

BAB I

PENDAHULUAN

1.1 Latar Belakang

Setiap negara memiliki mata uang sebagai alat tukar yang digunakan dalam transaksi ekonomi. Kegiatan transaksi ekonomi, seperti perdagangan dan investasi tidak hanya dilakukan antara masyarakat di suatu wilayah tertentu, tetapi juga dilakukan antar suatu negara dengan negara lainnya (Mokodongan, et al., 2018). Harga satu unit mata uang dalam unit mata uang asing dikenal sebagai nilai tukar mata uang atau tingkat kurs (*exchange rate*). Perubahan nilai tukar mata uang memengaruhi stabilitas dan aktivitas perekonomian suatu negara sehingga nilai tukar menjadi salah satu variabel penting dalam ekonomi makro. Pertumbuhan nilai tukar yang stabil menunjukkan bahwa suatu negara memiliki kondisi perekonomian yang relatif baik (Prasetyo, et al., 2021).

Indonesia memberlakukan *freely floating exchange rates*, yaitu sistem nilai tukar mengambang bebas sejak Agustus 1997. Dalam kondisi tersebut, nilai mata uang Rupiah bisa berubah setiap waktu di bursa valuta asing dan menjadi sangat rentan oleh pengaruh faktor ekonomi maupun non-ekonomi karena kurs hanya ditentukan oleh penawaran dan permintaan tanpa campur tangan pemerintah. Perubahan kurs terbagi menjadi dua, yaitu depresiasi dan apresiasi. Depresiasi Rupiah terhadap Dolar AS (Amerika Serikat) berarti nilai Rupiah melemah atau harga Dolar AS naik jika dibandingkan dengan Rupiah. Sebaliknya, apresiasi Rupiah terhadap Dolar AS berarti nilai Rupiah menguat atau harga Dolar AS turun terhadap Rupiah (Triyono, 2008).

Posisi Dolar AS sebagai mata uang yang dominan digunakan dalam perekonomian global memberikan dampak signifikan terhadap dinamika nilai tukar Rupiah. Ketika USD menguat, Rupiah cenderung mengalami depresiasi. Pelemahan nilai tukar ini secara langsung memengaruhi stabilitas ekonomi makro Indonesia. Sebagai negara yang bergantung pada impor bahan baku industri, depresiasi Rupiah mengakibatkan terjadinya lonjakan biaya produksi yang pada akhirnya mendorong kenaikan harga barang di tingkat konsumen atau disebut inflasi. Lebih lanjut, fluktuasi nilai tukar yang tajam dapat menggoyahkan stabilitas perekonomian, memicu ketidakpastian bagi para pelaku bisnis dan investor, serta berpotensi menggerus kepercayaan masyarakat terhadap mata uang domestik (Ramadhani, et al., 2021).

Mengingat besarnya dampak fluktuasi nilai tukar terhadap perekonomian Indonesia, maka peramalan nilai tukar Rupiah terhadap Dolar AS menjadi krusial untuk mendukung pengambilan keputusan strategis dan pertumbuhan ekonomi Indonesia. Bagi pelaku bisnis internasional, peramalan valuta asing (valas) merupakan komponen penting untuk meraih keuntungan dan menghindari kerugian akibat transaksi antar negara. Proyeksi pergerakan valas yang tidak akurat dapat

menghilangkan potensi laba secara signifikan. Oleh karena itu, kemampuan untuk meramalkan nilai tukar menjadi kunci bagi pengambilan keputusan yang melibatkan transfer dana lintas mata uang. Akan tetapi, peramalan nilai tukar secara akurat merupakan tantangan besar karena pergerakannya yang sangat kompleks dan dipengaruhi oleh banyak faktor (Ramadhani, et al., 2021). Selain faktor ekonomi yang memengaruhinya, permasalahan dalam peramalan nilai tukar juga terletak pada bagaimana membangun model matematis yang mampu menangkap pola non-linear dari data deret waktu secara efektif.

Pergerakan nilai tukar Rupiah dipengaruhi oleh berbagai faktor fundamental ekonomi. Beberapa faktor yang memengaruhinya, antara lain: tingkat inflasi, suku bunga, dan neraca perdagangan. Tingkat inflasi yang tinggi dapat menurunkan daya saing produk domestik, menekan ekspor, dan mendorong impor, sehingga pada akhirnya meningkatkan permintaan Dolar AS dan melemahkan rupiah (Prasetyo, et al., 2021). Selain itu, tingkat suku bunga memegang peranan krusial dalam memengaruhi aliran modal. Suku bunga yang kompetitif akan menarik modal asing (*capital inflow*) yang memperkuat Rupiah, sementara suku bunga rendah dapat memicu aliran modal keluar (*capital outflow*) yang akan menekan Rupiah. Selanjutnya, kinerja neraca perdagangan juga menjadi penentu, di mana surplus ekspor akan memperkuat rupiah melalui peningkatan permintaan sementara defisit akan menyebabkan sebaliknya (Sukirno, 2012).

Hubungan kompleks dan sifat non-linear dari berbagai faktor tersebut membuat pergerakan nilai tukar menjadi sangat fluktuatif sehingga metode peramalan konvensional sering kali tidak cukup akurat (Firman, 2020). Ramadhani et al., menyebutkan bahwa untuk mendapatkan hasil yang lebih akurat dan stabil, diperlukan metode yang mampu mempelajari pola dari data historis. Salah satu pendekatan yang dapat digunakan untuk menangani pola tersebut adalah metode dalam rumpun Jaringan Saraf Tiruan (JST). Penerapan JST untuk peramalan nilai tukar telah dilakukan oleh Firman (2020) menggunakan metode propagasi balik. Penelitian tersebut secara eksplisit menyarankan pengembangan metode untuk mengoptimalkan penentuan bobot dan bias awal guna meningkatkan kinerja model.

JST konvensional memiliki kelemahan dalam hal kecepatan pembelajaran atau *learning speed* (Ramadhani et al., 2021). Untuk mengatasinya, Huang et al. (2006) memperkenalkan Extreme Learning Machine (ELM) dengan *learning speed* dan tingkat akurasi yang jauh lebih baik. Implementasi ELM untuk prediksi nilai tukar Rupiah juga telah dilakukan oleh Maghfuroh (2020), yang menunjukkan relevansi metode ini untuk data keuangan yang kompleks, terbukti dengan hasil evaluasi *Mean Squared Error* sebesar 2319,92. Di samping keunggulan tersebut, ELM memiliki keterbatasan pada penentuan parameter bobot *input* dan bias yang dilakukan secara acak sehingga berisiko menghasilkan performa yang kurang optimal.

Meskipun berbagai metode peramalan telah digunakan untuk memprediksi nilai tukar Rupiah terhadap Dolar AS, hasil peramalan yang diperoleh sering kali bersifat tidak stabil dan kurang akurat akibat kompleksitas serta non-linearitas data. Oleh karena itu, diperlukan pendekatan peramalan yang tidak hanya cepat secara komputasi, tetapi juga mampu menghasilkan parameter model yang optimal.

Penentuan bobot *input* dan bias secara acak pada ELM dapat menyebabkan variasi hasil peramalan yang cukup besar antar percobaan, sehingga berpotensi menurunkan reliabilitas model ketika digunakan untuk data ekonomi yang sensitif seperti nilai tukar. Keterbatasan ini mendorong perlunya algoritma untuk mengoptimasi model ELM. Salah satu algoritma yang dapat digunakan untuk menemukan parameter terbaik pada ELM yaitu Particle Swarm Optimization (PSO) (Sukma, et al., 2021).

Penerapan algoritma PSO menawarkan solusi untuk mengoptimalkan pembobotan pada ELM. Namun, efektivitas algoritma ini juga bergantung pada konfigurasi *hyperparameter* yang digunakan. Kennedy dan Eberhart (1995) mengemukakan bahwa parameter seperti bobot inersia dan koefisien akselerasi memegang peranan krusial dalam menyeimbangkan kemampuan partikel untuk melakukan eksplorasi (pencarian global) dan eksploitasi (pencarian lokal). Dengan demikian, pemilihan parameter yang tidak tepat berisiko menyebabkan algoritma terjebak pada optimum lokal atau gagal untuk mencapai konvergensi yang diinginkan.

Di sisi lain, kinerja model ELM tidak terlepas dari pengaruh arsitektur jaringannya. Huang et.al (2026) menyatakan bahwa penentuan jumlah *hidden neuron* yang tepat menjadi hal yang sangat penting. Jumlah *hidden neuron* yang terlalu sedikit akan berpotensi menyebabkan *underfitting*, yaitu kondisi di mana kapasitas jaringan terlalu rendah sehingga model gagal untuk menangkap pola data. Sebaliknya, jumlah *hidden neuron* yang berlebihan dapat memicu *overfitting*, yakni keadaan di mana model menjadi terlalu kompleks dan mempelajari data latih secara berlebihan sehingga model kehilangan kemampuan untuk melakukan generalisasi saat memprediksi data uji (Huang, et. Al, 2006). Oleh karena itu, penentuan kombinasi *hyperparameter* PSO dan jumlah *hidden neuron* ELM yang tepat menjadi tantangan tersendiri untuk menghasilkan model peramalan yang optimal.

Efektivitas penggabungan metode ELM dan PSO untuk meningkatkan akurasi peramalan telah dibuktikan dalam berbagai penelitian terdahulu. Penelitian oleh Sukma, et al. (2021) yang menerapkan ELM-PSO untuk memprediksi harga cabai berhasil mencapai nilai *Mean Absolute Percentage Error* (MAPE) sebesar 1,26%. Demikian pula penelitian yang dilakukan oleh Darmayanti, et al. (2018) yang menunjukkan bahwa optimasi menggunakan PSO secara signifikan mampu meningkatkan performa ELM hingga mencapai nilai MAPE hanya sebesar 0,59%. Keberhasilan metode *hybrid* ELM-PSO dalam memberikan prediksi akurat pada studi kasus lain menunjukkan potensinya sebagai pendekatan yang relevan untuk diterapkan pada data nilai tukar yang kompleks. Berdasarkan uraian tersebut, penelitian ini mengusulkan metode Extreme Learning Machine yang dioptimasi menggunakan Particle Swarm Optimization untuk peramalan nilai tukar Rupiah terhadap Dolar AS dengan judul **“Peramalan Nilai Tukar Rupiah terhadap Dolar AS Menggunakan Model Extreme Learning Machine dengan Optimasi Particle Swarm Optimization”**.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, diperoleh rumusan masalah sebagai berikut.

1. Bagaimana pengaruh variasi *hyperparameter* algoritma PSO dan jumlah *hidden neuron* pada arsitektur ELM terhadap kinerja optimasi bobot *input* dan bias dalam peramalan nilai tukar Rupiah terhadap Dolar AS?
2. Bagaimana perbandingan kinerja dan akurasi peramalan nilai tukar Rupiah terhadap Dolar AS antara model ELM standar dengan model *hybrid* ELM-PSO?

1.3 Batasan Penelitian

Batasan masalah pada penelitian ini meliputi:

1. Penelitian ini dibatasi pada data historis harian yang mencakup nilai tukar Rupiah terhadap Dolar AS serta data faktor fundamental ekonomi, yaitu tingkat inflasi, suku bunga Indonesia dan AS, serta jumlah ekspor dan impor Indonesia dari periode 1 Juli 2015 sampai dengan 31 Juli 2025.
2. Model peramalan dibatasi pada penggunaan variabel *input* meliputi data kurs 5 sampai 1 hari sebelumnya, selisih tingkat inflasi Indonesia dan AS, selisih tingkat suku bunga Indonesia dan AS, serta rasio ekspor dan impor Indonesia.
3. Model peramalan yang digunakan dibatasi pada algoritma Extreme Learning Machine (ELM) sebagai model dasar dan Particle Swarm Optimization (PSO) sebagai algoritma optimasi untuk menentukan bobot *input* dan bias.
4. *Software* yang digunakan adalah Microsoft Excel untuk pengolahan data dasar dan Google Colaboratory (Phyton) untuk implementasi dan evaluasi model.
5. Metrik evaluasi kinerja yang digunakan untuk mengukur kinerja model dibatasi pada *Mean Absolute Percentage Error* (MAPE), *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), dan Koefisien Determinasi (R^2), tanpa membahas aspek kebijakan ekonomi atau implikasi makroekonomi lanjutan.
6. Penelitian ini menerapkan penggunaan *random seed* pada proses inisialisasi awal untuk memastikan hasil pengujian yang konsisten dan dapat direproduksi.

