimal pada model CNN.

2.2.6 Pengujian dan Evaluasi Model

Citra buah dan sayuran pada data uji digunakan untuk proses pengujian dan evaluasi model. Hasil dari evaluasi model akan direpresentasikan dalam *confusion matrix* untuk memberikan gambaran yang lebih komprehensif tentang performa model. Tahap evaluasi model seperti pada Gambar 4.



Gambar 4 Tahap Evaluasi Model CNN

3. HASIL DAN PEMBAHASAN

3.1 Skenario Uji Coba

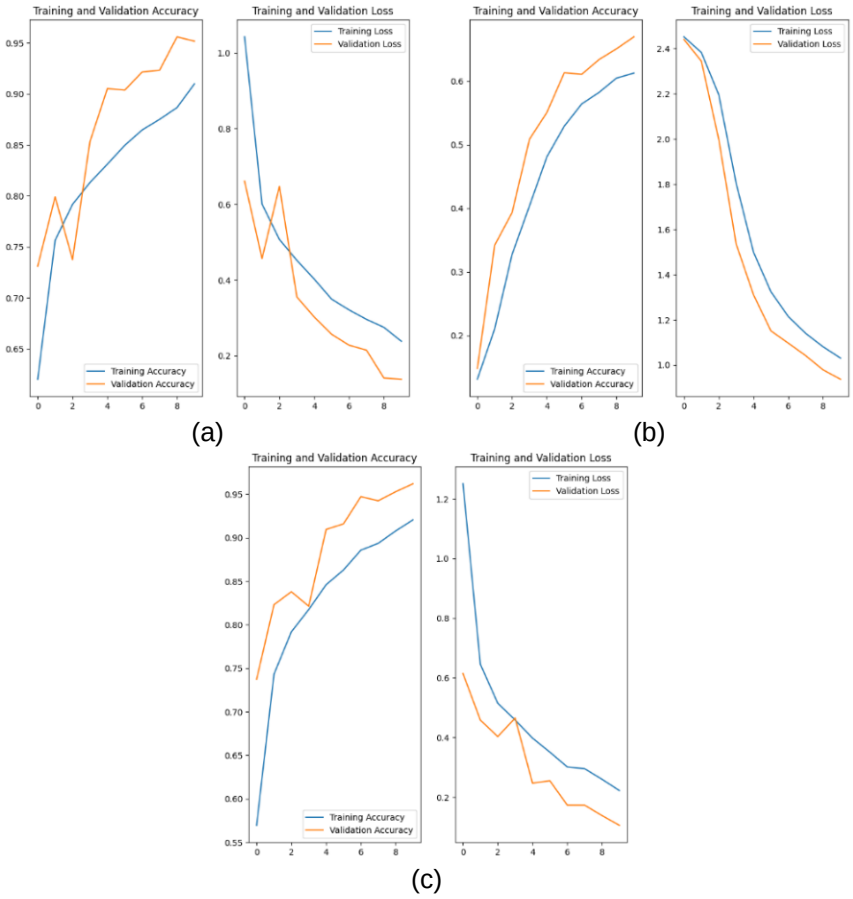
Dilakukan skenario uji coba terhadap *hyperparameter* untuk melihat performa model terbaik pada setiap skenario uji coba. Skenario uji coba terdiri dari skenario *optimizer* (ADAM, SGD, dan RMSprop), *learning rate* (0,01, 0,001, dan 0,0001), *batch size* (16, 32, dan 64), dan *epochs* (10, 15, dan 20). Setiap citra input akan di-*resize* menjadi 224x224x3 piksel. Hasil dari skenario uji coba adalah sebagai berikut.

3.1.1 Skenario Uji Coba *Optimizer*

Skenario pertama yaitu melakukan pengujian terhadap 3 jenis *optimizer* yaitu Adam, SGD, dan RMSprop. Pada pengujian *optimizer hyperparameter* seperti *epochs* diatur dengan jumlah 10 *epochs*, 32 *batch size*, dan 0,001 *learning rate.* Hasil dari skenario *optimizer* seperti pada Tabel 3.

Tabel 3 Perbandingan Akurasi dan *Loss* pada *Optimizer*

Optimizer	*Validation Acc*	*Loss*
Adam	0,9559	0,1413
SGD	0,6696	0,9368
RMSprop	0,9619	0,1053



Gambar 5 Grafik Pengujian Optimizer (a) Adam, (b) SGD, dan (c) RMSProp

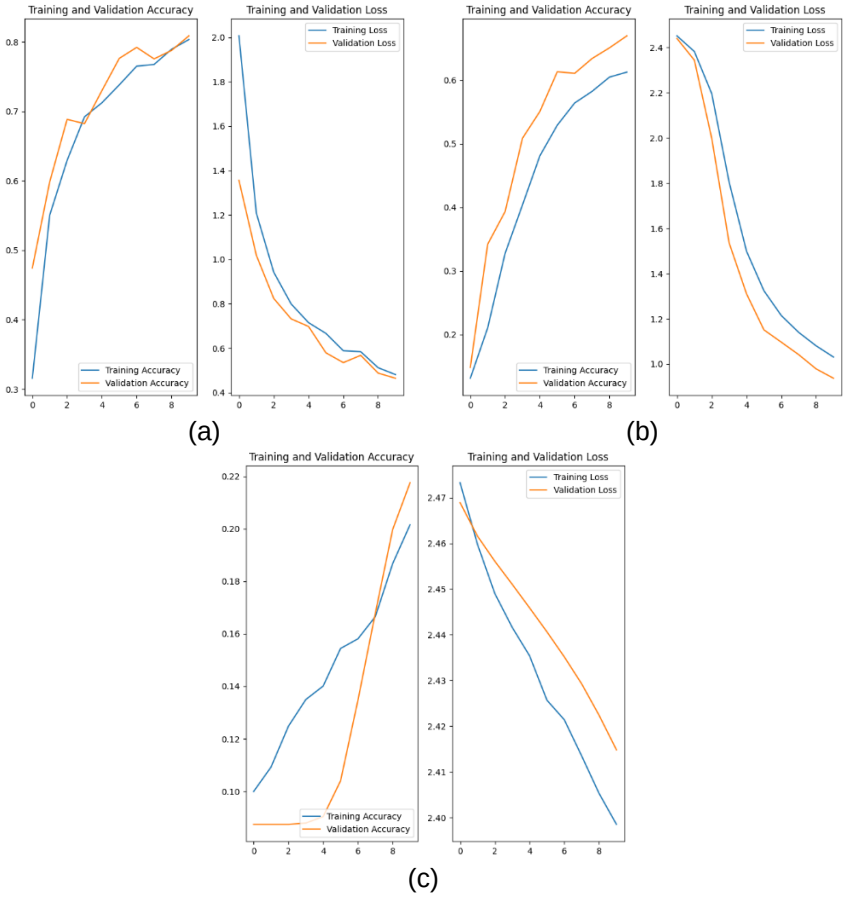
Berdasarkan Tabel 3, *optimizer* RMSprop memperoleh akurasi tertinggi disertai *loss* terendah. Sedangkan untuk *optimizer* Adam memperoleh akurasi dan *loss* sedikit lebih rendah dibandingkan dengan *optimizer* RMSprop. Namun, hasil terbaik yang dipilih untuk skenario *optimizer* adalah SGD, karena pada setiap iterasi (*epochs*) mengalami peningkatan nilai akurasi. Sedangkan *optimizer* Adam dan RMSprop masih muncul beberapa *spikes* atau lonjakan seperti pada Gambar 5.

3.1.2 Skenario Uji Coba *Learning Rate*

Skenario kedua yaitu pengujian terhadap pengaruh besaran nilai *learning rate*. Nilai *learning rate* yang digunakan terdiri dari nilai 0,01, 0,001, dan 0,0001. Sedangkan untuk *hyperparameter* lainnya seperti *optimizer* menggunakan RMSprop, 10 epochs, dan 32 *batch* size. Hasil skenario uji coba *learning rate* seperti pada Tabel 4.

Tabel 4 Perbandingan Akurasi dan *Loss* pada *Learning Rate*

Learning Rate	*Validation Acc*	*Loss*
0,01	0,8089	0,4644
0,001	0,6696	0,9368
0,0001	0,2176	2.4148



Gambar 6 Grafik Pengujian *Learning Rate* (a) 0,01, (b) 0,001, dan (c) 0,0001

Berdasakan Tabel 4, *learning rate* dengan nilai yang besar memperoleh akurasi yang sangat rendah disertai *loss* yang sangat tinggi. Hal ini disebabkan semakin besar nilai *learning rate* maka akan sangat berdampak pada ketelitian *neural network* pada saat proses pelatihan model dikarenakan model mengalami proses pelatihan yang terbilang lebih cepat. Sedangkan, semakin

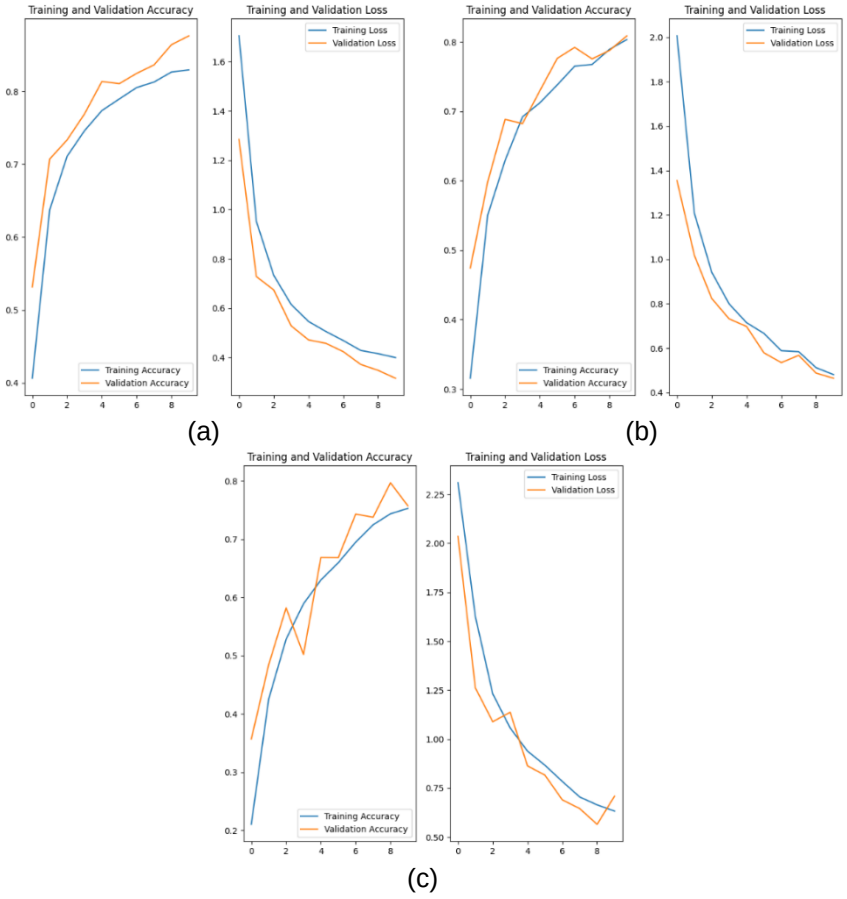
kecil nilai *learning rate* mengakibatkan *neural network* memiliki tingkat ketelitian yang cenderung lebih tinggi namun membutuhkan waktu *running* yang cenderung lebih lama. Sehingga pada skenario ini, nilai *learning rate* terbaik adalah 0,01. Grafik pengujian terhadap pengaruh nilai *learning rate* dapat dilihat pada Gambar 6.

3.1.3 Skenario Uji Coba *Batch Size*

Skenario ketiga yaitu pengujian terhadap pengaruh besaran *batch size*. Nilai *batch size* yang sesuai akan menghasilkan model dengan performa yang baik. Skenario *batch size* dilakukan dengan menggunakan besaran *batch size* di antaranya yaitu 16, 32, dan 64. *Hyperparameter* lainnya menggunakan *hyperparameter* terbaik yang diperoleh pada tahap sebelumnya yaitu *optimizer* RMSprop, 0,01 *learning* rate, dan 10 *epochs.* Hasil skenario *batch size* dapat dilihat pada Tabel 5.

Tabel 5 Perbandingan Akurasi dan *Loss* Terhadap Penggunaan Nilai *Batch Size*

Batch Size	*Validation Acc*	*Loss*
16	0,8762	0,3162
32	0,8089	0,4644
64	0,7968	0,5651



Gambar 7 Grafik Pengujian *Batch Size* (a) 16, (b) 32, dan (c) 64

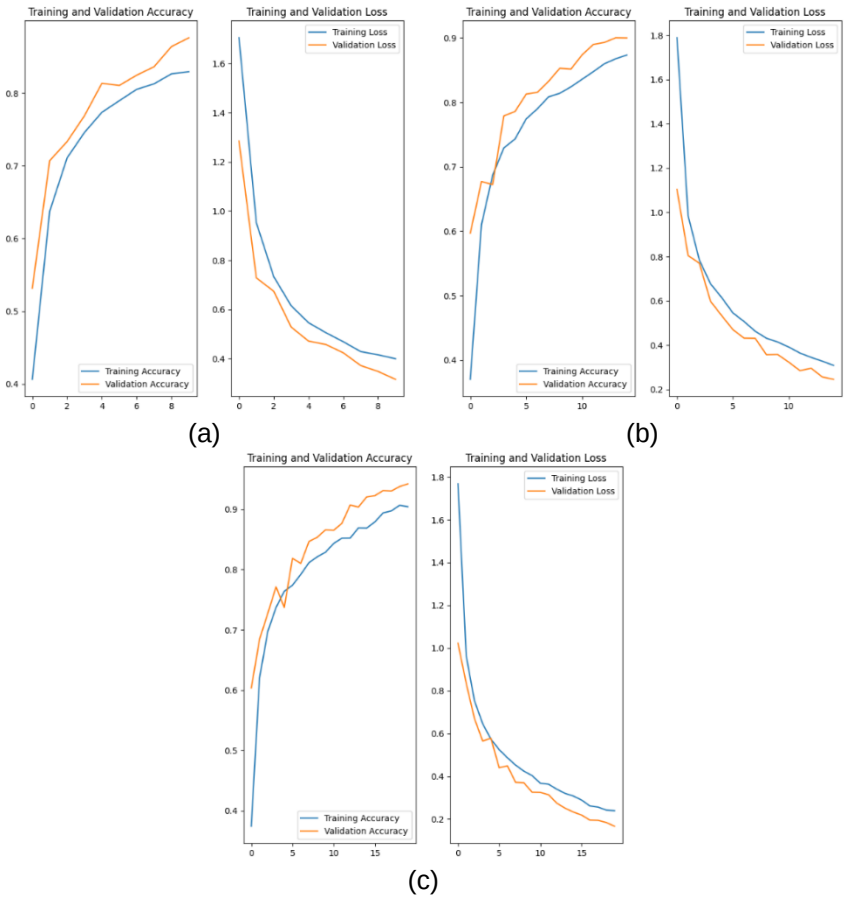
Berdasarkan Tabel 5, besaran *batch size* yang lebih kecil memperoleh akurasi yang cenderung lebih tinggi dibanding *batch size* yang lebih tinggi. *Batch size* yang lebih kecil memperoleh *loss* terendah dibandingkan dengan besaran *batch size* lainnya. Grafik pengujian terhadap pengaruh nilai *batch size* seperti pada Gambar 7.

3.1.4 Skenario Uji Coba *Epochs*

Skenario keempat yaitu pengujian terhadap penggunaan jumlah *epochs* pada saat pelatihan model dengan tujuan untuk memperoleh jumlah *epochs* yang optimal sehingga menghasilkan akurasi terbaik. Pengujian dilakukan dengan menggunakan beberapa jumlah *epoch* antara lain 10, 15, dan 20 *epochs*. *Hyperparameter optimizer* menggunakan RMSprop, 0,01 *learning rate,* dan 16 *batch size.* Hasil skenario *epochs* seperti pada Tabel 6.

Tabel 6 Perbandingan Akurasi dan *Loss* pada *Epochs*

Epochs	*Validation Acc*	*Loss*
10	0,8762	0,3162
15	0,8998	0,2454
20	0,9421	0,1661

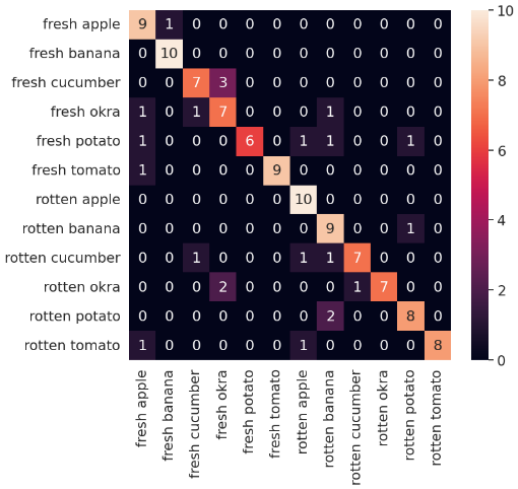


Gambar 8 Grafik Pengujian Jumlah *Epochs* (a) 10, (b) 15, dan (c) 20

Berdasarkan Tabel 6, bertambahnya jumlah *epochs* sangat berpengaruh terhadap tingkat akurasi dan *loss* yang diperoleh. *Epochs* dengan jumlah yang lebih banyak memperoleh akurasi yang semakin tinggi dan *loss* yang lebih rendah. Jumlah *epochs* terbaik dalam uji coba ini adalah 20 *epochs* dengan perolehan akurasi validasi mencapai 94,21%. Pada *epochs* yang berjumlah 20 memperlihatkan bahwasannya akurasi validasi memperoleh akurasi yang lebih tinggi dibandingkan akurasi *training* hampir pada setiap *epochs*. Hal ini menunjukan bahwasannya model mampu menggeneralisasi dengan baik. Masing-masing akurasi pada *training* dan validasi mengalami peningkatan yang fluktuatif. Hal ini berarti model belum mencapai konvergen, namun hal tersebut tidak selalu mengindikasikan adanya *overfitting* atau *underfitting* pada model. Grafik pengujian terhadap pengaruh jumlah *epochs* seperti pada Gambar 8.

3.2 Hasil Evaluasi Model

Berdasarkan skenario uji coba terhadap *hyperparameter* yang telah dilakukan maka diperoleh *hyperparameter* yaitu: *optimizer* SGD, 0,01 *learning rate,* 16 *batch size,* dan 20 *epochs*. Model CNN dengan akurasi tertinggi yang diperoleh akan dilakukan evaluasi menggunakan data uji/*testing* sehingga memperoleh *confusion matrix* seperti pada Gambar 9. Berdasarkan *confusion matrix* pada Gambar 9. maka diperoleh nilai *precision, recall, f1-score,* dan akurasi pada data uji seperti pada Tabel 7.



Gambar 9 *Confusion Matrix* pada Data Uji

Tabel 7 Hasil Evaluasi Model CNN pada Data Uji

	Precision	***Recall***	***F1-Score***
Fresh Apple	69,23%	90%	78,26%
Fresh Banana	90,91%	100%	95,24%
Fresh cucumber	77,78%	70%	73,68%
Fresh Okra	58,33%	70%	63,64%
Fresh Potato	100%	60%	75%
Fresh Tomato	100%	90%	94,74%
Rotten Apple	76,92%	100%	86,96%
Rotten Banana	64,29%	90%	75%
Rotten cucumber	87,5%	70%	77,78%
Rotten Okra	100%	70%	82,35%
Rotten Potato	80%	80%	80%
Rotten Tomato	100%	80%	88,89%
Rata-rata	**83,75%**	**80,83%**	**80,96%**
Akurasi		**80,83%**	

4. KESIMPULAN

Memisahkan antara buah dan sayuran yang segar dengan busuk dapat dilakukan dengan menggunakan teknologi *deep learning*. Salah satu teknologi *deep learning* adalah *Convolutional Neural Network* (CNN). Dalam penelitian ini, klasifikasi dengan CNN memperoleh akurasi pelatihan sebesar 90,42%, akurasi validasi mencapai 94,21%, dan akurasi pengujian mencapai 80,83%. Akurasi ini diperoleh dengan menggunakan *optimizer* SGD, 0,01 *learning rate,* 16 *batch size,* dan 20 *epochs*. Dengan demikian, penggunaan teknologi *deep learning*, khususnya CNN, dapat menjadi solusi yang andal dan efisien dalam melakukan klasifikasi objek visual, seperti membedakan antara buah dan sayuran yang segar dengan yang busuk untuk tujuan pemilihan dan manajemen kualitas produk.

