Jaringan Syaraf Tiruan untuk Memprediksi Pertumbuhan Penduduk Menggunakan Algoritma Extreme Learning Machine (ELM) (Studi Kasus di Kota Medan)

M. Fajar Alif
Matematika, Universitas Negeri Medan, Kota Medan, 20221, Indonesia
*Corresponding Author: fajaralif248@gmail.com

Abstrak

Perencanaan pembangunan kota yang efektif sangat bergantung pada data kependudukan yang akurat. Prediksi pertumbuhan jumlah penduduk menjadi salah satu pendekatan krusial untuk mengantisipasi kebutuhan infrastruktur, sosial, dan ekonomi di masa depan. Penelitian ini bertujuan untuk memprediksi pertumbuhan jumlah penduduk di Kota Medan menggunakan algoritma Extreme Learning Machine (ELM) sebagai salah satu pendekatan jaringan saraf tiruan yang unggul dalam kecepatan dan akurasi. Data yang digunakan merupakan data sekunder jumlah penduduk per kecamatan dari tahun 2001 hingga 2024 yang diperoleh dari BPS Kota Medan. Tahapan penelitian meliputi pre-processing data, pembagian data latih dan uji, normalisasi, pelatihan serta pengujian model, hingga evaluasi kinerja menggunakan MAPE. Hasil penelitian menunjukkan bahwa algoritma ELM mampu menghasilkan prediksi jumlah penduduk periode 2025–2029 dengan tingkat akurasi yang bervariasi antar kecamatan, di mana kecamatan dengan pertumbuhan stabil cenderung menghasilkan kesalahan prediksi lebih rendah dibandingkan kecamatan dengan pola fluktuatif. Secara keseluruhan, ELM terbukti efektif digunakan dalam peramalan jumlah penduduk dan dapat dijadikan dasar pertimbangan bagi pemerintah daerah dalam menyusun kebijakan kependudukan serta strategi pembangunan jangka panjang berbasis data.
Kata kunci: Extreme Learning Machine; Jaringan Saraf Tiruan; Prediksi; Pertumbuhan Penduduk

Abstract

Effective urban development planning relies heavily on accurate population data. Predicting population growth is a crucial approach to anticipating future infrastructural, social, and economic needs. This research aims to predict the population growth in Medan City using the Extreme Learning Machine (ELM) method, an artificial neural network approach known for its superior speed and accuracy. The study utilizes secondary data of the population per sub-district from 2001 to 2024, obtained from the Medan City Central Bureau of Statistics (BPS). The research stages include data pre-processing, splitting the data into training and testing sets, normalization, model training and testing, and performance evaluation using Mean Absolute Percentage Error (MAPE). The results indicate that the ELM method can predict the population for the 2025–2029 period with varying accuracy levels across sub-districts, where sub-districts with stable growth tend to yield lower prediction errors compared to those with fluctuating patterns. Overall, ELM proves to be effective for population forecasting and can serve as a basis for the local government in formulating population policies and data-driven, long-term development strategies.
Keywords: Extreme Learning Machine; Artificial Neural Network; Prediction; Population Growth.

Citation: Alif, M. F. (2025). Jaringan Syaraf Tiruan untuk Memprediksi Pertumbuhan Penduduk Menggunakan Algoritma Extreme Learning Machine (ELM) (Studi Kasus di Kota Medan). *Prodising Seminar Nasional Jurusan Matematika 2025*. 445 – 453.

PENDAHULUAN

Dalam merencanakan pembangunan suatu daerah, pertumbuhan penduduk adalah tolok ukur yang sangat penting. Data pertumbuhan penduduk diperlukan untuk mengambil keputusan yang tepat di berbagai bidang, mulai dari infrastruktur, pendidikan, ekonomi, hingga sosial, politik, transportasi, dan keamanan (Dewi et al., 2020). Kota Medan yang merupakan kota terbesar ketiga di Indonesia setelah Jakarta dan Surabaya adalah salah satu dari 33 Daerah tingkat II di Provinsi Sumatera Utara yang memiliki 21 kecamatan dan 151 kelurahan dengan luas wilayah mencapai sekitar 265,10 km². Sebagai kota metropolitan, Kota Medan dihadapkan pada berbagai tantangan akibat pertumbuhan penduduk, termasuk kebutuhan akan hunian, pendidikan, layanan kesehatan, serta pengembangan infrastruktur. Kondisi ini menuntut adanya pendekatan analitis dalam memproyeksikan laju pertumbuhan penduduk, agar pemerintah daerah dapat merumuskan langkah-langkah strategis yang efektif.