1.4 Tujuan Penelitian

Dari permasalahan yang telah dirumuskan sebelumnya, maka tujuan dari penelitian ini adalah sebagai berikut.

1. Menganalisis pengaruh variasi *hyperparameter* algoritma PSO dan jumlah *hidden neuron* pada arsitektur ELM terhadap kinerja optimasi bobot *input* dan bias.
2. Menganalisis dan membandingkan kinerja peramalan model ELM standar dengan model *hybrid* ELM-PSO berdasarkan model evaluasi yang digunakan.

1.5 Manfaat Penelitian

Beberapa manfaat yang diharapkan dari penelitian tugas akhir ini meliputi:

1. Manfaat praktis, yaitu memberikan gambaran mengenai kinerja model peramalan nilai tukar Rupiah terhadap Dolar AS berbasis ELM yang dioptimasi

- menggunakan PSO, sehingga dapat menjadi referensi teknis dalam analisis pergerakan nilai tukar.
2. Manfaat keilmuan, yaitu menambah wawasan dalam bidang matematika terapan dan komputasi mengenai penerapan metode *hybrid* ELM-PSO untuk peramalan data deret waktu yang bersifat non-linear.
 3. Manfaat akademik, yaitu menjadi bahan referensi dan perbandingan bagi penelitian selanjutnya yang membahas optimasi parameter pada model *machine learning* untuk peramalan.

1.6 Landasan Teori

1.6.1 Nilai Tukar Mata Uang

Berdasarkan Kamus Besar Bahasa Indonesia, nilai tukar atau biasa disebut dengan kurs merupakan nilai mata uang suatu negara yang dinyatakan dengan nilai mata uang negara lain. Misalnya, jika nilai kurs menunjukkan bahwa US\$1.00 setara dengan Rp16.000, maka artinya dibutuhkan sebanyak 16.000 Rupiah untuk memperoleh satu Dolar AS. Nilai tukar Rupiah terhadap Dolar AS memegang peranan penting dalam perdagangan internasional. Hal ini disebabkan karena nilai tukar Rupiah terhadap Dolar AS memungkinkan kita untuk mengetahui perbandingan antara harga semua barang dan jasa yang dihasilkan berbagai negara (Triyono, 2008).

Secara garis besar, sistem nilai tukar terbagi menjadi dua jenis (Gilpin, 2001), yaitu kurs tetap (*fixed exchange rate*), dan kurs bebas atau biasa disebut dengan kurs mengambang (*floating exchange rate*). Kurs tetap merupakan sistem dimana nilai tukar mata uang domestik secara resmi ditetapkan pada tingkat tertentu oleh bank sentral sebagai pemegang otoritas moneter tertinggi suatu negara, bukan oleh mekanisme pasar. Jadi, pada sistem ini, nilai tukar tidak ditetapkan berdasarkan aktivitas permintaan dan penawaran di pasar uang. Sistem nilai tukar ini pernah diberlakukan di Indonesia melalui UU No. 32 Tahun 1964 yang mana pada saat itu ditetapkan nilai tukar senilai Rp 250/USD. Selanjutnya, kurs bebas yaitu sistem kurs yang ditentukan secara bebas oleh kekuatan pasar, dalam hal ini permintaan dan penawaran. Sistem ini terbagi menjadi dua macam, yaitu kurs mengambang terkendali (*managed floating exchange rate*) dan kurs mengambang bebas (*freely floating exchange rate*). Kurs mengambang terkendali merupakan sistem kurs yang penetapannya tidak diserahkan sepenuhnya pada aktivitas pasar dan masih ada campur tangan dari pemerintah meskipun terbatas. Adapun kurs mengambang bebas adalah sistem yang penetapannya sepenuhnya dikendalikan oleh aktivitas pasar dan tidak ada campur tangan pemerintah. Sistem kurs mengambang bebas inilah yang diterapkan di Indonesia sejak 1997 hingga saat ini.

Sebagai konsekuensi dari penerapan sistem kurs mengambang bebas di Indonesia, pergerakan nilai tukar di pasar menjadi sangat rentan, baik oleh pengaruh faktor ekonomi maupun non-ekonomi. Secara umum, pergerakan nilai tukar diklasifikasikan menjadi dua, yaitu depresiasi dan apresiasi. Depresiasi merujuk pada pelemahan nilai uang domestik, dalam hal ini Rupiah terhadap mata uang asing

Dolar AS. Artinya, dibutuhkan lebih banyak mata uang Rupiah untuk memperoleh satu unit Dolar AS. Pada kondisi ini, harga barang-barang domestik menjadi relatif lebih murah bagi pihak asing. Sebaliknya, apresiasi merujuk pada penguatan nilai Rupiah terhadap Dolar AS, sehingga harga barang-barang domestik menjadi relatif lebih mahal bagi pihak asing (Sukirno, 1981).

1.6.1.1 Faktor yang Memengaruhi Nilai Tukar

Secara umum, pergerakan nilai tukar dipengaruhi oleh faktor permintaan dan penawaran di pasar valuta asing. Sukirno (2012) mengemukakan bahwa terdapat beberapa faktor yang memengaruhi perubahan kurs, di antaranya tingkat inflasi, perubahan harga ekspor dan impor, tingkat suku bunga dan tingkat pengembalian investasi, serta pertumbuhan ekonomi.

1. Tingkat inflasi

Inflasi merupakan kecenderungan naiknya harga barang dan jasa secara terus menerus. Inflasi dikatakan meningkat jika harga barang dan jasa domestik mengalami kenaikan. Hal ini sejalan dengan teori Paritas Daya Beli (PDB) yang dikemukakan oleh Gustav Cassel pada tahun 1918. Menurut teori ini, nilai tukar akan cenderung menyesuaikan diri untuk menyeimbangkan perbedaan tingkat inflasi antar negara. Apabila laju inflasi di dalam negeri meningkat relatif lebih tinggi dibanding laju inflasi negara mitra dagang, maka harga barang domestik akan menjadi lebih mahal. Kenaikan harga tersebut akan menyebabkan tekanan depresiasi atau pelemahan terhadap nilai tukar. Dengan kata lain, inflasi dapat diartikan sebagai penurunan nilai uang terhadap nilai barang dan jasa secara umum (Firman, 2020). Hal ini dibuktikan dengan penelitian yang dilakukan oleh Triyono (2018) yang menyatakan bahwa variabel inflasi memiliki pengaruh positif sebesar 0,0176 terhadap nilai kurs. Artinya, kenaikan variabel inflasi mengakibatkan naiknya variabel kurs sebesar 1,76%.

2. Perubahan harga ekspor dan impor

Ekspor merujuk pada kegiatan menjual barang atau jasa yang diproduksi di dalam negeri ke luar negeri. Sebaliknya, impor adalah kegiatan pembelian barang atau jasa dari luar negeri ke dalam negeri. Ekspor dan impor memegang peranan penting dalam perdagangan internasional serta memengaruhi ekonomi global (Rochani, et al., 2024). Ketika harga barang dan jasa domestik lebih mahal dibandingkan harga barang dan jasa di luar negeri atau dengan kata lain terjadi inflasi, maka impor cenderung bertambah. Keadaan ini akan meningkatkan permintaan valas dan nilai tukar rupiah akan melemah terhadap mata uang asing. Di sisi lain, inflasi juga akan menyebabkan harga ekspor mengalami peningkatan sehingga tingkat ekspor akan cenderung berkurang. Penurunan ekspor ini secara langsung akan mengurangi penawaran valas yang masuk ke dalam negeri yang juga menyebabkan pelemahan nilai tukar rupiah (Arifin dan Mayasya, 2018).

3. Tingkat suku bunga

Tingkat suku bunga merupakan salah satu hal yang sangat berpengaruh terhadap nilai tukar suatu mata uang terhadap mata uang lain. Semakin tinggi suku bunga suatu negara, maka semakin tinggi pula permintaan terhadap mata uang negara tersebut. Peningkatan permintaan ini tentunya akan mendorong apresiasi nilai tukar mata uang. Sebaliknya, penurunan suku bunga akan menurunkan permintaan dan cenderung memicu terjadinya depresiasi nilai tukar (Arifin dan Mayasya, 2018).

Berbagai faktor yang telah diuraikan di atas saling berhubungan untuk menentukan tingkat nilai tukar. Pendekatan moneter yang merupakan pengembangan dari teori PDB dan teori kuantitas uang meyakini bahwa faktor-faktor moneter, seperti inflasi dan suku bunga yang melandasi fungsi permintaan dan penawaran uang merupakan pendorong utama terjadinya pergerakan nilai tukar valas. Kholidin (2002) memodelkan pendekatan moneter dengan asumsi *sticky price* melalui rumus matematis sebagai berikut.

$$S_t = (m - m^*)_t - \alpha(y - y^*)_t + \left(\delta + \lambda - \frac{1}{\theta}\right)(r - r^*)_t + \left(\frac{1}{\theta}\right)(\pi_t^e - \pi^{e*})_t \quad (1)$$

dengan

S_t	: tingkat nilai tukar valuta asing di waktu t
m	: jumlah uang beredar negara domestik
m^*	: jumlah uang beredar negara asing
$(m - m^*)_t$	: selisih jumlah uang beredar di waktu t
y	: pendapatan riil negara domestik
y^*	: pendapatan riil negara asing
$(y - y^*)_t$	: selisih pendapatan riil di waktu t
r	: tingkat suku bunga negara domestik
r^*	: tingkat suku bunga negara asing
$(r - r^*)_t$	: selisih tingkat suku bunga di waktu t
π_t^e	: tingkat inflasi negara domestik
π^{e*}	: tingkat inflasi negara asing
$(\pi_t^e - \pi^{e*})_t$	: selisih tingkat inflasi di waktu t

Model pendekatan moneter pada Persamaan (1) mendeskripsikan arah hubungan antara variabel makroekonomi terhadap tingkat nilai tukar. Koefisien selisih jumlah uang beredar $(m - m^*)$ dan selisih tingkat inflasi $(\pi_t^e - \pi^{e*})$ memiliki tanda positif yang mengindikasikan bahwa peningkatan kedua variabel tersebut akan memicu depresiasi mata uang domestik. Sebaliknya, pada selisih pendapatan riil $(y - y^*)$, koefisien α yang merepresentasikan elastisitas pendapatan terhadap permintaan uang bertanda negatif. Hal ini berarti kenaikan pendapatan riil domestik relatif terhadap negara asing akan meningkatkan permintaan mata uang sehingga mendorong apresiasi mata uang.

