

**PREDIKSI KONSUMSI ENERGI MENGGUNAKAN *REGRESI*  
*LINIER* DAN *MULTILAYER PERCEPTRON* (MLP) PADA  
ALGORITMA KONSENSUS *PROOF OF WORK* (POW)**

**LAPORAN TUGAS AKHIR**

Proposal ini disusun untuk memenuhi salah satu syarat memperoleh Gelar  
Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi  
Industri Universitas Islam Sultan Agung  
Semarang



Disusun Oleh :

**WAHID IVAN SAPUTRA**

**NIM 32602100123**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM SULTAN AGUNG  
SEMARANG**

**2025**

**ENERGY CONSUMPTION PREDICTION USING *LINEAR REGRESSION* AND *MULTILAYER PERCEPTRON* (MLP) IN THE *PROOF OF WORK* (POW) CONSENSUS ALGORITHM**

**FINAL ASSIGNMENT REPORT**

This proposal was prepared to fulfill one of the requirements for obtaining a Bachelor's Degree (S1) in the Informatics Engineering Study Program, Industrial Technology, Sultan Agung Islamic University

Semarang,



Compiled by:

**WAHID IVAN SAPUTRA**

**NIM 32602100123**

**MAJORING OF INFORMATICS ENGINEERING  
INDUSTRIAL TECHNOLOGY FACULTY  
SULTAN AGUNG ISLAMIC UNIVERSITY  
SEMARANG**

**2025**

**LEMBAR PENGESAHAN  
TUGAS AKHIR**

**PREDIKSI KONSUMSI ENERGI MENGGUNAKAN *REGRESI LINIER*  
DAN *MULTILAYER PERCEPTRON (MLP)* PADA ALGORITMA \*  
*KONSENSUS PROOF OF WORK (POW)***

**WAHID IVAN SAPUTRA  
NIM 32602100123**

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir  
Program Studi Teknik Informatika  
Universitas Islam Sultan Agung  
Pada tanggal : 01 September 2025

**TIM PENGUJI UJIAN SARJANA :**

**IMAM M. IBNU  
SUBROTO, ST., MSC., PH.D.  
NIK. 210 600 017  
(Penguji 1)**

02-09-2025

**Sam Farisa Chaerul Haviana,  
ST., M.Kom  
NIK. 210616049  
(Penguji 2)**

03-11-2025

**Bagus SWP, S.Kom, M.Cs  
NIK. 210601024  
(Pembimbing)**

02-09-2025

Semarang, 13-11-2025  
Mengetahui,

Kaprodi Teknik Informatika  
Universitas Islam Sultan Agung

**Moch. Taufik, ST, MIT  
NIK. 210604034**

## SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini :

Nama : Wahid Ivan Saputra

Nim : 32602100123

Judul Tugas Akhir : Prediksi Konsumsi Energi Menggunakan *Regresi Linier* dan *Multilayer Perceptron* (MLP) pada Algoritma Konsensus *Proof of Work* (PoW)

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapa pun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.



Semarang,

Yang Menyatakan,



Wahid Ivan Saputra



## PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan di bawah ini :

Nama : Wahid Ivan Saputra

NIM : 32602100123

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Alamat Asal : Semarang

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Prediksi Konsumsi Energi Menggunakan *Regresi Linier* dan *Multilayer Perceptron* (MLP) pada Algoritma Konsensus *Proof of Work* (PoW).

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan di internet dan media lain untuk kepentingan akademis selama tetap mencantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

**UNISSULA**

جامعة سلطان أبوبوع الإسلامية

Semarang,

Yang menyatakan,



Wahid Ivan Saputra

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada ALLAH SWT, yang telah memberikan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Prediksi Konsumsi Energi Menggunakan *Regresi Linier* dan *Multilayer Perceptron* (MLP) pada Algoritma Konsensus *Proof of Work* (PoW)” ini dengan baik. Dengan penuh rasa hormat, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, SH., M.Hum yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T.
3. Dosen pembimbing penulis Bapak Bagus Satrio Waluyo Poetro, S.Kom, M.Cs yang telah memberikan arahan, bimbingan, dan saran yang berarti dalam penyelesaian tugas akhir ini.
4. Seluruh dosen Program Studi Teknik Informatika, Fakultas Teknologi Industri UNISSULA yang telah memberikan ilmunya kepada penulis.
5. Orang tua penulis, yang selalu memberikan segala doa, dukungan, dan motivasi dengan penuh limpahan kasih sayang sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik
6. Untuk Diri Sendiri yang tidak pernah menyerah dan berhenti mencoba sesulit apapun rintangan kuliah dan selama proses penyusunan laporan ini.
7. Rekan seperjuangan, yang telah memberikan dukungan moral, motivasi, serta semangat dalam proses penyelesaian tugas akhir ini
8. Dan kepada seluruh pihak yang tidak dapat saya sebutkan satu persatu.