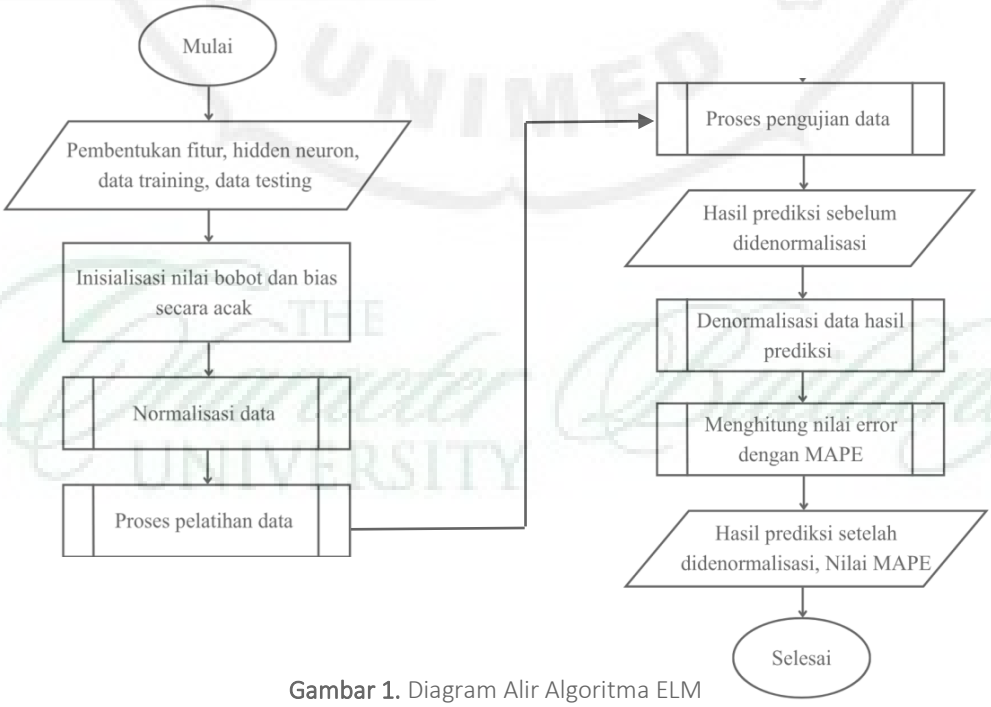
Terdapat korelasi yang erat antara dinamika kependudukan (bertambah atau berkurangnya jumlah penduduk) dengan tingkat perkembangan teknologi di suatu daerah. Pemanfaatan teknologi informasi di era digital saat ini menjadi sangat penting, tidak hanya karena perkembangan yang begitu cepat, tetapi juga karena keefektifannya dalam mendukung berbagai aktivitas manusia. Teknologi informasi memungkinkan data dikumpulkan, diolah, disimpan, dan dimanipulasi dengan berbagai cara untuk menghasilkan informasi yang berkualitas, relevan, akurat, dan tepat waktu. Informasi ini kemudian dapat dimanfaatkan dalam berbagai bidang, mulai dari bisnis dan pemerintahan hingga sektor swasta dan kebutuhan individu (Aji Sudarsono, 2016).

Perkembangan teknologi kecerdasan buatan (AI) kini telah merambah berbagai bidang, tak terkecuali dalam perencanaan demografi. Salah satu algoritma yang berkembang pesat adalah Jaringan Syaraf Tiruan (JST). Algoritma ini merupakan sistem pemrosesan informasi yang dirancang untuk meniru cara kerja jaringan saraf biologis pada makhluk hidup (Tambunan et al., 2019). JST merupakan bagian dari model pembelajaran mesin yang mampu menyelesaikan berbagai masalah melalui proses komputasi, dengan meniru aktivitas jaringan saraf pada otak manusia. Seperti halnya otak manusia, pada JST neuron-neuron saling berhubungan dan terdiri dari beberapa lapisan, termasuk input layer, output layer, serta hidden layer (Aulia et al., 2022).

Algoritma Extreme Learning Machine (ELM) merupakan algoritma jaringan saraf tiruan feed-forward yang memiliki satu lapisan tersembunyi, yang dikenal sebagai single hidden layer feed-forward neural networks (SLFNs) (Huang et al., 2004). Algoritma ELM memiliki keunggulan dalam kecepatan pembelajaran serta menghasilkan akurasi yang lebih tinggi dengan margin error yang lebih rendah (Pratiwi & Harianto, 2019). Dan akurasi yang dihasilkan ELM lebih baik dibandingkan metode konvensional seperti Exponential Smoothing dan Moving Average (Alfiyatin et al., 2019). Penelitian yang telah disebutkan di atas memberikan inspirasi pada penelitian ini sehingga dilakukannya penelitian yang berjudul "Jaringan Syaraf Tiruan Untuk Memprediksi Pertumbuhan Penduduk Menggunakan Algoritma Extreme Learning Machine (ELM) (Studi kasus di Kota Medan)". Prediksi ini bertujuan untuk memproyeksikan jumlah penduduk Kota Medan di masa depan. Hasil proyeksi tersebut diharapkan dapat menjadi dasar bagi pemerintah dalam menyusun langkah-langkah strategis untuk pembangunan kota.

METODE PENELITIAN

Penelitian ini dilaksanakan di Laboratorium Matematika Universitas Negeri Medan selama kurang lebih dua bulan dengan menggunakan pendekatan kuantitatif eksperimental. Fokus utama penelitian adalah penerapan model prediksi berbasis Jaringan Saraf Tiruan (JST) dengan algoritma Extreme Learning Machine (ELM) untuk menguji efektivitasnya dalam memprediksi pertumbuhan penduduk di Kota Medan berdasarkan data historis. Pendekatan kuantitatif dipilih karena melibatkan analisis numerik, statistik, dan komputasi guna memperoleh hasil akurasi yang tinggi. Adapun data yang digunakan dalam penelitian ini berjumlah 504 data, yang mencakup jumlah penduduk per kecamatan di Kota Medan dari tahun 2001 hingga 2024 yang bersumber dari situs resmi BPS Kota Medan. Adapun alur yang digunakan dalam penerapan algoritma ELM ini meliputi beberapa tahapan sebagai berikut:



Gambar 1. Diagram Alir Algoritma ELM

Proses Extreme Learning Machine (ELM) yang dijelaskan dalam Gambar 1 dimulai dengan memasukkan data jumlah penduduk dan jumlah hidden neuron sebagai fitur data dalam sistem. Selanjutnya, dilakukan pre-processing yang mencakup pembagian data menjadi data training dan data testing, serta perhitungan normalisasi data untuk mempersiapkan data agar siap digunakan dalam proses selanjutnya. Setelah itu, dilakukan inisialisasi bobot input dan bias secara acak. Proses berikutnya adalah training dan testing dengan menggunakan data yang telah dinormalisasi,

untuk mengoptimalkan model ELM. Setelah tahap training dan testing, dilakukan perhitungan denormalisasi data untuk mengembalikan data ke skala aslinya. Hasil output dari proses ini berupa prediksi jumlah penduduk yang diinginkan. Terakhir, untuk mengukur akurasi model, dilakukan perhitungan nilai MAPE (Mean Absolute Percentage Error), yang digunakan untuk menilai seberapa besar kesalahan prediksi dibandingkan dengan nilai aktual.