Di sisi lain, koefisien pada variabel suku bunga terdiri dari gabungan tiga parameter, yaitu δ , λ , dan $1/\theta$ yang menunjukkan adanya dua kekuatan berlawanan yang bekerja secara bersamaan. Koefisien positif $(\delta + \lambda)$ merepresentasikan

mekanisme pasar uang, dengan δ adalah penyesuaian penawaran uang dan λ adalah penyesuaian permintaan uang sebagai akibat dari perubahan tingkat suku bunga. Kenaikan suku bunga negara domestik mendorong lembaga keuangan untuk meningkatkan penawaran dana, sekaligus mengurangi keinginan masyarakat memegang uang, sehingga menciptakan kelebihan uang (*excess money balance*) yang menyebabkan depresiasi mata uang domestik. Sementara itu, koefisien $1/\theta$ merepresentasikan pengaruh perubahan aliran modal terhadap nilai tukar. Pada variabel selisih suku bunga koefisien $1/\theta$ bernilai negatif. Artinya, kenaikan suku bunga domestik dianggap menguntungkan bagi investor sehingga menarik aliran modal masuk (*capital inflow*) yang berdampak pada apresiasi nilai tukar domestik. Sebaliknya, koefisien $1/\theta$ bernilai positif pada selisih inflasi karena peningkatan inflasi domestik akan menyebabkan aliran modal keluar (*capital outflow*) sehingga nilai tukar domestik mengalami depresiasi.

Secara umum, Persamaan (1) menyatakan bahwa nilai tukar (S_t) ditentukan oleh selisih variabel-variabel moneter antara negara domestik dengan negara asing (*). Misalnya, kenaikan selisih inflasi ($\pi_t^e - \pi^{e*}$) dengan inflasi domestik lebih tinggi akan memberikan tekanan depresiasi terhadap nilai tukar. Demikian pula selisih suku bunga ($r - r^*$) memengaruhi aliran modal yang akan berdampak pada nilai mata uang. Oleh karena itu, model moneter tersebut memberikan landasan teoritis untuk pemilihan variabel *input* pada penelitian ini.

1.6.1.2 Kurs Referensi

Pasar valas memerlukan referensi harga *spot* USD/IDR yang kredibel untuk mendukung pembentukan harga yang efisien. Untuk menjawab kebutuhan tersebut, Bank Indonesia (BI) menerbitkan harga *spot* USD/IDR yang dikenal dengan sebutan *Jakarta Interbank Spot Dollar Rate* (JISDOR) sejak 20 Mei 2013. Penyusunan kurs referensi JISDOR mengacu pada data transaksi USD/IDR antar bank di pasar valas. Bank Indonesia memperoleh data transaksi tersebut secara *real-time* melalui Sistem Monitoring Transaksi Valuta Asing Terhadap Rupiah (SISMONTAVAR).

BI menghitung JISDOR berdasarkan rata-rata tertimbang dari volume seluruh transaksi USD/IDR antar bank yang berlangsung dalam rentang waktu 09.00-15.00 WIB. Kemudian, BI akan mempublikasikan kurs referensi ini secara resmi pada pukul 15.15 WIB. Data JISDOR tersedia untuk setiap hari kerja, tidak mencakup hari Sabtu, Minggu, hari libur nasional, atau hari lain ketika bank sedang tidak beroperasi. Apabila dalam rentang waktu tersebut tidak terdapat data transaksi, baik karena tidak ada aktivitas pasar maupun terdapat gangguan pada sistem, maka nilai JISDOR yang dipublikasikan adalah nilai kurs yang berlaku pada hari kerja sebelumnya (Bank Indonesia, 2020).

Meskipun JISDOR merupakan kurs acuan resmi, penggunaannya tidak bersifat mandatori. Artinya, JISDOR hanya menjadi acuan yang kredibel tetapi tidak wajib atau tidak mengikat secara hukum untuk digunakan. BI menerbitkan kurs referensi ini sebagai penyedia informasi kredibel untuk membantu pelaku pasar keuangan dalam mengambil keputusan. Di sisi lain, BI memanfaatkan data ini

sebagai alat monitoring dan asesmen untuk membentuk pengembangan pasar yang sehat serta memiliki resiliensi atau daya tahan yang tinggi terhadap gejolak pasar (Firman, 2020).

1.6.2 Peramalan

Peramalan (*forecasting*) secara umum didefinisikan sebagai ilmu yang digunakan untuk memperkirakan kejadian di masa depan. Peramalan merupakan suatu metode untuk memperkirakan sebuah nilai atau kondisi yang akan datang dengan menggunakan data masa lalu, misalnya data beberapa bulan atau beberapa tahun terakhir. Metode ini bertujuan untuk membantu dalam pengambilan keputusan dan menyusun perencanaan ke depan. Dalam dunia bisnis, peramalan biasanya digunakan sebagai dasar dalam melaksanakan perencanaan, pengawasan, dan pengambilan keputusan (Fauziah, et al., 2019).

Prinsip dasar dari peramalan adalah menganalisis data masa lalu, di mana data historis dikumpulkan, dipelajari, kemudian dianalisis untuk mengidentifikasi pola datanya. Selanjutnya pola tersebut diproyeksikan ke masa depan menggunakan model matematis. Metode peramalan secara umum terbagi menjadi dua, yaitu (Fauziah, et al., 2019):

1. Metode kualitatif, yaitu metode peramalan bersifat subjektif yang didasarkan pada pengalaman, target yang ingin dicapai, intuisi, atau opini seseorang, misalnya pimpinan, konsumen, atau ahli.
2. Metode kuantitatif, yaitu metode peramalan yang menggunakan data dan perhitungan statistik untuk menghilangkan unsur subjektif sehingga hasilnya lebih akurat dan dapat dipertanggungjawabkan.

1.6.3 Peramalan Data Deret Waktu (*Time Series Forecasting*)

Data deret waktu (*time series*) adalah serangkaian data yang dikumpulkan dan terurut berdasarkan urutan waktu, misalnya harian, mingguan, bulanan, atau tahunan. Berdasarkan hal tersebut, peramalan data deret waktu (*time series forecasting*) dapat didefinisikan sebagai metode untuk memperkirakan keadaan di masa depan melalui pengujian dan analisis data masa lalu yang bersifat periodik dan terurut berdasarkan urutan waktu dengan interval yang sama (Ukhra, 2014).

Untuk menentukan metode peramalan deret waktu yang sesuai, perlu diketahui pola dari data tersebut. Secara umum, pola data dibedakan menjadi empat jenis, yaitu pola *trend*, musiman, siklus, dan *irregular*. Pola *trend* merupakan kecenderungan arah data dalam jangka panjang, baik berupa kenaikan maupun penurunan. Pola musiman merupakan fluktuasi data yang terjadi secara periodik dan berulang dalam interval waktu tertentu, misalnya harian, mingguan, bulanan, atau triwulan. Pola siklus merupakan fluktuasi jangka panjang yang berkaitan dengan siklus bisnis yang biasanya sulit dideteksi dan sangat berkaitan dengan pola *trend*. Sedangkan pola *irregular* merupakan kejadian tidak terduga dan bersifat acak, akan tetapi kemunculannya dapat memengaruhi fluktuasi data (Ukhra, 2014).

1.6.4 Metrik Evaluasi

Peramalan selalu dihadapkan pada kondisi yang tidak pasti sehingga ada faktor akurasi yang perlu diperhitungkan. Tujuan dari model peramalan adalah untuk menemukan kesalahan peramalan (*forecast error*) yang paling minimum, karena peramalan yang baik adalah peramalan yang nilainya paling mendekati kenyataan. Untuk mengukur tingkat akurasi model, dapat digunakan ukuran statistik yang menyatakan perbedaan antara hasil peramalan dengan nilai aktual (Fauziah et al., 2019).

Terdapat beberapa metrik evaluasi yang umum digunakan untuk masalah regresi, antara lain (Setiawan, et al., 2025):

1. *Root Mean Square Error* (RMSE)

RMSE menyatakan rata-rata akar kuadrat dari eror prediksi. Kriteria model dikatakan baik jika nilai RMSE semakin mendekati 0. Nilai RMSE yang semakin kecil menandakan hasil prediksi semakin dekat dengan nilai aktual (Nurani et al., 2023). Secara matematis dituliskan sebagai berikut.

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (Y'_t - Y_t)^2}{n}} \quad (2)$$

dengan

Y'_t : nilai prediksi ke- t

Y_t : nilai data aktual ke- t

n : banyaknya data

2. *Mean Absolute Error* (MAE)

MAE didefinisikan sebagai rata-rata selisih mutlak dari nilai aktual dengan nilai prediksi yang dirumuskan sebagai berikut (Nurani et al., 2023).

$$MAE = \frac{1}{n} \sum_{t=1}^n |Y'_t - Y_t| \quad (3)$$

Nilai MAE merepresentasikan rata-rata kesalahan dalam satuan data yang sama, misalnya Rupiah untuk data nilai tukar. Model dikatakan memiliki kinerja baik jika nilai MAE semakin kecil (Nurani et al., 2023).

3. *Mean Absolute Percentage Error* (MAPE)

MAPE menyatakan persentase kesalahan rata-rata secara absolut (mutlak). Semakin kecil nilainya maka hasil yang diperoleh semakin akurat (Nurani et al., 2023). MAPE dihitung menggunakan rumus sebagai berikut.

$$MAPE = \frac{\frac{1}{n} \sum_{t=1}^n |Y'_t - Y_t|}{Y_t} \times 100\% \quad (4)$$

Adapun rentang nilai MAPE dapat dilihat pada Tabel 1 berikut.

Tabel 1. Rentang Nilai MAPE

Nilai MAPE	Akurasi Prediksi
<10%	Sangat baik
10% - 20%	Baik
20% - 50%	Cukup

>50%	Buruk
------	-------

4. *Mean Square Error (MSE)*

MSE menyatakan rata-rata selisih kuadrat dari nilai prediksi dan nilai aktual, di mana model dikatakan baik jika MSE semakin mendekati 0. Secara matematis, MSE ditunjukkan oleh rumus berikut (Sari dan Hasanuddin, 2020).

$$MSE = \frac{1}{n} \sum_t^n (Y'_t - Y_t)^2 \quad (5)$$

5. *Coefficient of Determination (R^2)*

R^2 menyatakan sebuah nilai yang merepresentasikan seberapa besar variabel independen berpengaruh terhadap variabel dependen dengan nilai berkisar antara 0 sampai 1. Nilai R^2 semakin tinggi mendekati 1 menunjukkan mode semakin baik. Kriteria interpretasi nilai R^2 dikategorikan menjadi tiga, meliputi: lemah ($R^2 \geq 0,19$), cukup ($R^2 \geq 0,33$), dan kuat ($R^2 \geq 0,67$). Secara matematis, R^2 dituliskan sebagai berikut (Nurani, et al., 2023).

$$R^2 = 1 - \frac{\sum_t^n (Y'_t - Y_t)^2}{\sum_t^n (Y_t - \bar{Y})^2} \quad (6)$$

1.6.5 Analisis Korelasi

Analisis korelasi didefinisikan sebagai sebuah metode analisis data penelitian yang bertujuan untuk mengetahui kekuatan hubungan, bentuk, arah, dan besarnya pengaruh antara variabel independen dengan variabel dependen. Dalam analisis korelasi, hubungan antar variabel dibedakan menjadi tiga, yaitu hubungan kausal, simetris, dan timbal balik. Hubungan kausal yaitu bentuk korelasi yang bersifat sebab-akibat, artinya keberadaan suatu variabel disebabkan oleh keadaan variabel lainnya. Hubungan simetris yaitu bentuk korelasi yang terjadi apabila dua atau lebih variabel muncul secara bersamaan, artinya keberadaan suatu variabel tidak disebabkan oleh variabel lain. Sedangkan hubungan timbal balik merupakan bentuk korelasi di mana dua atau lebih variabel saling memengaruhi dan variabel-variabel tersebut dapat berganti kedudukan, artinya variabel independen dapat bertindak sebagai variabel dependen dan sebaliknya (Hasan dan Misbahuddin, 2017).

Untuk menganalisis hubungan antar variabel dapat digunakan beberapa teknik statistik, antara lain analisis koefisien korelasi, analisis regresi, dan koefisien determinasi. Untuk mengukur derajat hubungan berupa kekuatan dan bentuk/arah hubungan antara variabel independen dan dependen, dapat digunakan teknik koefisien korelasi. Koefisien korelasi (r) memiliki nilai pada rentang -1 sampai dengan 1 (Hasan dan Misbahuddin, 2017).

