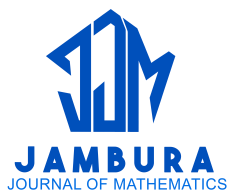


# Estimasi Cadangan Klaim Individu untuk Klaim *Short-Tailed* dan *Long-Tailed* Menggunakan Algoritma *Backpropagation*

Anas Satriya Rabdika dan Azizah Azizah



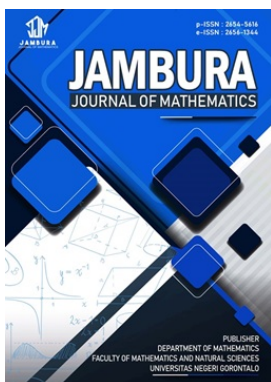
Volume 8, Issue 1, Pages 75–86, February 2026

Diterima 3 November 2025, Direvisi 5 Februari 2026, Disetujui 9 Februari 2026, Diterbitkan 13 Februari 2026

To Cite this Article : A. S. Rabdika dan A. Azizah, "Estimasi Cadangan Klaim Individu untuk Klaim *Short-Tailed* dan *Long-Tailed* Menggunakan Algoritma *Backpropagation*", *Jambura J. Math*, vol. 8, no. 1, pp. 75–86, 2026, <https://doi.org/10.37905/jjom.v8i1.35114>

© 2026 by author(s)

## JOURNAL INFO • JAMBURA JOURNAL OF MATHEMATICS

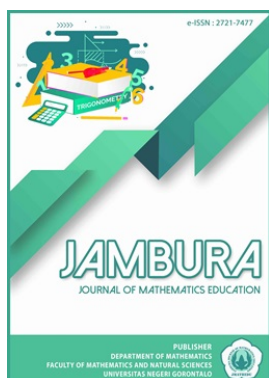


	Homepage	: <a href="http://ejurnal.ung.ac.id/index.php/jjom/index">http://ejurnal.ung.ac.id/index.php/jjom/index</a>
	Journal Abbreviation	: Jambura J. Math.
	Frequency	: Biannual (February and August)
	Publication Language	: English (preferable), Indonesia
	DOI	: <a href="https://doi.org/10.37905/jjom">https://doi.org/10.37905/jjom</a>
	Online ISSN	: 2656-1344
	Editor-in-Chief	: Hasan S. Panigoro
	Publisher	: Department of Mathematics, Universitas Negeri Gorontalo
	Country	: Indonesia
	OAI Address	: <a href="http://ejurnal.ung.ac.id/index.php/jjom/oai">http://ejurnal.ung.ac.id/index.php/jjom/oai</a>
	Google Scholar ID	: iWLjgaUAAAAJ
	Email	: <a href="mailto:info.jjom@ung.ac.id">info.jjom@ung.ac.id</a>

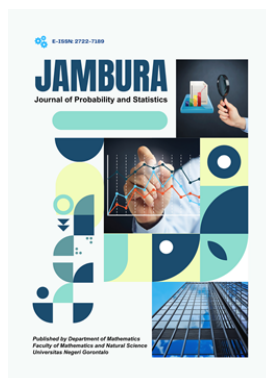
## JAMBURA JOURNAL • FIND OUR OTHER JOURNALS



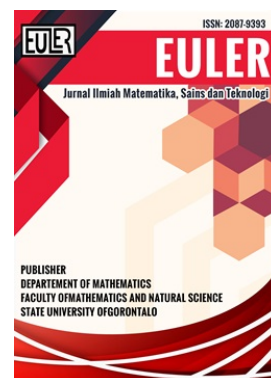
Jambura Journal of Biomathematics



Jambura Journal of Mathematics Education



Jambura Journal of Probability and Statistics



EULER : Jurnal Ilmiah Matematika, Sains, dan Teknologi

# Estimasi Cadangan Klaim Individu untuk Klaim *Short-Tailed* dan *Long-Tailed* Menggunakan Algoritma *Backpropagation*

Anas Satriya Rabdika<sup>1</sup>, Azizah Azizah<sup>1,\*</sup>

<sup>1</sup>Departemen Matematika, Universitas Negeri Malang, Malang 65145, Indonesia

## ARTICLE HISTORY

Diterima 3 November 2025  
Direvisi 5 Februari 2026  
Disetujui 9 Februari 2026  
Diterbitkan 13 Februari 2026

## KATA KUNCI

Algoritma *Backpropagation*  
Cadangan klaim individu  
*Machine learning*  
Klaim *Short-tailed*  
Klaim *long-tailed*

## KEYWORDS

*Backpropagation* Algorithm  
Individual claim reserving  
*Machine learning*  
*Short-tailed* claims  
*Long-tailed* claims

**ABSTRAK.** Dalam asuransi, risiko yang terjadi sering kali tidak menentu yang menyebabkan pengajuan atas risiko dapat memiliki nilai yang sangat besar sehingga perusahaan asuransi mungkin tidak dapat menyanggupi pembayaran klaim. Jika hal ini terjadi, maka asuransi membutuhkan cadangan klaim untuk mempersiapkannya. Sebelumnya, terdapat beberapa metode untuk mencadangkan klaim seperti metode pencadangan secara agregat, akan tetapi terdapat sifat klaim yang memiliki ketergantungan di antaranya sehingga menyebabkan kurangnya informasi mengenai masing-masing klaim. Selain itu, jumlah klaim yang semakin besar akan semakin sulit untuk dihitung secara tradisional. Oleh karena itu, penelitian ini bertujuan untuk menghitung cadangan klaim secara individu menggunakan salah satu cabang dari machine learning, yaitu Algoritma *Backpropagation*. Algoritma *Backpropagation* diyakini masih relevan jika dibandingkan dengan model algoritma lain karena dalam beberapa penelitian, Algoritma *Backpropagation* menghasilkan nilai Mean Absolute Percentage Error (MAPE) yang tergolong rendah, yakni sekitar 2,70%. Data untuk penelitian ini disimulasikan menggunakan software R dengan package Frank Copula sejumlah 10.000 klaim, dengan rincian 6.000 klaim *short-tailed* dan 4.000 klaim *long-tailed*. Model data dievaluasi kelayakannya menggunakan MAPE. Hasil kelayakan menggunakan MAPE sebesar 0,55% yang menunjukkan bahwa data tersebut sudah sangat baik untuk digunakan sebagai model prediksi. Hasil prediksi penelitian menunjukkan bahwa klaim yang harus dibayar pada tahun perkembangan ke-21 mencapai Rp22.945.450.000.000,- dengan rata-rata setiap klaim sebesar Rp2.294.545.152,-. Penelitian ini berkontribusi di bidang informatika sekaligus aktuaria dengan mengembangkan cara penghitungan cadangan klaim individu untuk memprediksi klaim yang perlu dibayar dengan lebih efisien.

**ABSTRACT.** In insurance, risk can occur at any time, causing claims to sometimes have a large amount of value, so that insurance companies may not be able to satisfy claim payments. If these situations occur, insurance companies need claim reserves to prepare for such events. There are several methods to calculate claim reserves, such as aggregate claim reserving. However, certain claim characteristics involve dependencies among claims, which result in a lack of detailed information for individual claims. In addition, an increasing number of claims becomes more difficult to compute using traditional methods. Therefore, this research aims to calculate individual claim reserves using one branch of machine learning, namely the *Backpropagation* Algorithm. The *Backpropagation* Algorithm is believed to remain relevant compared to other algorithmic models because, in several studies, it produces relatively low values of Mean Absolute Percentage Error (MAPE), at approximately 2.70%. The data used in this research are simulated using R software, generating 10,000 claims over 20 years, consisting of 6,000 *short-tailed* claims and 4,000 *long-tailed* claims. The data model is evaluated using MAPE. The resulting MAPE value is 0.55%, indicating that the data are highly suitable for predictive modeling. The prediction results show that the total claims to be paid in the 21st development year reach Rp22,945,450,000,000, with an average claim amount of approximately Rp2,294,545,152. This research contributes to both informatics and actuarial science by developing an individual claim reserving approach to predict claim payments more efficiently.



This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License. **Editorial of JJoM:** Department of Mathematics, Universitas Negeri Gorontalo, Jln. Prof. Dr. Ing. B. J. Habibie, Bone Bolango 96554, Indonesia.

## 1. Pendahuluan

Asuransi menurut UU No. 2 Tahun 1992 dapat didefinisikan sebagai suatu perjanjian di mana pihak penanggung mengikatkan diri kepada pihak tertanggung dengan menerima suatu premi, untuk memberikan penggantian kepadanya karena suatu kerugian maupun kerusakan yang mungkin akan dideritanya karena suatu peristiwa yang tidak tertentu. Dari definisi tersebut,

dapat disimpulkan bahwa suatu kerugian dapat terjadi kapan saja dan di mana saja, sehingga pihak tertanggung dapat mengajukan hak atas kerugian yang didapat. Berdasarkan waktu penyelesaian klaim, terdapat dua jenis klaim yaitu klaim *short-tailed* dan klaim *long-tailed*. Klaim *short-tailed* adalah klaim yang memiliki waktu relatif cepat dalam pembayarannya, umumnya kurang dari satu tahun, sedangkan klaim *long-tailed* memiliki waktu yang lebih lama dalam pembayarannya [1]. Nyatanya, terdapat produk

\*Penulis Korespondensi.

asuransi yang klaimnya tidak terbatas pada *short-tailed* atau *long-tailed* saja. Maka dari itu, perusahaan asuransi memerlukan sebuah metode khusus untuk menghitung cadangan klaim. Cadangan klaim sendiri merupakan sejumlah dana yang dipersiapkan oleh perusahaan asuransi guna membayar utang klaim [2].