HASIL DAN PEMBAHASAN

Data yang digunakan dalam penelitian ini diperoleh melalui website resmi Badan Pusat Statistik (BPS) Kota Medan, yang menyediakan data jumlah penduduk berdasarkan kecamatan. Data yang dimanfaatkan sebagai bahan uji adalah data jumlah penduduk tahunan di masing-masing kecamatan di Kota Medan selama periode tahun 2001 hingga 2024. Data tersebut kemudian digunakan sebagai input dalam proses pelatihan dan pengujian model prediksi. Secara keseluruhan, terdapat 504 sampel data yang digunakan. Sebagaimana dapat dilihat pada Tabel 1, data tersebut digunakan sebagai dataset dalam penelitian ini.

Tabel 1. Data Penduduk Kecamatan Kota Medan 2001-2024

Kecamatan	2001	2002	2003	2004	.....	2020	2021	2022	2023	2024
Medan Tuntungan	65645	66745	66752	68438	.....	97249	98561	100200	100132	101069
Medan Johor	101959	105109	107916	108911	.....	151756	154096	156957	154868	155854
Medan Amplas	88638	94012	100751	104455	.....	129726	130882	132458	131770	132403
Medan Denai	125505	129847	133539	133742	.....	169643	171908	174744	171896	172581
Medan Area	110432	101458	109057	108317	.....	117029	118710	120788	118057	118352
Medan Kota	84530	82486	83213	82901	.....	84666	85563	86738	84778	84779
Medan Maimun	48995	48329	47232	47137	.....	49231	50063	51066	49708	49847
Medan Polonia	46316	47842	48600	49048	.....	59915	60389	61056	60679	60910
Medan Baru	43415	43514	42093	42221	.....	36522	36545	36681	36191	36066
Medan Selayang	77783	78976	80102	81035	.....	103176	103208	103559	104144	104425
Medan Sunggal	103803	105517	106475	106759	.....	129063	130193	131741	133273	134650
Medan Helvetia	128144	130581	132699	136216	.....	164910	166332	168287	168292	169363
Medan Petisah	69778	70364	65938	66073	.....	71844	72587	73565	72432	72599
Medan Barat	86706	86640	91807	77839	.....	88602	90156	92021	89248	89427
Medan Timur	112888	114492	110158	110492	.....	116985	117314	118008	117035	117002
Medan Perjuangan	97699	99346	98822	99580	.....	103813	104432	105380	105317	105778
Medan Tembung	134113	136643	134303	135188	.....	146534	147209	148346	149274	150135
Medan Deli	130255	137496	135940	141787	.....	189321	190822	192933	191743	192476
Medan Labuhan	89245	95634	98335	100184	.....	133765	135589	137863	135622	136190
Medan Marelan	88790	95943	106710	112463	.....	182515	186391	190940	189469	191765
Medan Belawan	91881	92881	93160	93356	.....	108987	109908	111181	110238	110612

Setelah seluruh data berhasil dikumpulkan, langkah selanjutnya adalah melakukan proses *feature engineering* atau pembentukan fitur. Proses pembentukan fitur dilakukan dengan teknik jendela bergulir (*sliding window*), di mana sejumlah nilai berturut-turut dari kolom "jumlah" digunakan untuk membentuk vektor input, dan nilai setelahnya dijadikan sebagai target atau output yang dilambangkan sebagai y.

Tabel 2. Pembentukan Fitur Kecamatan Medan Tuntungan

	x_1	x_2	x_3	x_4	x_5	x_6	target/y
1	65645	66745	66752	68438	68738	68984	68817
2	66745	66752	68438	68738	68984	68817	69447
3	66752	68438	68738	68984	68817	69447	70073
4	68438	68738	68984	68817	69447	70073	80942
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
18	87939	88624	97249	98561	100200	100132	101069

Penjelasan untuk setiap parameter pada tabel-tabel di atas adalah sebagai berikut: Pada tabel-tabel di atas, digunakan enam fitur input, yakni X1-X6, yang masing-masing merepresentasikan data jumlah penduduk tahunan. Fitur X1 menunjukkan jumlah penduduk pada tahun 2001, X2 menunjukkan jumlah penduduk pada tahun 2002, X3 menunjukkan jumlah penduduk pada tahun 2003, X4 menunjukkan jumlah penduduk pada tahun 2004, dan seterusnya untuk X5 dan X6. Setiap proses pembentukan fitur selalu dilengkapi dengan nilai target. Target tersebut merepresentasikan jumlah penduduk yang ingin diprediksi. Tabel-tabel di atas memuat sebanyak 18 baris data. Jumlah ini disebabkan oleh penggunaan enam data sebelumnya sebagai input fitur untuk menghasilkan satu nilai target (sesuai

dengan jumlah fitur yang ditentukan). Dengan demikian, nilai target pertama adalah data jumlah penduduk pada tahun 2007, sementara data dari tahun 2001-2006 digunakan sebagai data input X1-X6.

Pada proses analisis prediksi dengan menggunakan algoritma *Extreme Learning Machine* (ELM), langkah awal yang dilakukan adalah membagi dataset menjadi dua bagian, yaitu data pelatihan (*training*) dan data pengujian (*testing*), dengan proporsi pembagian sebesar 70% untuk data pelatihan dan 30% untuk data pengujian. Pembagian 70:30 ini dilakukan untuk seluruh data kecamatan di kota Medan, di mana total data untuk setiap kecamatan yaitu 18 baris, sehingga jumlah data latih (70%) yaitu 13 baris dan jumlah data uji (30%) yaitu 5 baris. Selain itu, pembagian data sebelum normalisasi diharapkan dapat menghasilkan tingkat kesalahan prediksi (*error*) yang lebih optimal dan realistis. Adapun pembagian data tersebut ditampilkan secara rinci dalam tabel berikut.