Untuk mengukur kekuatan hubungan, koefisien korelasi dinyatakan dalam interval nilai berikut (Hasan dan Misbahuddin, 2017).

Tabel 2. Interval Nilai r dan Kekuatan Hubungan

r (+)	r (-)	Kekuatan Hubungan
0	0	Tidak ada
$0 < r < 0,20$	$0 < r < -0,20$	Sangat lemah

$0,20 < r < 0,40$	$-0,20 < r < -0,40$	Lemah
$0,40 < r < 0,70$	$-0,40 < r < -0,70$	Cukup/sedang
$0,70 < r < 0,90$	$-0,70 < r < -0,90$	Kuat
$0,90 < r < 1$	$-0,90 < r < -1$	Sangat kuat
1	-1	Sempurna

Untuk mengukur bentuk atau arah hubungan, koefisien korelasi dinyatakan dalam positif (+) dan negatif (-) (Hasan dan Misbahuddin, 2017).

1. Koefisien korelasi dengan nilai positif (+) menyatakan bahwa variabel-variabel berkorelasi positif. Hal ini berarti jika suatu variabel naik/turun maka variabel lainnya juga demikian. Nilai mendekati 1 menyatakan nilai korelasi positif yang kuat.
2. Koefisien korelasi negatif menyatakan bahwa variabel-variabel berkorelasi negatif. Hal ini berarti jika suatu variabel naik/turun maka variabel lainnya akan bersifat sebaliknya. Nilai mendekati -1 menyatakan nilai korelasi negatif yang kuat.
3. Koefisien korelasi dengan nilai nol menyatakan bahwa variabel tidak memiliki korelasi.
4. Koefisien korelasi bernilai sama dengan 1 atau -1 menyatakan korelasi positif atau negatif sempurna.

Tabel 3 menunjukkan beberapa jenis koefisien korelasi yang umum digunakan berdasarkan jenis variabelnya (Hasan dan Misbahuddin, 2017).

Tabel 3. Teknik Statistik untuk Analisis Koefisien Korelasi

Jenis Variabel		Jenis Koefisien Korelasi
Variabel I	Variabel II	
Nominal	Nominal	1. Kontingensi (C) 2. Lambda (λ) 3. Phi (ϕ)
Nominal	Ordinal	Theta (θ)
Nominal	Interval/rasio	1. Eta (η) 2. Point Biserial (r_{pbi})
Ordinal	Ordinal	1. Gamma (γ) 2. Spearman (r_s)
Ordinal	Interval/rasio	Jaspen's (M)
Interval/rasio	Interval/rasio	Pearson's (r)

1.6.5.1 Koefisien Korelasi Pearson

Koefisien korelasi Pearson digunakan untuk menganalisis hubungan antar variabel jika data yang digunakan berskala interval atau rasio. Dasar pemikiran dari koefisien korelasi ini adalah perubahan antar variabel. Jika perubahan sebuah variabel diikuti oleh perubahan dari variabel lainnya maka kedua variabel tersebut saling berhubungan (Sihombing dan Bangun, 2019). Rumus matematis koefisien korelasi Pearson adalah sebagai berikut (Hasan dan Misbahuddin, 2017).

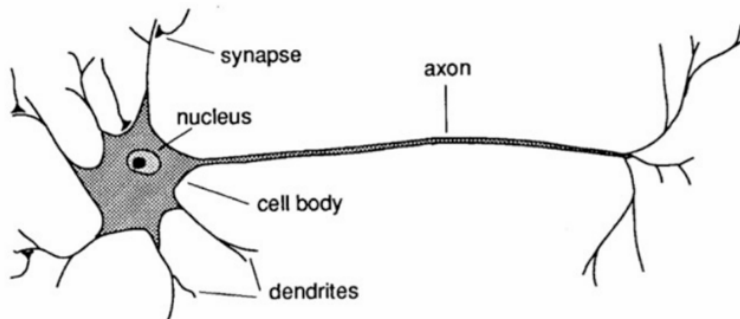
$$r = n \frac{\sum XY - (\sum X)(\sum Y)}{\sqrt{[n \sum X^2 - (\sum X^2)][n \sum Y^2 - (\sum Y)^2]}} \quad (7)$$

dengan

- r : koefisien korelasi Pearson
 X : variabel independen (bebas)
 Y : variabel dependen (terikat)

1.6.6 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (JST) terinspirasi dari jaringan saraf biologis yang terdapat pada otak manusia, sehingga JST mampu belajar dan melakukan generalisasi dari pengalaman yang diperoleh (Zhang, et al., 1998). Berdasarkan analogi sistem kerja jaringan saraf manusia pada Gambar 1, JST menerima vektor masukan (seperti halnya *dendrit* dalam otak manusia) dan memiliki sejumlah bobot yang merepresentasikan koneksi (seperti halnya *sinaps* dalam otak manusia). Unit pemroses utamanya adalah neuron (seperti halnya badan sel dalam otak manusia), yang berisi fungsi penambah (*adder*) dan fungsi aktivasi. Fungsi aktivasi berperan mengatur keluaran (melalui *akson*) yang diberikan oleh *neuron* (Prasetyo, 2012).

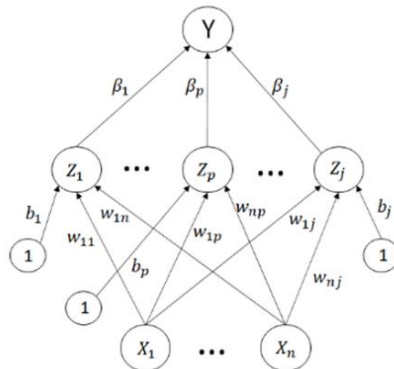


Gambar 1. Jaringan Saraf pada Otak Manusia

Jaringan Saraf Tiruan tersusun oleh beberapa bagian utama, di antaranya *input*, bias, fungsi aktivasi, *hidden layer*, dan *output*. *Input* merupakan bagian yang berperan memberikan informasi atau data masukan kepada neuron. Informasi atau data masukan tersebut diterima lalu diproses oleh neuron sehingga menghasilkan nilai bobot (Izati, et al., 2019). Adapun bias merupakan nilai *input* yang ditambahkan ke dalam JST, di mana nilai sinyal *input* bias akan selalu sama dengan 1 (Siang, 2005). Bias berfungsi mengubah nilai *threshold* menjadi sama dengan 0, di mana *threshold* merupakan konstanta penimbang dalam suatu hubungan dari suatu unit tertentu (Warsito, 2009). Selanjutnya, *hidden layer* merupakan bagian yang menyalurkan informasi dari *input layer* ke *output layer*. Keluaran proses yang terjadi pada *hidden layer* disebut *output hidden layer*, yang kemudian diteruskan ke *output layer*. Terakhir, *output* merupakan hasil akhir dari seluruh proses yang terjadi pada JST.

JST dapat direpresentasikan oleh sebuah graf yang memiliki beberapa lapisan, di mana setiap lapisan terdiri dari sejumlah *node/neuron/unit (vertex)* lalu

dihubungkan oleh sisi (*edge*) yang memiliki bobot (Firman, 2020). Arsitektur JST dapat dilihat pada Gambar 2 di bawah ini.



Gambar 2. Arsitektur Jaringan Saraf Tiruan

Pada Gambar 2, terlihat bahwa JST tersusun atas beberapa lapisan, di antaranya:

1. $X_1, \dots, X_n$ merupakan *input layer* (lapisan *input*) yang merepresentasikan variabel *input* atau fitur dari data,
2. $Z_1, \dots, Z_j$ merupakan *hidden layer* (lapisan tersembunyi), dan
3. Y merupakan *output layer* (lapisan *output*).

Selain itu, JST juga terdiri dari bobot dan bias, di antaranya:

1. $w_{1j}, \dots, w_{nj}$ merupakan bobot *input* yang menghubungkan neuron *input* X_n ke neuron tersembunyi Z_j ,
2. $b_1, \dots, b_j$ merupakan bias *hidden layer* yang ditambahkan ke neuron tersembunyi Z_j (Angka “1” pada diagram mewakili *input* konstan yang dikalikan dengan bobot bias b_j), dan
3. $\beta_1, \dots, \beta_j$ merupakan bobot *output* yang menghubungkan neuron tersembunyi Z_j ke neuron *output* Y .

Karena JST memiliki arsitektur yang diilustrasikan melalui struktur graf, operasi pada jaringan saraf tiruan dapat dijelaskan secara matematis dalam notasi aljabar (Putra, 2018). Model JST pada Gambar 1.2 dapat dituliskan seperti pada Persamaan (8) dan Persamaan (9).

$$Z_j = g(x_1 w_{1j} + \dots + x_n w_{nj}) \quad (8)$$

$$Y = g(Z_1 \beta_1 + \dots + Z_p \beta_p + \dots + Z_j \beta_j) \quad (9)$$

dengan

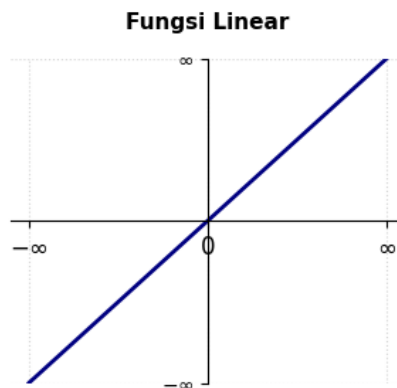
- | | |
|---------|---------------------------------|
| g | : fungsi aktivasi |
| x | : data masukan (<i>input</i>) |
| w | : bobot <i>input</i> |
| Z | : <i>hidden layer</i> |
| β | : bobot <i>output</i> |

Pada persamaan (8) dan (9), Z_j merepresentasikan nilai neuron pada *hidden layer* yang telah dikenakan fungsi aktivasi dan Y merepresentasikan nilai neuron pada *output layer* yang telah dikenakan fungsi aktivasi.

Dalam JST, fungsi aktivasi digunakan untuk menemukan *output* sebuah *neuron*. Warsito (2009) mengemukakan bahwa terdapat beberapa fungsi aktivasi yang sering digunakan pada JST, antara lain:

1. Fungsi linear atau fungsi identitas, yaitu fungsi aktivasi yang akan menghasilkan *output* sama dengan *inputnya*. Fungsi aktivasi ini biasanya digunakan untuk kasus yang *outputnya* berupa bilangan riil. Fungsi linear dinyatakan melalui persamaan berikut.

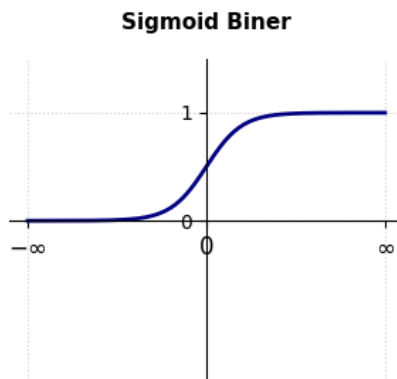
$$f(x) = x \quad (10)$$



Gambar 3. Grafik Fungsi Aktivasi Linear

2. Fungsi *sigmoid biner*, yaitu fungsi bernilai antara 0 sampai 1 yang digunakan apabila *output* yang diharapkan merupakan bilangan riil dalam rentang 0 sampai 1. Fungsi ini dinyatakan sebagai berikut.

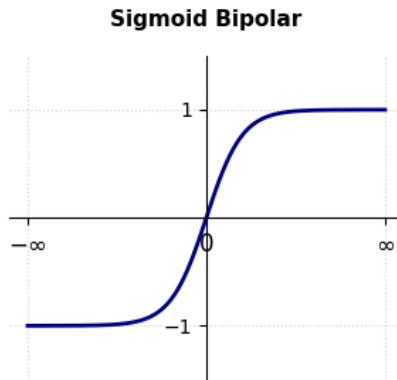
$$f(x) = \frac{1}{1 + e^{-x}} \quad (11)$$



Gambar 4. Grafik Fungsi Aktivasi Sigmoid Biner

3. Fungsi *sigmoid bipolar*, yaitu fungsi yang menghasilkan *output* berupa bilangan riil 1 atau -1 . Fungsi sigmoid bipolar dinyatakan melalui persamaan sebagai berikut.