Berdasarkan banyaknya data, teknik pencadangan klaim dibagi menjadi dua, yakni teknik pencadangan klaim secara agregat dan teknik pencadangan klaim secara individu. Sebelumnya, teknik pencadangan klaim masih dihitung secara agregat dan menggunakan teknik yang masih tradisional. Akan tetapi, dalam perhitungannya, teknik pencadangan klaim secara individu pada saat ini lebih dipercaya karena terdapat kelemahan saat melakukan perhitungan pada teknik pencadangan secara agregat. Hal ini dibuktikan oleh penelitian yang dilakukan oleh Vanegas dkk. [3] yang meragukan perhitungan cadangan klaim melalui metode klasik berupa *Chain-Ladder* karena kurangnya asumsi statistik yang kuat untuk menghitung metode tersebut. Bukti kekurangan perhitungan cadangan klaim secara agregat lainnya ditunjukkan oleh Chaoubi dkk. [4] yang menyatakan bahwa di dalam metode tradisional terlalu mengandalkan *Run-Off Triangle* yang berisi total cadangan klaim pada tahun kejadian dan tahun perkembangan pada periode tertentu, sehingga informasi detail mengenai masing-masing klaim pada periode tersebut tidak diperoleh. Oleh karena itu, penelitian ini akan berfokus pada teknik pencadangan klaim secara individu.

Dalam kasus pencadangan klaim, estimasi cadangan klaim individu menjadi suatu tantangan karena kompleksitas dalam perhitungannya terletak pada prediksi frekuensi dan severitas klaim yang belum dilaporkan, sehingga membutuhkan teknik pemodelan yang canggih untuk mengestimasi cadangan klaim secara akurat [5]. Dari semua pendapat tersebut dapat disimpulkan bahwa diperlukan adanya penghitungan cadangan klaim individu menggunakan *machine learning*. *Machine learning* merupakan algoritma dari sebuah komputer yang mendeteksi suatu pola dalam data yang kemudian kemampuannya akan meningkat seiring berjalannya waktu [6]. Pola yang diidentifikasi pada algoritma di *machine learning* tidak muncul begitu saja, tetapi membutuhkan pelatihan menggunakan data dan beberapa pengujian agar dapat menghasilkan suatu algoritma yang optimal. Dengan kata lain, metode ini dapat membantu manusia dalam menyelesaikan masalah yang membutuhkan pola yang kompleks walaupun membutuhkan pelatihan data terlebih dahulu. *Machine learning* memiliki beberapa kelebihan jika dibandingkan dengan metode tradisional. Salah satu kelebihan penggunaan *machine learning* adalah metode ini dapat menjadi solusi dalam pencadangan klaim secara individu karena dapat mendukung struktur regresi yang lebih fleksibel [7]. Selain itu, penggunaan *machine learning* dapat mengolah data yang terstruktur maupun tidak terstruktur [8]. Oleh karena itu, penelitian ini secara khusus akan membahas mengenai penggunaan *machine learning* dalam mencadangkan klaim individu.

Terdapat beberapa teknik untuk mencadangkan klaim secara individu menggunakan *machine learning*. Dalam penelitian ini, teknik yang digunakan adalah melalui *Neural Networks*. *Neural Networks* merupakan jaringan yang dirancang seperti cara kerja otak manusia yang digunakan untuk menyelesaikan masalah pengelompokan dan pengenalan suatu pola [9]. *Neural Networks* terdiri dari tiga jenis *layer*, yaitu *input layer*, *hidden layer*, dan *ou-*

*tput layer*. Menurut Syaharuddin dkk. [10], *input layer* berisikan data mentah, kemudian data tersebut diolah menggunakan fungsi kombinasi sekaligus ditambahkan *weight* di *hidden layer*. Terakhir, data yang sudah berada di *hidden layer* disubstitusikan ke dalam fungsi aktivasi sehingga diperoleh nilai akhir yang berada pada *output layer*.

Beberapa peneliti telah mencoba menerapkan *Neural Networks* menggunakan algoritma pelatihan tertentu, salah satunya dengan menggunakan Algoritma *Backpropagation*. Algoritma *Backpropagation* merupakan algoritma yang bekerja dalam dua langkah, yakni *feed forward* (perambatan maju) dan *feed backward* (perambatan mundur). *Feed forward* merupakan perhitungan algoritma secara maju dari *input layer* menuju *output layer*. *Feed backward* merupakan proses pembaruan *weight* yang dilakukan dari *output layer* kembali ke *input layer* untuk memperbarui nilai *weight* sehingga nilai prediksi diperkirakan akan mendekati nilai sebenarnya dan diperoleh nilai *weight* yang optimal untuk memprediksi nilai *output* dengan *error* seminimal mungkin. Metode Algoritma *Backpropagation* dipilih karena banyak digunakan, memiliki struktur yang sederhana, algoritma yang cukup matang, serta dapat diaplikasikan pada berbagai bidang [9]. Selain itu, meskipun telah digunakan beberapa *Neural Network* lain seperti *Convolutional Neural Networks* (CNN) dan *Recurrent Neural Networks* (RNN), algoritma ini masih relevan karena masih dapat memberikan model terbaik dengan nilai *MAPE* hanya sebesar 2,7% dan tingkat keakuratan sebesar 97% [11–13]. Berdasarkan informasi tersebut, penelitian ini menggunakan pendekatan Algoritma *Backpropagation* untuk mengestimasi cadangan klaim secara individu.

Berdasarkan pernyataan Vanegas dkk. [3] dan Chaoubi dkk. [4] yang menyatakan bahwa pendekatan agregat menggunakan metode *Chain-Ladder* memiliki kelemahan, terutama terkait asumsi statistik yang kuat serta keterbatasan dalam menangkap informasi detail masing-masing klaim pada suatu periode tertentu, serta dengan mempertimbangkan pendapat Wüthrich [8] yang menyatakan bahwa pendekatan *machine learning* mampu mengolah dan mengestimasi data secara cepat dan akurat, maka penelitian ini menggunakan pendekatan individu untuk mengestimasi cadangan klaim dengan Algoritma *Backpropagation*. Pendekatan ini dipilih karena dinilai mampu mengatasi keterbatasan yang terdapat pada metode agregat berbasis *Chain-Ladder*.

Penelitian ini menggunakan data asuransi perkapalan (*marine hull insurance*) sebagai studi kasus. Asuransi perkapalan dipilih karena memiliki karakteristik klaim yang tidak hanya bersifat *short-tailed*, yaitu klaim dengan waktu penyelesaian relatif cepat, tetapi juga bersifat *long-tailed*, di mana proses pembayaran klaim memerlukan waktu yang relatif lama, sebagaimana dijelaskan dalam Keputusan Resmi Komisi Eropa tahun 1992. Karakteristik tersebut menjadikan asuransi perkapalan sebagai contoh yang relevan untuk mengestimasi klaim individu menggunakan Algoritma *Backpropagation*.

## 2. Metode

### 2.1. Landasan Teori

#### 2.1.1. Cadangan Klaim

Cadangan klaim merupakan sejumlah dana yang dipersiapkan oleh perusahaan asuransi guna membayar hutang klaim [2]. Cadangan klaim perlu disiapkan oleh perusahaan asuransi karena terdapat aspek ketidakpastian terjadinya suatu klaim. Jika

klaim yang diajukan terlalu banyak dan melebihi cadangan klaim yang disiapkan, maka perusahaan dapat mengalami gagal bayar. Di sisi lain, jika cadangan klaim yang disiapkan terlalu besar, maka perusahaan perlu menyiapkan dana yang lebih banyak.

Selain itu, terdapat faktor waktu penyelesaian klaim sehingga klaim dibagi menjadi dua jenis, yaitu klaim *short-tailed* dan klaim *long-tailed*. Klaim *short-tailed* adalah klaim yang memiliki waktu relatif cepat dalam pembayarannya, umumnya kurang dari satu tahun, sedangkan klaim *long-tailed* memiliki waktu yang lebih lama dalam pembayarannya [1].

### 2.1.2. Pencadangan Klaim Individu

Terdapat dua jenis pencadangan klaim, yakni secara agregat dan secara individu. Dalam penghitungannya, pencadangan klaim secara agregat didasarkan pada data historis total klaim per periode tertentu, sedangkan pencadangan klaim secara individu menggunakan data historis setiap klaim per periode. Pencadangan klaim secara agregat seperti *Chain-Ladder* memiliki kelemahan yaitu tidak adanya asumsi statistik yang cukup kuat [3]. Selain itu, pencadangan klaim secara agregat hanya menunjukkan total klaim pada periode tertentu.

### 2.1.3. Copula

Copula merupakan metode statistika yang diperuntukkan untuk memodelkan struktur dependensi antara dua variabel [14]. Sebagai fungsi kumulatif dari distribusi multivariat, Copula biasanya digunakan untuk memisahkan ketergantungan antara dua pola data, sekaligus memberikan pendekatan yang efisien dalam menghitung ketergantungan antara risiko yang mungkin terjadi [15]. Dalam bidang asuransi, Copula digunakan untuk memisahkan ketergantungan antara *short-tailed claim* dan *long-tailed claim*.