DAFTAR PUSTAKA

Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning. *KARIMAH TAUHID*, *2*(1), 1–6. https://doi.org/10.30997/KARIMAHTAUHID.V2I1.7518

Apendi, S., Setianingsih, C., & Paryasto, M. W. (2023). Deteksi Bahasa Isyarat Sistem Isyarat Bahasa Indonesia Menggunakan Metode Single Shot Multibox Detector. *EProceedings of Engineering*, *10*(1), 249–255. https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/19322

Ayumi, V. (Vina), & Nurhaida, I. (Ida). (2021). Klasifikasi Chest X-Ray Images Berdasarkan Kriteria Gejala Covid-19 Menggunakan Convolutional Neural Network. *Journal Scientific and Applied Informatics*, *4*(2), 147–153. https://doi.org/10.36085/JSAI.V4I2.1513

Baloch, A., & Khalique, A. (2022). *Vegetables & Fruits fresh and Stale*. Kaggle. https://www.kaggle.com/datasets/alibaloch/vegetables-fruits-fresh-and-stale

Gregori, D., French, M., Gallipoli, S., Lorenzoni, G., & Ghidina, M. (2019). Global, Regional, and National Levels of Fruit and Vegetable Consumption from the ROUND (WoRld Map of COnsUmption of Fruit and Vegetables and Nutrient Deficits) Project (P18-067-19). *Current Developments in Nutrition*, *3*(Suppl 1), nzz039.P18-067-19. https://doi.org/10.1093/CDN/NZZ039.P18-067-19

Hawari, F. H., Fadillah, F., Alviandi, M. R., & Arifin, T. (2022). Klasifikasi Penyakit Tanaman Padi Menggunakan Algoritma CNN (Convolutional Neural Network). *Jurnal Responsif : Riset Sains Dan Informatika*, *4*(2), 184–189. https://doi.org/10.51977/JTI.V4I2.856

Iswantoro, D., & UN, D. H. (2022). Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN). *Jurnal Ilmiah Universitas Batanghari Jambi*, *22*(2), 900. https://doi.org/10.33087/jiubj.v22i2.2065

Jafaruddin, N. (2023). Pola Hidup Sehat dengan Konsumsi Buah dan Sayur. *Abdimas Galuh*, *5*(1), 569–577. https://doi.org/10.25157/AG.V5I1.9919

Magalhães, B., Gaspar, P. D., Corceiro, A., João, L., & Bumba, C. (2022). Fuzzy Logic Decision Support System to Predict Peaches Marketable Period at Highest Quality. *Climate 2022, Vol. 10, Page 29*, *10*(3), 29. https://doi.org/10.3390/CLI10030029

Mahaputri, C., Kristian, Y., & Setyati, E. (2022). Pengenalan Makanan Tradisional Indonesia Beserta Bahan-bahannya dengan Memanfaatkan DCNN Transfer Learning. *INSYST: Journal of Intelligent System and Computation*, *4*(2), 61–68. https://doi.org/10.52985/INSYST.V4I2.252

Mulyanto, A., Susanti, E., Rossi, F., Wajiran, W., & Borman, R. I. (2021). Penerapan Convolutional Neural Network (CNN) pada Pengenalan Aksara Lampung Berbasis Optical Character Recognition (OCR). *JEPIN (Jurnal Edukasi Dan Penelitian Informatika)*, *7*(1), 52–57. https://doi.org/10.26418/JP.V7I1.44133

Qotrunnada, Mufida, F., & Utomo, P. H. (2022). Metode Convolutional Neural Network untuk Klasifikasi Wajah Bermasker. *PRISMA, Prosiding Seminar Nasional Matematika*, *5*, 799–807. https://journal.unnes.ac.id/sju/index.php/prisma/article/view/54602

Rarastiti, C. N. (2022). Hubungan Pengetahuan Gizi Seimbang dengan Konsumsi Buah dan Sayur pada Remaja. *Jurnal Penelitian Inovatif*, *2*(2), 281–288. https://doi.org/10.54082/JUPIN.80

Sya'ban, D. R., Hamzah, A., & Susanti, E. (2022). Klasifikasi Buah Segar dan Busuk Menggunakan Algoritma Convolutional Neural Network Dengan TFLite Sebagai Media Penerapan Model Machine Learning. *PROSIDING SNAST*, F7-16. https://doi.org/10.34151/PROSIDINGSNAST.V8I1.4180

Wulandari, I., Yasin, H., & Widiharih, T. (2020). Klasifikasi Citra Digital Bumbu dan Rempah dengan Algoritma Convolutional Neural Network (CNN). *Jurnal Gaussian*, *9*(3), 273–282. https://doi.org/10.14710/J.GAUSS.9.3.273-282

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 1, JANUARI, 2024, Pp. 39 – 49
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Implementasi *Load Balancing* dengan HAProxy di Sistem Informasi Akademik UIN Sunan Kalijaga

Adi Wirawan [(1)], Rahmadhan Gatra [(2)*], Hendra Hidayat [(3)], Daru Prasetyawan [(4)]
Pusat Teknologi Informasi dan Pangkalan Data, UIN Sunan Kalijaga, Yogyakarta
e-mail : {adi.wirawan,rahmadhan.gatra,hendra.hidayat,daru.prasetyawan}@uin-suka.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 4 September 2023, direvisi 1 November 2023, diterima 7 November 2023, dan dipublikasikan 25 Januari 2024.

Abstract

Efficiently managing academic information systems (AIS) is essential for educational institutions to provide reliable services to students and faculty. This research explores the integration of HAProxy load balancing and file synchronization techniques to optimize the performance of AIS. HAProxy is employed to distribute incoming requests across multiple backend servers, and the backend will call web service to access the data saved in the database to facilitate seamless data sharing and access. Additionally, file synchronization mechanisms are implemented to maintain consistency across scripts used in the backend system. The study conducts performance evaluations and benchmarks to assess the impact of HAProxy load balancing and file synchronization on AIS responsiveness and reliability. The results reveal significant system scalability and fault tolerance improvements, reducing downtime and enhancing user experience. This research contributes to optimizing academic information systems, enhancing their ability to handle increased loads, and ensuring the efficient delivery of educational services.

Keywords: HAProxy, Load Balancing, File Synchronization, Web Service, Academic Information Systems

Abstrak

Pengelolaan sistem informasi akademik (SIA) secara efisien sangat penting bagi institusi pendidikan untuk memberikan layanan yang dapat diandalkan kepada mahasiswa dan dosen. Penelitian ini mengeksplorasi integrasi teknik *load balancing* HAProxy dan sinkronisasi *file* untuk mengoptimalkan kinerja SIA. HAProxy digunakan untuk mendistribusikan permintaan masuk ke beberapa server *backend*, dan *backend* akan memanggil *web service* untuk mengakses data yang disimpan dalam *database* untuk memfasilitasi berbagi dan akses data tanpa hambatan. Selain itu, mekanisme sinkronisasi *file* diterapkan untuk menjaga konsistensi antar skrip yang digunakan dalam sistem *backend*. Studi ini melakukan evaluasi kinerja dan tolok ukur untuk menilai dampak penyeimbangan beban HAProxy dan sinkronisasi file terhadap respons dan keandalan SIA. Hasilnya menunjukkan peningkatan signifikan dalam skalabilitas sistem dan toleransi kesalahan, mengurangi *downtime* dan meningkatkan pengalaman pengguna. Penelitian ini berkontribusi pada optimalisasi sistem informasi akademik, meningkatkan kemampuan mereka untuk menangani peningkatan beban, dan memastikan penyampaian layanan pendidikan yang efisien.

Kata Kunci: HAProxy, *Load Balancing,* Sinkronisasi *File*, *Web Service*, Sistem Informasi Akademik

1. PENDAHULUAN

Kebutuhan sebuah infrastruktur teknologi yang handal seperti server merupakan salah satu permasalahan yang sering dihadapi oleh instansi pemerintahan maupun swasta dalam melakukan pengelolaan data. Pengelolaan data tersebut dapat terjadi ribuan bahkan jutaan kali dalam setiap harinya. Server merupakan sebuah sistem perangkat *compute* yang menyediakan layanan-layanan tertentu seperti sistem operasi, program aplikasi, maupun data informasi kepada komputer lain yang saling terhubung dalam sebuah jaringan komputer (Harefa et al.,

2021). Pada dunia pendidikan seperti universitas, aplikasi akademik yang terinstal di server merupakan sebuah aplikasi yang banyak digunakan untuk memenuhi kegiatan administrasi dalam dunia pendidikan. Meningkatnya aktifitas kinerja pada saat pemilihan mata kuliah yang akan diambil di setiap awal semester membuat peningkatan permintaan data yang terjadi di server akademik sehingga selalu mengalami kelebihan kapasitas (*overload*).

Merealisasikan kebutuhan infrastruktur yang handal untuk dapat mengimplementasikan aplikasi berbasis web perlu adanya penyesuaian terlebih dahulu. Sebagai contoh pengimplementasian sebuah aplikasi berbasis web harus membutuhkan konfigurasi server yang handal dengan jaringan yang sudah terintegrasi. UPT. Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) sebagai unit pelaksana teknis yang mengelola semua infrastruktur dan sistem informasi termasuk sistem informasi akademik di UIN Sunan Kalijaga Yogyakarta mempunyai kewajiban untuk mengatur, mengelola dan melakukan manajemen di bidang sistem informasi akan dapat digunakan secara lancar oleh Civitas academica baik di dalam lingkungan kampus maupun di luar lingkungan kampus (internet). Ada waktu-waktu tertentu di mana kerja aplikasi begitu tinggi, yang akhirnya mengakibatkan sever menjadi *overload*, sebagai contoh saat ada proses KRS. Pada saat terjadinya *overload* ini yang dilakukan adalah meningkatkan kemampuan server dalam melayani permintaan dari *user* serta dengan mengatur ulang penjadwalan akses *user* untuk melakukan *request* ke aplikasi.

Implementasi aplikasi berbasis web membutuhkan suatu konfigurasi server yang handal dan dapat mengantisipasi kebutuhan masa yang akan datang (Rijayana, 2005). Beberapa solusi yang bisa diterapkan dalam mengatasi kejadian *overload* server ketika melayani transaksi permintaan data dari server ke *client* yakni dengan melakukan penambahan unit server baru serta menerapkan metode *clustering* (Rahmatulloh & MSN, 2017). Terdapat dua fungsi yang diterapkan pada metode *clustering* yaitu *failover cluster* dan *load balancing cluster*. *Failover* adalah situasi aplikasi melakukan *restart* aplikasi di *node* yang berbeda ketika yang seharusnya bisa melayani *request* gagal untuk memberikan *response* (M. López et al., 2012). Server berfungsi sebagai metode *failover* jika ada *node cluster* yang mengalami kerusakan maka akan digantikan dengan server yang lain, sehingga permintaan layanan tidak mengalami gangguan, sedangkan metode *load balancing* berfungsi ketika server menangani permintaan data maka akan terbagi secara merata ke semua *node* sehingga tidak mengalami kelebihan kapasitas (*overload*) dalam pemintaan data (Harefa et al., 2021).

Beban yang diterima oleh server bisa dibagi ke beberapa server untuk menghindari *bottleneck,* cara ini disebut dengan *load balancing. Load balancing* mengacu kepada sebuah kondisi di mana beban yang diterima oleh tiap server dalam kondisi sama (R. M. Zebari & O. Yaseen, 2011). Hal yang pertama dilakukan pada proses *load balancing* adalah membagi beban yang datang ke server kemudian melakukan pengecekan terhadap bentuk distribusi beban yang akan dijalankan. Apabila proses pengiriman beban telah dilakukan, maka beban yang datang ke server telah terbagi secara merata (Adil Yazdeen et al., 2021; Bathiya et al., 2016; Shukla & Kumar, 2018). Pemilihan metode yang sesuai dalam proses *load balancing* diperlukan untuk meningkatkan kemampuan distribusi kepada web server. Untuk mendapatkan hasil yang terbaik dalam membagi beban, *request* yang datang dari pengguna perlu dibagi melalui jalur DNS di antara web server yang ada di dalam *cluster* berdasar strategi yang diterapkan untuk melayani *request* yang dilakukan oleh pengguna. Selanjutnya, beban yang diterima oleh web server perlu dibagi kepada web server lain untuk meningkatkan kemampuan cluster dan memaksimalkan penggunaan sumber daya yang dimiliki oleh server (Abdulmohsin, 2016; Amanuel & Ameen, 2021).

HAProxy merupakan solusi yang murah, cepat, dan merupakan solusi handal yang menawarkan *high availability, failover,* dan skema *load balancing,* yang juga bisa digunakan sebagai *proxy* untuk aplikasi yang menggunakan TCP dan HTTP di dalam aplikasinya (Kaushal & Bala, 2011). HAProxy bisa digunakan untuk membagi beban *request* yang bisa dijalankan di sistem operasi Linux, Solaris, dan FreeBSD (Ahmad et al., 2021). HAProxy juga sudah umum untuk bisa digunakan dalam membagi beban kerja dari beberapa server seperti *web server*,

database server, smtp server, dan yang lainnya. Salah satu algoritma yang digunakan di HAProxy adalah Round Robin. Round Robin merupakan algoritma yang banyak digunakan di bidang *computer science*. Hal penting dari algoritma Round Robin ini adalah karena dia mudah untuk diimplementasikan dan mudah untuk dimengerti (Kumari, 2016). Diibaratkan terdapat dua server yang menunggu *request* yang masuk dari pengguna kepada sebuah server *load balancer.* Misalkan ada sebuah *request* masuk, maka berikutnya *load balancer* akan meneruskan *request* yang masuk kepada server yang pertama. Ketika ada *request* kedua masuk kepada *load balancer* maka *request* tersebut akan diteruskan kepada server yang kedua (Khiyaita et al., 2012).

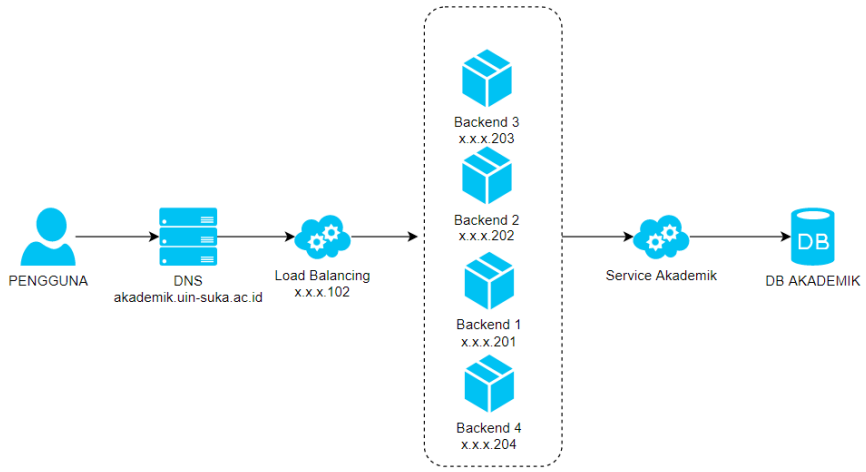
Server yang berada di belakang *load balancing* akan meneruskan setiap permintaan data menuju ke satu alamat *web service*. Menurut Chen et al. (2005), arsitektur *web service* mampu memodelkan interaksi antara tiga peran yang dimiliki yaitu peran penyedia layanan, peran konsumen layanan, dan peran pendaftar layanan. Dalam penggunaan *web service* ada beberapa *protocol* yang digunakan untuk melakukan proses transfer data, yakni Simple Object Access Protocol (SOAP) dan Representational State Transfer (REST) (Fauziah et al., 2022). Menurut penelitian Memeti et al. (2018) REST lebih baik dibandingkan dengan SOAP, karena REST bisa mendukung pertukaran data dengan format JSON maupun XML.

Tujuan dari penelitian ini adalah menerapkan fitur *load balancing* dengan menggunakan HAProxy pada *website* untuk membagi *request* yang masuk sehingga beban yang diterima *website* bisa dibagi ke beberapa server, serta untuk menghindari downtime layanan yang ada di *website* jika ada *node* server yang tidak bisa melayani *request* yang masuk.

2. METODE PENELITIAN

2.1 Arsitektur Yang Akan Diterapkan

Arsitektur yang akan diterapkan pada server sistem informasi akademik dengan menggunakan metode *load balancing* setidaknya terdapat beberapa bagian, di antaranya: DNS server, *load balancing* (HAProxy), server *backend*, *service* akademik, dan *database*. Dalam arsitektur HAProxy terdiri empat server yang berfungsi sebagai server *backend* dan satu server yang difungsikan sebagai server *load balancing* dalam mendistribusikan beban kinerja yang dilakukan oleh *end user*, sehingga koneksi yang didistribusikan merata ke masing-masing server *backend*. Penambahan server yang dijadikan sebagai *backend* bisa meningkatkan performa dari aplikasi, karena *request* yang datang ke server bisa dibagi ke beberapa server yang ada di dalam *cluster* (Data et al., 2019). Berikut ini tampilan dari alur arsitektur server *load balancing* seperti terlihat pada Gambar 1.



Gambar 1 Arsitektur Sistem

2.2 Kebutuhan *Software* dan *Hardware*

Sistem operasi yang digunakan untuk menjalankan *load balancing* berupa HAProxy, *backend,* dan *web service* yaitu Ubuntu 20.04.1 LTS, CPU 25 core, dengan menggunakan model Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz. Sedangkan untuk menjalankan *database* digunakan Windows Server 2012 R2, CPU 16 core, dengan seri Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz.

2.3 Konfigurasi HAProxy

Tahapan dalam menggunakan HAProxy sebagai *load balancer* dimulai dengan melakukan instalasi HAProxy di server yang akan digunakan sebagai *load balancer*. Perintah dimulai dengan menambahkan *repository* HAProxy, kemudian dilanjutkan dengan proses instalasi. Konfigurasi HAProxy pada server terlihat pada perintah pada Gambar 2.

```
sudo add-apt-repository ppa:vbernat/HAProxy-1.8
sudo apt-get update
sudo apt-get install HAProxy
```

```
global
        log /dev/log local0
        log /dev/log local1 notice
        chroot /var/lib/HAProxy
        stats socket /run/HAProxy/admin.sock mode 660 level admin expose-fd listeners
        user HAProxy
        group HAProxy
        daemon

frontend Local_Server
        mode http
        bind x.x.x.102:80
        bind x.x.x.102:443 ssl crt /etc/sertifikat2022/uin2022.pem
        http-request redirect scheme https unless { ssl_fc }
        default_backend nodes
        maxconn 4000

backend nodes
        balance roundrobin
        cookie SERVERID insert indirect nocache
        server s1 x.x.x.201:80 check cookie s1
        server s2 x.x.x.202:80 check cookie s2
        server s3 x.x.x.203:80 check cookie s3
        server s4 x.x.x.204:80 check cookie s4
```

Gambar 2 Konfigurasi HAProxy

Gambar 2 memberikan informasi konfigurasi yang dilakukan terhadap HAProxy. Pada bagian *backend nodes* konfigurasi yang dilakukan adalah menggunakan metode Round Robin dengan menggunakan *sticky cookie* untuk membuat pengguna tetap berada pada server yang sama ketika pengguna membuka kembali aplikasi di lain waktu (P. López & Baydal, 2018). Terdapat 4 buah *backend* yang digunakan yang diberikan nama s1, s2, s3, dan s4.

2.4 Konfigurasi Lsyncd

Dalam proses pengembangan aplikasi, hanya satu server yang dituju untuk diubah atau ditambahkan, server yang dituju tersebut berada di alamat x.x.x.201. Server yang dituju tersebut akan melakukan sinkron kepada 3 server lainnya, sehingga 4 server yang digunakan

akan selalu sinkron untuk perubahan atau penambahan yang terjadi di bagian *script* aplikasi. Untuk proses sinkronisasi *file* maka digunakan Lsyncd yang bisa dimanfaatkan untuk melakukan replikasi terhadap data-data yang berada dalam direktori tertentu dalam server (Pratama et al., 2021).

Aplikasi sistem informasi akademik yang dikembangkan di UIN Sunan Kalijaga tidak menyimpan data yang diunggah oleh pengguna di server aplikasi, tetapi semua data yang diunggah oleh pengguna disimpan di server *database*. Hal itulah yang menjadi alasan bahwa yang perlu dilakukan proses sinkron adalah bagian *script* saja.

```
sudo apt-get install lsyncd -y
```

```
settings {
      logfile = "/var/log/lsyncd/lsyncd.log",
      statusFile = "/var/log/lsyncd/lsyncd.status",
      statusInterval = 20,
      nodaemon = false,
      insist = true
}

sync {
      default.rsyncssh,
      source = "/var/www/html/",
      host = "x.x.x.202",
      targetdir = "/var/www/html/"
}

sync {
      default.rsyncssh,
      source = "/var/www/html/",
      host = "x.x.x.203",
      targetdir = "/var/www/html/"
}

sync {
      default.rsyncssh,
      source = "/var/www/html/",
      host = "x.x.x.204",
      targetdir = "/var/www/html/"
}
```

Gambar 3 Konfigurasi Lsyncd

Gambar 3 merupakan konfigurasi dari program Lsyncd, program tersebut akan melakukan sinkronisasi *file* kepada server yang lain. Terdapat tiga buah server yang menjadi tujuan proses sinkronisasi yang dijelaskan seperti yang ada di Gambar 3. Konfigurasi tersebut dimasukkan ke dalam program dengan inisial *sync,* di mana di dalamnya terdapat parameter di antaranya *source* yang menandakan sumber asal, *host* merupakan lokasi dari server tujuan, dan *tergetdir* yang merupakan folder tujuan.

3. HASIL DAN PEMBAHASAN

3.1 Pengujian Waktu *Response*

Tahap pertama yang dilakukan adalah melakukan pengecekan bahwa *request* yang masuk ke aplikasi bisa terbagi dengan menggunakan algoritma Round Robin yang disediakan oleh

HAProxy. Pengujian dilakukan dengan menggunakan Apache Jmeter, yang bisa memberikan hasil yang lebih baik dibanding *tools* yang lain (Abbas et al., 2017).