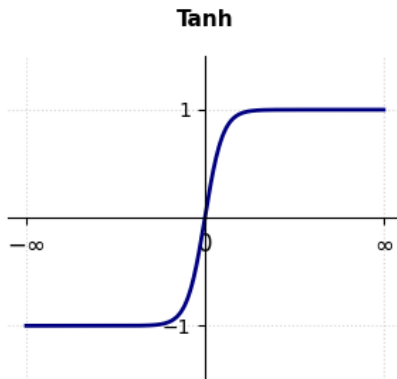
$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (12)$$



Gambar 5. Grafik Fungsi Aktivasi Sigmoid Bipolar

4. Fungsi aktivasi tangen hiperbolik (*tanh*), yaitu fungsi yang *outputnya* berkisar dari -1 hingga 1 . Fungsi ini dirumuskan sebagai berikut.

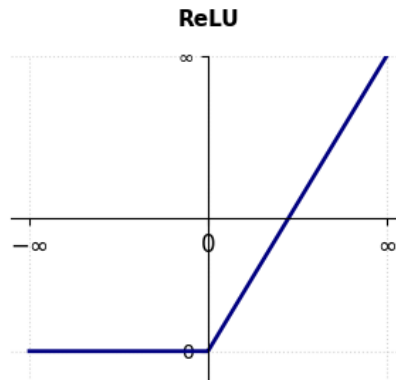
$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (13)$$



Gambar 6. Grafik Fungsi Aktivasi Tangen Hiperbolik

5. Fungsi aktivasi *Rectified Linear Unit* (ReLU), yaitu fungsi linear yang jika *inputnya* bernilai positif, misalnya x , maka *outputnya* juga akan menjadi x . Jika tidak, maka akan menghasilkan nol.

$$f(x) = \max(0, x) \quad (14)$$



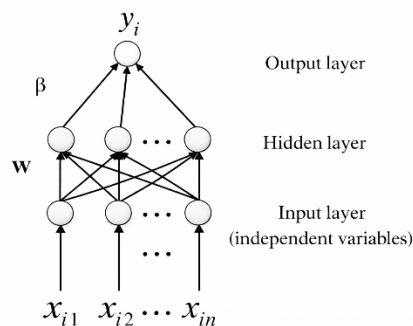
Gambar 7. Grafik Fungsi Aktivasi ReLU

1.6.7 Extreme Learning Machine (ELM)

Metode ELM merupakan salah satu metode pembelajaran baru dari JST dengan arsitektur *Single Hidden Layer Feedforward Neural Networks* (SLFNs). Artinya, ELM merupakan jaringan saraf tiruan *feedforward* yang tersusun oleh satu *hidden layer*. ELM pertama kali dikenalkan pada tahun 2004 oleh Guang-Bin Huang untuk mengatasi permasalahan metode JST, khususnya dalam hal kecepatan pembelajaran atau *learning speed*. Huang, et al. (2006) mengemukakan terdapat dua hal yang menyebabkan *learning speed* pada JST *feedforward* relatif rendah, yaitu:

1. Proses pelatihan data menggunakan *slow gradient based learning algorithm*.
2. Parameter pada jaringan ditentukan secara iteratif menggunakan metode pembelajaran tersebut.

Secara struktural, ELM terdiri dari tiga lapisan utama, yaitu *input layer*, *hidden layer*, dan *output layer*. Arsitektur ELM ditunjukkan pada gambar berikut.



Gambar 8. Arsitektur ELM

Pada ELM, parameter-parameter seperti bobot *input* dan bias dipilih dan dibangkitkan secara acak sehingga memiliki *learning speed* yang lebih baik dan mampu menghasilkan generalisasi performa yang baik (Nurhaedah, 2023). ELM juga

memiliki model matematis yang sederhana dan lebih efektif dibanding metode JST *feedforward* lainnya.

1.6.7.1 Algoritma Extreme Learning Machine

Untuk N sampel berbeda (x_i, y_i) , di mana setiap sampel terdiri dari *input* $x_i = [x_{i1}, x_{i2}, \dots, x_{in}] \in \mathbb{R}^n$ dan target (*output*) $y_i = [y_{i1}, y_{i2}, \dots, y_{im}] \in \mathbb{R}^m$, sebuah SLFNs standar dengan $\tilde{N}$ *hidden neuron* dan fungsi aktivasi $g(x)$ dirumuskan sebagai berikut (Huang, 2006).

$$\sum_{j=1}^{\tilde{N}} \beta_j g(x_i) = \sum_{j=1}^{\tilde{N}} \beta_j g(w_j \cdot x_i + b_j) = o_i, i = 1, \dots, N \quad (15)$$

dengan

w_j : matriks bobot *input*, yang menghubungkan neuron *input* ke neuron tersembunyi ke- j

b_j : bias dari neuron tersembunyi ke- j

β_j : vektor bobot *output*, yang menghubungkan neuron tersembunyi ke- j ke neuron *output*

$w_j \cdot x_i$: hasil kali dalam matriks bobot *input* (w_j) dan matriks *input* (x_i)

$g(w_j \cdot x_i + b_j)$: hasil dari fungsi aktivasi pada neuron tersembunyi ke- j

Dengan mengasumsikan bahwa SLFNs standar dapat mengaproksimasi *output* dengan error nol, artinya $\sum_{i=1}^N \|o_i - y_i\| = 0$, Persamaan (15) menjadi:

$$\sum_{j=1}^{\tilde{N}} \beta_j g(w_j \cdot x_i + b_j) = y_i, i = 1, \dots, N \quad (16)$$

Lebih lanjut, Persamaan (16) dapat disederhanakan dalam bentuk sistem persamaan berikut.

$$H \beta = Y \quad (17)$$

dengan

$Y = \begin{bmatrix} y_1 \\ \dots \\ y_N \end{bmatrix}_{N \times m}$ merepresentasikan matriks yang berisi seluruh target *output* y_i ,

$\beta = \begin{bmatrix} \beta_1 \\ \dots \\ \beta_{\tilde{N}} \end{bmatrix}_{\tilde{N} \times m}$ merupakan matriks yang berisi semua bobot β_j , dan

$$H(w_1, \dots, w_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \dots & \dots & \dots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}$$

adalah matriks *hidden layer output* dari JST. Kolom ke- i dari H merupakan *output* dari *neuron* pada *hidden layer* yang bersesuaian dengan $x_1, x_2, \dots, x_N$.

Pada JST konvensional, bobot *input* (w_j) dan bias *hidden layer* (b_j) harus dibangkitkan secara iteratif, misalnya menggunakan *gradient descent*. Namun,

Huang et al. (2006) membuktikan bahwa parameter-parameter tersebut dapat ditentukan langsung secara acak jika fungsi aktivasi yang digunakan memenuhi syarat tertentu.

Teorema 1. Diberikan sebuah SLFNs standar dengan N *hidden* neuron dan fungsi aktivasi $g: R \rightarrow R$ yang dapat diturunkan tak terbatas (*infinitely differentiable*) pada setiap interval. Untuk N sampel data berbeda, bobot *input* w_j dan bias b_j dapat dipilih secara acak dari sembarang distribusi kontinu, maka dengan probabilitas satu, matriks *output hidden layer* H akan bersifat *invertible* (jika $\tilde{N} = N$) dan jaringan dapat mencapai error pelatihan nol ($\|H\beta - Y\| = 0$).

Dalam praktiknya, jumlah *hidden* neuron $\tilde{N}$ biasanya jauh lebih kecil dibandingkan jumlah data sampel N ($\tilde{N} \ll N$). Pada kondisi ini, berlaku Teorema 2 berikut.

Teorema 2. Diberikan nilai $\epsilon > 0$ yang sangat kecil dan fungsi aktivasi $g(x)$ yang dapat diturunkan tak terbatas (*infinitely differentiable*). Terdapat jumlah *hidden* neuron $\tilde{N} \leq N$ sedemikian sehingga untuk bobot *input* w_j dan bias b_j yang dipilih secara acak, error pelatihan dapat diminimalkan mendekati ϵ , yaitu $\|H\beta - Y\| < \epsilon$.

Berdasarkan Teorema 1 dan Teorema 2, karena w_j dan b_j telah ditetapkan secara acak dan dibiarkan tetap, maka Persamaan (17) berubah menjadi sistem persamaan linear yang hanya perlu mencari $\beta = H^{-1}Y$. Namun, matriks H pada merupakan matriks non-persegi berukuran $N \times \tilde{N}$, sehingga tidak memiliki invers standar H^{-1} . Selain itu, sistem persamaan tersebut akan menjadi *overdetermined* karena jumlah persamaan (N) lebih banyak dibandingkan jumlah variabel yang dicari ($\tilde{N}$).

Secara umum, tidak ada solusi eksak β yang memenuhi secara sempurna. Karena H^{-1} tidak terdefinisi, proses pelatihan ELM beralih mencari nilai taksiran $\hat{\beta}$ yang dapat meminimalkan error kuadrat terkecil (*Least Square Solution*) di mana $\|H\hat{\beta} - Y\| = \min_{\beta} \|H\beta - Y\|$.

Secara matematis, solusi ini diperoleh dengan memanipulasi persamaan agar matriks H memiliki invers. Karena H bukan matriks persegi, langkah pertama yaitu mengalikan kedua ruas Persamaan (17) dengan transpose dari matriks H (H^T) dari sebelah kiri.

$$\begin{aligned} H^T(H\beta) &= H^TY \\ (H^TH)\beta &= H^TY \end{aligned}$$

Matriks hasil perkalian (H^TH) menjadi matriks persegi $\tilde{N} \times \tilde{N}$. Berdasarkan Teorema 1, karena parameter *input* dibangkitkan secara acak, kolom-kolom H saling bebas linear (*full column rank*), sehingga (H^TH) bersifat *invertible* (memiliki invers). Langkah selanjutnya yaitu mengalikan kedua ruas dengan $(H^TH)^{-1}$.

$$\begin{aligned} ((H^TH)^{-1}(H^TH))\beta &= (H^TH)^{-1}H^TY \\ I \cdot \beta &= (H^TH)^{-1}H^TY \\ \beta &= (H^TH)^{-1}H^TY \end{aligned}$$

Menurut Huang et al. (2006), solusi di atas merupakan solusi yang optimal untuk masalah *least-squares* dengan norma terkecil (*minimum norm least-squares*

solution). Solusi ini bersifat unik dan diberikan oleh rumus pada Persamaan (18) berikut.

$$\hat{\beta} = H^+ Y \quad (18)$$

Dengan

$Y = \begin{bmatrix} y_1 \\ \dots \\ y_N \end{bmatrix}_{N \times m}$ merepresentasikan matriks yang berisi seluruh target *output* y_i ,

$\hat{\beta} = \begin{bmatrix} \hat{\beta}_1 \\ \dots \\ \hat{\beta}_{\tilde{N}} \end{bmatrix}_{\tilde{N} \times m}$ merupakan matriks yang berisi semua bobot $\hat{\beta}_j$, dan

$H^+ = (H^T H)^{-1} H^T$ adalah matriks *Moore Penrose Generalized Inverse* dengan orde $\tilde{N} \times N$ dari matriks H (Huang, 2006).

Algoritma ELM terdiri dari dua tahapan utama, yaitu pelatihan dan pengujian. Langkah-langkah proses pelatihan adalah sebagai berikut (Sukma, et al., 2021).