*Short-tailed claim* membutuhkan waktu penyelesaian pembayaran klaim yang relatif cepat [1]. Sebaliknya, *long-tailed claim* membutuhkan waktu penyelesaian pembayaran yang lebih lama. Dalam perhitungannya, *short-tailed claim* dan *long-tailed claim* dipisah menggunakan Copula. Salah satu jenis Copula yang dapat digunakan untuk menghitung ketergantungan ini adalah Frank Copula. Frank Copula merupakan bagian dari keluarga *Archimedean Copula* karena Copula ini dapat memodelkan ketergantungan antara *short-tailed* dan *long-tailed* dengan bentuk yang lebih sederhana [16]. Pada penelitian ini, penerapan Frank Copula dibantu menggunakan *software R* untuk mempermudah simulasi data *short-tailed claim* dan *long-tailed claim*.

### 2.1.4. Algoritma Backpropagation

Algoritma *Backpropagation* merupakan salah satu cabang dari *Neural Networks*. Algoritma ini menggunakan nilai *error* untuk mengatur *weight* yang optimal [17]. Algoritma *Backpropagation* memiliki dua langkah, yaitu *feed forward* (perambatan maju) dan *feed backward* (perambatan mundur) [18].

Tahap *feed forward* dimulai dengan menginput data-data yang tersedia sebagai titik-titik di *input layer*. Kemudian, digunakan fungsi kombinasi untuk mendapatkan nilai neuron pada *hidden layer* sebelum akhirnya diproses lebih lanjut menggunakan fungsi aktivasi. Adapun rumus untuk fungsi kombinasi adalah sebagai berikut:

$$\delta_j = \sum (\omega_{i,j} x_i), \quad (1)$$

dengan keterangan:

- $\delta_j$  : nilai neuron pada setiap *hidden layer*,
- $\omega_{i,j}$  : nilai *weight* yang menghubungkan *input layer* dengan *hidden layer*,
- $x_i$  : nilai neuron pada setiap *input layer*.

Setelah menghasilkan nilai di *hidden layer*, digunakan fungsi aktivasi yang bertujuan untuk mengaktifkan dan menonaktifkan neuron atau titik-titik pada *layer* [19]. Fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid, yang memiliki rentang nilai  $[0, 1]$ . Rumus fungsi aktivasi sigmoid adalah sebagai berikut [20]:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (2)$$

Sedangkan tahap *feed backward* merupakan langkah sebaliknya dari *output layer* ke *input layer*. Langkah ini bertujuan untuk mengatur nilai *weight* sehingga mendapatkan nilai *output* yang diinginkan. Tahap *feed backward* diawali dengan menghitung setiap neuron di *output layer* menggunakan persamaan berikut:

$$\delta_k = y_k(1 - y_k)(y_k - t_k), \quad (3)$$

dengan:

- $\delta_k$  : nilai neuron baru pada *output layer*,
- $y_k$  : nilai *output layer* sebenarnya,
- $t_k$  : hasil *output* dari tahap *feed forward*.

Selanjutnya, untuk setiap neuron di *hidden layer*, nilai baru ditentukan menggunakan persamaan berikut:

$$\delta_j = y_j(1 - y_j) \sum_{k \in K} (\delta_k \omega_{jk}), \quad (4)$$

dengan:

- $\delta_j$  : nilai neuron baru pada *hidden layer*,
- $y_j$  : nilai neuron pada *hidden layer*,
- $\omega_{jk}$  : vektor *weight* antara *hidden layer* dan *output layer*.

Langkah terakhir adalah memperbarui nilai *weight* di setiap *layer* sebagai berikut:

$$\omega_{l-1,l} = \omega_{l-1,l} - \eta \delta_l y_{l-1}, \quad (5)$$

atau

$$\theta = \theta - \eta \delta_l, \quad (6)$$

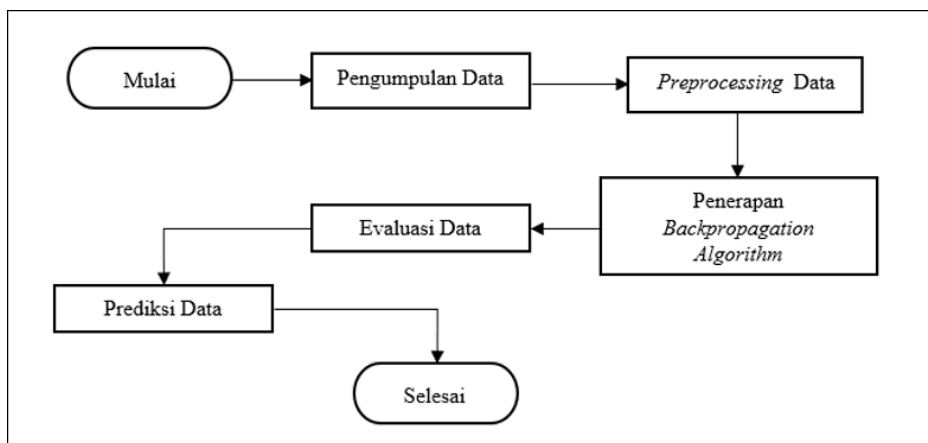
dengan:

- $\omega_{l-1,l} = \theta$  : nilai *weight*,
- $\eta$  : *learning rate*, yaitu laju pembelajaran yang digunakan untuk mempercepat atau memperlambat iterasi,
- $l$  : *layer*. Jika  $l$  adalah *output layer*, maka  $l - 1$  adalah *hidden layer*. Jika  $l$  adalah *hidden layer*, maka  $l - 1$  adalah *input layer*.

## 2.2. Data Penelitian

Data yang digunakan dalam penelitian ini adalah data banyaknya klaim dan besar masing-masing klaim asuransi perkapalan suatu perusahaan asuransi dalam kurun waktu 20 tahun, mulai tahun 2005 sampai 2024. Data tersebut merupakan hasil simulasi menggunakan *package Frank Copula* pada *software R*.

Data besar klaim diasumsikan berdistribusi *lognormal* karena distribusi ini memiliki ekor tebal (*heavy-tailed*) yang merupakan salah satu karakteristik data klaim asuransi [21]. Distribusi *lognormal* memiliki dua parameter, yaitu rata-rata ( $\mu$ ) dan standar deviasi ( $\sigma$ ). Nilai parameter  $\mu$  ditentukan sebesar Rp 2.354.500.000,- dan standar deviasi diasumsikan bernilai 2% dari rata-rata.



Gambar 1. Rancangan susunan algoritma Backpropagation

Nilai tersebut didasarkan pada Laporan Asuransi Umum dan Reasuransi tahun 2022, di mana total besar klaim yang dibayar asuransi perkapalan (*marine hull*) di Indonesia mencapai 967,70 miliar Rupiah dari 411 klaim [22]. Data simulasi dibangkitkan sebanyak 10.000 data untuk 20 tahun perkembangan dengan rincian 6.000 data *short-tailed claim* dan 4.000 data *long-tailed claim*.

### 2.3. Rancangan Penelitian

Metode yang dilakukan pada penelitian ini melalui berbagai tahapan secara sistematis agar mendapatkan hasil yang sesuai dengan tujuan yang diinginkan. Untuk lebih jelasnya diberikan *flowchart* pada Gambar 1.

#### 2.3.1. Pengumpulan Data

Pengumpulan data pada tahapan ini dilakukan menggunakan simulasi di *software R*. Adapun tahapan untuk membuat program simulasi data tersebut adalah sebagai berikut:

1. Menentukan nilai parameter distribusi, antara lain nilai rata-rata *ultimate claim* ( $\mu$ ), standar deviasi ( $\sigma$ ), banyaknya klaim ( $U$ ), dan jenis distribusi. Nilai rata-rata *ultimate claim* yang ditetapkan pada penelitian ini sebesar Rp 2.354.500.000,- yang merupakan rata-rata klaim yang dibayarkan asuransi perkapalan di Indonesia pada tahun 2022 [22]. Standar deviasi ditetapkan sebesar 2% dari rata-rata, sedangkan banyaknya klaim ditetapkan sebanyak 10.000. Data hasil simulasi diasumsikan mengikuti distribusi lognormal karena distribusi tersebut mampu menggambarkan karakteristik dari data klaim asuransi [21].
2. Membuat fungsi untuk membangkitkan rata-rata *ultimate claim* masing-masing tahun perkembangan berdasarkan *short-tailed claim* dan *long-tailed claim*. Rumus untuk fungsi tersebut adalah sebagai berikut [20]:

Fungsi *short-tailed claim* untuk *paid*:

$$1 - \left[ \exp \left( -\frac{t+1}{2} \right) \right]^{1.5} \tag{7}$$

Fungsi *long-tailed claim* untuk *paid*:

$$1 - \left[ \exp \left( -\frac{t+3}{6} \right) \right]^3, \tag{8}$$

dengan  $t$  merupakan waktu tahun perkembangan (*development year*),  $t = 1, 2, \dots, 20$ .

3. Membuat program *loop* untuk mensimulasikan *short-tailed claim* dan *long-tailed claim* berdistribusi lognormal menggunakan *Frank Copula*.
4. Setelah terbentuk masing-masing data *short-tailed claim* dan *long-tailed claim*, langkah selanjutnya adalah mensortir data klaim berdasarkan masing-masing tahun perkembangan.
5. Menyimpan data dalam format CSV sebagai data simulasi untuk Algoritma *Backpropagation*.