Tabel 3. Data Pelatihan Kecamatan Medan Tuntungan

	x1	x2	x3	x4	x5	x6	target/y
1	65645	66745	66752	68438	68738	68984	68817
2	66745	66752	68438	68738	68984	68817	69447
3	66752	68438	68738	68984	68817	69447	70073
4	68438	68738	68984	68817	69447	70073	80942
...	...	...	...	...	...	...	...
13	82534	84775	85613	86425	87123	87939	88624

Tahapan selanjutnya adalah adalah normalisasi data. Tujuan dari tahapan ini adalah untuk mengubah skala nilai data agar berada dalam rentang yang seragam, yakni antara 0 hingga 1. Proses normalisasi ini bertujuan agar model dapat melakukan pelatihan secara lebih efektif dan konsisten. Teknik normalisasi yang digunakan adalah metode *Min-Max Normalization*. Di bawah ini ditunjukkan salah satu contoh perhitungan normalisasi (persamaan 1) terhadap data jumlah penduduk kecamatan medan tuntungan. Pada persamaan 1, Dimana x_{min} = 65645, dan $x_{\max}$ = 82534. Sehingga $x_{1,1}$ = 0 dan $x_{2,1}$ = 0,0651

$$x_{ij} = \frac{(x_i - x_{min})}{x_{max} - x_{min}} \quad (1)$$

Nilai untuk matriks bobot W diinisialisasi secara acak dalam rentang -1 hingga 1, dengan ordo (jumlah *hidden neuron* × jumlah input layer). Sementara itu, matriks bias yang berordo 1 × jumlah *hidden neuron* diisi dengan nilai acak antara 0 dan 1. Seluruh parameter ini dimanfaatkan pada tahap pelatihan (*training*) untuk menghasilkan matriks bobot keluaran β, yang kemudian digunakan pada tahap pengujian (*testing*) untuk menghitung matriks prediksi Y. Matriks bobot awal dan matriks bias dapat dilihat pada Tabel.

Tabel 4. Nilai Bobot Awal (*Weight*)

W	1	2	3	4	5	6
1	-0,2509	0.9014	0.4640	0.1973	-0.6880	-0.6880
2	-0.8838	0.7324	0.2022	0.4161	-0.9588	0.9398
3	0.6649	-0.5753	-0.6364	-0.6332	-0.3915	0.0495
4	-0.1361	-0.4175	0.2237	-0.7210	-0.4157	-0.2673
5	-0.0879	0.5704	-0.6007	0.0285	0.1848	-0.9071
6	0.2151	-0.6590	-0.8699	0.8978	0.9313	0.6168
7	-0.3908	-0.8047	0.3685	-0.1197	-0.7559	-0.0096

Tabel 5. Nilai Bias

Bias	1	2	3	4	5	6	7
1	0.5164	0.5707	0.0285	0.1715	0.6853	0.8339	0.307

Dari tabel di atas ditunjukkan bahwa digunakan 6 input neuron dan 7 *hidden neuron*. Konfigurasi ini secara langsung menentukan ukuran matriks yang digunakan: matriks bobot (*hidden neuron × input neuron*) berukuran 7×6, dan matriks bias (1 × *hidden neuron*) berukuran 1×7. Kemudian untuk menghitung matriks Hinit, diperlukan matriks transpose dari bobot masukan. Kemudian dengan rumus$H_{init} = X_{training} * W^T + bias$, akan diperoleh tabel 7.

Tabel 6. Nilai Bobot Awal (*Weight*) Transpose

W^T	1	2	3	4	5	6	7
1	-0.2509	-0.8838	0.6649	-0.1361	-0.0879	0.2151	-0.3908
2	0.9014	0.7324	-0.5753	-0.4175	0.5704	-0.6590	-0.8047
3	0.4640	0.2022	-0.6364	0.2237	-0.6007	-0.8699	0.3685
4	0.1973	0.4161	-0.6332	-0.7210	0.0285	0.8978	-0.1197
5	-0.6880	-0.9588	-0.3915	-0.4157	0.1848	0.9313	-0.7559
6	-0.6880	0.9398	0.0495	-0.2673	-0.9071	0.6168	-0.0096

Tabel 7. Matriks H_{init}

H_{init}	1	2	3	4	5	6	7
1	0,5104	0,5789	0,0289	0,1692	0,6774	0,8393	0,3069
2	0,5360	0,5256	-0,0011	0,1649	0,6290	0,7973	0,3020
3	0,6138	0,6423	-0,0682	0,1145	0,6416	0,7461	0,2374
4	0,5619	0,5629	-0,0257	0,0806	0,6109	0,7890	0,1647
5	0,1580	1,0715	-0,0234	-0,1209	0,1141	1,2053	0,1081
6	-0,2691	0,5483	-0,2810	-0,3943	0,1571	1,7949	-0,3293
7	-0,1413	0,8082	-0,7290	-0,8586	0,1722	2,3346	-0,4490
8	0,1306	0,9485	-1,1249	-0,796	-0,1769	1,8894	-0,2951
9	0,5894	1,4565	-1,4892	-1,0948	0,0354	1,5700	-0,8010
10	0,3681	0,8608	-1,1598	-1,2814	-0,0183	1,8310	-1,1817
11	0,3437	0,8792	-1,2451	-1,4083	-0,0572	1,9908	-1,2526
12	0,3735	0,9218	-1,3718	-1,4568	-0,1379	1,9780	-1,2757
13	0,4522	1,0186	-1,4935	-1,5624	-0,1267	1,9660	-1,4053

Setelah memperoleh nilai H_{init}, langkah selanjutnya adalah menghitung nilai H menggunakan fungsi aktivasi sigmoid biner (Persamaan 2) dan akan diperoleh tabel 8.

$$H = \frac{1}{1 + exp^{(-H_{init})}} \tag{2}$$

Tabel 8. Matriks Keluaran *Hidden Layer* (H)

H	1	2	3	4	5	6	7
1	0,6249	0,6408	0,5072	0,5422	0,6631	0,6983	0,5761
2	0,6309	0,6285	0,4997	0,5411	0,6523	0,6894	0,5749
3	0,6488	0,6553	0,4829	0,5286	0,6551	0,6783	0,5591
4	0,6369	0,6371	0,4936	0,5201	0,6481	0,6876	0,5411
5	0,5394	0,7449	0,4941	0,4698	0,5285	0,7695	0,527
6	0,4331	0,6338	0,4302	0,4027	0,5392	0,8575	0,4184
7	0,4647	0,6917	0,3254	0,2976	0,5429	0,9117	0,3896
8	0,5326	0,7208	0,2451	0,3109	0,4559	0,8687	0,4267
9	0,6432	0,811	0,184	0,2507	0,5088	0,8278	0,3098
10	0,591	0,7028	0,2387	0,2173	0,4954	0,8619	0,2347
11	0,5851	0,7067	0,2236	0,1965	0,4857	0,8798	0,2223
12	0,5923	0,7154	0,2023	0,189	0,4656	0,8785	0,2183
13	0,6112	0,7347	0,1834	0,1733	0,4684	0,8772	0,197