1. Tentukan jumlah *hidden neuron* ($\tilde{N}$) dan fungsi aktivasi $g(x)$, serta inialisasi bobot *input* (W) dan bias (b) secara *random*.
2. Hitung matriks inialisasi training (H_{init}) dengan rumus:

$$H_{init} = X_{train} \cdot W^T + b \quad (19)$$

Keterangan:

X_{train} : matriks data masukan (*input*) pelatihan

W^T : tranpose matriks bobot *input*

b : bias

3. Terapkan fungsi aktivasi $g(x)$ pada setiap elemen matriks inialisasi (H_{init}) untuk memperoleh matriks *output hidden layer* (H). Fungsi aktivasi yang digunakan pada penelitian ini adalah fungsi aktivasi sigmoid biner.

$$H = g(H_{init}) = \frac{1}{1 + \exp(-H_{init})} \quad (20)$$

4. Perhitungan *Moore Penrose Inverse*

Moore Penrose Inverse digunakan karena memiliki kelebihan untuk mendapatkan nilai invers dari matriks yang memiliki ordo berbeda. Berikut rumus matematis untuk menghitung *Moore Penrose Inverse*:

$$H^+ = (H^T H)^{-1} \cdot H^T \quad (21)$$

Keterangan:

H^T : transpose matriks *output hidden layer*

$(H^T * H)^{-1}$: invers dari perkalian matriks H^T dan H

5. Hitung matriks *output weights* menggunakan rumus sebagai berikut

$$\hat{\beta} = H^+ \cdot Y_{train} \quad (22)$$

Selanjutnya, langkah-langkah proses pengujian adalah sebagai berikut.

1. Ambil nilai parameter W, b , dan β dari proses *training*.
2. Hitung matriks inisialisasi testing dengan rumus:

$$H_{init_testing} = X_{testing} \cdot W^T + b \quad (23)$$

3. Terapkan fungsi aktivasi $g(x)$ pada $H_{init_testing}$ untuk memperoleh matriks *output hidden layer* data uji menggunakan rumus berikut:

$$H = g(H_{init_testing}) = \frac{1}{1 + \exp(-H_{init_testing})} \quad (24)$$

4. Hitung *output layer* (prediksi) dengan rumus:

$$\hat{Y} = H \cdot \hat{\beta} \quad (25)$$

1.6.7.2 Moore Penrose Inverse

Penyelesaian sistem persamaan linear umum $Ax = b$ ketika A bersifat singular dan non-persegi dapat dipecahkan menggunakan *Moore Penrose Generalized Inverse*.

Definisi 1. Diberikan matriks $A \in R_{m \times n}$. Terdapat matriks *unique* $A^+ \in R_{m \times n}$ yang dinamakan *Moore Penrose Generalized Inverse* dari matriks jika memenuhi (Macausland, 2014):

- (i) $AA^+A = A$
- (ii) $A^+AA^+ = A^+$
- (iii) $(AA^+)^T = AA^+$
- (iv) $(A^+A)^T = A^+A$

Macausland (2014) menyatakan bahwa jika matriks A memiliki *full column rank*, dimana kolom-kolom matriksnya linear independen, maka *Moore Penrose Generalized Inverse* dapat dihitung secara langsung melalui formula yang lebih sederhana.

Teorema 3. Jika matriks $A \in R_{m \times n}$ dengan $m \geq n$ memiliki *full column rank*, maka matriks $(A^T A)$ bersifat *invertible* (non-singular) dan *Moore Penrose Generalized Invers* dari A diberikan oleh:

$$A^+ = (A^T A)^{-1} A^T \quad (26)$$

Bukti:

Untuk membuktikan Teorema 3, perlu ditunjukkan dua hal, yaitu bahwa matriks $A^T A$ bersifat *invertible* (non-singular) dan $A^+ = (A^T A)^{-1} * A^T$ memenuhi keempat sifat *Moore Penrose Generalized Inverse* pada Definisi 1.

Langkah 1: Membuktikan “Jika matriks $A \in R_{m \times n}$ dengan $m \geq n$ memiliki *full column rank*, maka matriks $(A^T A)$ bersifat *invertible* (non-singular)”

Diketahui matriks A berukuran $m \times n$ memiliki kolom-kolom yang bebas linear (*full column rank*). Akan ditunjukkan bahwa matriks $A^T A$ bersifat *invertible*.

Karena matriks A memiliki kolom-kolom yang bebas linear, maka sistem persamaan linear homogen $Ax = 0$ hanya memiliki satu solusi (solusi trivial), yaitu vektor nol ($x = 0$). Dengan kata lain, *null space* (ruang nol) dari matriks A hanya berisi vektor nol.

$$Ax = 0 \Rightarrow x = 0$$

Matriks A berukuran $m \times n$, sehingga matriks A^T akan berukuran $n \times m$. Berdasarkan aturan perkalian matriks, perkalian antara A^T dan A akan menghasilkan dimensi:

$$(m \times n) \times (n \times m) \rightarrow (m \times m)$$

Karena jumlah kolom dan baris sama ($m \times m$), maka $A^T A$ merupakan sebuah matriks persegi. Ini memenuhi syarat mutlak bagi sebuah matriks untuk dapat memiliki invers.

Untuk membuktikan bahwa matriks persegi $A^T A$ adalah *invertible*, perlu ditunjukkan bahwa kolom-kolom dari $A^T A$ juga bebas linear. Artinya, perlu dibuktikan bahwa *null space* dari $A^T A$ hanya berisi vektor nol.

Misalkan terdapat sebuah vektor sembarang v yang merupakan anggota *null space* dari $A^T A$, sehingga memenuhi persamaan homogen berikut.

$$(A^T A)v = 0$$

dengan 0 merepresentasikan vektor nol.

Kalikan kedua ruas persamaan dengan v^T dari sebelah kiri, diperoleh

$$v^T (A^T A)v = v^T 0$$

$$v^T (A^T A)v = 0$$

Berdasarkan sifat transpose perkalian matriks, berlaku $(AB)^T = B^T A^T$. Maka, $v^T A$ dapat ditulis ulang sebagai $(Av)^T$. Substitusikan kembali ke dalam persamaan sehingga diperoleh

$$(Av)^T (Av) = 0$$

Perkalian di atas ekuivalen dengan hasil kali titik (*dot product*) dari vektor Av dengan dirinya sendiri, yaitu $(Av) \cdot (Av)$. Secara matematis, hasil kali titik sebuah vektor dengan dirinya sendiri adalah sama dengan kuadrat dari panjang atau norma vektor tersebut, sehingga persamaan dapat ditulis menjadi

$$\|Av\|^2 = 0$$

Satu-satunya vektor di dalam ruang riil yang memiliki panjang sama dengan 0 adalah vektor nol itu sendiri. Oleh karena itu, persamaan di atas mengimplikasikan bahwa:

$$Av = 0$$

Persamaan $Av = 0$ menunjukkan bahwa vektor v berada di dalam *null space* dari matriks A . Berdasarkan informasi bahwa kolom-kolom matriks A saling bebas linear, maka satu-satunya solusi yang memungkinkan untuk persamaan $Av = 0$ adalah $v = 0$. Oleh karena itu terbukti bahwa:

$$v = 0$$

Hal ini membuktikan bahwa sistem $(A^T A)v = 0$ hanya memiliki solusi trivial yang berarti kolom-kolom pada matriks $A^T A$ saling bebas linear.

Karena $A^T A$ adalah matriks persegi $m \times m$ dengan kolom-kolom bebas linear atau memiliki *full rank*, maka berdasarkan Teorema Matriks *Invertible*, matriks tersebut dipastikan non-singular. Kondisi ini secara ekuivalen menjamin bahwa matriks dapat direduksi menjadi matriks identitas (I) melalui Operasi Baris Elementer yang secara mutlak membuktikan bahwa matriks $A^T A$ bersifat *invertible*.

Langkah 2: Membuktikan “Jika matriks $A \in R_{m \times n}$ dengan $m \geq n$ memiliki *full column rank*, maka *Moore Penrose Generalized Invers* dari A diberikan oleh $A^+ = (A^T A)^{-1} A^T$ ”

Akan ditunjukkan bahwa $A^+ = (A^T A)^{-1} A^T$ memenuhi keempat sifat *Moore Penrose Generalized Inverse* pada Definisi 1.

(i) $AA^+A = A$

$$A[(A^T A)^{-1} A^T]A = A$$

$$A(A^T A)^{-1} (A^T A) = A$$

$$A \cdot I = A$$

$$A = A$$

(ii) $A^+AA^+ = A^+$

$$[(A^T A)^{-1} A^T]A[(A^T A)^{-1} A^T] = A^+$$

$$(A^T A)^{-1} (A^T A) [(A^T A)^{-1} A^T] = A^+$$

$$I[(A^T A)^{-1} A^T] = A^+$$

$$(A^T A)^{-1} A^T = A^+$$

$$A^+ = A^+$$

(iii) $(AA^+)^T = AA^+$

$$(A(A^T A)^{-1} A^T)^T = AA^+$$

$$((A^T A)^{-1} A^T)^T A^T = AA^+$$

$$(A^T)^T ((A^T A)^{-1})^T A^T$$

$$A((A^T A)^{-1})^T A^T = AA^+ \text{ (sifat transpose inverse)}$$

$$A(A^T A)^{-1} A^T = AA^+ \text{ (karena } A^T A \text{ simetris)}$$

$$AA^+ = AA^+$$

(iv) $(A^+A)^T = A^+A$

Karena $A^+A = (A^T A)^{-1} A^T A = (A^T A)^{-1} (A^T A) = I$ dan matriks identitas $I^T = I$, maka

$$((A^T A)^{-1} A^T A)^T = A^+A$$

$$(A^+A)^T = A^+A$$

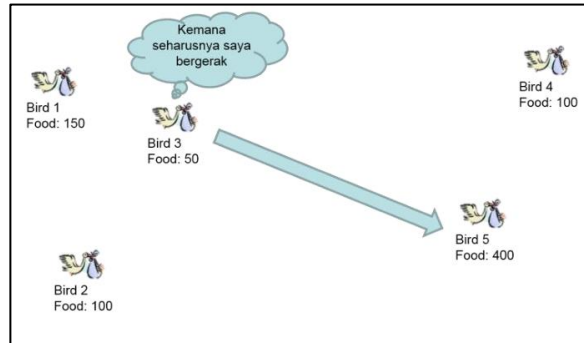
$$A^+A = A^+A$$

Karena keempat sifat terpenuhi, maka terbukti bahwa $A^+ = (A^T A)^{-1} A^T$ adalah *Moore Penrose Generalized Inverse* dari A jika A memiliki *full column rank*. Algoritma ELM memanfaatkan formula $H^+ = (H^T H)^{-1} H^T$ untuk menghitung *Moore Penrose Generalized Inverse* dengan asumsi bahwa matriks H yang dihasilkan dari *hidden neuron* acak memenuhi kondisi *full column rank*.

1.6.8 Particle Swarm Optimization

Algoritma PSO pertama kali diperkenalkan pada tahun 1995 oleh Kennedy dan Eberhart sebagai suatu metode heuristik yang dapat digunakan sebagai solusi dari

masalah optimasi non-linier. PSO didefinisikan sebagai sebuah algoritma optimasi berbasis populasi yang diadopsi dari perilaku kawanan burung dalam mencari makanan. Gambar 9 berikut merupakan ilustrasi cara kerja algoritma PSO.