#### 2.3.2. Preprocessing Data

Tahapan *preprocessing* merupakan tahapan di mana data mentah diubah menjadi data terstruktur yang siap digunakan dengan tujuan untuk meningkatkan kualitas data agar dapat dimanfaatkan secara optimal dalam analisis lanjutan [23]. Tahapan ini mencakup dua langkah sebagai berikut:

1. Pembersihan Data  
Pembersihan data merupakan tahap deteksi sekaligus menghilangkan *error* dalam data, seperti salah pengejaan atau penginputan, terdapat *missing value*, ataupun ketidakvalidan data lainnya yang bertujuan untuk meningkatkan kualitas data [24]. Dalam penelitian ini tidak diperlukan pembersihan data karena data berupa simulasi sehingga tidak memiliki *missing value*.
2. Normalisasi Data  
Normalisasi data merupakan salah satu teknik penskalaan nilai sehingga data berada pada selang  $[0, 1]$  [25]. Hal ini diperlukan karena fungsi aktivasi sigmoid hanya dapat digunakan ketika bilangannya berada pada selang tersebut [26]. Normalisasi data dapat dihitung sebagai berikut:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{9}$$

dengan:

- $x$ : data sebenarnya
- $x'$ : data hasil normalisasi
- $x_{\max}$ : nilai maksimum pada data
- $x_{\min}$ : nilai minimum pada data.

#### 2.3.3. Penerapan Algoritma Backpropagation

Dalam tahapan ini, data dilatih menggunakan Algoritma *Backpropagation*. Adapun untuk ketentuannya, ditetapkan parameter-parameter yang digunakan untuk menjalankan Algo-

Tabel 1. Hasil simulasi data

Tahun Kejadian	Klaim ke-	Tahun Perkembangan			
		1	2	...	20
1	1	1.190.390.372	1.622.049.406	...	2.369.807.501
	...	...	...	...	...
2	500	271.569.807	426.287.133	...	2.209.007.099
	1	1.189.784.847	1.620.146.879	...	2.367.642.069
3	...	...	...	...	...
	500	260.936.134	409.646.230	...	2.123.534.011
...	1	1.174.641.475	1.600.378.863	...	2.336.223.900
	...	...	...	...	...
20	500	273.047.945	428.668.896	...	2.221.040.811
	...	...	...	...	...
20	500	266.642.260	418.354.128	...	2.166.701.013

ritma *Backpropagation*. Pertama, *hidden layer* ditetapkan sebanyak 2 neuron berdasarkan hasil penelitian yang telah dilakukan oleh Windarto [27] yang menyatakan bahwa *hidden layer* ini memiliki nilai *error* yang paling rendah. Kemudian, iterasi maksimum ditetapkan sebanyak 500 yang merupakan iterasi dengan keakuratan tertinggi [28]. Penetapan *weight* awal berkisar di antara selang  $[-0, 50; 0, 50]$  karena selang tersebut yang paling biasa digunakan [29]. Terakhir, *learning rate* ditetapkan sebesar 0,20 yang didasarkan pada hasil analisis yang dilakukan oleh [10]. Fungsi untuk *input layer* menggunakan fungsi kombinasi dengan inisialisasi *random weight*, dan fungsi untuk *hidden layer* serta *output layer* menggunakan fungsi aktivasi sigmoid. Fungsi ini dipilih karena memiliki rata-rata keakuratan model sebesar 99,60% dan *error* maksimum hanya sebesar 1% [30].

Tahapan penerapan Algoritma *Backpropagation* adalah sebagai berikut:

1. Melakukan perambatan maju (*feed forward*) dari *input layer* menuju *hidden layer* menggunakan pers. (1), dilanjutkan dengan fungsi aktivasi sigmoid menuju *output layer* menggunakan pers. (2).
2. Melakukan perambatan mundur (*feed backward*) dari *output layer* kembali ke *input layer* menggunakan pers. (3)–pers. (6) secara berurutan.
3. Mengulangi proses *feed forward* dan *feed backward* hingga mencapai iterasi maksimum sebanyak 500 kali.

Setelah proses Algoritma *Backpropagation* selesai, data dikembalikan ke interval awal menggunakan denormalisasi data sebagai berikut:

$$\hat{x} = x'(x'_{\max} - x'_{\min}) + x'_{\min}, \tag{10}$$

dengan:

- $\hat{x}$ : nilai data setelah denormalisasi
- $x'$ : data hasil Algoritma *Backpropagation*
- $x'_{\max}$ : nilai maksimum data hasil Algoritma *Backpropagation*
- $x'_{\min}$ : nilai minimum data hasil Algoritma *Backpropagation*.

### 2.3.4. Evaluasi Data

Tahapan ini bertujuan untuk mengecek keakuratan model data hasil Algoritma *Backpropagation* terhadap data sebenarnya menggunakan *Mean Absolute Percentage Error (MAPE)* [31]. Terdapat empat kriteria kelayakan *MAPE*, yaitu sangat baik, baik, layak,

dan tidak layak [32]. Jika nilai *MAPE* kurang dari 10%, maka data dikatakan sangat baik; jika berada pada rentang 10–20% maka dikatakan baik; jika berada pada rentang 20–50% maka dikatakan layak; dan jika melebihi 50% maka data tidak layak. Rumus *MAPE* adalah sebagai berikut:

$$MAPE = \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \times 100\%, \tag{11}$$

dengan:

- $y$ : nilai data hasil Algoritma *Backpropagation*
- $x$ : nilai data sebenarnya
- $n$ : banyaknya data
- $i$ : indeks data ke- $i$ ,  $i = 1, 2, \dots, n$ .

### 2.3.5. Prediksi Data

Setelah data dievaluasi, dilakukan prediksi menggunakan Algoritma *Backpropagation* untuk mengestimasi cadangan klaim pada tahun perkembangan ke-21 dengan tahapan yang sama seperti pada *training*. Proses prediksi dimulai dengan *feed forward* menggunakan pers. (1) dan pers. (2), kemudian dilanjutkan dengan *feed backward* menggunakan pers. (3) dan pers. (4). Selanjutnya, nilai *weight* diperbarui menggunakan pers. (5) dan pers. (6) dan proses diulangi hingga mencapai iterasi maksimum sebanyak 500 kali. Hasil prediksi disajikan dalam bentuk interval dengan tingkat kepercayaan 95% karena menghasilkan nilai *MAPE* sekitar 1–2% [33, 34].

## 3. Hasil dan Pembahasan

Setelah melalui berbagai tahapan pada bagian metode, diperoleh hasil masing-masing tahapan penelitian sebagai berikut.

### 3.1. Simulasi Data

Pada tahapan ini, dilakukan simulasi data menggunakan *software R* untuk membuat data historis klaim asuransi perkapalan menggunakan pers. (7) dan pers. (8). Hasil simulasi data yang diperoleh disajikan pada Tabel 1.

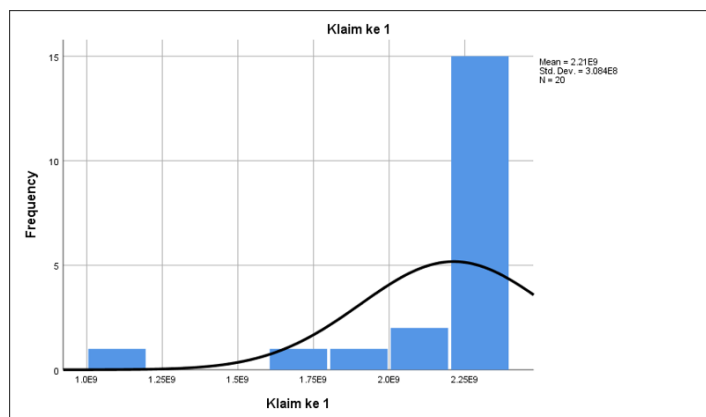
Hasil simulasi menghasilkan masing-masing 500 klaim dari 20 tahun kejadian dengan 20 tahun perkembangan yang berbeda. Sebagai contoh, baris pertama kolom pertama menunjukkan cadangan klaim yang harus dipersiapkan untuk tahun kejadian

Tabel 2. Statistik deskriptif

Tahun Kejadian	Klaim ke-	Nilai Minimum	Nilai Maksimum	Mean	Skewness	Kurtosis
1	1	1.190.390.372	2.371.195.305	2.213.331.586	-2,522	6,312
	⋮	⋮	⋮	⋮	⋮	⋮
	500	271.569.807	2.209.007.099	1.505.602.969	-0,678	-0,830
2	1	1.189.784.847	2.367.642.069	2.211.319.527	-2,517	6,284
	⋮	⋮	⋮	⋮	⋮	⋮
	500	1.062.360.410	9.053.077.553	2.746.495.040	2,074	3,566
⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	500	1.085.058.888	9.242.504.693	2.804.902.466	2,073	3,561

pertama yang akan dibayarkan pada tahun perkembangan pertama senilai Rp 1.190.390.372,-. Selain itu, dari hasil simulasi pada Tabel 1, diperoleh beberapa informasi terkait klaim yang disajikan pada Tabel 2.

Hasil Statistik Deskriptif menunjukkan masing-masing banyaknya data, range, nilai minimum, nilai maksimum, mean, standar deviasi, variansi, skewness, dan kurtosis untuk masing-masing klaim pada masing-masing tahun perkembangan berdasarkan tahun kejadian. Sebagai contoh, pada tahun perkembangan ke-1 memiliki nilai minimum di angka 1.190.390.372 dan nilai maksimum pembayarannya sebesar 2.371.195.305. Nilai mean dari tahun perkembangan ini sekitar 2.213.331.586 dengan standar deviasi 447.500.000, yang menunjukkan data menyimpang sekitar 54% dari mean. Skewness dan kurtosis akan dijelaskan pada Gambar 2.