Setelah menyelesaikan perhitungan matriks H, tahap berikutnya adalah menentukan Matriks H^+ *Pseudo-Inverse* melalui pendekatan *Moore-Penrose* (persamaan 3).

$$\boldsymbol{H^+} = (H^T * H)^{-1} * H^T \tag{3}$$

Tabel 9. Matriks H^T

H^T	1	2	3	4	5	⋯	12	13
1	0,6249	0,6309	0,6488	0,6369	0,5394	⋯	0,5923	0,6112
2	0,6408	0,6285	0,6553	0,6371	0,7449	⋯	0,7154	0,7347
3	0,5072	0,4997	0,4829	0,4936	0,4941	⋯	0,2023	0,1834
4	0,5422	0,5411	0,5286	0,5201	0,4698	⋯	0,189	0,1733
5	0,6631	0,6523	0,6551	0,6481	0,5285	⋯	0,4656	0,4684
6	0,6983	0,6894	0,6783	0,6876	0,7695	⋯	0,8785	0,8772
7	0,5761	0,5749	0,5591	0,5411	0,527	⋯	0,2183	0,197

Tabel 10. Matriks Hasil Perkalian $H^T.H$

$H^T.H$	1	2	3	4	5	6	7
1	4,4229	5,2327	2,6168	2,7087	4,1438	6,0352	3,0216
2	5,2327	6,298	3,067	3,1578	4,8979	7,3085	3,5506
3	2,6168	3,067	1,7945	1,8478	2,5808	3,5179	2,0314
4	2,7087	3,1578	1,8478	1,9193	2,6645	3,6019	2,1129
5	4,1438	4,8979	2,5808	2,6645	3,9635	5,6583	2,96
6	6,0352	7,3085	3,5179	3,6019	5,6583	8,5524	4,0591
7	3,0216	3,5506	2,0314	2,1129	2,96	4,0591	2,3414

Tabel 11. Matriks *Invers* Hasil Perkalian $H^T.H$

$(H^T * H)^{-1}$	1	2	3	4	5	6	7
1	97,4906	-77,5881	70,8971	-112,115	-84,9124	32,9564	81,7165
2	-77,5881	100,7498	-21,0931	43,8835	63,3957	-56,1856	-56,6917
3	70,8971	-21,0931	201,4225	-358,1022	-18,8201	-28,5926	162,2453
4	-112,115	43,8835	-358,1022	868,5404	-73,1112	85,0289	-449,9105
5	-84,9124	63,3957	-18,8201	-73,1112	150,2415	-58,8035	7,7569
6	32,9564	-56,1856	-28,5926	85,0289	-58,8035	49,8718	-21,3703
7	81,7165	-56,6917	162,2453	-449,9105	7,7569	-21,3703	273,4108

Tabel 12. Matriks *Pseudo-Inverse* (H^+)

H^+	1	2	3	⋯	10	11	12	13
1	0,1584	1,8199	-0,1977	⋯	1,1652	1,9458	2,6509	1,3879
2	-0,6847	-2,4034	0,4119	⋯	-0,8692	-1,5342	-2,0702	-0,4224
3	-0,1818	-0,3617	-0,8524	⋯	1,4574	2,9966	1,5237	0,7786
4	-0,9698	0,1538	0,4178	⋯	-0,7106	-4,6605	-0,8949	0,2798
5	1,4142	-0,7771	1,5884	⋯	-0,4336	-0,4968	-2,5783	-1,1258
6	-0,2916	0,9445	-0,9388	⋯	0,4746	0,4632	1,3752	0,3496
7	0,8305	1,0617	-0,1488	⋯	-0,9833	1,3413	0,1724	-1,1773

Nilai H^+ yang telah diperoleh selanjutnya diaplikasikan dalam perhitungan nilai output $\hat{\beta}$, di mana pada tahapan ini nilai output $\hat{\beta}$ dihasilkan dari perkalian nilai H^+ dengan matriks target $Y_{training}$. Sehingga diperoleh nilai bobot; $\hat{\beta}_1$=0,4024; $\hat{\beta}_2$= 1,1252; $\hat{\beta}_3$ = 0,8632; $\hat{\beta}_4$ = 0,871; $\hat{\beta}_5$ = -2,4958; $\hat{\beta}_6$ = 1,2926; dan $\hat{\beta}_7$ -1,6896

Tahapan pertama yang dilakukan adalah mengkalkulasi nilai matriks H dengan memanfaatkan nilai *weight* dan bias yang sama seperti yang digunakan pada tahap pelatihan, serta menggunakan dataset pengujian sebagai input. Pada proses uji, contoh data yang dipakai adalah data uji Kecamatan Medan Tuntungan yang sebanyak 5 data dengan jumlah *hidden neuron* seperti pada proses latih yaitu 7 serta menggunakan fungsi aktivasi *sigmoid biner*. Persamaan yang dipakai pada proses pengujian dalam mencari H_{init} sama dengan persamaan yang dipakai proses pelatihan dalam mencari hasil H_{init}. Setelah memperoleh nilai H_{init}, langkah selanjutnya adalah menghitung nilai H menggunakan fungsi aktivasi

Tabel 13. Matriks H_{init}

H_{init}	1	2	3	4	5	6	7
1	0,4333	0,9513	-1,4996	-1,6463	-0,1609	2,0247	-1,5172
2	0,1516	1,355	-1,537	-1,8323	-0,5628	2,3271	-1,5971
3	-0,1679	0,9801	-1,7593	-2,0861	-0,5453	2,7877	-1,991
4	-0,1341	1,1957	-2,1083	-2,5009	-0,5959	3,285	-2,1423
5	0,0557	1,2148	-2,4703	-2,5097	-0,8313	3,0189	-2,0994