Gambar 9. Ilustrasi Algoritma PSO

Proses optimasi pada algoritma PSO dimulai dengan menginisialisasi sekumpulan solusi kandidat secara acak yang disebut sebagai partikel. Setiap partikel merupakan individu dalam populasi (kawanan) yang merepresentasikan sebuah solusi alternatif dalam ruang pencarian dan posisinya dapat dievaluasi menggunakan sebuah fungsi objektif untuk mengukur kualitas solusinya. Selama proses iterasi, setiap partikel akan menyesuaikan posisi dan kecepatannya. Penyesuaian ini dilakukan oleh mekanisme berbagi informasi, di mana setiap partikel mengkomunikasikan posisi terbaik yang telah ditemukan kepada partikel lain dalam kawanannya. Hal ini menyebabkan setiap partikel memiliki kecenderungan untuk bergerak menuju posisi-posisi yang dianggap terbaik dalam ruang pencarian tersebut (Darmayanti, et al., 2018).

Cholissodin dan Riyandani (2016) mengemukakan bahwa algoritma PSO terdiri atas beberapa faktor penyusun, di antaranya:

1. *Swarm*, yaitu jumlah partikel yang ada dalam suatu populasi. Ukurannya tergantung pada kompleksitas permasalahan yang akan diselesaikan.
2. Partikel, didefinisikan sebagai suatu individu yang ada dalam suatu *swarm*. Setiap partikel merepresentasikan solusi penyelesaian masalah, di mana representasi solusi tersebut akan menentukan lokasi dan kecepatannya. Berikut notasi dari partikel.

$$S_j = \{\bar{x}_j^{(1)}, \bar{x}_j^{(2)}, \dots, \bar{x}_j^{(N)}\}$$

Keterangan:

S_j : populasi partikel

$\bar{x}_j^{(i)}$: lokasi partikel ke- i pada iterasi ke- j

N : banyaknya partikel

3. Kecepatan (*velocity*), yaitu vektor penentu arah perpindahan lokasi partikel. Perubahan kecepatan dilakukan pada setiap iterasi untuk memperbaiki posisi partikel semula. Kecepatan disimbolkan sebagai berikut.

$$\bar{v}_j^{(i)} = \{\bar{v}_j^{(1)}, \bar{v}_j^{(2)}, \dots, \bar{v}_j^{(N)}\}$$

Keterangan:

$\bar{v}_j^{(i)}$: kecepatan partikel ke- i pada iterasi ke- j

4. Personal best (P_{best}), yaitu posisi terbaik yang ditemukan oleh suatu partikel dengan membandingkan nilai *fitness* pada posisi partikel saat ini dengan posisi partikel sebelumnya. P_{best} dipersiapkan untuk memperoleh nilai terbaik.

$$\bar{P}_{best_j}^{(i)} = \min_{\bar{x}} \{f(\bar{x}_j^{(1)}), f(\bar{x}_j^{(2)}), \dots, f(\bar{x}_j^{(N)})\}, \quad \bar{P}_{best_j}^{(i)} \in \mathbb{R}^n$$

Keterangan:

$\bar{P}_{best_j}^{(i)}$: P_{best} ke- i pada iterasi ke- j

5. Global best (G_{best}), yaitu posisi terbaik partikel yang diperoleh melalui perbandingan nilai *fitness* terbaik dari keseluruhan partikel yang ada dalam *swarm*.

$$\bar{P}_{best} = \min\{\bar{P}_{best_j}^{(1)}, \bar{P}_{best_j}^{(2)}, \dots, \bar{P}_{best_j}^{(N)}\}$$

6. Bobot inersia (w), digunakan untuk mengontrol dampak dari perubahan kecepatan (*velocity*) yang diberikan oleh partikel.
7. Koefisien akselerasi, yaitu faktor yang mengontrol sejauh mana partikel melakukan perpindahan dalam satu iterasi. Secara umum, nilai koefisien akselerasi C_1 dan C_2 berada dalam rentang 0 sampai 4. Meskipun demikian, nilainya dapat ditentukan sendiri sesuai dengan konteks permasalahan pada penelitian.

Selanjutnya, terdapat dua rumus yang digunakan pada algoritma PSO, meliputi:

Rumus memperbarui kecepatan dinyatakan sebagai berikut.

$$\bar{v}_{j+1}^{(i)} = w \cdot \bar{v}_j^{(i)} + C_1 \cdot r_1 (\bar{P}_{best_j}^{(i)} - \bar{x}_j^{(i)}) + C_2 \cdot r_2 (\bar{G}_{best} - \bar{x}_j^{(i)}) \quad (27)$$

Rumus memperbarui lokasi dinyatakan sebagai berikut.

$$\bar{x}_{j+1}^{(i)} = \bar{x}_j^{(i)} + \bar{v}_{j+1}^{(i)} \quad (28)$$

dengan

$\bar{v}_{j+1}^{(i)}$: nilai *update* kecepatan (kecepatan partikel ke- i pada iterasi ke- $j + 1$)

$\bar{v}_j^{(i)}$: kecepatan partikel ke- i pada iterasi ke- j

$\bar{x}_j^{(i)}$: posisi partikel ke- i pada iterasi ke- j

$\bar{P}_{best_j}^{(i)}$: posisi optimal ke- i pada iterasi ke- j

$\bar{G}_{best}$: global optimal ke- i pada iterasi ke- j

w : nilai vektor bobot inersia

C_1, C_2 : koefisien akselerasi

r_1, r_2 : bilangan acak $[0, 1]$

1.6.8.1 Algoritma Particle Swarm Optimization

Penerapan algoritma PSO melalui serangkaian proses dengan langkah-langkah sebagai berikut (Darmayanti, et al., 2018).

1. Menginisialisasi nilai awal, meliputi kecepatan awal, posisi awal, Pbest, dan Gbest.
 - a) Inisialisasi kecepatan awal partikel dengan nilai 0.
 - b) Inisialisasi posisi awal partikel dibangkitkan secara acak.
 - c) Hitung nilai *fitness*, di mana nilai *fitness* didefinisikan sebagai nilai MAPE sehingga tujuan dari algoritma ini adalah menemukan nilai MAPE atau *fitness* terendah.
 - d) Inisialisasi Pbest dan Gbest awal, dimana Pbest memiliki nilai yang sama dengan nilai posisi awal partikel dan Gbest memakai nilai dari Pbest dengan hasil *fitness* terendah.
2. Memperbaharui (*update*) nilai partikel
 - a) *Update* kecepatan partikel menggunakan rumus pada Persamaan (27).
 - b) *Update* posisi partikel menggunakan rumus pada Persamaan (28).
 - c) *Update* nilai Pbest dan Gbest. Nilai Pbest diperoleh melalui perbandingan nilai *fitness* Pbest pada iterasi sebelumnya dengan nilai *fitness* pada *update* posisi, di mana nilai *fitness* terkecil akan menjadi nilai Pbest yang baru. Sedangkan nilai Gbest diperoleh melalui nilai Pbest dengan nilai *fitness* terendah.
3. Evaluasi kondisi pemberhentian. Jika kondisi pemberhentian terpenuhi, maka algoritma dihentikan dan diperoleh *output* algoritma dalam hal ini Gbest. Sebaliknya jika kondisi pemberhentian tidak terpenuhi, maka seluruh algoritma pada langkah ke-2 akan diulangi sampai diperoleh *output* dengan nilai terbaik. Terdapat beberapa kondisi pemberhentian yang umum digunakan, antara lain:
 - a) Iterasi maksimum telah tercapai.
 - b) Iterasi berhenti akan tetapi tidak ada perubahan yang signifikan.
 - c) Telah dicapai waktu maksimum.

1.6.9 Hybrid ELM-PSO

ELM merupakan salah satu metode JST yang memiliki kelemahan pada pemilihan bobot *input* dan bias yang ditentukan secara acak sehingga sering kali menyebabkan hasil yang diperoleh tidak optimal. Untuk mengoptimalkan hasil peramalan pada ELM, dapat digunakan algoritma optimasi PSO, di mana PSO berperan untuk menemukan nilai bobot *input* dan bias yang optimal. Gabungan dari kedua metode ini telah terbukti berhasil diterapkan dalam berbagai kasus, seperti peramalan dan klasifikasi.

Algoritma *hybrid* ELM-PSO secara umum melalui proses sebagai berikut (Darmayanti, et al., 2018):

1. Menentukan panjang dimensi partikel dan iterasi maksimum. Panjang dimensi partikel diperoleh dari $\text{jumlah fitur} \times \text{jumlah hidden neuron} + \text{bias}$. Sedangkan iterasi maksimum ditentukan sendiri sesuai dengan kebutuhan.
2. Menginisialisasi kecepatan partikel sesuai dengan langkah-langkah algoritma PSO yang telah dijelaskan sebelumnya, yaitu kecepatan awal partikel bernilai 0.

3. Menginisialisasi posisi partikel secara acak dengan panjang dimensi partikel yang dijelaskan pada langkah 1.
4. Menjalankan proses pelatihan pada ELM dilanjutkan dengan proses pengujian melalui mekanisme yang telah dijelaskan pada Subbab 1.6.3.1, menggunakan nilai bobot *input* dan bias yang diperoleh dari PSO untuk mendapatkan nilai MAPE yang nantinya akan digunakan sebagai *fitness* pada posisi partikel, *update* Pbest dan Gbest.
5. Menjalankan langkah-langkah algoritma PSO pada Subbab 1.6.4.1 hingga terpenuhi kondisi pemberhentian dan diperoleh Gbest dengan *fitness* terendah.

BAB II

METODOLOGI PENELITIAN

2.1 Jenis dan Sumber Data

Data yang digunakan pada penelitian ini adalah data kuantitatif bersifat deret waktu (*time series*). Data yang digunakan merupakan data sekunder yang diperoleh dari berbagai sumber publikasi resmi. Periode data yang dikumpulkan yaitu 10 tahun, terhitung sejak Juli 2015 hingga Juli 2025. Seluruh data yang diperoleh kemudian diolah untuk kepentingan penelitian. Adapun perincian data yang digunakan adalah sebagai berikut.

1. Nilai Tukar (Kurs) USD/IDR

Data Kurs USD/IDR yang digunakan pada penelitian ini adalah data harian kurs referensi Jakarta Interbank Spot Dollar Rate (JISDOR). Data ini digunakan sebagai variabel *output* (target) sekaligus sebagai data dasar untuk 5 variabel masukan, yaitu *lag* deret data historis waktu sebelumnya ($y_{t-1}, y_{t-2}, \dots, y_{t-k}$). Dalam penelitian ini digunakan sebanyak 5 lag data Kurs USD/IDR, artinya terdapat sebanyak 5 periode (hari) sebelumnya yang digunakan sebagai *input* untuk meramalkan kurs satu periode berikutnya. Data Kurs USD/IDR yang digunakan diperoleh dari situs resmi Bank Indonesia (BI).

2. Suku Bunga

Suku bunga merupakan salah satu variabel makroekonomi yang memengaruhi pergerakan nilai tukar, sehingga penelitian ini menggunakan data suku bunga. Data yang digunakan adalah data bulanan suku bunga acuan Indonesia dan Amerika Serikat. Untuk Indonesia, digunakan data BI-Rate yang diperoleh dari situs resmi Bank Indonesia. Sedangkan untuk Amerika Serikat, digunakan data Federal Funds Rate yang diperoleh dari situs Trading Economics.

3. Inflasi

Penelitian ini menggunakan data inflasi karena Inflasi juga merupakan variabel yang memengaruhi nilai tukar seperti yang dijelaskan pada Subbab 1.6.1.1. Data yang digunakan adalah inflasi bulanan dari situs resmi Bank Indonesia untuk inflasi Indonesia dan data *annual inflation rates* dari situs Bureau of Labor Statistics untuk Amerika Serikat.

4. Data Ekspor dan Impor

Data yang dikumpulkan adalah data bulanan total ekspor dan impor Indonesia yang diperoleh dari situs resmi Satu Data Perdagangan Kementerian Perdagangan Indonesia.

2.2 Variabel Penelitian

Variabel dalam penelitian ini terbagi menjadi dua, yaitu variabel masukan (*input*) yang digunakan untuk proses peramalan dan variabel keluaran (*output*) yang menjadi target peramalan.