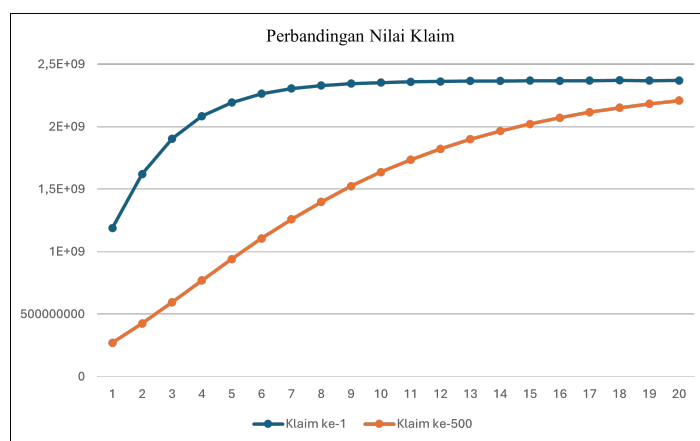


Gambar 2. Perbandingan klaim pada tahun kejadian pertama

Skewness dari tahun perkembangan ini sebesar -2,522 yang artinya rata-rata (mean) pada data dipengaruhi oleh nilai ekstrem yang lebih kecil dari mean, dan pada gambar di atas nilai ekstremnya sangat jelas dipengaruhi oleh nilai yang jauh lebih kecil dari mean. Jika dilihat pada plot histogram di atas, nilai mean terdapat pada hampir kebanyakan data. Akibat dari pengaruh nilai ekstrem dan banyaknya data yang berada pada nilai mean tersebut, nilai kurtosis menjadi positif, yakni sekitar 6,312. Akibat dari nilai skewness dan kurtosis tersebut, grafik ini merupakan salah satu contoh klaim long-tailed.

Hasil simulasi pada Tabel 1 menunjukkan bahwa terdapat 10.000 klaim yang masing-masing dibagi menjadi 500 klaim per tahun dengan 20 tahun perkembangan. Terdapat perbedaan yang

cukup signifikan antara klaim ke-1 dan klaim ke-500 pada masing-masing tahun kejadian. Diberikan satu contoh berikut untuk mengilustrasikan lebih jelas mengenai perbedaan kedua data tersebut (lihat Gambar 3).

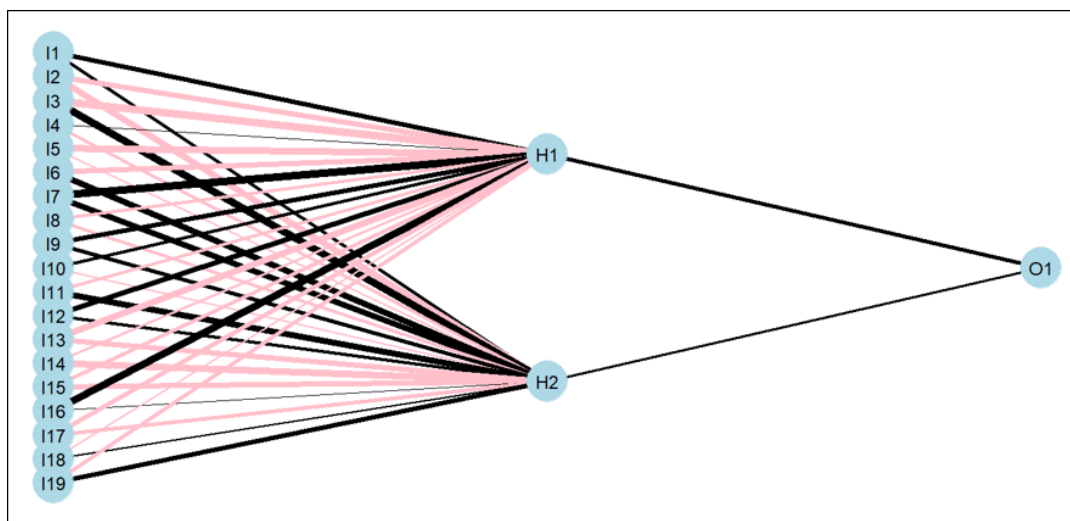


Gambar 3. Perbandingan klaim pada tahun kejadian pertama

Berdasarkan grafik distribusi data pada Gambar 2, klaim ke-1 lebih besar dibandingkan klaim ke-500. Perbedaan karakteristik ini menjadi pertimbangan adanya data klaim short-tailed dan long-tailed. Data klaim ke-1 merupakan klaim jenis short-tailed yang cenderung memiliki nilai pembayaran yang jauh lebih besar pada tahun perkembangan pertama jika dibandingkan dengan data klaim ke-500 yang bersifat long-tailed. Akan tetapi, pada perkembangan tahun pembayarannya, nilai pembayaran yang dilakukan oleh klaim long-tailed lebih besar jika dibandingkan dengan perkembangan pembayaran klaim short-tailed. Selain itu, nilai klaim ke-500 pada tahun perkembangan pertama sangat jauh di bawah nilai pada tahun perkembangan lainnya. Hal ini menunjukkan bahwa salah satu karakteristik data klaim long-tailed adalah memiliki peluang yang cukup besar untuk menghasilkan outlier, yaitu nilai ekstrem yang berada jauh dari sebagian besar nilai lainnya dalam suatu data.

### 3.2. Hasil Preprocessing Data

Setelah data simulasi berhasil dibangkitkan, tahap selanjutnya adalah preprocessing data menggunakan metode normalisasi sebagaimana ditunjukkan pada Persamaan (9). Normalisasi ini bertujuan untuk mentransformasikan nilai besar klaim ke dalam selang [0, 1].



Gambar 4. Arsitektur algoritma Backpropagation

Dalam implementasinya menggunakan perangkat lunak R, proses normalisasi hanya dapat dilakukan secara per kolom. Oleh karena itu, pada penelitian ini normalisasi data dilakukan berdasarkan tahun perkembangan, di mana setiap kolom merepresentasikan satu tahun perkembangan.

Pada tahapan normalisasi data, diperlukan nilai minimum dan nilai maksimum untuk setiap tahun perkembangan. Nilai minimum dan maksimum yang diperoleh untuk masing-masing tahun perkembangan disajikan pada Tabel 3.

Tabel 3. Nilai minimum dan nilai maksimum pada data

Tahun Perkembangan	Nilai Minimum	Nilai Maksimum
1	251.745.359	1.266.543.943
2	395.225.203	1.725.291.111
3	551.683.248	2.026.954.746
4	713.627.608	2.216.051.766
5	873.239.975	2.334.696.164
6	1.024.502.285	2.406.843.586
7	1.167.177.031	2.450.442.830
8	1.295.529.486	2.477.582.093
9	1.413.283.221	2.495.369.637
10	1.518.908.859	2.505.026.148
11	1.609.996.690	2.511.026.744
12	1.690.588.060	2.515.032.400
13	1.761.656.104	2.516.731.686
14	1.822.780.766	2.518.784.738
15	1.875.584.156	2.520.275.723
16	1.920.681.419	2.519.180.407
17	1.960.024.116	2.519.258.227
18	1.992.124.973	2.519.319.718
19	2.023.094.101	2.519.189.023
20	2.046.730.800	2.519.097.708

Nilai maksimum dan minimum pada Tabel 3, digunakan dalam persamaan (9) untuk menghasilkan data preprocessing pada Tabel 4.

Hasil preprocessing data mengubah nilai klaim ke bentuk selang  $[0, 1]$  sehingga dapat dilanjutkan ke penerapan algoritma Backpropagation. Pada Tabel 1, dapat dilihat bahwa nilai klaim pada tahun perkembangan ke-2 lebih besar daripada nilai klaim pada tahun perkembangan pertama. Akan tetapi, setelah dinor-

Tabel 4. Hasil preprocessing data

Tahun Kejadian	Klaim ke-	Tahun Perkembangan			
		1	2	...	20
1	1	0,924957	0,922379	...	0,683953
	500	0,01953535	0,02335368	...	0,3435387
2	1	0,92436027	0,92094810	...	0,6793687
	500	0,00905675	0,0108423	...	0,162592
3	1	0,90943773	0,9060857	...	0,612856
	500	0,02099194	0,0251444	...	0,369014
...	...	...	...	...	...
20	500	0,01467966	0,01738931	...	0,253977

malisasi pada Tabel 4, nilai klaim pada tahun perkembangan pertama memiliki nilai yang lebih besar daripada nilai klaim pada tahun perkembangan kedua.

### 3.3. Hasil Penerapan Algoritma Backpropagation

Setelah data simulasi melewati tahap preprocessing, maka Algoritma Backpropagation dapat diterapkan pada data. Penerapan Algoritma Backpropagation dilakukan menggunakan pers. (1) dan pers. (2) untuk feed forward, pers. (3) dan pers. (4) untuk feed backward, serta pers. (5) dan pers. (6) untuk mengubah nilai weight untuk setiap iterasi. Hasil penerapan Algoritma Backpropagation diberikan pada Gambar 4.