Langkah yang pertama dilakukan adalah menyiapkan satu server untuk menangani di bagian *backend* aplikasi. Dari server tersebut kemudian dibandingkan waktu *response* yang dihasikan dari penambahan server yang ada. Semakin rendah waktu yang diperlukan oleh aplikasi untuk menerima *response*, maka akan semakin membuat pengguna tetap berada di aplikasi yang sama (Nah, 2004). Rekap waktu *response* ditampilkan di Tabel 1. Dilakukan beberapa percobaan *request* ke server pertama dengan jumlah *request* yang berbeda-beda. Dengan meningkatnya jumlah *request* yang dilakukan, maka waktu yang diberikan oleh server untuk melakukan *response* menjadi meningkat.

Tabel 1 Rekap Distribusi *Request* oleh HAProxy dengan Satu Server

No.	Jumlah *Request*	*Response*	Beban Server 1
1	150	1523 ms	150
2	200	1698 ms	200
3	250	1721 ms	250
4	300	2092 ms	300

Langkah yang kedua adalah menambahkan satu buah server, sehingga menjadi 2 buah server yang menangani *request* di bagian *backend* aplikasi. Rekap waktu *response* dan pembagian server yang dilakukan oleh HAProxy ditampilkan pada Tabel 2. Diperlihatkan di Tabel 2 bahwa dengan menambahkan server di bagian *backend*, maka *response* yang diberikan oleh aplikasi menjadi lebih rendah dibandingkan ketika hanya menggunakan 1 server saja. Pada Tabel 2 juga diperlihatkan bahwa pembagian beban yang diterima tiap server adalah merata.

Tabel 2 Rekap Distribusi *Request* oleh HAProxy dengan Dua Server

No.	Jumlah *Request*	*Response*	Beban Server 1	Beban Server 2
1	150	1319 ms	75	75
2	200	1373 ms	100	100
3	250	1544 ms	125	125
4	300	1572 ms	150	150

Langkah yang ketiga adalah menambahkan satu buah server kembali, sehingga menjadi 3 buah server yang menangani *request* di bagian *backend* aplikasi. Rekap waktu *response* dan pembagian server yang dilakukan oleh HAProxy ditampilkan pada Tabel 3. Di mana pada tabel *response* yang diberikan aplikasi menjadi berkurang dibandingkan dengan hanya menggunakan 2 buah server dan tiap-tiap server menerima beban yang relatif merata.

Tabel 3 Rekap Distribusi *Request* oleh HAProxy dengan Tiga Server

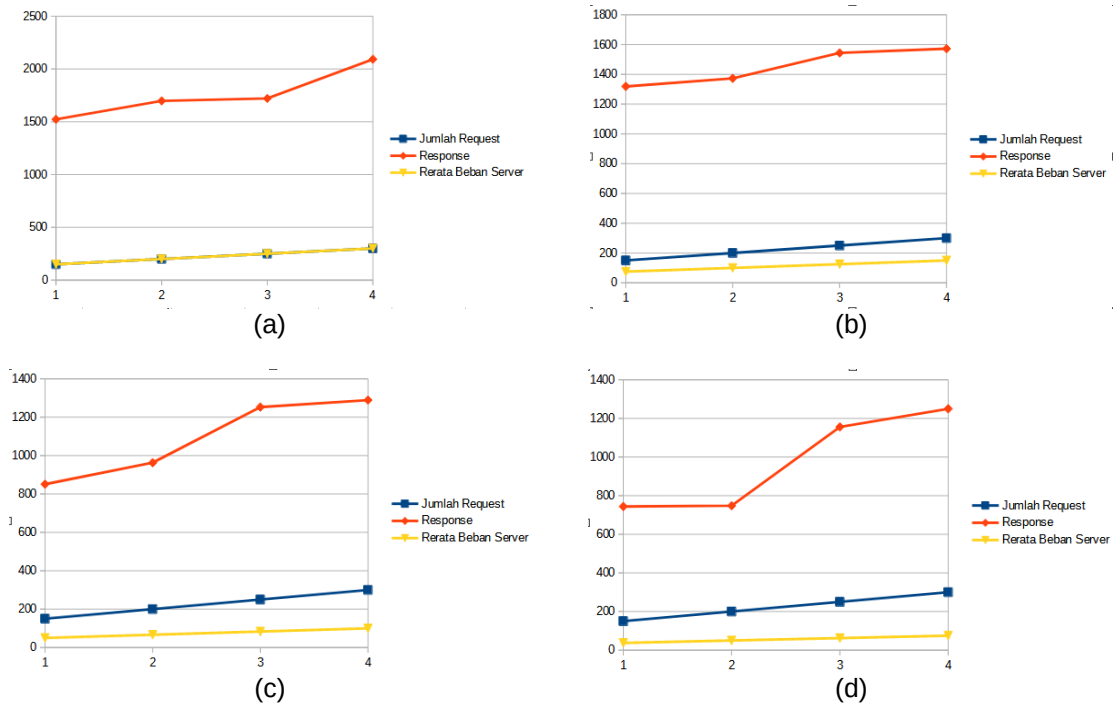
No.	Jumlah *Request*	*Response*	Beban Server 1	Beban Server 2	Beban Server 3
1	150	851 ms	50	50	50
2	200	963 ms	67	67	66
3	250	1253 ms	83	84	83
4	300	1289 ms	100	100	100

Langkah yang keempat adalah menambahkan satu buah server, sehingga menjadi 4 buah server yang menangani *request* di bagian *backend* aplikasi. Rekap waktu *response* dan pembagian server yang dilakukan oleh HAProxy ditampilkan pada Tabel 4. Di mana *response* yang diberikan oleh aplikasi menjadi berkurang dibandingkan hanya dengan menggunakan 3 buah server dan tiap-tiap server mendapatkan beban yang relatif merata.

Tabel 4 Rekap Distribusi *Request* oleh HAProxy dengan Empat Server

No.	Jumlah *Request*	*Response*	Beban Server 1	Beban Server 2	Beban Server 3	Beban Server 4
1	150	744 ms	37	37	38	38
2	200	748 ms	50	50	50	50
3	250	1156 ms	62	62	63	63
4	300	1250 ms	75	75	75	75

Dari seluruh pengujian yang dilakukan, dilakukan perbandingan *response time* yang diberikan dengan menambahkan jumlah server di bagian *backend* aplikasi. Dengan menambahkan server *backend* pada aplikasi, maka waktu yang diperlukan untuk aplikasi memberikan *response* semakin berkurang. Gambar 4 memberikan gambaran perbandingan *response time* pada seluruh pengujian *response time*.

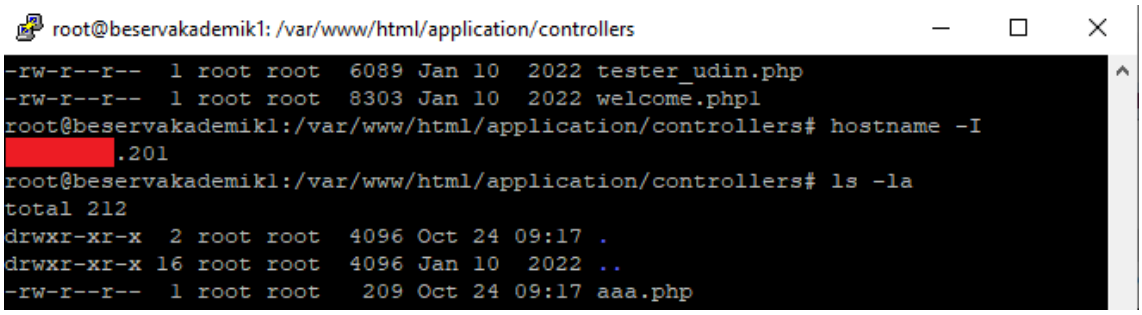


Gambar 4 Perbandingan *Response Time* pada (a) Jumlah Server 1, (b) Jumlah Server 2, (c) Jumlah Server 3, dan (d) Jumlah Server 4

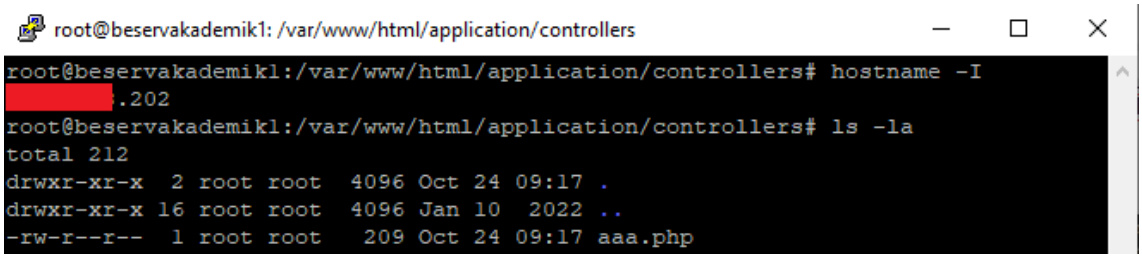
3.2 Pengujian Proses Sinkronisasi *File*

Proses pengujian berikutnya adalah memastikan bahwa proses sinkronisasi *file* yang terjadi antara server utama dalam hal ini adalah server dengan ip x.x.x.201 dengan ketiga server lainnya bisa berjalan dengan baik. Proses pengujian seperti yang tampak di Gambar 5 dilakukan dengan membuat sebuah *file* dengan nama ‘aaa.php’ dan diisikan *script* untuk melakukan pengecekan server yang sedang berjalan. Proses sinkronisasi dianggap berhasil apabila di ketiga server yang lain, yakni server x.x.x.202, x.x.x.203, dan server x.x.x.204 terbentuk *file* dan dengan isi yang sama.

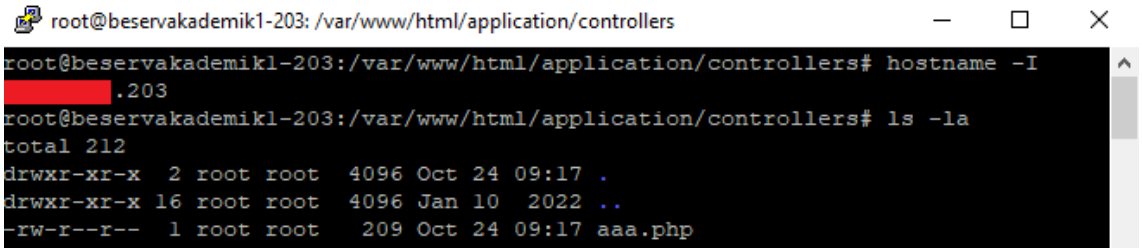
Hasil menunjukkan pada Gambar **Error! Reference source not found.**, 7, dan 8 bahwa secara otomatis akan tercipta *file* ‘aaa.php’ di tiga server yang lain dengan isi yang sama dengan *file* yang berada di server x.x.x.201. Selanjutnya pada Gambar 9 dan 10 menunjukkan bahwa apabila *script* dijalankan, maka akan memberikan informasi ip server yang berbeda-beda tergantung lokasi server yang menjalankan *script* tersebut.



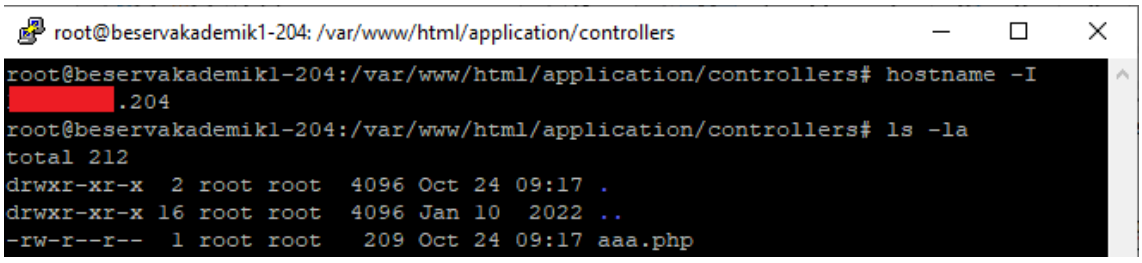
Gambar 5 Server x.x.x.201



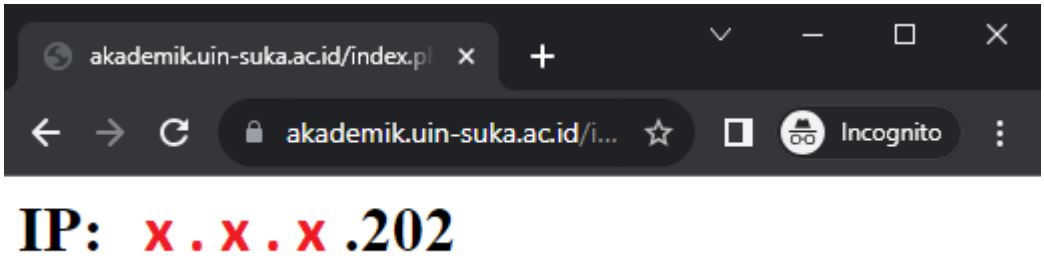
Gambar 6 Server x.x.x.202



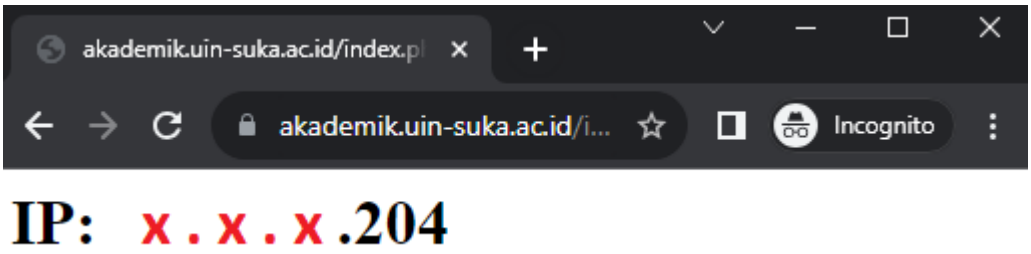
Gambar 7 Server x.x.x.203



Gambar 8 Server x.x.x.204



Gambar 9 Tampilan *Script* di Server x.x.x.202



Gambar 10 Tampilan *Script* di Server x.x.x.204

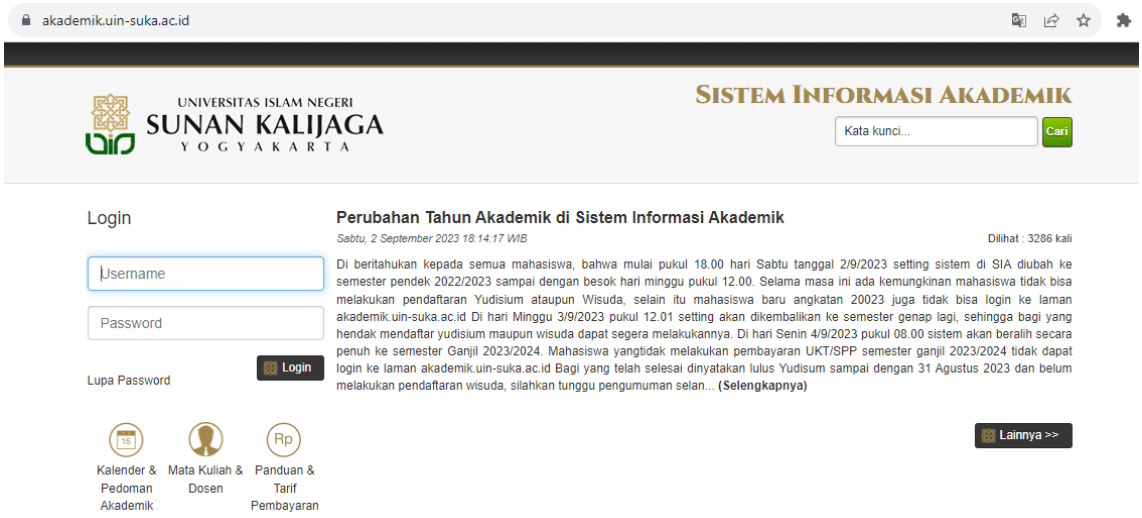
3.3 Pengujian Failover

Node server yang memiliki IP x.x.x.202 akan dimatikan untuk *apache service* yang dimiliki untuk melakukan pengujian proses *failover*. Gambar 11 menunjukkan untuk server s2 yang memiliki IP x.x.x.202 ditampilkan dengan *background* warna merah yang menunjukkan bahwa status server s2 dalam posisi mati atau tidak bisa melayani *request* yang masuk. Gambar 12 menunjukkan bahwa aplikasi tetap bisa melayani *request* yang masuk dengan mengarahkan *request* menuju ke *node* yang masih hidup dan siap melayani *request*.

nodes

	Queue			Session rate			Sessions						Bytes		Denied	
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp
s1	0	0	-	4	804		7	1 035	-	5 453 273	835 890	1s	8 206 661 006	92 412 222 938		0
s2	0	0	-	2	2 724		0	3 999	-	5 769 689	826 446	1s	7 958 436 325	87 811 585 945		0
s3	0	0	-	4	805		0	966	-	5 351 760	835 520	1s	8 236 587 137	88 207 501 403		0
s4	0	0	-	3	805		1	1 070	-	5 317 905	835 754	1s	8 230 182 167	92 419 079 469		0
Backend	0	0		11	2 623		8	3 999	400	21 824 143	3 333 610	1s	32 631 917 954	360 850 402 131	0	0

Gambar 11 *Statistic* Server *Backend* yang *Running*



Gambar 12 Tampilan Aplikasi Akademik

4. KESIMPULAN

Hasil penelitian menunjukkan bahwa dengan menggunakan HAProxy dan penambahan server *backend* dapat mengoptimalisasi kerja aplikasi sistem informasi akademik UIN Sunan Kalijaga. Hal ini ditunjukkan dengan ketika ada *request* yang datang kepada suatu aplikasi, maka beban aplikasi bisa didistribusikan secara merata kepada setiap *backend* yang dimiliki oleh aplikasi serta mengurangi *response time* yang diberikan aplikasi. Penggunaan Lsyncd untuk melakukan proses sinkronisasi *file* antar server juga bekerja dengan baik pada server. Dilihat dari

keberhasilan dalam membuat *script* yang ada di tiap *backend* aplikasi menjadi sama antara satu server dengan server yang lain.

DAFTAR PUSTAKA

Abbas, R., Sultan, Z., & Bhatti, S. N. (2017). Comparative analysis of automated load testing tools: Apache JMeter, Microsoft Visual Studio (TFS), LoadRunner, Siege. *2017 International Conference on Communication Technologies (ComTech)*, 39–44. https://doi.org/10.1109/COMTECH.2017.8065747

Abdulmohsin, H. A. (2016). A Load Balancing Scheme for a Server Cluster Using History Results. *Iraqi Journal of Science*, 2121–2130. https://ijs.uobaghdad.edu.iq/index.php/eijs/article/view/6946

Adil Yazdeen, A., Zeebaree, S. R. M., Mohammed Sadeeq, M., Kak, S. F., Ahmed, O. M., & Zebari, R. R. (2021). FPGA Implementations for Data Encryption and Decryption via Concurrent and Parallel Computation: A Review. *Qubahan Academic Journal*, *1*(2), 8–16. https://doi.org/10.48161/qaj.v1n2a38

Ahmad, U. A., Saputra, R. E., & Harahap, R. M. (2021). Implementasi High Availability Server Menggunakan Platform Haproxy (studi Kasus: Aplikasi Zammad Untuk Online Help Desk). *EProceedings of Engineering*, *8*(5). https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/16305/16011

Amanuel, S. V. A., & Ameen, S. Y. A. (2021). Device-to-Device Communication for 5G Security : A Review. *Journal of Information Technology and Informatics*, *1*(1), 26–31. https://www.qabasjournals.com/index.php/jiti/article/view/27

Bathiya, B., Srivastava, S., & Mishra, B. (2016). Air pollution monitoring using wireless sensor network. *2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, 112–117. https://doi.org/10.1109/WIECON-ECE.2016.8009098

Chen, M., Zhang, D., & Zhou, L. (2005). Providing web services to mobile users: the architecture design of an m-service portal. *International Journal of Mobile Communications*, *3*(1), 1. https://doi.org/10.1504/IJMC.2005.005870

Data, M., Kartikasari, D. P., & Bhawiyuga, A. (2019). The Design of High Availability Dynamic Web Server Cluster. *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, 181–186. https://doi.org/10.1109/SIET48054.2019.8986069

Fauziah, A., Noer, S., Junaedi, J., & Riyandi, M. A. (2022). Sistem Penjualan Sayur Menggunakan Framework Laravel. *Jurnal RESTIKOM : Riset Teknik Informatika Dan Komputer*, *3*(1), 42–50. https://doi.org/10.52005/restikom.v3i1.80

Harefa, H. S., Triyono, J., & Raharjo, S. (2021). Implementasi Load Balancing Web Server untuk Optimalisasi Kinerja Web Server dan Database Server. *Jurnal Jarkom*, *9*(1), 10–20. https://ejournal.akprind.ac.id/index.php/jarkom/article/view/3670

Kaushal, V., & Bala, A. (2011). Autonomic Fault Tolerance Using HAProxy in Cloud Enviorment. *(IJAEST) International Journal of Advanced Engineering Sciences and Technologies*, *7*(2), 222–227. https://www.semanticscholar.org/paper/Autonomic-Fault-Tolerance-Using-HAProxy-in-Cloud-Kaushal-Bala/84c5afc5d807fd76ddf7b7ae27e3e44887f5cfe6

Khiyaita, A., Bakkali, H. El, Zbakh, M., & Kettani, D. El. (2012). Load balancing cloud computing: State of art. *2012 National Days of Network Security and Systems*, 106–109. https://doi.org/10.1109/JNS2.2012.6249253

Kumari, P. (2016). A Round-Robin based Load balancing approach for Scalable demands and maximized Resource availability. *International Journal Of Engineering And Computer Science*, *5*(8), 17375–17380. https://doi.org/10.18535/ijecs/v5i8.04

López, M., Castro, L. M., & Cabrero, D. (2012). Failover and takeover contingency mechanisms for network partition and node failure. *Proceedings of the Eleventh ACM SIGPLAN Workshop on Erlang Workshop*, 51–60. https://doi.org/10.1145/2364489.2364498

López, P., & Baydal, E. (2018). Teaching high-performance service in a cluster computing course. *Journal of Parallel and Distributed Computing*, *117*, 138–147. https://doi.org/10.1016/j.jpdc.2018.02.027

Memeti, A., Imeri, F., & Cico, B. (2018). *REST vs. SOAP: Choosing the best web service while developing in-house web applications* (Vol. 2).