.Tabel 14. Matriks Keluaran *Hidden Layer* (H)

1	1	2	3	4	5	6	7
1	0,6067	0,7214	0,1825	0,1616	0,4599	0,8834	0,1799
2	0,5378	0,7949	0,177	0,138	0,3629	0,9111	0,1684
3	0,4581	0,7271	0,1469	0,1105	0,3669	0,942	0,1202
4	0,4665	0,7678	0,1083	0,0758	0,3553	0,9639	0,1051
5	0,5139	0,7712	0,078	0,0752	0,3034	0,9534	0,1092

Tahap selanjutnya adalah menghitung nilai *output* prediksi ($\hat{Y}$). Nilai ini didapatkan melalui perkalian matriks antara matriks keluaran hidden layer (H) dengan matriks bobot *output* ($\hat{\beta}$), yang nilainya telah ditentukan pada proses pelatihan ELM sebelumnya, sehingga diperoleh Y_1 =1,0443; Y_2 = 1,3712; Y_3 = 1,3243; Y_4 = 1,3928; dan Y_5 = 1,4981. Denormalisasi adalah proses untuk mengembalikan nilai prediksi ke skala data aslinya. Untuk melakukannya, nilai prediksi dikalikan dengan rentang data (nilai maksimal dikurangi nilai minimal) dari target data uji. Hasil Denormalisasi Data Kecamatan Medan Tuntungan; D_1 = 101238; D_2 = 102487; D_3 =102308; D_4 = 102570; dan D_5= 102972. Adapun perbandingan dan selisih hasil prediksi yang telah didenormalisasi dengan target prediksi dapat dilihat pada tabel berikut.

Tabel 15. Perbandingan Hasil Prediksi Dalam Satuan Asli (Jiwa)

Tahun	Target Prediksi	Hasil Prediksi (Denormalisasi)	Selisih
2025	102732	101238	1494
2026	102985	102487	498
2027	103302	102308	994
2028	103288	102570	719
2029	103469	102972	497

Pengujian terhadap jumlah *hidden neuron* dilakukan karena parameter ini memiliki peran penting dalam tahap awal pembentukan bobot input secara acak, yang selanjutnya digunakan sepanjang proses pelatihan dan prediksi dengan algoritma ELM. Dalam tahapan ini, eksperimen dibatasi dengan dilakukan sebanyak 7 kali dengan variasi jumlah neuron tersembunyi, mulai dari 2, 3, 4, 5, 6, 7 dan 8. Setiap percobaan bertujuan untuk mengamati bagaimana perubahan jumlah neuron memengaruhi akurasi model, khususnya terhadap nilai MAPE. Dataset yang digunakan terdiri dari 18 baris data, yang kemudian dibagi menjadi 13 baris data untuk pelatihan dan 5 baris data untuk pengujian.

Didapatkan hasil dari serangkaian eksperimen yang bertujuan untuk menemukan arsitektur model Jaringan Saraf Tiruan ELM yang paling optimal untuk setiap Kecamatan. "Optimal" dalam konteks ini diukur dengan metrik MAPE, di mana nilai MAPE yang lebih rendah menunjukkan model yang lebih baik dan lebih akurat. Eksperimen ini dilakukan dengan mengubah-ubah satu parameter kunci, yaitu jumlah *Hidden neuron* (h), dari 2 hingga 8 neuron. Dari tabel tersebut menunjukkan perbedaan nilai MAPE di setiap *hidden neuron* pada masing-masing kecamatan. Ini menunjukkan bahwa karakteristik data (pola) di setiap kecamatan berbeda-beda. Contoh A (Medan Amplas): Nilai MAPE terendah adalah 5,2981% yang dicapai saat menggunakan 7 neuron. Contoh B (Medan Deli): Nilai MAPE terendah adalah 5,0224% yang dicapai saat menggunakan 4 neuron.

Menambahkan lebih banyak neuron tidak selalu berarti hasilnya akan lebih baik. Seringkali, setelah mencapai titik optimal, penambahan neuron justru dapat memperburuk kinerja model (MAPE meningkat). Fenomena ini dikenal sebagai *overfitting*, di mana model menjadi terlalu kompleks dan mulai "menghafal" *noise* dari data latihan, sehingga kinerjanya buruk pada data baru (data pengujian). Contoh (Medan Tuntungan): Kinerja terus membaik dari 2 neuron (MAPE 36,7%) hingga 7 neuron (MAPE 14,4%). Namun, ketika jumlah neuron ditambah menjadi 8, MAPE justru naik kembali menjadi 26,7%. Ini menunjukkan bahwa 7 neuron adalah arsitektur terbaik untuk kecamatan ini.

Dengan jumlah data historis yang terbatas, yaitu hanya 24 titik data dari tahun 2001 hingga 2024, model ELM mengalami kesulitan untuk menangkap pola berfluktuasi yang kompleks ini, kecamatan-kecamatan yang menghasilkan prediksi dengan tingkat kesalahan (MAPE) terendah, yaitu Medan Amplas, Medan Deli dan Medan Marelan. Data historisnya menunjukkan pola yang cenderung meningkat. Pola yang jelas dan konsisten ini memungkinkan model *Extreme Learning Machine* (ELM) untuk mempelajari pola pertumbuhan dengan efektif, tiga kecamatan dengan tingkat kesalahan (MAPE) terbesar, yaitu Medan Barat, Medan Timur dan Medan Tembung. Data historisnya menunjukkan pola yang sangat berfluktuasi dengan pergerakan naik-turun yang tajam. Keterbatasan jumlah data menyebabkan model tidak memiliki cukup contoh untuk membedakan antara pola fluktuasi yang sesungguhnya dengan *noise* acak, sehingga menghasilkan *error* prediksi yang tinggi. Temuan ini mengindikasikan bahwa efektivitas model ELM pada studi kasus ini sangat sensitif terhadap stabilitas pola data historis.