Arsitektur Algoritma Backpropagation yang digunakan dalam penelitian ini terdiri atas 19 neuron pada input layer, 2 neuron pada hidden layer, dan 1 neuron pada output layer. Jumlah neuron pada input layer ditetapkan sebanyak 19 karena Algoritma Backpropagation dilatih menggunakan data perkembangan klaim selama 20 tahun, sehingga satu tahun perkembangan dijadikan sebagai target keluaran. Berdasarkan hasil plotting jaringan Algoritma Backpropagation, ketebalan garis antar neuron merepresentasikan besar kontribusi masing-masing neuron. Garis yang

**Tabel 5.** Nilai *weight* akhir pada Algoritma *Backpropagation*

Hubungan Antar Layer	H1	H2	Output
I1	0,29133454	0,186489418	–
I2	-0,33736238	-0,407189220	–
I3	-0,49355206	0,499670625	–
I4	0,02439172	-0,177625924	–
I5	-0,46357095	-0,055277091	–
I6	-0,34712172	0,410478562	–
I7	0,50307012	0,384435594	–
I8	-0,18917483	-0,166544333	–
I9	0,27509210	0,233760342	–
I10	0,15786453	-0,086326629	–
I11	-0,18403253	0,388552815	–
I12	0,34239662	0,187849730	–
I13	-0,45426756	-0,361353040	–
I14	-0,13022378	-0,443317920	–
I15	-0,20625366	-0,317147851	–
I16	0,48756212	0,009325002	–
I17	-0,29305041	-0,211805552	–
I18	-0,05439015	0,039542496	–
I19	-0,23829705	0,256991953	–
H1	–	–	0,2405697
H2	–	–	0,1119946

**Tabel 6.** Hasil penerapan Algoritma *Backpropagation*

Tahun Kejadian	Klaim ke-	Tahun Perkembangan			
		1	2	...	20
1	1	0,9249570	0,9223785	...	0,6933681
	⋮	⋮	⋮	⋮	⋮
	500	0,019535353	0,023353677	...	0,3330666
	⋮	⋮	⋮	⋮	⋮
2	1	0,924360265	0,920948104	...	0,6895384
	⋮	⋮	⋮	⋮	⋮
	500	0,009056749	0,01084234	...	0,1924684
	⋮	⋮	⋮	⋮	⋮
3	1	0,909437726	0,90608567	...	0,6241160
	⋮	⋮	⋮	⋮	⋮
	500	0,020991936	0,02514439	...	0,3536233
	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
20	500	0,01915212	0,02290502	...	0,2906086

lebih tebal menunjukkan kontribusi yang lebih besar terhadap proses pembelajaran jaringan. Dapat diamati bahwa neuron input pertama (I1), ketiga (I3), keenam (I6), ketujuh (I7), kesembilan (I9), dan kesebelas (I11) pada *input layer* memiliki kontribusi yang lebih besar terhadap *hidden layer* dibandingkan dengan neuron input lainnya, seperti neuron kedua (I2). Sebaliknya, kontribusi neuron pada *hidden layer* terhadap *output layer* tidak menunjukkan perbedaan ketebalan garis yang signifikan. Hal ini mengindikasikan bahwa kedua neuron pada *hidden layer*, yaitu H1 dan H2, memiliki kontribusi yang relatif sama besar dalam menghasilkan keluaran jaringan. Adapun hasil penerapan Algoritma *Backpropagation* disajikan pada [Tabel 5](#).

[Tabel 5](#) menunjukkan hubungan antara *input layer* dengan *hidden layer* melalui nilai *weight*. Apabila nilai *weight* semakin positif mendekati angka 1, maka kontribusi nilai tersebut adalah untuk mendorong nilai fungsi kombinasi. Sebagai contoh, nilai I7 memiliki *weight* sebesar 0.50307012 terhadap H1. Artinya, I7 mendorong fungsi kombinasi untuk mendekati angka 1 sehingga

neuron pada H1 semakin aktif. Sebaliknya, apabila nilai *weight* semakin negatif mendekati angka  $-1$ , maka kontribusi *weight* tersebut adalah untuk menekan fungsi kombinasi mendekati angka 0 yang menyebabkan neuron menjadi tidak aktif. Contohnya nilai *weight* antara I14 dengan H2 sebesar  $-0.443317920$  yang menekan fungsi mendekati 0 sehingga menyebabkan H2 semakin tidak aktif. Adapun hasil penerapan Algoritma *Backpropagation* dapat dilihat pada [Tabel 6](#).

Berdasarkan hasil [Tabel 6](#), tidak ada perubahan yang cukup signifikan jika dibandingkan dengan [Tabel 4](#). Misalkan, besar klaim hasil normalisasi pada tahun kejadian pertama yang dibayarkan pada tahun perkembangan pertama menunjukkan nilai 0,9249570 pada [Tabel 4](#) dan [Tabel 6](#). Begitu juga dengan klaim tahun kejadian pertama yang dibayarkan pada tahun ke-2 dan ke-3 menunjukkan nilai yang sama pada masing-masing tabel. Akan tetapi, nilai yang diperoleh pada [Tabel 6](#) masih berada dalam selang  $[0, 1]$ .

Tabel 7. Hasil denormalisasi data

Tahun Kejadian	Klaim ke-	Tahun Perkembangan			
		1	2	...	20
1	1	1.190.390.372	1.622.049.406	...	2.374.254.934
	...	...	...	...	...
500	500	271.569.807	426.287.133	...	2.204.060.431
	...	...	...	...	...
2	1	1.189.784.847	1.620.146.879	...	2.372.445.902
	...	...	...	...	...
500	500	260.936.134	409.646.230	...	2.137.646.512
	...	...	...	...	...
3	1	1.174.641.475	1.600.378.863	...	2.341.542.546
	...	...	...	...	...
500	500	273.047.945	428.668.896	...	2.213.770.731
	...	...	...	...	...
...	...	...	...	...	...
20	500	271.180.903	425.690.384	...	2.184.004.707

Tabel 8. Perbandingan klaim ke-500 di tahun kejadian ke-20

Tahun Perkembangan	Sebelum Algoritma <i>Backpropagation</i>	Setelah Algoritma <i>Backpropagation</i>
1	266.642.260	271.180.903
2	418.354.128	425.690.384
3	584.532.329	594.745.195
4	755.828.426	768.791.060
5	924.250.469	940.231.628
6	1.085.058.888	1.103.696.720
7	1.235.563.833	1.256.717.610
8	1.372.915.340	1.395.995.504
9	1.496.372.220	1.519.847.431
10	1.605.781.238	1.629.488.099
11	1.703.670.165	1.726.487.106
12	1.789.344.642	1.811.879.991
13	1.865.196.417	1.886.297.036
14	1.930.232.288	1.951.628.009
15	1.985.010.126	2.008.835.810
16	2.033.546.089	2.054.664.272
17	2.076.063.316	2.095.893.906
18	2.113.214.327	2.130.179.841
19	2.143.303.315	2.160.205.083
20	2.166.701.013	2.184.004.707

3.4. Hasil Denormalisasi Data

Untuk mengembalikan ke nilai awal sebelum *preprocessing* data maka perlu digunakan *denormalisasi* data seperti pada pers. (10). Hasil *denormalisasi* data dapat dilihat pada Tabel 7.

Berdasarkan Tabel 7, dilakukan perbandingan antara data hasil denormalisasi menggunakan *Backpropagation* Algorithm dengan data awal sebelum dilakukan normalisasi, sebagaimana disajikan pada Tabel 1. Proses perbandingan ini serupa dengan analisis yang dilakukan pada perbandingan antara Tabel 4 dan Tabel 6. Hasil pengamatan menunjukkan bahwa secara umum belum terdapat selisih yang signifikan antara nilai klaim pada Tabel 1 dan Tabel 7. Namun demikian, perbedaan yang cukup mencolok terlihat pada klaim ke-500 pada tahun kejadian ke-20, di mana nilai klaim yang dibayarkan pada Tabel 7 lebih rendah dibandingkan dengan nilai klaim pada Tabel 1. Untuk memberikan gambaran yang lebih lengkap, perbandingan antara nilai klaim sebelum dan setelah penerapan Algoritma *Backpropagation* disajikan pada Tabel 8.

Berdasarkan Tabel 8, dapat diketahui bahwa semakin lama tahun perkembangan, maka semakin besar selisih antara nilai cadangan klaim sebelum penerapan Algoritma *Backpropagation* dengan nilai cadangan klaim setelah penerapan Algoritma *Backpropagation*. Kondisi ini mengindikasikan adanya potensi error dalam hasil pemodelan menggunakan Algoritma *Backpropagation*. Oleh karena itu, untuk mengetahui besarnya error yang dihasilkan oleh model, pada tahapan selanjutnya akan digunakan ukuran evaluasi berupa *Mean Absolute Percentage Error (MAPE)*.

3.5. Hasil Evaluasi Data

Tahapan sebelum melakukan prediksi data adalah mengevaluasi kelayakan model menggunakan *Mean Absolute Percentage Error (MAPE)* pada pers. (11). Suatu model dikatakan layak digunakan sebagai acuan prediksi apabila nilai *MAPE* kurang dari 50%. Hasil evaluasi model menggunakan *MAPE* disajikan pada Tabel 9.