Nah, F. F.-H. (2004). A study on tolerable waiting time: how long are Web users willing to wait? *Behaviour & Information Technology*, *23*(3), 153–163. https://doi.org/10.1080/01449290410001669914

Pratama, K. A., Subagio, R. T., Hatta, M., & Asih, V. (2021). Implementasi Laod Balancing pada Web Server Menggunakan Apache dengan Server Mirror Data Secara Real Time. *Jurnal Digit*, *11*(2), 178. https://doi.org/10.51920/jd.v11i2.203

R. M. Zebari, S., & O. Yaseen, N. (2011). Effects of Parallel Processing Implementation on Balanced Load-Division Depending on Distributed Memory Systems. *Journal of University of Anbar for Pure Science*, *5*(3), 50–56. https://doi.org/10.37652/juaps.2011.44313

Rahmatulloh, A., & MSN, F. (2017). Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *3*(2), 241–248. https://doi.org/10.25077/TEKNOSI.v3i2.2017.241-248

Rijayana, I. (2005). Teknologi Load Balancing untuk Mengatasi Beban Server. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*. https://journal.uii.ac.id/Snati/article/view/1385

Shukla, P., & Kumar, A. (2018). CLUE Based Load Balancing in Replicated Web Server. *2018 8th International Conference on Communication Systems and Network Technologies (CSNT)*, 104–107. https://doi.org/10.1109/CSNT.2018.8820224

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 1, JANUARI, 2024, Pp. 50 – 58
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Server *Redundancy*: Performa Jaringan Mengunakan DNS *Failover* MikroTik pada Kasus *Private Server* dan *Public Server*

Tommi Alfian Armawan Sandi [(1)*], Firmansyah [(2)], Ahmad Fauzi [(3)] , Hanggoro Aji Al Kautsar [(3)]
[1,2,4] Informatika, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika, Jakarta
[3] Sistem Informasi, Fakultas Teknologi Informasi, Universitas Nusa Mandiri, Jakarta
e-mail : {tommi.taf,firmansyah.fmy,hanggoro.hgr}@ bsi.ac.id, ahmad.azy@nusamandiri.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 26 Oktober 2023, direvisi 14 Desember 2023, diterima 15 Desember 2023, dan dipublikasikan 25 Januari 2024.

Abstract

The digitization of user services has increased, in line with the need for VPS and cloud computing services, which are rampant among application and platform developers. Quite a few companies that create applications or application users have servers to handle user needs. Testing is carried out using the ICMP Protocol to get real-time results and can be measured. From Scenario 1, carrying out 20 test requests, we get a packet loss of 5% with RTT Avarage of 195,838ms and Mdev 4,103ms. If you apply DNS failover in scenario 2, the client will likely access the web a little slower, as evidenced by the packet loss being 25% greater in value. Compared to scenario 1, having a high standard deviation (Mdev) of round-trip times is not desirable. This variation is also known as jitter. Increased jitter can cause a bad user experience, especially in real-time audio and video streaming applications. However, this is still understandable because it only has a 1-5 second effect on the service. Next, in scenario 3, we can see that private and public servers have relatively high gab with 0% packet loss, which has a small Mdev value of 0.309ms. Therefore, the DNS failover method is a solution for network administrators who have problems related to server migration between public servers and private servers so that services can run even if a server is maintaining or downlinking.

Keywords: Server Redundancy, DNS Failover, RTT, Mdev, Private Server, Public Server

Abstrak

Digitalisasi layanan pengguna mengalami peningkatan, selaras dengan kebutuhan VPS dan layanan *cloud computing* yang merajalela di kalangan *development* aplikasi dan *platfrom*. Tidak sedikit dari perusahaan yang membuat aplikasi atau pengguna aplikasi memiliki server untuk mengangani kebutuhan penguna. Pengujian dilakukan dengan *protocol* ICMP untuk mendapatkan hasil yang *real-time* dan dapat diukur. Dari skenario 1 melakukan 20 kali *request* pengujian didapatkan *packet loss* 5% dengan *avarage* RTT sebesar 195,838ms dan Mdev 4,103ms, jika menerapkan *failover* DNS pada skenario 2, *client* kemungkinan mengakses web sedikit lambat terbukti dari *packet loss* yang hilang sebesar 25% lebih besar nilainya dibanding skenario 1 dan memiliki standar deviasi (Mdev) *round-trip times* yang tinggi tidaklah diinginkan. Variasi ini juga dikenal sebagai *jitter*. *Jitter* yang tinggi dapat menyebabkan pengalaman pengguna yang buruk, terutama dalam aplikasi *real-time* seperti *streaming audio* dan video. Walaupun begitu hal ini masih dalam kondisi dapat dimaklumkan karna hanya 1-5 detik pengaruhnya terhadap layanan. Selanjutnya pada skenario 3, kita dapat melihat jika perbedaan *private server* dan *public server* memiliki gap yang cukup tinggi dengan 0% *packet loss* yang tentunya memiliki nilai Mdev yang kecil 0,309ms. Oleh karena itu, jika metode *failover* DNS pastinya menjadi solusi administrator jaringan yang memiliki masalah terkait migrasi server antara *public server* dan *private server* supaya layanannya bisa berjalan walaupun ada server yang *maintenace* atau *downlink*.

Kata Kunci: *Server Redundancy,* DNS *Failover,* RTT, Mdev*, Private Server, Public Server*

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 1, JANUARI, 2024: 50 – 58

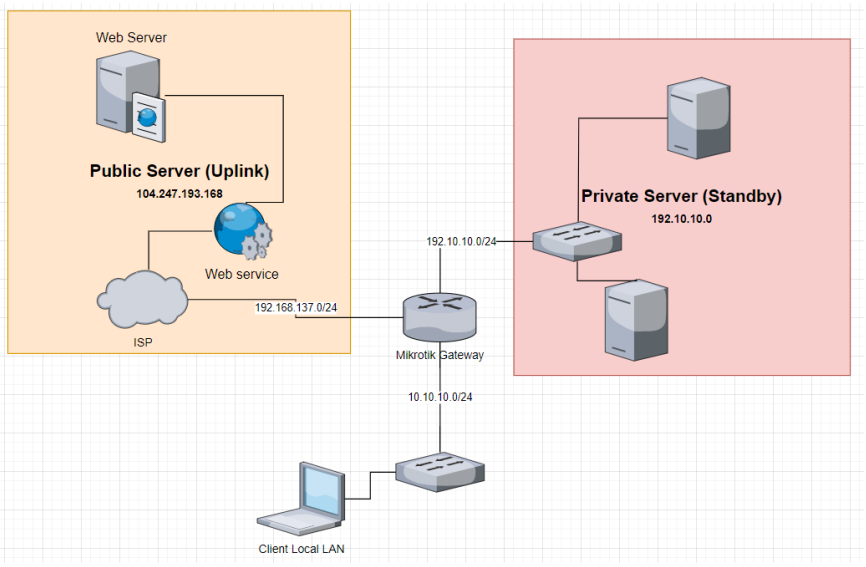
1. PENDAHULUAN

Digitalisasi layanan pengguna mengalami peningkatan, selaras dengan kebutuhan VPS dan layanan *cloud computing* (’Abidah et al., 2020) yang merajalela di kalangan *development* aplikasi dan *platfrom*. Tidak sedikit dari perusahaan yang membuat aplikasi atau pengguna aplikasi memiliki server untuk mengangani kebutuhan penguna. Terdapat 3 (tiga) model utama dalam komputasi awan yang dimanfaatkan untuk membangun sebuah *platfrom* di antaranya ada IaaS (*Infrastructure as a Service*), PaaS (*Platform as a Service*), dan SaaS (*Software as a Service*) (Nadeem, 2022). Pemanfaatan server telah meningkat secara signifikan karena virtualisasi sistem komputer, yang menyebabkan peningkatan konsumsi daya dan kebutuhan penyimpanan oleh pusat data dan dengan demikian juga meningkatkan biaya operasional (Alyas et al., 2022). Penyediaan IaaS penyedia *cloud* untuk menyediakan mesin virtual (VM) dan menggunakannya dengan cara yang mirip dengan klaster lokal (Malla & Christensen, 2020).

Fenomena migrasi server bukanlah hal yang baru dalam bidang ini (Hosseini Shirvani et al., 2020), walaupun banyaknya layanan yang diberikan oleh server ke pengguna membuat server mendapat perhatian khusus terlebih aksesbilitas dan performa (Bhardwaj & Rama Krishna, 2022). Manajemen migrasi pun tidak lagi menjadi bagian *system development* (Khan et al., 2022) namun juga bagian dari administrator jaringan. Seperti penelitian sebelumnya yang dilakukan mengenai proses migrasi *cloud computing* dari lingkungan Amazon EC2 ke VMware mengalami kendala yang terdapat di IaaS yang tidak maksimal dalam proses migrasi IaaS, konversi *image virtual machine* dari satu sistem ke sistem lain tidak menjamin keberhasilan proses migrasi (Jupriyadi et al., 2021; Wijayanto et al., 2021). Terdapat kecenderungan beberapa parameter seperti konfigurasi *central processing unit*, *random access memory*, jaringan, serta struktur aplikasi dan data yang mungkin bisa berubah dalam proses migrasi (Mafakhiri, 2019). Penelitian selanjutnya tentang implementasi *load balancing* dan *failover* pada proses migrasi *container* Docker menyebutkan ketersediaan web server yang tidak didukung dengan metode dapat menghambat kinerja dari suatu server, dikarenakan terjadinya penumpukan request pada server dapat membuat kegagalan jaringan (Ri et al., 2023; Rifiera & Nurwarsito, 2022). Pada analisis perbandingan layanan data server menggunakan *failover cluster* pada *platform* nginx dan apache menegaskan jika web server yang efektif untuk mendukung teknik *failover* serta perlu dibangun suatu *high availability web server cluster* (Marzuki et al., 2022) yang dapat menjalankan teknik *failover*, sehingga dapat meminimalisir resiko kegagalan *service* pada web server yang dapat menghambat efektifitas kerja penyedia layanan (Maila et al., 2020; Nannipieri et al., 2019). Pada saat ini dengan menjalankan *failover routing* tidak bisa memilah antara *traffic* ISP atau layanan dari server (Sandi et al., 2021). Oleh karena itu perlunya pengujian *failover* dengan DNS yang tersedia di masing-masing server untuk melakukan peralihan server tanpa mengurangi keandalan dari layanan tersebut (Dooley, n.d.; Rouse, 2013).

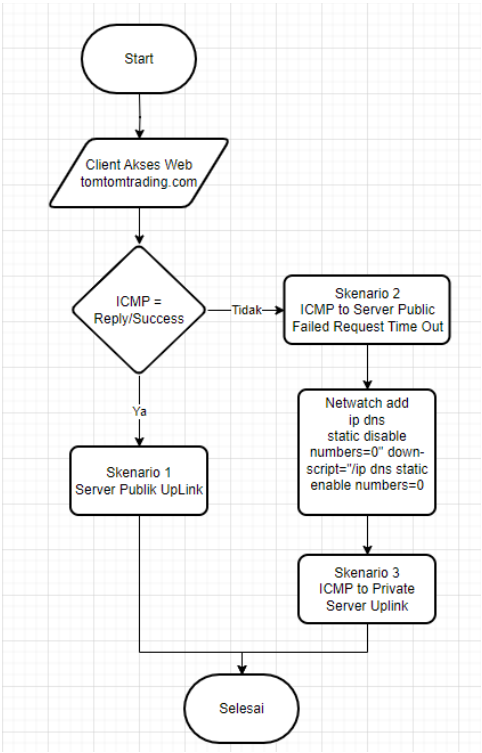
2. METODE PENELITIAN

Penelitian ini dilakukan dengan melakukan eksperimen data yang dipadukan oleh catatan observasi pada jaringan *Local Area Network* (Hae, 2021), perangkat yang digunakan dalam penelitian ini di antaranya: MikroTik dengan arsitektur x86 yang diunduh pada *virtual machine*, web server terbagi atas 2 (dua) yang terunduh pada VPS dan *private server* (*local*) pada setiap server menggunakan *operating system* Centos 7 dan terinstall CWP (Centos Web Panel/Control-WebPanel) *latest version*, serta *client* menggunakan perangkat laptop dan PC. Penelitian ini bersifat instrumental dan mengumpulkan semua data dengan menghasilkan data secara deskriptif dan menambilkan hasil server *redundancy* pada jaringan ketika mengalami *downlink*/*maintenance* server. Terlihat pada Gambar 1 merupakan skema jaringan yang digunakan pada penelitian server *redundancy*, performa jaringan menggunakan DNS *failover* MikroTik pada kasus *private server* dan *public server*.

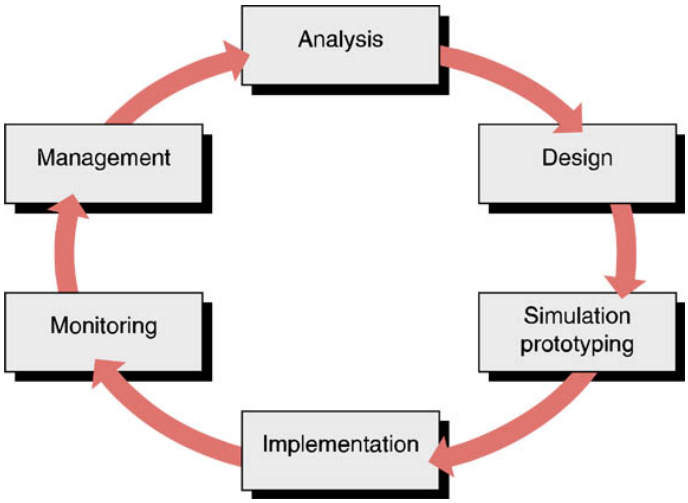


Gambar 1 Skema Jaringan

Tahapan pengujian yang akan digunakan dalam penelitian dapat dilihat pada Gambar 2. Diagram pengujian mengedepankan performa terhadap DNS *failover* terhadap kedua server tersebut. Detail pengujian di antaranya melakukan uji konektifitas terhadap *client* yang mengakses *public server* dalam keadaan normal, selanjutnya pengujian DNS *failover* dalam keadaan *public server*. Pada kondisi *downlink* kemudian dilakukan pengujian terakhir dengan *client* mengakses *private server* dalam keadaan normal. Pada implementasi *failover* ini berbeda dengan *failover* pada *routing*, parameter yang digunakan adalah *check gateway* dan ICMP. Dengan menggunakan ICMP dapat mengetahui secara *real-time* dan *responsive* terhadap konektivitas pada suatu perangkat (Ďulík et al., 2023).



Gambar 2 Diagram Alur Pengujian



Gambar 3 SPDLC

Dapat dilihat pada Gambar 3 merupakan metode penelitian yang digunakan dalam penelitian ini. Penelitian menggunakan metode *The Security Development Life Cycle (SPDLC)* yang memiliki 6 tahapan yaitu analisis, *design*, simulasi, implementasi, *monitoring*, dan manajemen (W & Fitriana, 2021).

1) Analisis yang diterapkan dengan mengidentifikasikan kebutuhan penguna dan permasalahan yang dihadapi sebelum melakukan tahapan berikutnya
2) *Design* dengan melakukan *re-map* (topologi) *Local Area Network* yang ada di lapangan dan disesuaikan dengan kebutuhan penguna.
3) Simulasi yang dilakukan pada penelitian ini dengan mengumpulkan kofigurasi-konfigurasi, mulai dari konfigurasi MikroTik maupun server dan dijalankan menggunakan *virtual box*.
4) Implementasi merupakan tahapan di mana penulis melakukan penerapan pada perangkat *rill* yang disesuaikan pada kebutuhan penguna, baik itu instalasi di sisi server maupun *router*.
5) *Monitoring* pada penelitian ini yaitu dengan melihat *log activity* pada MikroTik apakah jaringan telah berjalan dengan semestinya dan *failover* DNS berjalan dengan baik.
6) Manajemen melakukan evaluasi terhadap layanan jaringan yang telah diterapkan sesuai dengan tujuan awal penelitian.

3. HASIL DAN PEMBAHASAN

Skenario pengujian server *redundancy* performa jaringan menggunakan DNS *failover* MikroTik pada kasus *private server* dan *public server* dilakukan dengan cara pembuktian uji konektifitas jaringan kantor yang normal dengan konfigurasi dasar *gateway* sampai DNS yang sesuai dengan ISP yang diberikan. Skenario 2 yaitu menghentikan layanan dari *public server* dan melihat dari *client* sebelum dan setelah menggunakan *failover* ke *private server*. Skenario 3 adalah melakukan redundansi dari *private server* ke *public server* dan mengetahui *quality of service* terhadap layanan-layanan tersebut. Untuk mengimplenetasikan layanan jaringan menggunakan DNS *failover* pada MikroTik peneliti menggunakan spesifikasi IP *address* pada Tabel 1.

Tabel 1 IP *Address*

Perangkat	Interface	IP Address	Gateway
Router 1	Ether 1	192.168.137.2/24	192.168.137.1
	Ether 2	192.10.10.1/24	
	Ether 3	10.10.10.1/24	
Public server	NIC	104.247.193.168	
Private server	NIC	192.10.10.2	192.10.10.1
PC *Client* (1…24)	NIC	10.10.10.xxx/24	

3.1 Implementasi Server *Redundancy* dengan DNS *Failover*

Pada tahapan ini setiap alamat jaringan pada server harus diketahui dan diperlukan IP *static*. Hal ini dilakukan untuk mempermudah *router* mencari *route* mana yang harus dituju dengan memanfaatkan IP tersebut. Pada impementasi *failover* ini berbeda dengan *failover* pada *routing*, parameter yang digunakan adalah *check gateway* dan ICMP. Berdasarkan topologi yang diimplementasi terdapat beberapa *client* dan server yang harus mendapatkan layanan dari *router* yang sebagai *gateway* onfigurasi. *Basic config* yang harus dilakukan di antaranya IP *address*, IP *route*/*gateway*, IP server DNS, dan *firewall* NAT.