Hasil akhir dari serangkaian proses pemodelan dan peramalan jumlah penduduk untuk setiap kecamatan di Kota Medan. Tujuannya adalah untuk memberikan proyeksi atau perkiraan jumlah penduduk (Hasil Prediksi) untuk periode lima tahun ke depan, yaitu dari tahun 2025 hingga 2029. Data ini merupakan output dari model prediksi yang telah dipilih sebagai yang paling optimal untuk masing-masing kecamatan (berdasarkan analisis MAPE dari tabel sebelumnya). Dari data pada tabel tersebut secara mayoritas, sebagian besar kecamatan diprediksi akan mengalami peningkatan jumlah penduduk selama periode 2025-2029. Ini sejalan dengan pola umum urbanisasi dan pertumbuhan populasi alami. Contoh: Kecamatan Medan Labuhan menunjukkan pola kenaikan yang jelas, dari 138.191 jiwa pada tahun 2025 menjadi 140.525 jiwa pada tahun 2029. Begitu pula dengan Medan Marelana yang diprediksi naik dari 193.176 menjadi 195.242 jiwa.

Laju pertumbuhan penduduk tidak seragam di semua kecamatan, beberapa kecamatan diprediksi akan tumbuh lebih cepat dibandingkan yang lain. Menariknya, tidak semua kecamatan menunjukkan pola kenaikan yang lurus. Beberapa kecamatan menunjukkan pola yang lebih dinamis, di mana model memprediksi adanya puncak pertumbuhan diikuti oleh stabilisasi atau bahkan sedikit penurunan. Contoh: Kecamatan Medan Johor diprediksi mencapai puncak populasi pada tahun 2027 (161.228 jiwa) sebelum sedikit menurun pada tahun 2028 dan 2029. Demikian pula dengan Medan Denai yang mencapai puncaknya pada tahun 2028 (190.931 jiwa) sebelum sedikit menurun. Ini bisa berarti model menangkap pola data historis yang lebih kompleks, seperti saturasi populasi atau migrasi keluar. Secara keseluruhan, tabel ini bukan hanya sekadar kumpulan angka, melainkan sebuah alat strategis yang merangkum output dari analisis data kompleks menjadi informasi yang dapat ditindaklanjuti untuk perencanaan dan pengambilan keputusan yang lebih baik.

KESIMPULAN

Dari hasil penelitian terdapat kesimpulan, yaitu, model Jaringan Saraf Tiruan dengan algoritma *Extreme Learning Machine* (ELM) berhasil menghasilkan prediksi jumlah penduduk untuk 21 kecamatan di Kota Medan untuk periode lima tahun mendatang, yaitu dari tahun 2025 hingga 2029. Hasil peramalan menunjukkan bahwa sebagian besar kecamatan diprediksi akan mengalami peningkatan jumlah penduduk, sejalan dengan pola umum urbanisasi. Namun, laju pertumbuhan tidak seragam di setiap wilayah, dan beberapa kecamatan seperti Medan Johor dan Medan Denai menunjukkan pola yang lebih dinamis di mana model memprediksi adanya titik puncak populasi yang diikuti oleh stabilisasi, mengindikasikan bahwa model mampu menangkap pola historis data yang lebih kompleks daripada sekadar pola linier.

Kemudian tingkat akurasi penerapan algoritma ELM sangat bervariasi dan menunjukkan hubungan yang kuat antara karakteristik data historis, jumlah *hidden neuron* yang optimal, dan performa prediksi akhir. Analisis menunjukkan bahwa model mencapai akurasi tertinggi pada data kecamatan yang memiliki pertumbuhan cenderung linear, di mana arsitektur dengan jumlah *hidden neuron* yang sesuai (misalnya, 4 neuron untuk Medan Deli dan 6 untuk Medan Marelan) mampu menangkap pola stabil tersebut dengan sangat baik, menghasilkan tingkat kesalahan (MAPE) terendah. Sebaliknya, pada kecamatan yang menghasilkan tingkat kesalahan (MAPE) terbesar (Medan Barat, Medan Timur, dan Medan Tembung), data historisnya menunjukkan pola yang sangat fluktuatif. Pada kasus ini, proses optimasi *hidden neuron* menghadapi tantangan: untuk data seperti di Medan Timur, model dengan kompleksitas tinggi (7 *hidden neuron*) dipilih sebagai yang terbaik, namun kemungkinan besar mengalami overfitting terhadap data training yang terbatas (24 titik data), sehingga gagal melakukan generalisasi dengan baik. Sementara itu, untuk Medan Tembung, model yang sangat sederhana (2 *hidden neuron*) menjadi pilihan terbaik, yang mengindikasikan bahwa tidak ada tingkat kompleksitas yang mampu memodelkan pola data yang terlalu acak secara efektif. Dengan demikian, dapat disimpulkan bahwa tingkat akurasi penerapan ELM sangat bergantung pada interaksi antara kompleksitas model (ditentukan oleh *hidden neuron*) dan karakteristik data. Algoritma ini terbukti sangat andal untuk data dengan pola yang jelas, namun terbatas kemampuannya saat dihadapkan pada data yang sangat fluktuatif dengan jumlah data yang kecil.

DAFTAR PUSTAKA

Kurniasih, I. H., Furqon, M. T., & Adinugroho, S. (2020). Prediksi Pertumbuhan Penduduk di Kota Malang menggunakan Metode Extreme Learning Machine (ELM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4(2), 509-516.

Saad, M. I. S. (2023). Perbandingan Algoritma Extreme Learning Machine dan Multilayer Perceptron Dalam Prediksi Mahasiswa Drop Out. *Bulletin of Information Technology (BIT)*, 4(3), 369-376.

Fahrizal, M. A., Adinugroho, S., & Wihandika, R. C. (2021). Prediksi Volume Penggunaan Air Bulanan Kota Batu Menggunakan Metode Extreme Learning Machine (ELM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(7), 3078-3086.

Padhila, P. H., Cholissodin, I., & Adikara, P. P. (2022). Prediksi Harga Bitcoin berdasarkan Data Historis Harian dan Google Trend Index menggunakan Algoritme Extreme Learning Machine. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(7), 3515-3524.