Hasil evaluasi menunjukkan bahwa error tertinggi berada pada tahun perkembangan ke-20, yakni sebesar 1,04%, sedangk-

**Tabel 9.** Hasil uji kelayakan data

Tahun Perkembangan	MAPE (%)	Tahun Perkembangan	MAPE (%)
1	0,03777484	11	0,63511420
2	0,08133588	12	0,69299815
3	0,12331978	13	0,77729078
4	0,17157014	14	0,78006041
5	0,22482131	15	0,77957393
6	0,26612559	16	0,88828949
7	0,30444249	17	0,91943936
8	0,36532423	18	0,97344275
9	0,45328384	19	0,98271553
10	0,55559066	20	1,04392417

**Tabel 10.** Hasil prediksi data

Tahun Kejadian	Klaim ke-	Tahun Perkembangan		
		21	Rata-rata	Total
1	1	2.374.254.934	2.295.555.384	1.147.778.000.000
	⋮	⋮		
	500	2.204.060.431		
2	1	2.372.445.902	2.298.394.099	1.149.197.000.000
	⋮	⋮		
	500	2.137.646.512		
3	1	2.341.542.546	2.299.138.409	1.149.569.000.000
	⋮	⋮		
	500	2.213.770.731		
⋮	⋮	⋮	⋮	⋮
20	500	2.184.004.707	2.294.436.230	1.147.218.000.000
Rata-rata			2.294.545.152	1.147.273.000.000
Total				22.945.450.000.000

an error terkecil berada pada tahun perkembangan pertama sebesar 0,04%. Nilai error yang ditampilkan merupakan nilai MAPE per tahun perkembangan. Secara keseluruhan, nilai MAPE dapat dihitung berdasarkan rata-rata nilai MAPE per tahun perkembangan. Hasil rata-rata dari MAPE diperoleh sebesar 0,56%, yang menunjukkan bahwa nilai error kurang dari 10%, sehingga Algoritma yang digunakan sangat baik. Selanjutnya, model ini dapat digunakan untuk memprediksi cadangan klaim pada tahun perkembangan ke-21.

**3.6. Hasil Prediksi Data**

Hasil evaluasi data menunjukkan nilai MAPE yang sangat kecil sehingga Algoritma yang digunakan dapat diaplikasikan untuk memprediksi cadangan klaim pada tahun perkembangan ke-21 dengan menggunakan pers. (1)–pers. (6). Hasil prediksi cadangan klaim pada tahun perkembangan ke-21 disajikan pada Tabel 10.

Berdasarkan Tabel 10, diperoleh hasil prediksi besar pembayaran klaim untuk tahun perkembangan ke-21 berdasarkan tahun kejadian ke-1 sampai ke-20 yang diasumsikan terdapat 500 klaim pada masing-masing tahun kejadian. Hasil prediksi menunjukkan bahwa rata-rata pembayaran atas klaim yang terjadi pada tahun kejadian pertama adalah sebesar Rp2.295.555.384. Untuk klaim yang terjadi pada tahun kejadian ke-2, rata-rata pembayaran sebesar Rp2.298.394.099,- dan seterusnya hingga tahun kejadian ke-20, yang dapat dilihat pada kolom rata-rata pada Tabel 10. Selain itu, estimasi jumlah cadangan klaim

yang harus disiapkan oleh perusahaan asuransi adalah sebesar Rp22.945.450.000.000,- dengan rata-rata pembayaran per tahun kejadian sebesar Rp1.147.273.000.000,-. Dengan tingkat kepercayaan interval sebesar 95%, diperoleh hasil estimasi klaim yang harus dibayarkan sebagaimana ditunjukkan pada Tabel 11.

Hasil klaim dengan tingkat kepercayaan 95% mengestimasi bahwa klaim pertama pada tahun kejadian pertama berada di antara Rp2.314.898.561,- hingga Rp2.433.611.308,-. Begitu pula dengan klaim pertama pada tahun kejadian kedua yang berada di antara Rp2.313.134.755,- sampai dengan Rp2.431.757.050,- dan seterusnya, sehingga diperoleh hasil utama berupa rata-rata klaim yang berada pada rentang Rp2.237.181.523,- hingga Rp2.351.908.780,- jika dihitung per klaim dan berada di antara Rp1.118.591.000.000,- dan Rp1.175.954.000.000,- jika dihitung per tahun kejadian. Total keseluruhan cadangan klaim yang harus dibayarkan mulai dari tahun kejadian pertama hingga tahun kejadian ke-20 pada tahun perkembangan ke-21 diperkirakan berada pada rentang Rp22.371.820.000.000,- hingga Rp23.519.090.000.000,-.

**4. Kesimpulan**

Sebagai bagian dari machine learning, Algoritma Backpropagation dapat digunakan pada bisnis asuransi untuk mengestimasi cadangan klaim. Penelitian ini berhasil memodelkan pola pembayaran klaim berjenis short-tailed maupun long-tailed secara individu untuk masing-masing tahun perkembangan

Tabel 11. Hasil prediksi data menggunakan kepercayaan interval 95%

Tahun Kejadian	Klaim ke-	Perkiraan Tahun Perkembangan ke-21	Rentang Minimum	Rentang Maksimum
1	1	2.374.254.934	2.314.898.561	2.433.611.308
	⋮	⋮	⋮	⋮
	500	2.204.060.431	2.148.958.921	2.259.161.942
2	1	2.372.445.902	2.313.134.755	2.431.757.050
	⋮	⋮	⋮	⋮
	500	2.137.646.512	2.084.205.349	2.191.087.675
3	1	2.341.542.546	2.283.003.983	2.400.081.110
	⋮	⋮	⋮	⋮
	500	2.213.770.731	2.158.426.463	2.269.114.999
⋮	⋮	⋮	⋮	⋮
20	500	2.184.004.707	2.129.404.589	2.238.604.824
Rata-rata (per klaim)		2.294.545.152	2.237.181.523	2.351.908.780
Rata-rata (per tahun kejadian)		1.147.273.000.000	1.118.591.000.000	1.175.954.000.000
Total		22.945.450.000.000	22.371.820.000.000	23.519.090.000.000

an dari bisnis asuransi perkapalan. Algoritma *Backpropagation* diterapkan untuk membentuk pola pembayaran tersebut dan menghasilkan nilai *MAPE* sebesar 0,55%, sehingga model dikatakan sangat baik. Model tersebut digunakan untuk memprediksi cadangan klaim pada tahun pembayaran ke-21 dan diperoleh hasil prediksi mulai dari Rp22.371.820.000.000,- sampai dengan Rp23.519.090.000.000,- untuk total keseluruhan klaim yang harus dibayarkan pada tahun perkembangan ke-21. Rata-rata nilai klaim yang harus dibayarkan berada pada rentang Rp2.237.181.523,- hingga Rp2.351.908.780,- dengan rata-rata pembayaran per tahun kejadian sebesar Rp1.118.591.000.000,- sampai dengan Rp1.175.954.000.000,-.

Karena keterbatasan dalam pencarian data asuransi perkapalan, penelitian ini hanya menggunakan simulasi sehingga belum dapat digeneralisasi ke praktik industri riil. Selanjutnya, karena terdapat keterbatasan parameter pada penelitian ini, diharapkan pada penelitian selanjutnya dapat menggunakan estimasi parameter seperti metode momen untuk memperoleh nilai  $\mu$  dan  $\sigma$  yang lebih akurat. Selain itu, kekurangan lain yang ditemukan pada penelitian ini adalah tidak adanya pengujian parameter *hidden layer*, *learning rate*, dan parameter lainnya yang dapat digunakan untuk membentuk model yang lebih baik. Oleh karena itu, peneliti merekomendasikan penelitian selanjutnya untuk menguji beberapa parameter lain sehingga diperoleh model yang lebih akurat dalam menghitung cadangan klaim. Penelitian ini juga terbatas pada data yang variansinya tidak terlalu jauh dari *mean*, karena ketika dicoba menggunakan data dengan standar deviasi yang berbeda dari 2%, nilai *MAPE* yang dihasilkan terlalu besar sehingga model ini tidak layak digunakan. Hal ini dapat dikaji lebih lanjut pada penelitian-penelitian berikutnya.

**Kontribusi Penulis.** Anas Satriya Rabdika: Konseptualisasi, metodologi, perangkat lunak, analisis formal, investigasi, sumber daya, penulisan—persiapan draf asli. Azizah: Konseptualisasi, metodologi, validasi, kurasi data, penulisan—tinjauan dan penyuntingan, supervisi. Semua penulis telah membaca dan menyetujui versi manuskrip yang diterbitkan.

**Ucapan Terima Kasih.** Ucapan terima kasih ditujukan kepada Universitas Negeri Malang yang telah memfasilitasi dengan berbagai sarana selama

pelaksanaan penelitian, serta kepada reviewer dan editor Jurnal yang telah memberikan masukan serta dukungan dalam meningkatkan kualitas naskah ini.