3.1.1 Konfigurasi IP *Address*

Konfigurasi IP *address* yang digunakan dalam penelitian ini yaitu seperti pada Gambar 4. Pada konfigurasi tersebut menambahkan IP *Address* di masing-masing *interface*, dengan melihat tabel IP yang telah ditetapkan.

```
[admin@MikroTik] >ip address add address=192.168.137.2/24 interface=ether1
[admin@MikroTik] >ip address add address=192.10.10.1/24 interface=ether2
[admin@MikroTik] >ip address add address=10.10.10.1/24 interface=ether3
[admin@MikroTik] > ip address print
Columns: ADDRESS, NETWORK, INTERFACE
#     ADDRESS          NETWORK       INTERFACE
0     192.168.137.2/24 192.168.137.0 ether1
1     192.10.10.1/24   192.10.10.0   ether2
2     10.10.10.1/24    10.10.10.0    ether3
```

Gambar 4 Konfigurasi IP *Address*

3.1.2 Konfigurasi DNS Server

Pada konfigurasi DNS masih menunggu semua IP server yang digunakan *public server* meliliki IP 104.247.193.168 dan *private server* 192.10.10.2 dimasukkan dalam DNS *list* dengan konfigurasi seperti pada Gambar 5. Selain itu, perlu ditambahkan IP server pada DNS *static*, Hasil dari DNS server bisa menentukan ke mana *client* dapat mendapatkan layanannya. Konfigurasi yang dilakukan seperti pada Gambar 6. Pada Gambar 7, Menunjukkan bahwa ada 2 DNS *static* yang berhasil di input dengan *private server* dan *public server*.

```
[admin@MikroTik] > ip dns set servers=8.8.8.8 allow-remote-request=yes
```

Gambar 5 Konfigurasi DNS Server

```
[admin@MikroTik] > ip dns static add address=192.10.10.2 name=tomtomtrading.com type=A ttl=1d
[admin@MikroTik] > ip dns static add address=104.247.193.168 name=tomtomtrading.com type=A ttl=1d
```

Gambar 6 Konfigurasi DNS *Static*

```
[admin@MikroTik] > ip dns static/print
Flags: X - DISABLED
Columns: NAME, ADDRESS, TTL
#   NAME                ADDRESS          TTL
;;; ServerPrivate
0 X tomtomtrading.com   192.10.10.2      1d
;;; Server Public
1   tomtomtrading.com   104.247.193.168  1d
[admin@MikroTik] >
```

Gambar 7 DNS *Static Print*

3.1.3 Konfigurasi *Route*

Konfigurasi *route* yang digunakan dalam penelitian ini yaitu seperti pada Gambar 8.

```
[admin@MikroTik] > ip route add gateway=192.168.137.1 distance=1
```

Gambar 8 Konfigurasi *Route*

3.1.4 Konfigurasi NAT

Konfigurasi NAT yang digunakan dalam penelitian ini yaitu seperti pada Gambar 9.

```
[admin@MikroTik] > ip firewall nat add chain=srcnat out-interface=ether1 action=masquerade
```

Gambar 9 Konfigurasi NAT

3.1.5 Konfigurasi Netwatch

Netwatch difungsikan untuk memonitor kondisi *host*. Hal ini dikarenakan setiap server kemungkinan tidak dapat dimonitor *real-time* oleh administrator jaringan. Dengan memanfaatkan Netwatch dapat melakukan tindakan pencegahan dan keputusan kondisi yang diperlukan. Konfigurasi Netwatch ditunjukkan pada Gambar 10. Pada Gambar 11 terdapat 2 konfigurasi yang bertugas untuk melakukan *ping reply* pada IP 104.247.193.168, jika kondisi IP tersebut *downlink* maka akan di alihkan otomatis ke *private server*.

```
[admin@MikroTik] > tool netwatch add host=104.247.193.168 type=simple interval=10m timeout=1 up-script="/ip dns static disable numbers=0" down-script="/ip dns static enable numbers=0"
[admin@MikroTik] > tool netwatch add host=104.247.193.168 type=simple interval=10m timeout=1 up-script="/ip dns static enable numbers=1" down-script="/ip dns static disable numbers=1"
```

Gambar 10 Konfigurasi Netwatch

```
[admin@MikroTik] > tool netwatch print
Flags: X - DISABLED
Columns: TYPE, HOST, TIMEOUT, INTERVAL, STATUS, SINCE
#    TYPE    HOST             TIMEOUT  INTERVAL  STATUS    SINCE
0    icmp    104.247.193.168                     down      oct/23/2023 13:23:26
1 X simple   192.50.10.2      1s       10s       unknown
[admin@MikroTik] >
```

Gambar 11 *Tool* Netwatch *Print*

3.2 Pengujian Konektifitas Skenario 1: *Client* Mengakses *Public Server* dalam Keadaan Normal

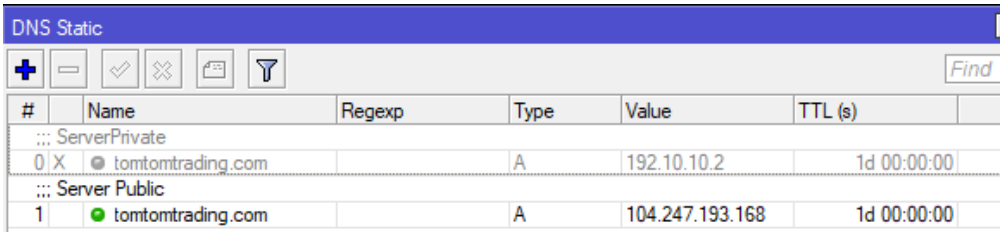
Skenario dalam uji yang pertama ialah melakukan tes terhadap layanan *public server* berupa tampilan web yang berisi informasi-informasi *trading*. *Protocol* ICMP dipilih lebih tepat digunakan dikarenakan fungsinya untuk mendiagnosis masalah komunikasi jaringan. Telihat pada Tabel 2 dan 3 perhitungan dengan melakukan 20 kali *request* layanan menggunakan *ping* pada *client*. Keadaan *public server uplink* terlihat pada Gambar 12.

Tabel 2 *Ping Statistics* Skenario 1

Packets		
Sent	*Received*	*Loss*
20	19	1 (5% *loss*)

Tabel 3 RTT (*Round Trip Times*) Skenario 1

Min	Avg	Max	Mdev
189,341 ms	195,838 ms	201,594 ms	4,103 ms



Gambar 12 *Public Server Uplink*

3.3 Pengujian Konektifitas Skenario 2: DNS *Failover* dalam Keadaan *Public Server Downlink*

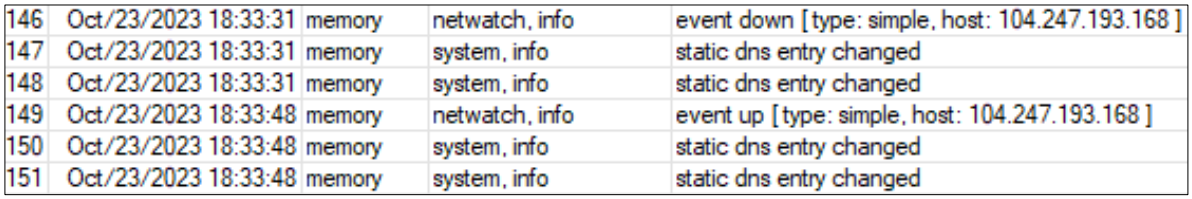
Skenario dalam uji yang kedua ialah melakukan tes terhadap layanan *public server* dengan menonaktifkan layanan atau 503 *Service Unavailable* dengan metode tes *ping* ICMP ke domain tomtomtrading.com dan dilakukan 20 kali *request* yang hasilnya terlihat pada Tabel 4 dan 5. Pada proses ini juga terjadi *failover* DNS yang terlihat pada Gambar 13. Waktu peralihan atau redundansi antara *public server* dan *private server* berlangsung 17 detik dari total 20 detik *request*. Perubahan *private server uplink* terlihat pada Gambar 14.

Tabel 4 *Ping Statistics* Skenario 2

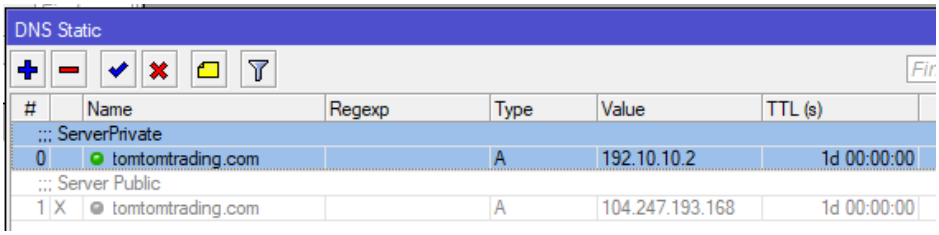
Packets		
Sent	*Received*	*Loss*
20	15	5 (25% *loss*)

Tabel 5 RTT (*Round Trip Times*) Skenario 2

Min	Avg	Max	Mdev
190,995 ms	205,844 ms	245,656 ms	13,160 ms



Gambar 13 *Failover* DNS *Success*



Gambar 14 Perubahan *Private Server Uplink*

3.4 Pengujian Konktifikas Skenario 3: Client Mengakses *Private Server* dalam Keadaan Normal

Skenario dalam uji yang terakhir ialah melakukan tes terhadap layanan *private server* yang letaknya dalam satu area yang terjangkau LAN, pengetesan menggunakan metode *ping* ICMP *private server* dengan 20 kali *request* terhadap layanan tersebut. Hasil pengujian terdapat pada Tabel 6 dan 7.

Tabel 6 *Ping Statistics* Skenario 3

Packets		
Sent	*Received*	*Loss*
20	20	0 (0% loss)

Tabel 7 RTT (*Round Trip Times*) Skenario 3

Min	Avg	Max	Mdev
2,154 ms	2,741 ms	3,100 ms	0,309ms

4. KESIMPULAN

Dengan melakukan beberapa uji coba dan eksperimen terhadap layanan pada *public server* dan *private* dapat disimpulkan dengan menggunakan *protocol* ICMP 20 kali *request* dari data skenario 1 pengujian didapatkan *packet loss* 5% dengan *avarage* RTT sebesar 195,838ms dan Mdev 4,103ms. Jika menerapkan *failover* DNS pada skenario 2, *client* kemungkinan mengakses web sedikit lambat. Terbukti dari *packet loss* yang hilang sebesar 25% lebih besar nilainya dibanding skenario 1 dan memiliki standar deviasi (Mdev) *round-trip times* yang tinggi tidaklah diinginkan. Variasi ini juga dikenal sebagai *jitter*. *Jitter* yang tinggi dapat menyebabkan pengalaman pengguna yang buruk, terutama dalam aplikasi *real-time* seperti *streaming* audio dan video. Walaupun begitu hal ini masih dalam kondisi dapat dimaklumkan karna hanya 1-5 detik pengaruhnya terhadap layanan. Selanjutnya pada skenario 3, kita dapat melihat jika perbedaan *private server* dan *public server* memiliki gap yang cukup tinggi dengan 0% *packet loss* yang tentunya memiliki nilai Mdev yang kecil 0,309ms. Oleh karena itu, jika metode *failover* DNS pastinya menjadi solusi administrator jaringan yang memiliki masalah terkait migrasi server antara *public server* dan *private server* supaya layanannya bisa berjalan walaupun ada server yang *maintenance* atau *downlink*.

DAFTAR PUSTAKA

'Abidah, I. N., Hamdani, M. A., & Amrozi, Y. (2020). Implementasi Sistem Basis Data Cloud Computing pada Sektor Pendidikan. *KELUWIH: Jurnal Sains Dan Teknologi*, *1*(2), 77–84. https://doi.org/10.24123/saintek.v1i2.2868

Alyas, T., Javed, I., Namoun, A., Tufail, A., Alshmrany, S., & Tabassum, N. (2022). Live Migration of Virtual Machines Using a Mamdani Fuzzy Inference System. *Computers, Materials & Continua*, *71*(2), 3019–3033. https://doi.org/10.32604/cmc.2022.019836

Bhardwaj, A., & Rama Krishna, C. (2022). A Container-Based Technique to Improve Virtual Machine Migration in Cloud Computing. *IETE Journal of Research*, *68*(1), 401–416. https://doi.org/10.1080/03772063.2019.1605848

Dooley, K. (n.d.). *What is Network Redundancy & Why is It Important?* Auvik. Retrieved January 25, 2024, from https://www.auvik.com/franklyit/blog/simple-network-redundancy/

Ďulík, M., Harakaľ, M., & Javurek, M. (2023). Advanced Methods for Network Infrastructure Analysis and Security. *2023 Communication and Information Technologies (KIT)*, 1–7. https://doi.org/10.1109/KIT59097.2023.10297110

Hae, Y. (2021). ANALISIS KEAMANAN JARINGAN PADA WEB DARI SERANGAN SNIFFING DENGAN METODE EKSPERIMEN. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *8*(4), 2095–2105. https://doi.org/10.35957/jatisi.v8i4.1196

Hosseini Shirvani, M., Rahmani, A. M., & Sahafi, A. (2020). A survey study on virtual machine migration and server consolidation techniques in DVFS-enabled cloud datacenter:

Taxonomy and challenges. *Journal of King Saud University - Computer and Information Sciences*, *32*(3), 267–286. https://doi.org/10.1016/j.jksuci.2018.07.001

Jupriyadi, J., Hijriyanto, B., & Ulum, F. (2021). Komparasi Mod Evasive dan DDoS Deflate Untuk Mitigasi Serangan Slow Post. *Techno.Com*, *20*(1), 59–68. https://doi.org/10.33633/tc.v20i1.4116

Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2022). Systematic Literature Review on Security Risks and its Practices in Secure Software Development. *IEEE Access*, *10*, 5456–5481. https://doi.org/10.1109/ACCESS.2022.3140181

Mafakhiri, J. (2019). Proses Migrasi Cloud Computing Dari Lingkungan Amazon Ec2 Ke Vmware. *Jurnal Bangkit Indonesia*, *8*(1), 1. https://doi.org/10.52771/bangkitindonesia.v8i1.85

Maila, H. H., Indra, D., & Satra, R. (2020). Analisis Perbandingan Layanan Data Server Menggunakan Failover Cluster pada Platform Nginx dan Apache. *Buletin Sistem Informasi Dan Teknologi Islam*, *1*(2), 87–91. https://doi.org/10.33096/busiti.v1i2.829

Malla, S., & Christensen, K. (2020). HPC in the cloud: Performance comparison of function as a service (FaaS) vs infrastructure as a service (IaaS). *Internet Technology Letters*, *3*(1), e137. https://doi.org/10.1002/itl2.137

Marzuki, K., Hanif, N., & Hariyadi, I. P. (2022). Application of Domain Keys Identified Mail, Sender Policy Framework, Anti-Spam, and Anti-Virus: The Analysis on Mail Servers. *International Journal of Electronics and Communications Systems*, *2*(2), 65–73. https://doi.org/10.24042/ijecs.v2i2.13543

Nadeem, F. (2022). Evaluating and Ranking Cloud IaaS, PaaS and SaaS Models Based on Functional and Non-Functional Key Performance Indicators. *IEEE Access*, *10*, 63245–63257. https://doi.org/10.1109/ACCESS.2022.3182688

Nannipieri, L., Cacciaguerra, S., Mirenna, S., Locati, M., Marletta, M., & Gucciardi, E. (2019). Making Linked Data more reliable with a failover server system: a case study with seismological data at INGV. *Annals of Geophysics*, *62*(Vol 62 (2019)), DM567. https://doi.org/10.4401/ag-8050

Ri, O.-C., Kim, Y.-J., & Jong, Y.-J. (2023). *Hybrid load balancing method with failover capability in server cluster using SDN*. http://arxiv.org/abs/2307.05552

Rifiera, S. N., & Nurwarsito, H. (2022). Implementasi Load Balancing dan Failover pada Proses Migrasi Container Docker. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *6*(5), 2025–2033. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/10967

Rouse, M. (2013, April 15). *Network Redundancy*. Techopedia. https://www.techopedia.com/definition/29305/network-redundancy

Sandi, T. A. A., Heristian, S., & Leksono, I. N. (2021). OPTIMALISASI FAILOVER DENGAN NETWATCH PADA MIKROTIK. *CONTEN : Computer and Network Technology*, *1*(1), 23–30. https://doi.org/10.31294/conten.v1i1.388

Wijayanto, D., Firdonsyah, A., Adhinata, F. D., & Jayadi, A. (2021). Rancang Bangung Private Server Menggunakan Platform Proxmox dengan Studi Kasus: PT.MKNT. *Journal ICTEE*, *2*(2), 41. https://doi.org/10.33365/jictee.v2i2.1333

W, Y., & Fitriana, Y. B. (2021). Analisis Network Security Komputer Tingkat Desa Menggunakan Metode Security Policy Development Life Cycle (SPDLC). *Jurnal Teknik Juara Aktif Global Optimis*, *1*(2), 11–21. https://doi.org/10.53620/jtg.v1i2.28

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 1, JANUARI, 2024, Pp. 59 – 69
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Pemeringkatan Kinerja Dosen pada Perguruan Tinggi Swasta Menggunakan Algoritma *Simple Additive Weighting*

Elly Mufida [(1)*], Nandang Iriadi [(2)], Doni Andriansyah [(3)]
[1] Informatika, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika, Jakarta
[2] Teknologi Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika, Jakarta
[3] Informatika, Fakultas Teknologi Informasi, Universitas Nusa Mandiri, Jakarta
e-mail : {elly.elm,nandang.ndi}@bsi.ac.id, doni.dad@nusamandiri.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 10 Oktober 2023, direvisi 13 Desember 2023, diterima 13 Desember 2023, dan dipublikasikan 25 Januari 2024.

Abstract

The Tri Dharma of Higher Education is an obligation for lecturers while carrying out their duties as lecturers at higher education institutions, the implementation of which is regulated in Law Number 20 of 2003 concerning the National Education System. The Tri Dharma of Higher Education is a lecturer's obligation, including Education and Teaching, Research and Development, and Community Service. Lecturers need Support and motivation to implement quality Tri Dharma, especially at "X" Private Universities. Providing rewards or awards can motivate lecturers to give their best performance to Tri Dharma. Student feedback is also needed as evaluation material for lecturers to measure their teaching abilities. Rewarding lecturers can be done by ranking lecturer performance, especially at private universities. The SAW (Simple Additive Weighting) algorithm ranks lecturer performance through the criteria of education and teaching, research, community service, and student feedback. An assessment of several subcriteria presents each criterion. The normalized scoring matrix is the ranking preference. From the results of data processing on lecturer performance and feedback from students, with a sample of 25 lecturers, a ranking score was obtained on a scale of 0 to 1, where a score of 1 is the highest ranking. The lecturer performance ranking process involves lecturers, students, and the Study Program Management Unit. A lecturer performance rating information system is needed to facilitate all actors' involvement in the lecturer performance rating process and provide valid and timely rating results to stakeholders.

Keywords: Algorithms, Simple Additive Weighting, SAW, Tri Dharma of Higher Education, Student Feedback

Abstrak

Tri Dharma Perguruan Tinggi adalah kewajiban bagi dosen selama menjalankan tugasnya sebagai dosen pada perguruan tinggi, yang pelaksanaannya diatur dalam UU Nomor 20 tahun 2003 mengenai Sistem Pendidikan Nasional. Tri Dharma Perguruan tinggi merupakan kewajiban dosen, meliputi Pendidikan dan Pengajaran, Penelitian dan Pengembangan, serta Pengabdian Kepada Masyarakat. Diperlukan dukungan dan motivasi kepada dosen dalam melaksanakan Tri Dharma yang berkualitas, khususnya pada Perguruan Tinggi Swasta “X”. Pemberian *reward* atau penghargaan dapat memotivasi dosen untuk memberikan kinerja terbaiknya pada Tri Dharma. Selain itu umpan balik dari mahasiswa juga diperlukan sebagai bahan evaluasi bagi dosen untuk mengukur kemampuan mengajarnya. Pemberian *reward* kepada dosen dapat dilakukan melalui pemeringkatan kinerja dosen, khususnya pada perguruan tinggi swasta. Algoritma SAW (*Simple Additive Weighting*) digunakan pada pemeringkatan kinerja dosen melalui kriteria pendidikan dan pengajaran, penelitian, pengabdian masyarakat, serta umpan balik dari mahasiswa. Setiap kriteria dipresentasikan oleh penilaian pada beberapa subkriteria. Matriks penilaian ternormalisasi menjadi preferensi pada pemeringkatan. Dari hasil pengolahan data kinerja dosen dan hasil umpan balik dari mahasiswa, dengan sample 25 dosen, didapat skor pemeringkatan dengan skala nilai 0 sampai 1, di mana skor 1 adalah peringkat paling tinggi. Proses pemeringkatan kinerja dosen ini melibatkan aktor Dosen, Mahasiswa, dan Unit Pengelola Program Studi. Untuk memudahkan keterlibatan semua aktor pada proses pemeringkatan kinerja

dosen dan tersedianya hasil pemeringkatan yang valid dan tepat waktu kepada pemangku kepentingan, maka diperlukan sistem informasi pemeringkatan kinerja dosen.

Kata Kunci: Algoritma, *Simple Additive Weighting*, SAW, Tri Dharma Perguruan Tinggi, Umpan Balik Mahasiswa

1. PENDAHULUAN

Perguruan tinggi sebagai bagian dari sistem pendidikan nasional memiliki peranan yang sangat penting sebagai wadah penerapan Tri Dharma Perguruan Tinggi (Ariani, 2019). Di dalam Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional, Pasal 20 Ayat 2 tertulis bahwa "Perguruan Tinggi berkewajiban menyelenggarakan pendidikan, penelitian, dan pengabdian kepada masyarakat". Dalam Undang-undang Nomor 12 Tahun 2012 tentang pendidikan tinggi, juga menyatakan bahwa "dosen adalah pendidik profesional dan ilmuwan dengan tugas utama mentransformasikan, mengembangkan, dan menyebarluaskan ilmu pengetahuan dan teknologi melalui pendidikan, penelitian, dan pengabdian kepada masyarakat". Pendidikan merupakan faktor utama dalam membentuk pribadi memperbaiki masyarakat dan membangun bangsa yang beradab (Ariani, 2019). Dosen sebagai salah satu komponen yang sangat berperan dalam pendidikan tinggi diharapkan dapat memberikan kinerja terbaiknya untuk dapat membantu terbentuknya proses pembelajaran yang berkualitas. Peran dosen yang strategis, menuntut kerja dosen yang profesional, dan mampu mengembangkan ragam potensi yang terpendam dalam diri mahasiswa sehingga menjadi memiliki kompetensi (Azhari & Alaren, 2017). Undang-undang No 12 Tahun 2021 menyatakan bahwa: "Pembelajaran adalah proses interaksi mahasiswa dengan dosen dan sumber belajar pada suatu lingkungan belajar". Pada proses pembelajaran yang dilakukan dosen, umpan balik dari mahasiswa menjadi sangat penting sebagai bahan evaluasi bagi dosen.