Prayoga, B. (2023). pengaruh pertumbuhan ekonomi dan indeks pembangunan manusia terhadap penyerapan tenaga kerja kota medan. *Journal Economics and Strategy*, *4*(1), 42-51.

Dewi, A. K., Furqon, M. T., & Wihandika, R. C. (2020). Prediksi Laju Pertumbuhan Penduduk Menggunakan Metode Support Vector Regression (Studi Kasus : Kota Malang). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(1), 421–427.

Sudarsono, A. (2016). Jaringan Syaraf Tiruan Untuk Memprediksi Laju Pertumbuhan Penduduk Menggunakan Metode Backpropagation (Studi Kasus di Kota Bengkulu). *Media Infotama*, 12(1), 61–69.

Tambunan, H. S., Gunawan, I., & Sumarno, S. (2019). Prediksi Jumlah Pendapatan Beasiswa PPA dan BBP Menggunakan Jaringan Syaraf Tiruan Backpropagation. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 3(4), 346.

Safira, A. J., Cholissodin, I., & Adikara, P. P. (2022). Prediksi Penerimaan Mahasiswa Baru dengan Menggunakan Metode Extreme Learning Machine (ELM) (Studi Kasus pada Universitas 17 Agustus 1945 Surabaya). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, *6*(9), 4526-4533.

Huang, G.-B., Zhu, Q.-Y. & Siew, C.-K. (2004). Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. *Proceedings of International Joint Conference on Neural Networks*.

Pratiwi, H. & Harianto, K. (2019). Perbandingan Algoritma ELM Dan Backpropagation Terhadap Prestasi Akademik Mahasiswa. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 3(2), pp. 282-294.

Alfiyatin, A. N., Mahmudy, W. F., Ananda, C. F. & Anggodo, Y. P. (2019). Penerapan Extreme Learning Machine (ELM) Untuk Peramalan Laju Inflasi Di Indonesia. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 6(2), pp. 179-186.

Hajjah, A., & Marlim, Y. N. (2021). Analisis error terhadap peramalan data penjualan. *Techno. Com*, 20(1), 1-9.

B. P. Statistik, "Badan Pusat Statistik," Badan Pusat Statistik, [Online]. Available: https://www.bps.go.id/subject/12/kependudukan.html. [Accessed 8 02 2025].

Anjani, R. P., Prianto, C., & Saputra, M. H. K. (2020). *Buku Laporan Forecasting barang Inbound dan outbound menggunakan single Exponential Smoothing dan MAPE* (Pertama). Kreatif Industri Nusantara.

Azmi, U., Hadi, Z. N., & Soraya, S. (2020). ARDL METHOD: Forecasting Data Curah Hujan Harian NTB. *Jurnal Varian*, 3(2), 73–82.

Effendi, N. (2018). SATIN – Sains dan Teknologi Informasi Penerapan Jaringan Syaraf Tiruan untuk Memprediksi Efektifitas Pembelajaran dengan E-Learning di Universitas Muhammadiyah Riau. *Satin*, 4(1), 1–10.

Haris, M. F. Al, & Kuswadi, S. (2019). *Kecerdasan Buatan dan Aplikasinya* (Pertama). CV. Absolute Media.

Huang, G. Bin, Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing* 70(1– 3), 489–501.

Humaini, Q. (2015). Jaringan Syaraf Tiruan Extreme Learning Machine (ELM) Untuk Memprediksi Kondisi Cuaca. *Skripsi*, 1–86.

Jumiyatun, Mahmudi, I., & Mustari, A. (2019). *Kontrol Power Elektronik dan Aplikasinya*. Media Nusa Creative.

Fikriya, Z. A., Irawan, M. I., & Soetrisno, S. (2017). Implementasi Extreme Learning Machine Untuk Pengenalan Objek Citra Digital. *Jurnal Sains dan Seni ITS*, *6*(1), A1-A6.

Ashar, N.M., Cholissodin, I. and Dewi, C., 2018. Penerapan Metode Extreme Learning Machine (ELM) Untuk Memprediksi Jumlah Produksi Pipa Yang Layak (Studi Kasus Pada PT. KHI Pipe Industries). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, 2(11), pp.4621– 4628.

Fachrony, A., Cholissodin, I. and Santoso, E., 2018. Implementasi Algoritme Extreme Learning Machine (ELM) untuk Prediksi Beban Pemanasan dan Pendinginan Bangunan. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, 2(9), pp.3043–3049.

Najar, A. M. (2018). *Penerapan Metode Extreme Learning Machine Untuk Prediksi Tingkat Berdasarkan Keadaan Cuaca (Studi Kasus : Wilayah Dki Jakarta)*.

Relief, S. (2013). Jaringan Syaraf Tiruan & Implementasinya. *Informaticrazy*.

Siang, J. J. (2005). *Jaringan SyarafTiruan & Pemrogramannya Menggunakan Matlab*. CV. Andi Offset.

Solikhun, & Wahyudi, M. (2020). *Jaringan Syaraf Tiruan Backpopagation*. Yayasan Kita Menulis.

Suryanto, A. A., & Muqtadir, A. (2019). Penerapan Metode Mean Absolute Error (MAE) Dalam Algoritma Regresi Linear Untuk Prediksi Produksi Padi. *Saintekbu*, 11(1), 78–83.

Windarto, A. P., Nasution, D., Wanto, A., Tambunan, F., Hasibuan, M. S., Siregar, M. N. H., Lubis, M. R., Solikhun, Fadhillah, Y., & Nofriansyah, D. (2020). *Jaringan Syaraf Tiruan : Algoritma Prediksi dan Implementasi* (J. Simarmata (ed.); 1st ed.). Yayasan Kita Menulis.

Zheng, A., & Casari, A. (2018). *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media, Inc.

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.

Gu, S., Kelly, B., & Xiu, D. (2018). Em pirical Asset Pricing via Machine Learning. *SSRN Electronic Journal*.

Muhardina, R. (2024). Prediksi Risiko Penyakit Hipertensi Menggunakan Metode Extreme Learning Machine (ELM). *Coding: Jurnal Komputer dan Aplikasi*, 12(2), 159-165.