**Pembiayaan.** Penelitian ini tidak menerima pendanaan eksternal.

**Konflik Kepentingan.** Para penulis menyatakan tidak ada konflik kepentingan yang terkait dengan artikel ini.

**Ketersediaan Data.** Tidak tersedia.

Referensi

- [1] A. N. Mandia, J. K. Mungatu, and P. G. O. Weke, "Measuring multivariate negative dependence in higher dimensions using copula-based approach," *UJAM*, vol. 12, no. 2, pp. 50–61, Apr. 2024, doi: 10.13189/ujam.2024.120202.
- [2] S. Srun, T. Kean, C. Khim, M. S. Rakitina, I. A. Dmitrieva, and N. A. Ivanova, "Evaluation of claims reserve methods: claims payment pattern and standard-Bühlmann, case study at prévoir insurance company (agriculture Cambodia)," *E3S Web Conf.*, vol. 486, p. 03015, 2024, doi: 10.1051/e3sconf/202448603015.
- [3] S. C. Vanegas, A. L. Badescu, and X. S. Lin, "Claim reserving via inverse probability weighting: a micro-level Chain-Ladder method," *Eur. Actuar. J.*, vol. 15, no. 1, pp. 89–135, Apr. 2025, doi: 10.1007/s13385-024-00395-3.
- [4] I. Chaoubi, C. Besse, H. Cossette, and M. Côté, "Micro-level reserving for general insurance claims using a long short-term memory network," *Appl. Stoch. Models Bus. Ind.*, vol. 39, no. 3, pp. 382–407, May 2023, doi: 10.1002/asmb.2750.
- [5] H. Abdelrahman, A. Badescu, R. Craiu, and S. Lin, "Marked Cox models for IBNR claims count: continuous and discretized approaches with Dirichlet-driven reporting delays," arXiv:2409.12896, Sep. 2024, doi: 10.48550/arXiv.2409.12896.
- [6] H. Surden, "Machine learning and law: An overview," in *Research Handbook on Big Data Law*, R. Vogl, Ed. Edward Elgar Publishing, 2021, doi: 10.4337/9781788972826.00014.
- [7] Ł. Delong, M. Lindholm, and M. V. Wüthrich, "Collective reserving using individual claims data," *Scandinavian Actuarial Journal*, vol. 2022, no. 1, pp. 1–28, Jan. 2022, doi: 10.1080/03461238.2021.1921836.
- [8] M. V. Wüthrich, "Machine learning in individual claims reserving," *Scandinavian Actuarial Journal*, vol. 2018, no. 6, pp. 465–480, Jul. 2018, doi: 10.1080/03461238.2018.1428681.
- [9] S. T. Hassan, Parvin, Asha, A. Datta, and S. Mondal, "Use of machine learning classifiers on structured vs unstructured data," *IJSREM*, vol. 8, no. 3, pp. 1–5, Mar. 2024, doi: 10.55041/IJSREM29771.
- [10] S. Syaharuddin, F. Fatmawati, and H. Suprajitno, "Best architecture recommendations of ANN backpropagation based on combination of learning rate,

- momentum, and number of hidden layers," *JTAM*, vol. 6, no. 3, p. 629, Jul. 2022, doi: [10.31764/jtam.v6i3.8524](https://doi.org/10.31764/jtam.v6i3.8524).
- [11] E. Bravo-López, T. Fernández Del Castillo, C. Sellers, and J. Delgado-García, "Landslide susceptibility mapping with artificial neural networks: Multi-approach analysis of backpropagation algorithm applying the neuralnet package in Cuenca, Ecuador," *Remote Sensing*, vol. 14, no. 14, p. 3495, Jul. 2022, doi: [10.3390/rs14143495](https://doi.org/10.3390/rs14143495).
- [12] G. G. Tolun and Y. A. Kaplan, "Development of backpropagation algorithm for estimating solar radiation: A case study in Turkey," *RRST-EE*, vol. 68, no. 3, pp. 313–316, Oct. 2023, doi: [10.59277/RRST-EE.2023.3.11](https://doi.org/10.59277/RRST-EE.2023.3.11).
- [13] J. M. Ahuja, M. Sharma, and V. Birchha, "Solar energy prediction using LM-back-propagation in ANN," *IOSR Journal of Computer Engineering*, vol. 21, no. 1, pp. 1–7, 2019.
- [14] M. Li, "Comprehensive review of backpropagation neural networks," *AJST*, vol. 9, no. 1, pp. 150–154, Jan. 2024, doi: [10.54097/51y16r47](https://doi.org/10.54097/51y16r47).
- [15] M. Hofert, "The philosophy of copula modeling: A conversation with ChatGPT," *Journal of Data Science*, vol. 21, no. 4, pp. 619–637, 2023, doi: [10.6339/23-JDS1114](https://doi.org/10.6339/23-JDS1114).
- [16] M. G. Cruz, G. W. Peters, and P. V. Shevchenko, *Fundamental Aspects of Operational Risk and Insurance Analytics: A Handbook of Operational Risk*. Hoboken, NJ, USA: John Wiley & Sons, 2015.
- [17] I. M. Widiartha, A. A. N. Gunawan, E. R. N. A. Sanjaya, and K. Sari, "A hybrid method of backpropagation and particle swarm optimization for enhancing accuracy performance," *CJAST*, vol. 42, no. 6, pp. 10–18, Mar. 2023, doi: [10.9734/cjast/2023/v42i64072](https://doi.org/10.9734/cjast/2023/v42i64072).
- [18] K. Markov, "Multilayer perceptron with backpropagation, HDL coder, and FPGA technology: An integrated approach for efficient neural network implementation," *PECR*, vol. 80, Dec. 2023, doi: [10.7546/PECR.80.23.02](https://doi.org/10.7546/PECR.80.23.02).
- [19] A. Apicella, F. Isgrò, and R. Prevete, "Hidden classification layers: Enhancing linear separability between classes in neural network layers," *Pattern Recognition Letters*, vol. 177, pp. 69–74, Jan. 2024, doi: [10.1016/j.patrec.2023.11.016](https://doi.org/10.1016/j.patrec.2023.11.016).
- [20] D. Qiu, "Individual claims reserving: Using machine learning methods," M.S. thesis, Concordia Univ., Montreal, QC, Canada, Dec. 2019. [Online]. Available: <https://spectrum.library.concordia.ca/id/eprint/986258/>
- [21] Mukhaiyar, Utriweni, A. Dianpermatasari, A. Dzakiya, S. B. Widayani, and H. K. Syam, "The value at risk analysis using heavy-tailed distribution on the insurance claims data," *JTAM*, vol. 8, no. 4, p. 1233, Oct. 2024, doi: [10.31764/jtam.v8i4.25053](https://doi.org/10.31764/jtam.v8i4.25053).
- [22] Asosiasi Asuransi Umum Indonesia, "Asuransi umum & reasuransi tahun 2022 (unaudited)," Mar. 2023.
- [23] B. Konda, "The impact of data preprocessing on data mining outcomes," *World J. Adv. Res. Rev.*, vol. 15, no. 3, pp. 540–544, Sep. 2022, doi: [10.30574/wjarr.2022.15.3.0931](https://doi.org/10.30574/wjarr.2022.15.3.0931).
- [24] D. Chicco, L. Oneto, and E. Tavazzi, "Eleven quick tips for data cleaning and feature engineering," *PLoS Comput. Biol.*, vol. 18, no. 12, p. e1010718, Dec. 2022, doi: [10.1371/journal.pcbi.1010718](https://doi.org/10.1371/journal.pcbi.1010718).
- [25] I. Izonin, R. Tkachenko, N. Shakhovska, B. Ilchyshyn, and K. K. Singh, "A two-step data normalization approach for improving classification accuracy in the medical diagnosis domain," *Mathematics*, vol. 10, no. 11, p. 1942, Jun. 2022, doi: [10.3390/math10111942](https://doi.org/10.3390/math10111942).
- [26] D. Marlina and F. Arifin, "Predicting the number of tourists based on backpropagation algorithm," *RESTI*, vol. 5, no. 3, pp. 439–445, Jun. 2021, doi: [10.29207/resti.v5i3.3061](https://doi.org/10.29207/resti.v5i3.3061).
- [27] A. P. Windarto, "Implementasi JST dalam menentukan kelayakan nasabah pinjaman KUR pada Bank Mandiri Mikro Serbelawan dengan metode backpropagation," *J-SAKTI*, vol. 1, no. 1, pp. 12–23, Aug. 2017, doi: [10.30645/j-sakti.v1i1.25](https://doi.org/10.30645/j-sakti.v1i1.25).
- [28] A. B. Pratama, "Rainfall forecasting in Lampung Province using backpropagation neural network," B.S. thesis, Univ. Lampung, Bandar Lampung, Indonesia, 2023.
- [29] F. Khairati and H. Putra, "Prediksi kuantitas penggunaan obat pada layanan kesehatan menggunakan algoritma backpropagation neural network," *JSI-SFOOTEK*, pp. 128–135, Aug. 2022, doi: [10.37034/jsisfotek.v4i3.158](https://doi.org/10.37034/jsisfotek.v4i3.158).
- [30] V. Shatravin, D. Shashev, and S. Shidlovskiy, "Sigmoid activation implementation for neural networks hardware accelerators based on reconfigurable computing environments for low-power intelligent systems," *Applied Sciences*, vol. 12, p. 5216, May 2022, doi: [10.3390/app12105216](https://doi.org/10.3390/app12105216).
- [31] R. G. Santosa, A. R. Chrismanto, Y. Lukito, and W. S. Raharjo, "The MAPE analysis of ARIMA (p,d,q) on LQ45 stock price to determine training data period," *IJITCSA*, vol. 2, no. 3, pp. 182–201, Dec. 2024, doi: [10.58776/ijitcsa.v2i3.168](https://doi.org/10.58776/ijitcsa.v2i3.168).
- [32] I. Nabillah and I. Ranggadara, "Mean absolute percentage error untuk evaluasi hasil prediksi komoditas laut," *J. Inform. Syst.*, vol. 5, no. 2, pp. 250–255, Nov. 2020, doi: [10.33633/joins.v5i2.3900](https://doi.org/10.33633/joins.v5i2.3900).
- [33] N. Susyanto and J. P. Arcede, "Unveiling SIR model parameters: Empirical parameter approach for explicit estimation and confidence interval construction," *Jambura J. Biomath.*, vol. 5, no. 1, pp. 54–62, Jul. 2024, doi: [10.37905/jjbm.v5i1.26287](https://doi.org/10.37905/jjbm.v5i1.26287).
- [34] D. Rodin, Y. Shapiro, A. Pinhasov, A. Kreinin, and M. Kirby, "An accurate wearable hydration sensor: Real-world evaluation of practical use," *PLoS ONE*, vol. 17, no. 8, p. e0272646, Aug. 2022, doi: [10.1371/journal.pone.0272646](https://doi.org/10.1371/journal.pone.0272646).