Pemberian penghargaan kepada dosen adalah salah satu upaya memotivasi dosen untuk memberikan kinerja terbaiknya pada setiap semester. Pemberian penghargaan dimaksudkan sebagai dorongan agar karyawan mau bekerja dengan lebih baik dan membangkitkan motivasi sehingga mendorong kinerja karyawan yang lebih baik (Wijaya, 2021). Untuk memberikan penghargaan kepada dosen yang telah melaksanakan Tri Darma Perguruan Tinggi, maka diperlukan sebuah sistem penunjang keputusan untuk menentukan dosen terbaik, dan dapat dilakukan dengan pemeringkatan kinerja dosen melalui penilaian kinerja Tri Dharma Perguruan Tinggi dan umpan balik dari mahasiswa. Sistem penunjang keputusan adalah suatu sistem di mana sistem tersebut dapat mengambil sebuah keputusan yang memberikan dampak setelahnya pada lingkungan sistem itu sendiri atau bisa juga pada luar sistem (Ratama, 2020). Oleh karena hasil sistem penunjang keputusan harus didasari dengan suatu metode yang dibangun secara ilmiah untuk meminimalisir dampak negatif (Gunawan & Yunus, 2021).

Simple Additive Weighting atau penjumlahan terbobot sederhana adalah algoritma yang dapat digunakan untuk melakukan pemeringkatan. Algoritma *Simple Additive Weighting* dilakukan dengan "mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif", dan membutuhkan proses normalisasi matriks keputusan ke suatu skala yang dapat diperbandingkan dengan semua *rating* alternatif yang ada (Setiadi et al., 2018). Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi *Multiple Attribute Decision Making* (MADM) untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu (Jufri, 2022). Semakin banyak pilihan dan semakin spesifik kriteria yang digunakan, semakin akurat sistem menentukan nilai proses seleksi (Sukaryati et al., 2022). Metode *Simple Additive Weighting* ini mengharuskan "pembuat keputusan" menentukan bobot bagi setiap atribut (Hery Sunandar & Denni M Rajagukguk, 2021). Skor total untuk alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating (yang dapat dibandingkan lintas atribut) dengan bobot tiap atribut. *Rating* tiap atribut haruslah bebas dimensi dalam arti telah melewati proses normalisasi matriks sebelumnya.

Penggunaan algoritma *Simple Additive Weighting* pada penelitian sebelumnya di antaranya adalah pada pemilihan calon karyawan, pengangkatan karyawan terbaik, pemberian bonus pada karyawan, pemberian dana hibah pada siswa kurang mampu, pemilihan lokasi proyek pembangunan strategi, pembelian mobil bekas, dan pemberian dana kredit koperasi (Ratama, 2020). Metode *Simple Additive Weighting* mampu menangani masalah pengambilan keputusan pemilihan guru terbaik pada SMKN 1 Kadipaten, dengan menggunakan 5 kriteria, yaitu: ijazah, sertifikasi, absen, kedisiplinan, dan tanggung jawab (Apriani et al., 2021). Metode *Simple Additive Weighting* dapat digunakan untuk menentukan pemilihan tempat bimbingan belajar (bimbel) terbaik menggunakan 4 kriteria, yaitu: menggunakan 4 kriteria, yaitu: biaya, lokasi, fasilitas, dan kualitas (Jufri, 2022). Sistem pendukung keputusan dengan metode *Simple Additive Weighting* mampu memberikan alternatif dalam menentukan karyawan terbaik pada AMIK Mahaputra Riau, dengan menggunakan 7 kriteria, yaitu: kedisplinan, inisiatif, prestasi, kerjasama, ketertiban, kinerja, dan sosial (Simatupang, 2018). Pemilihan karyawan berprestasi dengan metode *Simple Additive Weighting* dapat membantu manager PT Indomarco Prismatama cabang Tangerang 1 dalam memilih karyawan yang berprestasi karena prosesnya lebih cepat dan mudah (Syam & Rabidin, 2019). SPK (sistem penunjang keputusan) pemilihan karyawan terbaik di PT Gracias Mitra Selaras menggunakan metode *Simple Additive Weighting* membantu pengambilan keputusan dalam masalah pemilihan karyawan terbaik dengan tepat berdasarkan kriteria yang sudah ditentukan (Yusuf & Triyono, 2022). Metode *Simple Additive Weighting* dapat menyelesaikan masalah pemilihan karyawan terbaik pada PT Wahana Internet Nusantara dengan nilai prioritas atau bobot yang ditentukan sesuai dengan kebutuhan masing-masing (Sukaryati et al., 2022). Penentuan peminatan dan lintas minat siswa dengan menggunakan metode *Simple Additive Weighting* pada SMA Negeri Dharma Pendidikan mampu memberikan manfaat bagi pihak Sekolah SMA Negeri Dharma Pendidikan yang terlihat dari hasil pengujian impelementasi menggunakan metode UAT dengan nilai rata-rata tingkat penerimaan pengguna sebesar 82.5% (E. B. Serelia & Adin Saf, 2020).

Pada penelitian ini penulis merancang Sistem Informasi Pemeringkatan Kinerja Dosen pada *Perguruan Tinggi Swasta* "X" dengan menggunakan metode *Simple Additive Weighting*. Kriteria yang digunakan adalah kinerja dosen yang dinilai dari Tri Dharma Perguruan Tinggi ditambah penilaian umpan balik dari mahasiswa. Sistem informasi diperlukan untuk memudahkan interaksi aktor-aktor yang terlibat, yaitu: Dosen, Mahasiswa, dan Unit Pengelola Program Studi. Melalui sistem informasi, pemeringkatan kinerja dosen dapat memberikan hasil pemeringkatan dosen dengan cepat dan tepat waktu kepada semua pemangku kepentingan.

2. METODE PENELITIAN

Pada penelitian ini penulis merancang sebuah instrumen penilaian pemeringkatan dosen untuk Perguruan Tinggi Swasta "X", yang digunakan untuk pemeringkatan kinerja dosen. Pemeringkatan kinerja dosen diambil dari penilaian terhadap kinerja Tri Dharma Perguruan Tinggi, ditambah dengan penilaian umpan balik terhadap dosen yang mengajar pada semester berjalan. Hasil pemeringkatan kinerja dosen selanjutnya digunakan untuk memutuskan pemilihan dosen terbaik. Berikut adalah tahapan yang penulis lakukan dalam penelitian.

2.1 Analisa kebutuhan

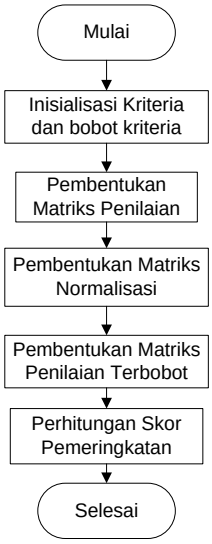
Berdasarkan analisa penulis mengenai pemberian *reward* bagi dosen pada Perguruan Tinggi Swasta "X" yang berprestasi, maka dibutuhkan suatu mekanisme untuk menentukan dosen berpresetasi melalui pemeringkatan. Pemberian *reward* ini dilakukan dalam rangka memotivasi dosen untuk lebih meningkatkan kualitas kinerjanya, dan dapat digunakan oleh Unit Pengelola Program Studi untuk memantau kinerja dosennya.

1) **Merancang instrumen pemeringkatan kinerja dosen dengan metode *Simple Additive Weighting***: Instrumen pemeringkatan kinerja dosen dibuat dengan mengacu pada algoritma *Simple Additive Weighting*. Penulis menetukan kriteria berdasarkan unsur yang ada pada Tri Dharma Perguruan Tinggi ditambah umpan balik dari mahasiswa.

2) **Membangun rancangan sistem informasi**: Aktor yang terlibat pada sistem peringkatan kinerja dosen ini adalah: Dosen, Mahasiswa, dan Unit Pengelola Program Studi. Untuk itu sistem informasi sangat diperlukan agar proses pemeringkatan dapat berjalan dengan mudah, cepat, dan tepat. Pada perancangan sistem ini penulis menggunakan *tools* berupa *use case diagram*, *activity diagram*, rancangan *database*, serta rancangan *user interface*.
3) **Menguji rancangan sistem**: Pengujian sistem dilakukan menggunakan metode *black box*, yaitu hanya menguji program melalui *user interface*. Pengujian juga dilakukan hanya dengan menggunakan data *sample*, yaitu data kinerja dari 25 dosen.

Konsep dasar metode *Simple Additive Weighting* adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode *Simple Additive Weighting* membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada (Yulianti & Z, 2018). Berikut adalah tahapan yang dilakukan oleh penulis dalam merancang instrumen pemeringkatan dengan menggunakan metode *Simple Additive Weighting*. Gambar 1 menyajikan *flowchart* dari proses yang dilakukan pada algoritma *Simple Additive Weighting*.

1) Menentukan kriteria dan subkriteria.
2) Menentukan bobot nilai untuk setiap kriteria dan subkriteria.
3) Memberikan bobot penilaian pada semua subkriteria untuk semua alternatif, kemudian membentuk hasil penillaian dalam bentuk matriks peniaian.
4) Membuat matriks penilaian ternormalisasi.
5) Membuat matriks penilaian terbobot.
6) Preferensi pemeringkatan.



Gambar 1 *Flowchart* Proses Algoritma *Simple Additive Weighting*

3. HASIL DAN PEMBAHASAN

Hasil penelitian berdasarkan pembentukan algoritma *Simple Additive Weighting* dan perancangan *user interface* sistem informasi pemeringkatan kinerja dosen yang dilakukan sebagai berikut.

3.1 Pembentukan Algoritma *Simple Additive Weighting*

3.1.1 Kriteria dan Subkriteria

Penentukan kriteria didasarkan pada Tri Dharma Perguruan Tinggi Dosen dan umpan balik mahasiswa mengenai dosen. Dari Tri Dharma Perguruan Tinggi terbentuk tiga kriteria, yaitu:

- C1: Pengajaran, dengan tiga subkriteria, yaitu: mengampu matakuliah sesuai keilmuan, melakukan pengajaran sebanyak 16 kali pertemuan untuk setiap matakuliah yang diampunya, dan mengumpulkan nilai tepat waktu.
- C2: Penelitian, dengan tiga subkriteria, yaitu: tema penelitian sesuai keilmuan, publikasi penelitan pada *ejournal*, dan membuat laporan penelitian sesuai gaya selingkung pada perguruan tinggi *homebase*.
- C3: Pengabdian Masyarakat, dengan subkriteria, yaitu: tema pengabdian sesuai dengan keilmuan, membuat laporan kegiatan sesuai dengan gaya selingkung *homebase* perguruan tinggi, memiliki luaran lain selain laporan kegiatan.

Sedangkan umpak balik dari mahasiswa hanya membentuk satu kriteria, yaitu:

- C4: Umpan Balik Mahasiswa, dengan 5 subkriteria, yaitu: keandalan (*realibility*), daya tanggal (*responsiveness*), kepastian (*assurance*), empati (*emphaty*), dan *tanggible*.

3.1.2 Nilai dan Bobot untuk Setiap Kriteria dan Subkriteria

Tabel 1 Bobot Nilai Subkriteria

	Kriteria	Kategori	Bobot
C1	**Pengajaran**		
	C1-1. Mengampu matakuliah sesuai keilmuan	cost	1=YES; 2=NO; 3=Tidak mengajar
	C1-2. Melakukan pengajaran sebanyak 16 kali	cost	1=16 pertemuan; 2=15 sampai 12 pertemuan; 3=Kurang dari 12 pertemuan; 4=Tidak mengajar
	C1-3. Mengumpulkan nilai tepat waktu	cost	1=YES; 2=NO; 3=Tidak mengajar
C2	**Penelitian**		
	C2-1. tema penelitian sesuai keilmuan	cost	1=YES; 2=NO; 3= Tidak melakukan penelitian
	C2-2. publikasi jurnal penelitian	cost	1=Jurnal terkareditasi sinta 2 atau 1; 2= Jurnal terkareditasi sinta 4 atau 3; 3= Jurnal terakreditas sinta 6 atau 5; 4= Jurnal tidak terakreditasi; 5=Tidak ada publikasi
	C2-3. Laporan penelitian	cost	1=lengkap; 2= Tidak lengkap; 3= Tidak ada
C3	**Pengabdian**		
	C3-1. Tema kegiatan sesuai dengan keilmuan	cost	1=YES; 2=NO; 3= Tidak melakukan pengabdian masyarakat
	C3-2. Laporan kegiatan	cost	1=Lengkap; 2= Tidak lengkap; 3=tidak ada
	C3-3. Luaran kegiatan	cost	1=Publikasi ejournal atau media massa elektronik; 2= Tidak dipubilkasikan; 3=Tidak melakukan pengabdian
C4	**Umpan Balik dari Mahasiswa**		
	C4-1. Keandalan/ Reliability C4-2. Daya tanggal/ Responsiveness C4-3. Kepastian/Assurance C4-4. Empati/Empathy C4-5. Tangible	benefit	1=Sangat tidak baik; 2=Tidak baik; 3=Baik; 4=Sangat baik (menggunakan Likert skala 1 sampai 4)

Tabel 1 menyajikan matriks penilaian penilaian subkriteria. Tabel ini menjadi acuan untuk menghitung nilai masing-masing kriteria. Tabel 2 menyajikan bobot dari setiap kriteria, masing-masing kriteria diberi bobot sebesar 0,25 atau 25%.

Tabel 2 Bobot Kriteria

Kriteria	Keterangan	Bobot
C1	Pengajaran	25%
C2	Penelitian	25%
C3	Pengabdian	25%
C4	Umpan balik Mahasiswa	25%

3.1.3 Bobot Penilaian pada Semua Subkriteria untuk Semua Kriteria

Tabel 3 Skor Penilaian Dosen Berdasarkan Bobot pada Masing-Masing Subkriteria dan Krieria

Nama	Bobot Penilaian Berdasarkan Subkriteria														Total Bobot Berdasarkan Kriteria			
	C1-1	C1-2	C1-3	C2-1	C2-2	C2-3	C3-1	C3-2	C3-3	C4-1	C4-2	C4-3	C4-4	C4-5	C1	C2	C3	C4
Dosen_01	2	4	1	2	4	1	2	2	1	3	3	2	3	4	7	7	5	15
Dosen_02	1	3	3	2	5	3	1	3	4	4	4	3	3	1	7	10	8	15
Dosen_03	1	3	2	2	2	2	2	1	2	4	3	4	3	1	6	6	5	15
Dosen_04	2	2	3	1	5	3	3	3	4	3	3	2	3	4	7	9	10	15
Dosen_05	2	3	1	2	2	1	1	1	1	2	4	2	2	4	6	5	3	14
Dosen_06	3	4	3	2	3	1	2	1	2	3	3	3	3	4	10	6	5	16
Dosen_07	1	4	1	2	4	1	3	4	4	4	4	2	4	2	6	7	11	16
Dosen_08	2	4	3	2	5	3	2	3	1	3	4	4	2	2	9	10	6	15
Dosen_09	2	4	2	2	4	2	1	2	4	1	1	3	4	3	8	8	7	12
Dosen_10	1	1	1	1	2	1	1	1	1	4	3	4	3	4	3	4	3	18
Dosen_11	2	1	2	2	2	2	1	2	2	4	3	4	2	5	5	6	5	18
Dosen_12	1	3	1	1	2	1	1	1	2	5	2	1	3	5	5	4	4	16
Dosen_13	1	3	2	2	3	2	2	1	2	5	2	3	5	5	6	7	5	20
Dosen_14	1	1	2	1	2	1	2	2	2	5	4	4	5	4	4	4	6	22
Dosen_15	1	2	1	1	1	2	1	2	2	5	4	4	3	5	4	4	5	21
Dosen_16	2	2	1	2	2	2	1	1	1	4	4	3	5	4	5	6	3	20
Dosen_17	1	1	2	1	3	1	1	2	2	3	2	5	4	5	4	5	5	19
Dosen_18	1	3	2	1	2	1	2	2	1	3	3	3	4	3	6	4	5	16
Dosen_19	1	2	2	1	3	2	1	2	1	2	5	4	1	5	5	6	4	17
Dosen_20	2	2	2	1	4	2	2	2	1	3	4	3	1	4	6	7	5	15
Dosen_21	1	2	2	1	2	1	1	1	1	3	2	3	3	5	5	4	3	16
Dosen_22	1	3	2	2	3	1	2	1	1	5	3	5	3	3	6	6	4	19
Dosen_23	1	2	1	1	3	1	1	1	1	3	5	4	3	4	4	5	3	19
Dosen_24	2	2	2	1	3	2	1	1	1	3	3	5	5	5	6	6	3	21
Dosen_25	2	2	1	1	2	2	1	2	1	5	4	3	5	5	5	5	4	22

```
[ 7  7  5  15
  7  10 8  15
  6  6  5  15
  7  9  10 15
  6  5  3  14
  10 6  5  16
  6  7  11 16
  9  10 6  15
  8  8  7  12
  3  4  3  18
  5  6  5  18
  5  4  4  16
  6  7  5  20
  4  4  6  22
  4  4  5  21
  5  6  3  20
  4  5  5  19
  6  4  5  16
  5  6  4  17
  6  7  5  15
  5  4  3  16
  6  6  4  19
  4  5  3  19
  6  6  3  21
  5  5  4  22 ]
```

Gambar 2 Matriks Penilaian Kriteria

Tabel 3 adalah hasil penilaian 25 dosen yang disajikan dalam bentuk matriks terhadap empat kriteria yang digunakan sesuai dengan bobot dari masing-masing kriteria. Matriks penilaian pada Gambar 2 yang terbentuk dari hasil penilaian yang telah diinputkan pada sistem. Matrik penilaian

berordo $i \times j$ di mana i adalah jumlah alternatif (dalam penelitian ini adalah jumlah dosen yang akan diperingkatkan) dan j adalah jumlah kriteria. Gambar 2 menyajikan hasil matriks penilaian yang terbentuk dari Tabel 3.

3.1.4 Membuat Matriks Penilaian Ternormalisasi

Jika i adalah jumlah alternatif (dosen yang akan diperingkatkan), dan j adalah jumlah kriteria yang digunakan, maka matriks penilaian ternormalisas M_{ij} didapat dari rumus pada Pers. (1) (Jufri, 2022). Gambar 3 adalah hasil perhitungan matriks penilaian ternormalisasi berdasarkan rumus tersebut.

$$M_{ij} = \begin{cases} \dfrac{x_{ij}}{Max\_x_{ij}}; \text{untuk Kriteria berkatgori Benefit} \\ \dfrac{Min\_x_{ij}}{x_{ij}}; \text{untuk Kriteria berkategori Cost} \end{cases} \quad (1)$$

0,43	0,40	0,38	0,80
0,50	0,67	0,60	0,80
0,43	0,44	0,30	0,80
0,50	0,80	1,00	0,86
0,30	0,67	0,60	0,75
0,50	0,57	0,27	0,75
0,33	0,40	0,50	0,80
0,38	0,50	0,43	1,00
1,00	1,00	1,00	0,67
0,60	0,67	0,60	0,67
0,60	1,00	0,75	0,75
0,50	0,57	0,60	0,60
0,75	1,00	0,50	0,55
0,75	1,00	0,60	0,57
0,60	0,67	1,00	0,60
0,75	0,80	0,60	0,63
0,50	1,00	0,60	0,75
0,60	0,67	0,75	0,71
0,50	0,57	0,60	0,80
0,60	1,00	1,00	0,75
0,50	0,67	0,75	0,63
0,75	0,80	1,00	0,63
0,50	0,67	1,00	0,57
0,60	0,80	0,75	0,55

Gambar 3 Matriks Penilaian Ternormalisasi

3.1.5 Membuat Matriks Penilaian Terbobot

0,11	0,14	0,15	0,20
0,11	0,10	0,09	0,20
0,13	0,17	0,15	0,20
0,11	0,11	0,08	0,20
0,13	0,20	0,25	0,21
0,08	0,17	0,15	0,19
0,13	0,14	0,07	0,19
0,08	0,10	0,13	0,20
0,09	0,13	0,11	0,25
0,25	0,25	0,25	0,17
0,15	0,17	0,15	0,17
0,15	0,25	0,19	0,19
0,13	0,14	0,15	0,15
0,19	0,25	0,13	0,14
0,19	0,25	0,15	0,14
0,15	0,17	0,25	0,15
0,19	0,20	0,15	0,16
0,13	0,25	0,15	0,19
0,15	0,17	0,19	0,18
0,13	0,14	0,15	0,20
0,15	0,25	0,25	0,19
0,13	0,17	0,19	0,16
0,19	0,20	0,25	0,16
0,13	0,17	0,25	0,14
0,15	0,20	0,19	0,14

Gambar 4 Matriks Penilaian Terbobot

Skoring pemeringkatan dihasilkan dengan mengalikan matriks penilaian yang ternormalisasi dengan bobot masing-masing kriteria. Gambar 4 adalah hasil pembentukan matriks penilaian terbobot.