2.2.1 Variabel Masukan (*Input*)

Variabel masukan (X) atau variabel independen adalah data-data yang menjadi *input* bagi model *hybrid* ELM-PSO untuk mempelajari pola pergerakan nilai tukar. Berdasarkan studi literatur dan analisis korelasi, ditetapkan 8 variabel masukan yang terdiri dari data variabel internal (data historis kurs USD/IDR) dan variabel eksternal (data fundamental makroekonomi). Variabel masukan tersebut dijabarkan sebagai berikut.

Tabel 4. Variabel *Input* Penelitian

Variabel	Satuan	Deskripsi
Kurs ke $t - 1$	Rupiah	Nilai kurs referensi JISDOR 1 hari sebelumnya.
Kurs ke $t - 2$	Rupiah	Nilai kurs referensi JISDOR 2 hari sebelumnya.
Kurs ke $t - 3$	Rupiah	Nilai kurs referensi JISDOR 2 hari sebelumnya.
Kurs ke $t - 4$	Rupiah	Nilai kurs referensi JISDOR 2 hari sebelumnya.
Kurs ke $t - 4$	Rupiah	Nilai kurs referensi JISDOR 2 hari sebelumnya.
Selisih Suku Bunga	Persen (%)	Selisih tingkat suku bunga antara Indonesia dan Amerika Serikat.
Selisih Inflasi	Persen (%)	Selisih tingkat inflasi antara Indonesia dan Amerika Serikat.
Rasio Ekspor Impor	-	Perbandingan antara total ekspor dan impor Indonesia.

2.2.2 Variabel Keluaran (*Output*)

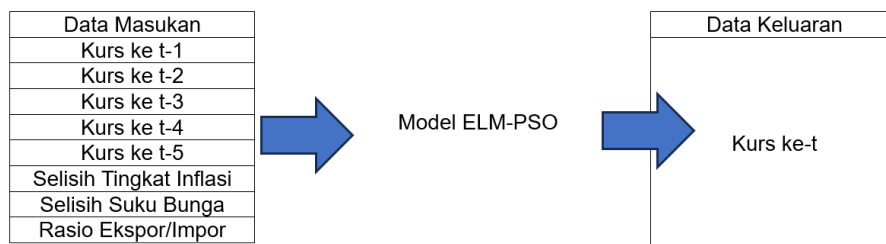
Variabel keluaran (Y) atau variabel dependen adalah target yang ingin diramalkan oleh model. Dalam penelitian ini variabel keluaran yang digunakan adalah sebagai berikut.

Tabel 5. Variabel *Output* Penelitian

Variabel	Satuan	Deskripsi
Kurs ke t	Rupiah	Nilai kurs referensi JISDOR pada hari ke- t .

2.2.3 Skema Data Masukan dan Keluaran

Skema pola data masukan dan keluaran pada penelitian ini ditunjukkan pada gambar berikut.



Gambar 10. Skema Data Masukan dan Keluaran

2.3 Tahapan Penelitian

Penelitian ini terdiri dari beberapa tahapan sistematis, mulai dari identifikasi masalah hingga pengujian dan evaluasi dengan rincian sebagai berikut.

2.3.1 Identifikasi Masalah

Tahap awal penelitian adalah identifikasi masalah untuk memperoleh gambaran umum yang relevan untuk penelitian. Melalui tahapan ini, diperoleh informasi bahwa fluktuasi nilai tukar (kurs) Rupiah Terhadap Dollar AS yang tinggi menimbulkan ketidakpastian dan risiko, terutama bagi para pelaku ekonomi. Oleh karena itu, diperlukan model peramalan yang akurat untuk memprediksi pergerakan kurs. Penelitian ini kemudian diusulkan untuk menerapkan metode *hybrid* Extreme Learning Machine (ELM) yang dioptimasi oleh Particle Swarm Optimization (PSO) untuk membangun model peramalan.

2.3.2 Studi Literatur

Studi literatur dilakukan dengan mengumpulkan landasan teori dan meninjau penelitian terdahulu untuk memperdalam pemahaman penulis terkait dengan penelitian yang akan dilakukan. Studi literatur dilakukan dengan mengumpulkan sumber-sumber bacaan seperti artikel jurnal, skripsi, buku, dan lain sebagainya.

2.3.3 Pengumpulan Data

Tahap ini berfokus pada pengumpulan data sekunder yang relevan dan aktual untuk digunakan dalam membangun model. Data yang dikumpulkan adalah seluruh data yang telah dipaparkan pada Subbab 2.1.

2.3.4 Pengolahan Data Awal (*Pre-processing Data*)

Data mentah yang telah dikumpulkan tidak dapat digunakan secara langsung oleh model. Data yang terkumpul perlu melalui tahap *Pre-processing* untuk memastikan kualitas data, meliputi penyelarasan frekuensi, pembersihan, normalisasi, dan pembagian data.

2.3.5 Perancangan dan Pelatihan Model

Tahap ini adalah inti dari penelitian di mana model *hybrid* ELM-PSO dirancang dan dilatih. Berbeda dengan ELM standar yang bobot *input* serta biasanya dibangkitkan secara acak, pada penelitian ini parameter tersebut dioptimasi oleh PSO untuk menemukan hasil yang lebih optimal. Perancangan optimasi dimulai dengan mendefinisikan partikel PSO untuk membangkitkan satu *set* bobot *input* dan *hidden* bias. Selanjutnya, ditetapkan fungsi *fitness* untuk PSO sehingga akan ditemukan satu *set* bobot dan bias dengan *error* peramalan terkecil pada data latih. Keseluruhan proses pelatihan juga mencakup pengujian *hyperparameter* PSO dan jumlah *hidden neuron* pada ELM. Pengujian *hyperparameter* dilakukan menggunakan pendekatan *One Factor at A Time* (OFAT) yang diadopsi dari penelitian Ramos, et al. (2024). Prinsip kerja metode OFAT adalah memvariasikan nilai satu parameter tertentu pada suatu rentang pengujian, sementara parameter-parameter lainnya dijaga agar tetap

konstan (Frey, et al., 2023). Pendekatan ini digunakan untuk mengidentifikasi dampak dari setiap *hyperparameter* terhadap stabilitas dan akurasi model peramalan tanpa ada bias interaksi dari perubahan parameter lain. Pada penelitian ini, pengujian *hyperparameter* dilakukan berbasis pada nilai *fitness* dalam hal ini MAPE, di mana nilai parameter optimal akan diambil dari nilai parameter dengan *fitness* terbaik (MAPE terendah).

2.3.6 Pengujian dan Evaluasi

Setelah model dilatih dan ditemukan parameter terbaik, selanjutnya dilakukan tahap pengujian untuk mengevaluasi kinerja model pada data uji.

2.3.6.1 Skenario Pengujian dan Evaluasi

Untuk memastikan keunggulan metode yang diusulkan, penelitian ini membandingkan tiga skenario model, yaitu:

1. ELM Standar: Model ini adalah model dasar ELM yang tidak dioptimasi oleh PSO sehingga bobot *input* dan biasnya dibangkitkan secara acak.
2. ELM-PSO (*Baseline*): Model ini adalah model ELM yang dioptimasi oleh PSO tetapi parameter pada PSO ditentukan sendiri berdasarkan parameter yang umumnya digunakan pada literatur penelitian sebelumnya.
3. ELM-PSO (OFAT): Model ini adalah model ELM yang dioptimasi bobot dan biasnya oleh PSO serta telah menggunakan parameter PSO yang diperoleh dari hasil pengujian *hyperparameter*.

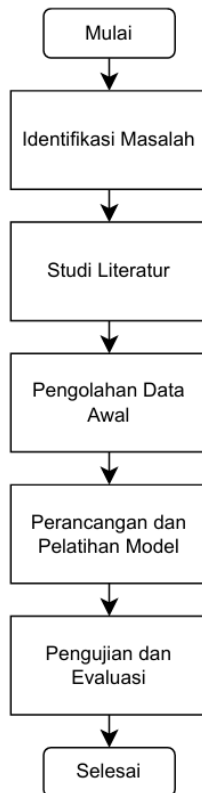
2.3.6.2 Metrik Evaluasi

Kinerja dari ketiga skenario model tersebut diukur dan dibandingkan menggunakan empat metrik evaluasi yang umumnya digunakan untuk regresi, antara lain: MAPE, MAE, RMSE, dan R^2 .

2.4 Diagram Alir

2.4.1 Diagram Alir Penelitian

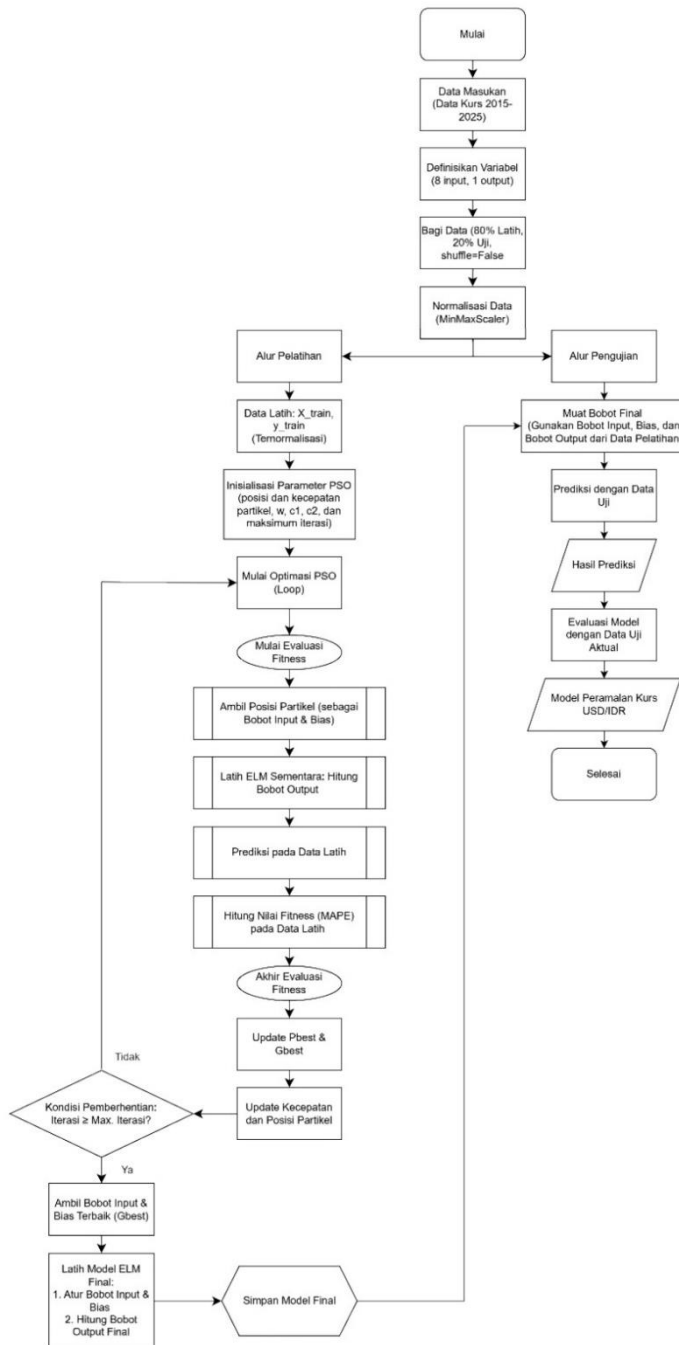
Berdasarkan penjelasan pada tahap penelitian, alur kerja penelitian ditunjukkan oleh diagram alir berikut.



Gambar 11. Diagram Alir Penelitian

2.4.2 Diagram Alir Proses ELM-PSO

Model peramalan menggunakan *hybrid* ELM-PSO melalui serangkaian proses yang dapat digambarkan pada diagram alir sebagai berikut.



Gambar 12. Diagram Alir Proses ELM-PSO