3.1.6 Preferensi Pemeringkatan

Preferensi pemeringkatan dilakukan dengan menjumlahkan skor penilaian semua kriteria untuk n-alternatif. Skor pada preferensi pemeringkatan dibuat dengan menggunakan rumus pada Pers. (2) (Jufri, 2022). Di mana V_i adalah skor preferensi untuk alternatif i, w_j adalah bobot kriteria-j, dan r_{ij} eleman matriks penelilaian ternormalisasi. Tabel 4 menyajikan hasil pemeringkatan yang sudah terurut berdasarkan preferensi pemeringkatan menggunakan Pers. (2).

$$V_i = \sum_{i=1}^{n} w_j r_{ij} \quad (2)$$

Tabel 4 Preferensi Pemeringkatan

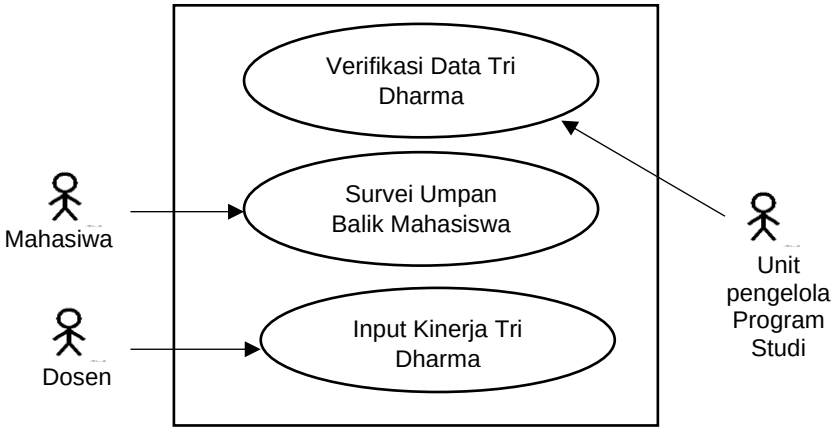
Peringkat	Alternatif	Bobot Pemeringkatan
1	Dosen_10	0,92
2	Dosen_21	0,84
3	Dosen_23	0,80
4	Dosen_05	0,79
5	Dosen_12	0,78
6	Dosen_15	0,73
7	Dosen_16	0,72
8	Dosen_18	0,71
9	Dosen_14	0,70
10	Dosen_17	0,70
11	Dosen_24	0,68
12	Dosen_19	0,68
13	Dosen_25	0,67
14	Dosen_03	0,64
15	Dosen_22	0,64
16	Dosen_11	0,63
17	Dosen_20	0,62
18	Dosen_01	0,60
19	Dosen_06	0,58
20	Dosen_09	0,58
21	Dosen_13	0,57
22	Dosen_07	0,52
23	Dosen_08	0,51
24	Dosen_02	0,50
25	Dosen_04	0,49

3.2 Perancangan *User Interface* Sistem Informasi

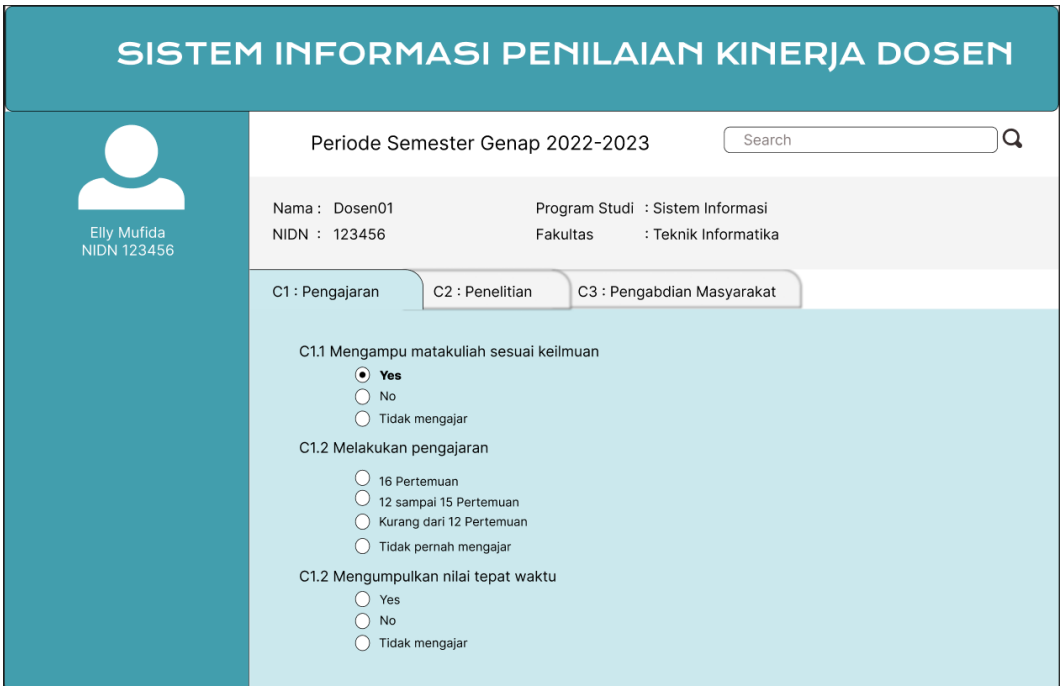
Perancangan *user interface* pada sistem informasi menggunakan *usecase diagram* dan rancangan tampilan *user interface*. Gambar 5 adalah *usecase diagram* dari sistem pemeringkatan kinerja dosen. Aktor yang terlibat adalah; Dosen, Mahasiswa, dan Unit Pengelola Program Studi. *Case* yang terbentuk adalah: input kinerja dosen, survei umpan balik mahasiswa, dan penilaian dan verifikasi.

Dalam sistem ini, dosen dapat melakukan penginputan data kinerja pada bidang pengajaran, penelitian dan pengabdian masyarakat. Pimpinan Unit Pengelola Program Studi dapat melakukan verifikasi dan penilaian kinerja data pengajaran dosen yang telah diinputkan oleh dosen. Mahasiswa dapat mengisi kuesioner survei umpan balik mahasiswa mengenai dosen.

Gambar 6 menyajikan rancangan *user interface* untuk penilaian kinerja dosen yang dilakukan oleh Unit Pengelola Program Studi.



Gambar 5 *Use Case Diagram* Sistem Pemeringkatan Kinerja Dosen



Gambar 6 *User Interface* Penilaian Kinerja Dosen

4. KESIMPULAN

Dari hasil penelitian yang telah dilakukan, penulis menyimpulkan bahwa sistem keputusan pemeringkatan kinerja dosen untuk Perguruan Tinggi Swasta “X” dengan menggunakan algoritma *Simple Additive Weighting* adalah sebagai berikut:

1) Penulis menggunakan *sample* data kinerja dari 25 dosen.
2) Kinerja dosen dinilai berdasarkan kriteria dari unsur Tri Dharma Perguruan Tinggi dan umpan balik mahasiswa mengenai dosen.
3) Pemeringkatan dosen didapat melalui hasil preferensi dengan skor antara 1 sampai 0, di mana skor 1 untuk peringkat paling tinggi. Diperlukan penambahan subkriteria penilaian untuk mempertajam penilaian dan meminimalisir kemungkinan hasil bobot pemeringkatan yang sama.

4) Masih ada subkriteria yang belum bisa memenuhi semua kemungkinan yang terjadi, misalnya untuk subkriteria C2-2 belum ada nilai untuk publikasi hasil penelitian dosen pada jurnal internasional.
5) Proses pemeringkatan kinerja dosen membutuhkan sebuah sistem informasi yang dapat mengintegrasikan aktifitasi semua aktor yang terlibat, yaitu: proses penginputan kinerja oleh dosen, pengisian kuesioner umpan balik mahasiswa, serta penilaian oleh Unit Pengelola Program Studi, sehingga dapat menjamin ketersediaan hasil pemeringkatan kinerja dosen dengan tepat waktu kepada seluruh pemangku kepentingan.

Pada penelitian selanjutnya perlu analisa lebih dalam lagi untuk mengembangkan subkriteria pada masing-masing kriteria, agar penilaian kinerja dosen menjadi lebih objektif lagi. Perlu ada perbandingan penggunaan algoritma lain yang dapat digunakan untuk kasus pemeringkatan kinerja dosen untuk mengevaluasi kelebihan dan kekurangan metode *Simple Additive Weighting*. Perlu dilakukan kajian lebih dalam lagi pada mekanisme pengambilan data umpan balik mahasiswa mengenai dosen.

DAFTAR PUSTAKA

Apriani, N. D., Krisnawati, N., & Fitrisari, Y. (2021). Implementasi Sistem Pendukung Keputusan Dengan Metode SAW Dalam Pemilihan Guru Terbaik. *Journal Automation Computer Information System*, *1*(1), 37–45. https://doi.org/10.47134/jacis.v1i1.5

Ariani, S. S. (2019). Persepsi Mahasiswa dalam Pengimplementasian Tri Daharma Perguruan Tinggi. *At-Tadbir : Jurnal Manajemen Pendidikan Islam*, *3*(1), 59–77. https://doi.org/10.3454/AT-TADBIR.V3I1.3414

Azhari, D. S., & Alaren, A. (2017). Peran Dosen dalam Mengembangkan Karakter Mahasiswa. *Jurnal Pelangi*, *9*(2), 88–97. https://doi.org/10.22202/jp.2017.v9i2.1856

Gunawan, V. S., & Yunus, Y. (2021). Sistem Penunjang Keputusan dalam Optimalisasi Pemberian Insentif terhadap Pemasok Menggunakan Metode TOPSIS. *Jurnal Informatika Ekonomi Bisnis*, 101–108. https://doi.org/10.37034/infeb.v3i3.86

Jufri, H. Al. (2022). Perhitungan Manual Dengan Menggunakan Metoda SAW (Simple Additive Weighting). *Jurnal Ilmiah Sistem Informasi*, *2*(1), 59–68. https://doi.org/10.46306/SM.V2I1.21

Ratama, N. (2020). *Sistem Penunjang Keputusan dan Sistem Pakar dengan Pemahaman Studi Kasus*. Uwais Inspirasi Indonesia. http://uwaismall.com

Serelia, E. B., & Adin Saf, M. R. (2020). Sistem Pendukung Keputusan Penentuan Peminatan Siswa Dengan Menggunakan Metode SAW (Simple Additive Weighting) Pada SMA Negeri Dharma Pendidikan. *Techno.Com*, *19*(3), 227–236. https://doi.org/10.33633/tc.v19i3.3498

Setiadi, A., Yunita, Y., & Ningsih, A. R. (2018). Penerapan Metode Simple Additive Weighting (SAW) Untuk Pemilihan Siswa Terbaik. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, *7*(2), 104–109. https://doi.org/10.32736/sisfokom.v7i2.572

Simatupang, J. (2018). Sistem Pendukung Keputusan Penentuan Karyawan Terbaik Menggunakan Metode SAW Studi Kasus AMIK Mahaputra Riau. *Jurnal Intra Tech*, *2*(1), 73–82. https://doi.org/10.37030/jit.v2i1.27

Sukaryati, L. N., Voutama, A., Karawang, U. S., & Ronggo, J. H. (2022). Penerapan Metode Simple Additive Weighting Pada Sistem Pendukung Keputusan Untuk Memilih Karyawan Terbaik. *Jurnal Ilmiah Matrik*, *24*(3), 260–267. https://doi.org/10.33557/JURNALMATRIK.V24I3.2029

Hery Sunandar, & Denni M Rajagukguk. (2021). Sistem Pendukung Keputusan Penentuan Instruktur Bahasa Inggris Dengan Metode Simple Additive Weighting (SAW). *JUKI : Jurnal Komputer Dan Informatika*, *1*(2), 59–65. https://doi.org/10.53842/juki.v1i2.18

Syam, S., & Rabidin, M. (2019). Metode Simple Additive Weighting dalam Sistem Pendukung Keputusan Pemilihan Karyawan Berprestasi (Studi Kasus : PT. Indomarco Prismatama cabang Tangerang 1). *UNISTEK*, *6*(1), 14–18. https://doi.org/10.33592/unistek.v6i1.168

Wijaya, L. F. (2021). Sistem Reward dan Punishment Sebagai Pemicu dalam Meningkatkan Kinerja Karyawan. *Journal MISSY (Management and Business Strategy)*, *2*(2), 25–28. https://doi.org/10.24929/missy.v2i2.1681

Yulianti, E., & Z, R. (2018). Sistem Pendukung Keputusan Seleksi Penerima Bedah Rumah Menggunakan Metode Simple Additive Weighting (SAW) (Studi Kasus : Dinas Sosial Dan Tenaga Kerja Kota Padang). *JURNAL TEKNOIF*, *6*(2), 64–73. https://doi.org/10.21063/JTIF.2018.V6.2.64-73

Yusuf, R., & Triyono, G. (2022). Sistem Penunjang Keputusan Menggunakan Metode SAW di PT Gracias Mitra Selaras. *Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, *1*(1), 2003–2010. https://senafti.budiluhur.ac.id/index.php/senafti/article/view/369

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 1, JANUARI, 2024, Pp. 70 – 78
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Algoritma *Decision Tree* untuk Prediksi Deteksi Penyakit Kanker Payudara

Ayu Dian Fitri Mellina [(1)*], Suhartono [(2)], M. Ainul Yaqin [(3)]
Teknik Informatika, Fakultas Sains dan Teknologi, UIN Maulana Malik Ibrahim, Malang
e-mail : 17650087@student.uin-malang.ac.id, {suhartono,yaqinov}@ti.uin-malang.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 27 Juni 2023, direvisi 12 September 2023, diterima 6 Oktober 2023, dan dipublikasikan 25 Januari 2024.

Abstract

Cancer is a deadly disease that is difficult to cure. Early cancer detection can be done through laboratory tests to identify the cancer type. Breast cancer is a type of cancer with initial symptoms in the form of a lump. Data mining and classification methods, such as decision trees with ID3 and C5.0 algorithms, are used to categorize breast cancer. The dataset used is Breast Cancer Coimbra, which was downloaded from UCI Machine Learning in 2018. ID3 has limitations in handling unstructured data and continuous attributes, while C5.0 is better. Both algorithms produce tree models with different levels of accuracy. This study shows that the C5.0 algorithm has the best classification results with 80% accuracy, 84.2% precision, 80% recall, and 80% F1 score. 80% accuracy shows the system's classification ability, so the C5.0 model can be used to predict breast cancer.

Keywords: Breast Cancer, Classification, Prediction, Decision Tree, Machine Learning

Abstrak

Kanker merupakan penyakit mematikan yang sulit untuk disembuhkan. Deteksi dini pada kanker dapat dilakukan melalui uji laboratorium yang dapat mengidentifikasi jenis kanker. Kanker payudara merupakan salah satu jenis kanker ganas dengan gejala awal berupa benjolan. *Data mining* dan metode klasifikasi, seperti *decision tree* dengan algoritma ID3 dan C5.0, digunakan untuk mengkategorikan kanker payudara. *Dataset* yang digunakan adalah Breast Cancer Coimbra yang diunduh dari UCI Machine Learning tahun 2018. ID3 memiliki keterbatasan dalam menangani data tidak terstruktur dan atribut kontinu, sementara C5.0 lebih baik dalam hal tersebut. Kedua algoritma menghasilkan model pohon dengan tingkat keakuratan yang berbeda. Penelitian ini menunjukkan bahwa algoritma C5.0 memiliki hasil klasifikasi terbaik dengan akurasi 80%, presisi 84,2%, *recall* 80%, dan F1 *score* 80%. Akurasi 80% menunjukkan kemampuan sistem dalam klasifikasi, sehingga model C5.0 dapat digunakan untuk memprediksi kanker payudara.

Kata Kunci: Kanker Payudara, Klasifikasi, Prediksi, *Decision Tree, Machine Learning*

1. PENDAHULUAN

Kanker payudara termasuk pada golongan penyakit kanker ganas yang mana kasusnya banyak dijumpai di kalangan wanita. Penyakit ini dapat menyerang wanita pada usia berapapun, tetapi resiko terkenanya penyakit ini meningkat dengan bertambahnya usia. Penyakit ini juga dapat menyerang pria, meskipun hal ini sangat jarang terjadi. Jumlah kasus penyakit kanker payudara di seluruh dunia semakin bertambah setiap tahunnya. Salah satu gejala awal pada kanker payudara adalah munculnya benjolan kecil yang semakin lama akan bertambah semakin besar. Perubahan atau mutasi pada DNA sel payudara merupakan penyebab awal timbulnya kanker payudara. Mutasi gen biasanya terjadi karena diwariskan dari generasi sebelumnya, akan tetapi mutasi ini dapat juga terjadi tanpa penyebab yang pasti. Perempuan dengan resiko terkena kanker lebih besar adalah perempuan yang mengalami siklus menstruasi lebih banyak daripada perempuan normal lainnya, terlambat menopause, serta menstruasi dini. Peningkatan jumlah kasus kanker payudara di seluruh dunia juga disebabkan oleh perubahan pola gaya hidup yang tidak sehat serta kurangnya aktivitas fisik (Musa & Aliyu, 2020).

Terdapat beberapa atribut yang menjadi acuan dalam mendeteksi jenis kanker pada penyakit kanker payudara. Atribut tersebut membentuk sebuah pola yang kemudian dikategorikan sesuai dengan kelas yang sudah ada. Dalam menentukan jenis kanker payudara yang dialami oleh pasien di rumah sakit, pihak laboratorium rumah sakit tentunya membutuhkan waktu yang cukup lama untuk menganalisis hasil diagnosa mengenai jenis kanker tersebut. Hasil data laboratorium tidak sepenuhnya memberikan hasil yang konkrit, tentunya diperlukan hipotesis yang dapat memperkuat hasil laboratorium tersebut. Hipotesis disini bertujuan agar dapat membantu dokter menentukan jenis kanker pasien sehingga dapat ditangani dengan segera. Terdapat banyak cara atau metode dalam menentukan pola pada data mengenai penyakit kanker, salah satu cara tersebut adalah menggunakan metode *data mining*. Pada proses *data mining* terdapat beberapa metode di dalamnya, salah satunya adalah metode prediktif yang memiliki teknik yang dapat digunakan, yakni regresi dan klasifikasi (Sunjana, 2010). Dalam penelitian ini, proses klasifikasi dapat diterapkan untuk mengolah hasil data uji laboratorium dengan mengkategorikannya menjadi dua kategori, yakni kategori jinak (*benign*) dan ganas (*malignant*).

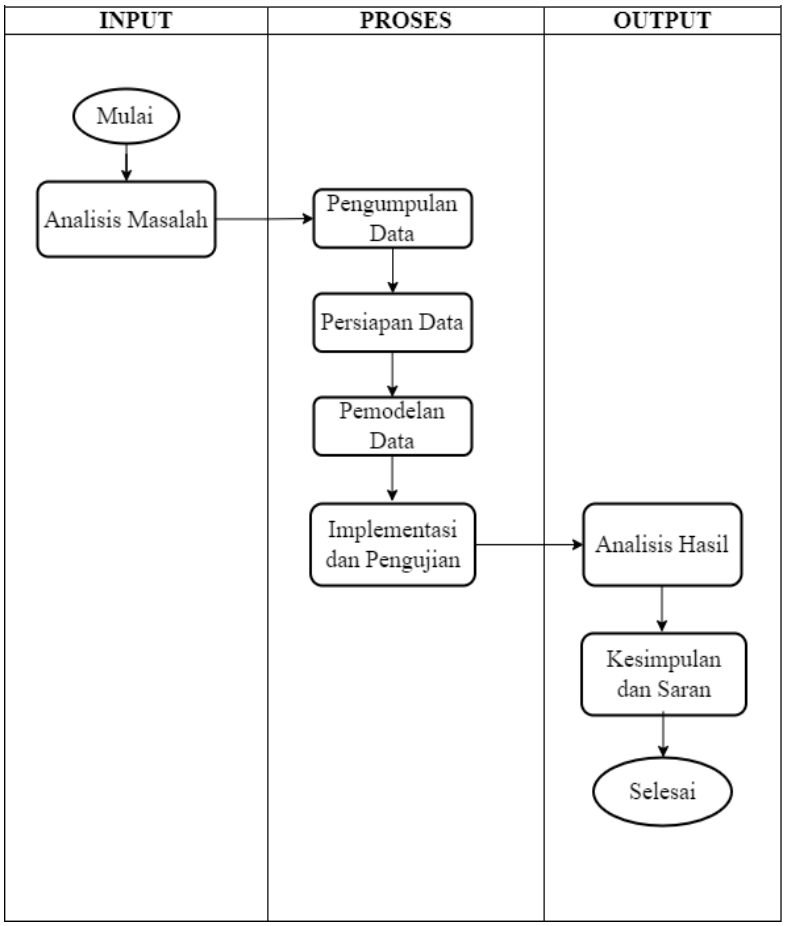
Klasifikasi yang digunakan pada penelitian ini menggunakan metode *decision tree*. *Decision tree* merupakan sebuah metode sistem prediksi yang strukturnya menyerupai pohon bercabang atau biasa juga disebut dengan struktur hierarki sehingga metode ini cocok untuk diterapkan pada permasalah penelitian ini yang di dalamnya menggambarkan sebuah persoalan dan mencari atau membutuhkan sebuah solusi dari persoalan tersebut (Wahyudin, 2009). Kedua algoritma Iterative Dichotomiser-3 (ID3) dan C5.0 merupakan salah satu algoritma *decision tree* yang dapat digunakan untuk tujuan tersebut (Wei, 2011).

Kedua algoritma Iterative Dichotomiser-3 (ID3) dan C5.0 merupakan algoritma *decision tree* yang dapat digunakan untuk deteksi penyakit kanker payudara. Algoritma Iterative Dichotomiser-3 (ID3) merupakan algoritma yang pertama kali dikembangkan oleh *decision tree* yang memiliki kemampuan yang cukup baik dalam menangani data yang terstruktur. Namun, algoritma ini tidak dapat menangani data yang tidak terstruktur dan tidak dapat menangani atribut yang bernilai kontinu. Sedangkan, C5.0 merupakan pengembangan dari algoritma Iterative Dichotomiser-3 (ID3) yang memiliki kemampuan yang lebih baik dalam menangani data yang tidak terstruktur dan dapat menangani atribut yang bernilai kontinu. Algoritma ini juga memiliki kemampuan untuk membangun model yang lebih akurat dan memiliki fitur pruning yang dapat membantu mengurangi overfitting pada model. Kedua algoritma tersebut dapat menghasilkan model pohon (*tree*) yang berbeda dengan dataset yang sama. Model yang dihasilkan dari kedua algoritma tersebut tentunya memilki tingkat keakuratan yang berbeda.

Latar belakang penelitian ini adalah untuk mengembangkan suatu sistem prediksi menggunakan metode *decision tree* serta membandingkan model yang dihasilkan oleh algoritma yang ada pada *decision tree* yakni Iterative Dichotomiser-3 (ID3) dan C5.0 untuk deteksi penyakit kanker payudara. Sistem ini diharapkan dapat membantu dokter dalam mengambil keputusan yang lebih akurat dan tepat waktu dalam menegakkan diagnosis kanker payudara.

2. METODE PENELITIAN

Dalam menjalankan program yang dibangun, penelitian ini menggunakan bahasa pemrograman R, serta menggunakan beberapa *library* yang disediakan oleh R untuk melakukan proses *machine learning.* Alur penelitian dilakukan dengan memulai dari menganalisis masalah, melakukan pengumpulan data, kemudian mempersiapkan data yang akan digunakan, dan dilanjutkan dengan pemodelan data dengan metode yang sudah ada. Langkah selanjutnya adalah melakukan implementasi dan pengujian, sehingga bisa didapatkan hasil dan dapat ditarik menjadi sebuah kesimpulan. Adapun alur penelitian ditunjukan oleh Gambar 1.



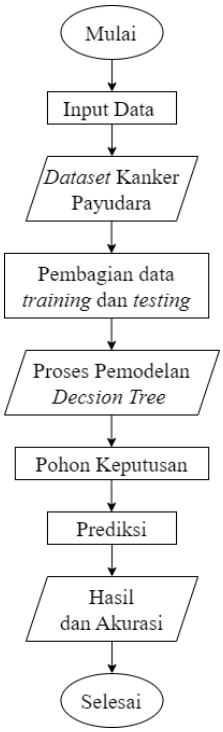
Gambar 1 Prosedur Penelitiain

2.1 Pengumpulan Data

Data yang digunakan pada penelitian ini diperoleh dari *website* resmi UCI *Machine Learning Repository*. Data tersebut merupakan *dataset breast cancer* coimbra tahun 2018 (Patrcio et al., 2018). Total jumlah keseluruhan *dataset* yang digunakan sebanyak 116 data, dengan perincian 52 data merupakan kelas *benign* (jinak) serta 64 data merupakan kelas *malign* (ganas). *Missing value* pada keseluruhan data yang digunakan berjumlah 0 atau tidak ada. Pada *dataset* tersebut terdapat 9 atribut dan 1 kelas klasifikasi, 9 atribut tersebut yaitu umur, glukosa, resistin, *Homeostatic Model Assessment* (HOMA), insulin, leptin, *Body Mass Index* (BMI), adiponectin, dan MCP-1.

2.2 Desain Sistem

Gambaran umum desain sistem prediksi deteksi penyakit kanker payudara menggunakan metode *decision tree* yakni metode Iterative Dichotomiser-3 (ID3) dan algoritma C5.0 yang dijabarkan pada *flowchart* Gambar 2. Alur dari desain sistem yang dibangun dimulai dari penginputan *dataset* kanker payudara, kemudian pembagian *dataset* menjadi dua bagian yakni data *training* dan *testing.* Kemudian proses *learning* dengan menerapkan model *decision tree* dengan menggunakan algoritma ID3 dan C5.0, proses pembentukan pohon keputusan, prediksi, kemudian hasil dan akurasi dari sistem yang dibangun.



Gambar 2 Desain Sistem

2.3 Pembagian *Dataset*

Pada langkah ini, *dataset* yang digunakan adalah data terstruktur sebanyak 116 *item*. *Dataset* terdiri dari usia, BMI, kadar glukosa dalam darah, insulin, HOMA, Leptin, Adiponectin, Resistin, dan MCP.1. *Dataset* pelatihan diklasifikasikan menjadi dua, yaitu kanker jinak dan kanker ganas. *Dataset* tersebut dibagi menjadi dua bagian. Bagian pertama digunakan sebagai data pelatihan (*training*) untuk membuat model klasifikasi, sementara bagian kedua digunakan sebagai data uji (*testing*) untuk mengevaluasi model yang telah dibuat. Pembagian *dataset* pada penelitian ini dilakukan dengan jumlah persentase yang beragam untuk mengetahui variasi nilai akurasi yang yang ada pada kedua metode *decision tree.* Adapun pembagian persentase pengujian data dijelaskan pada Tabel 1.

Tabel 1 Skenario Pengujian

Skenario	Jumlah Data Latih	Jumlah Data Uji	Keterangan
1	80%	20%	Dataset sebanyak 80% akan menjadi data latih (data training), sedangkan 20% sisanya akan menjadi data uji (data testing).
2	75%	25%	Dataset sebanyak 75% akan menjadi data latih (data training), sedangkan 25% sisanya akan menjadi data uji (data testing).
3	70%	30%	Dataset sebanyak 70% akan menjadi data latih (data training), sedangkan 30% sisanya akan menjadi data uji (data testing).
4	50%	50%	Dataset sebanyak 50% akan menjadi data latih (data training), sedangkan 50% sisanya akan menjadi data uji (data testing).
5	25%	75%	Dataset sebanyak 25% akan menjadi data latih (data training), sedangkan 75% sisanya akan menjadi data uji (data testing).

2.4 *Modeling*

Proses pemodelan pada algoritma *decision tree* Iterative Dichotomiser-3 (ID3) dilakukan dengan cara pembentukan pohon klasifikasi menggunakan dua langkah. Langkah pertama yang dilakukan adalah dengan cara menentukan nilai *entropy*, kemudian dilanjut dengan langkah kedua yakni menghitung nilai *information gain* pada tiap variabel (Pribadi et al., 2018). *Entropy* pada proses ini berfungsi untuk mengukur node yang digunakan sebagai parameter pada sampel data. Perhitungan nilai *entropy* pada algoritma ID3 yakni sebagaimana pada Pers. (1). Di mana S merupakan data sampel yang digunakan sebagai training, P_+ adalah probablitas sampel S dengan kelas positif, dan P_- adalah probablitas sampel S dengan kelas positif.

$$Entropy(S) = -P_+ log_2 P_+ - P_- log_2 P_- \quad (1)$$

Pengurangan *entropy* pada algoritma ID3 disebut dengan *information gain*. Pembagian sampel S terhadap atribut X dihitung dengan menggunakan rumus *information gain* yakni sebagaimana pada Pers. (2). Di mana X adalah atribut, V yaitu nilai yang memungkin untuk atribut X, dan $value(X)$ himpunan yang mungkin untuk atribut (X).

$$Gain(S, X) = Entropy(S) - \sum_{V\ \epsilon\ value(X)} \frac{|S_V|}{|S|}\ Entropy(S_V) \quad (2)$$

Setelah *information gain* pada semua atribut dihitung, kemudian dipilih nilai *information gain* tertinggi untuk dijadikan *root* pada suatu pohon keputusan. Hal ini dilakukan seterusnya hingga parameter pada tiap-tiap atribut terklasifikasi dengan sempurna. Proses perhitungan tersebut memiliki kesamaan dengan algoritma ID-3 yakni mulai dari perhitungan *entropy* dan *information gain*, akan tetapi pada algoritma C5.0 atribut dengan *gain ratio* tertinggi akan dipilih sebagai *root node* (Wei, 2011). Adapun rumus perhitungan *gain ratio* yakni sebagaimana pada Pers. (3).

$$Gain\ Ratio = \frac{Gain(S, X)}{\sum_{i=1}^{m} Entropy(S_i)} \quad (3)$$

2.5 Evaluasi Sistem

Evaluasi sistem dilakukan dengan menggunakan *binary class confusion matrix* karena penelitian ini termasuk dalam klasifikasi dua kelas. Evaluasi pada sistem yang dibangun meliputi nilai akurasi, presisi, *recall*, dan F-1 *score.* Akurasi klasifikasi mengacu pada persentase data pengujian yang diklasifikasikan dengan benar oleh model. Jika akurasi klasifikasi dianggap memadai, maka model dapat digunakan untuk mengklasifikasikan set data di masa mendatang yang memiliki label kelas yang belum diketahui (Agarwal, 2013). Presisi atau yang dikenal juga sebagai *precision*, menggambarkan proporsi unit yang diprediksi sebagai positif oleh model yang juga benar-benar positif dalam data yang sebenarnya. Presisi dapat diinterpretasikan sebagai tingkat kesesuaian antara permintaan informasi dan respons terhadap permintaan tersebut (Mayadewi & Rosely, 2015). *Recall* adalah hasil perhitungan yang menunjukan sejauh mana semua data uji yang positif telah diprediksi dengan benar sebagai positif dalam klasifikasi. *Recall* juga dikenal sebagai *True Positive Rate* (TPR), sensitivitas, dan probabilitas deteksi (Grandini et al., 2020). Dalam klasifikasi *binary class* di mana setiap observasi hanya memiliki satu label, skor F1 yang dihitung dengan metode mikro (*micro-averaged* F1) sama dengan akurasi klasifikasi secara keseluruhan (Zhang et al., 2015). Rumus dari akurasi, presisi, *recall* dan F1 *score* ditunjukan oleh Pers. (4) sampai (7).

$$accuracy = \frac{TP}{Total\ data\ testing} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1\ Score = \left(\frac{2 \times Recall \times Precision}{Recall + Precision}\right) \quad (7)$$

3. HASIL DAN PEMBAHASAN

Hasil uji coba tingkat akurasi pada skenario pengujian yang dilakukan mulai dari iterasi pertama hingga iterasi kelima dikelompokan dalam Tabel 2. Dari hasil pemaparan pada Tabel 2 tersebut nilai akurasi yang paling optimal didapatkan oleh skenario pengujian dengan pembagian data dengan perbandingan rasio yakni sebesar 70:30 dengan artian 70% dari total keseluruhan data atau sebanyak 81 data digunakan sebagai data latih (*training*), sementara 30% dari total keseluruhan data atau sebanyak 35 data digunakan sebagai data uji (*testing*). Tabel 3 menunjukan lima sampel data *training*, sedangkan Tabel 4 menunjukan lima sampel data *testing*.

Tabel 2 Hasil Akurasi pada Skenario Pengujian

No.	Skenario Pengujian	Akurasi	
		Iterative Dichotomiser-3 (ID3)	C5.0
1	Skenario 1 (80%-20%)	69.57%	75.00%
2	Skenario 2 (75%-25%)	72.41%	68.97%
3	Skenario 3 (70%-30%)	77.14%	80.00%
4	Skenario 4 (50%-50%)	72.41%	63.79%
5	Skenario 5 (25%-75%)	68.57%	70.11%

Tabel 3 Sampel Data *Training*

No.	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1	Classification
1	29	32,270	84	5,81	1,203	45,619	6,209	24,603	904,981	1
2	66	36,212	101	15,533	3,869	74,706	7,539	22,320	864,968	1
3	86	21,111	92	3,549	0,805	6,699	4,819	10,576	773,92	1
4	69	28,444	108	8,808	2,346	14,748	5,288	16,485	353,568	2
5	51	22,892	103	2,74	0,696	8,016	9,349	11,554	359,232	2

Tabel 4 Sampel Data *Testing*

No.	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1	Classification
1	68	21,367	77	3,226	0,612	9,882	7,169	12,766	928,22	1
2	49	22,854	92	3,226	0,732	6,831	13,679	10,317	530,41	1
3	34	21,47	78	3,469	0,667	14,57	13,11	6,92	354,6	1
4	29	23,01	82	5,663	1,145	35,59	26,72	4,58	174,8	1
5	25	22,86	82	4,09	0,827	20,45	23,67	5,14	313,73	1

Tabel 5 Hasil Prediksi Klasifikasi ID3

No.	*Classification*	Prediksi ID3
1	1	2
2	1	1
3	1	1
4	1	1
5	1	1

Hasil prediksi dari klasifikasi kemudian ditunjukan oleh *confusion matrix* pada Tabel 6. Seluruh nilai tersebut diperlukan untuk proses perhitungan nilai *accuracy*, *precision*, *recall*, dan *micro* F1 pada *confusion matrix*. Pada Tabel 7 berisi konfusi *matrix* beserta deskripsi dari nilai-nilai

tersebut. Selanjutnya dilakukan perhitungan nilai akurasi, presisi, *recall*, dan F1 *score*, dapat dilakukan dengan mengunakan Pers. (4) sampai Pers. (7).

Tabel 6 Hasil Prediksi Algoritma ID3

		Prediksi ID3	
		1	**2**
Aktual	**1.**	14	6
	2.	2	13

Tabel 7 *Confussion Matrix* Algoritma ID3

		Prediksi ID3	
		1	**2**
Aktual	**1.**	14 (TP)	6 (FN)
	2.	2 (FP)	13 (TN)

$$Accuracy = \frac{14+13}{35} = \frac{27}{35} = 0{,}7714 = 77{,}14\%$$

$$Precision = \frac{14}{14+2} \times 100\% = \frac{14}{16} \times 100\% = 87{,}5\%$$

$$Recall = \frac{14}{14+6} \times 100\% = \frac{14}{20} \times 100\% = 70\%$$

$$\text{F1 } score = \frac{2 \times 70\% \times 87{,}5\%}{70\% + 87{,}5\%} = \frac{12{,}250}{157} = 78\%$$

Prediksi selanjutnya menggunakan *decision tree* C5.0 yang hasil klasifikasi dan prediksinya ditunjukan pada Tabel 8 dan 9. Pada Tabel 10 berisi *confusion matrix* pada algoritma C5.0 beserta deskripsi dari nilai-nilai pada tiap prediksi tersebut, maka perhitungan nilai akurasi, presisi, *recall,* dan F1 *score*, dapat dilakukan dengan mengunakan Pers (4) sampai Pers. (7).

Tabel 8 Hasil Prediksi Klasifikasi Algoritma C5.0

No.	***Classification***	**Prediksi C5**
1	1	1
2	1	2
3	1	1
4	1	1
5	1	1

Tabel 9 Hasil Prediksi Algoritma ID3

		Prediksi ID3	
		1	**2**
Aktual	**1.**	16	4
	2.	3	12

Tabel 10 *Confussion Matrix* Algoritma ID3

		Prediksi ID3	
		1	**2**
Aktual	**1.**	16 (TP)	4 (FN)
	2.	3 (FP)	12 (TN)

$$Accuracy = \frac{16+12}{35} = \frac{28}{35} = 0{,}8 = 80\%$$

$$Precision = \frac{16}{16+3} \times 100\% = \frac{16}{19} \times 100\% = 84{,}2\%$$

$$Recall = \frac{16}{16+4} \times 100\% = \frac{16}{20} \times 100\% = 80\%$$

$$\text{F1 } score = \frac{2 \times 84{,}2\% \times 80\%}{84{,}2\% + 80\%} = \frac{13{,}472}{164{,}2} = 82\%$$

Hasil evaluasi yang telah dilakukan pada uji coba skenario pertama hingga kelima dengan menggunakan *ratio* yang berbeda-beda pada data latih (*training*) dan data uji (*testing*), serta tidak saling beririsan. Dapat dilihat bahwasanya algoritma Iterative Dichotomiser-3 (ID3) memiliki tingkat nilai akurasi yang konsisten pada ketiga skenario uji coba, yang berkisar 72.41%-77.14%. Hal ini disebabkan karna algoritma Iterative Dichotomiser-3 (ID3) sulit memprediksi model pohon klasifikasi dengan data uji (*testing*) yang berubah-ubah secara signifikan dan bervariasi seperti pada skenario uji coba kedua sampai keempat, hal inilah yang menyebabkan hasil kinerja pada algoritma tersebut lebih konsisten, karena banyaknya data latih yang digunakan tidak mempengaruhi nilai akurasi. Serta pohon keputusan yang dihasilkan cenderung sederhana dan lebih umum. Sedangkan pada algoritma C5.0, nilai akurasi yang didapatkan lebih bervariasi pada tiap uji coba skenario. Hal ini dikarenakan algoritma C5.0 lebih memiliki kecenderungan untuk mempelajari pola yang sangat spesifik pada data latih (*training*) yang digunakan. Hal inilah yang menyebabkan hasil pohon keputusan pada algoritma C5.0 lebih sederhana jika dibandingkan dengan ID3, karena algoritma C5.0 akan melakukan pemangkasan (*pruning*) terhadap cabang-cabang yang tidak signifikan dari pohon keputusan untuk menghindari mempelajari pola yang terlalu spesifik pada data pelatihan. Algoritma C5.0 memiliki lebih banyak *hyper-parameter* dan aturan (*rule*) yang dapat mempengaruhi hasil akurasi. Jika *ratio* pembagian data yang digunakan berbeda dalam setiap skenario pengujian, maka hasil akurasi C5.0 dapat bervariasi secara signifikan.

4. KESIMPULAN

Berdasarkan hasil perbandingan akurasi dari dua algoritma pada metode *decision tree* yakni Iterative Dichotomiser-3 (ID3) dan C5.0 secara umum diambil dari semua skenario pengujian yang telah dilakukan, maka algoritma C5.0 dengan perolehan nilai akurasi sebesar 80% mengungguli algortima Iterative Dichotomiser-3 (ID3) dengan hasil nilai akurasi sebesar 77,14%. Algortima C5.0 memiliki keunggulan dalam kompleksitas model yan dihasilkan dengan menggunakan teknik *pruning* sehingga dapat meningkatkan kinerja dan akurasi model. Sementara untuk hasil lain dari Iterative Dichotomiser-3 (ID3) nilai presisi yang didapat sebesar 87,5%, *recall* sebesar 70%, dan F1 *score* sebesar 78%. Sementara pada algoritma C5.0 nilai presisi sebesar 84,2%, *recall* sebesar 80%, dan nilai F1 *score* sebesar 82%. Hasil tersebut menunjukan bahwa nilai akurasi, presisi, *recall*, dan F1 *score* pada C5.0 termasuk ke dalam kategori baik. Sehingga dapat ditarik kesimpulan bahwa pemodelan sistem dengan menggunakan algoritma C5.0 memiliki tingkat akurasi yang lebih baik dibandingkan dengan algoritma Iterative Dichotomiser-3 (ID3).

DAFTAR PUSTAKA

Agarwal, S. (2013). Data Mining: Data Mining Concepts and Techniques. *2013 International Conference on Machine Intelligence and Research Advancement*, 203–207. https://doi.org/10.1109/ICMIRA.2013.45

Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*. http://arxiv.org/abs/2008.05756

Mayadewi, P., & Rosely, E. (2015). Prediksi Nilai Proyek Akhir Mahasiswa Menggunakan Algoritma Klasifikasi Data Mining. *Seminar Nasional Sistem Informasi Indonesia*

(SESINDO) 2015, *2015*. https://is.its.ac.id/pubs/oajis/index.php/home/detail/1582/PREDIKSI-NILAI-PROYEK-AKHIR-MAHASISWA-MENGGUNAKAN-ALGORITMA-KLASIFIKASI-DATA-MINING

Musa, A. A., & Aliyu, U. M. (2020). Application of Machine Learning Techniques in Predicting of Breast Cancer Metastases Using Decision Tree Algorithm, in Sokoto Northwestern Nigeria. *Journal of Data Mining in Genomics & Proteomics*, *11*(1). https://www.walshmedicalmedia.com/open-access/application-of-machine-learning-techniques-in-predicting-of-breast-cancer-metastases-using-decision-tree-algorithm-in-sokoto-north-53078.html

Patrcio, M., Pereira, J., Crisstomo, J., Matafome, P., Seia, R., & Caramelo, F. (2018). *Breast Cancer Coimbra*. UCI Machine Learning Repository. https://doi.org/https://doi.org/10.24432/C52P59

Pribadi, D., Athiry, S., Saputra, R. A., Supiandi, A., Prayudi, D., Nusa, S., & Sukabumi, M. (2018). Sistem Pakar Diagnosa Penyakit Demam Berdarah Dengue Menggunakan Algoritma Iterative Dichotomiser 3 (ID3). *SNIT 2018*, *1*(1), 129–133. https://seminar.bsi.ac.id/snit/index.php/snit-2018/article/view/37

Sunjana. (2010). Aplikasi Mining Data Mahasiswa dengan Metode Klasifikasi Decision Tree. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 1907–5022. https://journal.uii.ac.id/Snati/article/view/1857

Wahyudin. (2009). *Metode Iterative Dichotomizer 3 (ID3) Untuk Penerimaan Mahasiswa Baru*. Universitas Pendidikan Indonesia.

Wei, W. (2011). ID3 Algorithm and C4.5 Algorithm Based on Decision Tree. *Journal of Hubei University of Technology*.

Zhang, D., Wang, J., & Zhao, X. (2015). Estimating the Uncertainty of Average F1 Scores. *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, 317–320. https://doi.org/10.1145/2808194.2809488

9 772527 583007