Semarang, 26 November 2025



Wahid Ivan Saputra

## DAFTAR ISI

<b>PREDIKSI KONSUMSI ENERGI MENGGUNAKAN <i>REGRESI LINIER</i> DAN <i>MULTILAYER PERCEPTRON</i> (MLP) PADA ALGORITMA KONSENSUS <i>PROOF OF WORK</i> (POW)</b>	<b>i</b>
<b>LEMBAR PENGESAHAN TUGAS AKHIR</b>	<b>ii</b>
<b>SURAT PERNYATAAN KEASLIAN TUGAS AKHIR</b>	<b>iii</b>
<b>PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH</b>	<b>iv</b>
<b>KATA PENGANTAR</b>	<b>v</b>
<b>DAFTAR ISI</b>	<b>vi</b>
<b>DAFTAR GAMBAR</b>	<b>viii</b>
<b>DAFTAR TABEL</b>	<b>ix</b>
<b>ABSTRAK</b>	<b>x</b>
<b>BAB I PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Pembatasan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	6
<b>BAB II TINJAUAN PUSTAKA DAN DASAR TEORI</b>	<b>7</b>
2.1 Tinjauan Pustaka	7
2.2 Dasar Teori	9
2.2.1 Blockchain dan Algoritma Konsensus	9
2.2.2 <i>Proof of Work</i> (PoW)	10
2.2.3 Penambangan <i>Bitcoin</i> (Mining)	12
2.2.4 <i>Application-Specific Integrated Circuit</i> (ASIC)	12
2.2.5 Deep Learning	14
2.2.6 Regresi Linier	16
2.2.7 Multilayer Perceptron	17
<b>BAB III METODOLOGI PENELITIAN</b>	<b>20</b>
3.1 METODE PENELITIAN	20
3.2 Pengumpulan Data	22
3.3 Preprocessing Data	23

3.4	Pelatihan Model .....	27
3.4.1	Pemodelan <i>Regresi Linier</i> .....	28
3.4.2	Pemodelan <i>Multilayer Perceptron</i> (MLP) .....	32
3.5	Evaluasi Model .....	37
3.6	Visualisasi Model.....	39
3.7	Perancangan User Interface.....	40
3.7.1	Halaman Awal Sistem.....	41
3.7.2	Tampilan Hasil Sistem .....	42
3.8	Deployment.....	43
3.9	Analisa Kebutuhan .....	44
<b>BAB IV HASIL DAN ANALISIS PENELITIAN .....</b>		<b>46</b>
4.1	Hasil penelitian.....	46
4.1.1	Pengumpulan Data .....	46
4.1.2	Preprocessing data.....	47
4.1.3	Pembuatan model <i>Regresi Linier</i> dan <i>Multilayer Perceptron</i> (MLP) 50	
4.1.4	Evaluasi Model .....	60
4.2	Visualisasi Hasil.....	62
4.2.1	Visualisasi hasil model <i>Regresi Linier</i> .....	62
4.2.2	Visualisasi hasil model <i>Multilayer Perceptron</i> .....	63
4.3	Hasil Deployment .....	64
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>70</b>
5.1	Kesimpulan .....	70
5.2	Saran.....	70
<b>DAFTAR PUSTAKA .....</b>		<b>72</b>
<b>LAMPIRAN.....</b>		<b>75</b>



## DAFTAR GAMBAR

Gambar 2 1 Alur algoritma Proof of Work (PoW) .....	11
Gambar 2 2 Bonanza Mine System.....	13
Gambar 2 3 Arsitektur Multilayer perceptron.....	18
Gambar 3. 1 Alur tahapan penelitian .....	20
Gambar 3. 2 Pengumpulan data .....	22
Gambar 3. 3 Flowchart analisis menggunakan regresi linier .....	28
Gambar 3. 4 Alur prediksi menggunakan MLP .....	32
Gambar 3. 5 Flowchart perancangan system .....	40
Gambar 3. 6 Tampilan dashboard .....	41
Gambar 3. 7 Tampilan halaman prediksi .....	42
Gambar 4. 1 Hasil pengecekan NaN .....	48
Gambar 4. 2 Mengecek duplikasi data .....	48
Gambar 4. 3 Distribusi y_train .....	52
Gambar 4. 4 Nilai Intercept dan Koefisien Regresi Linier .....	52
Gambar 4. 5 Prediksi vs aktual gap_twh menggunakan regresi linier .....	53
Gambar 4. 6 Residu plot .....	54
Gambar 4. 7 Grafik training loss .....	58
Gambar 4. 8 Residual Plot .....	59
Gambar 4. 9 Distribusi Residu .....	60
Gambar 4. 10 Hasil grafik prediksi Regresi Linier .....	62
Gambar 4. 11 Hasil grafik prediksi Multilayer Perceptron .....	63
Gambar 4. 12 Halaman awal .....	65
Gambar 4. 13 Hasil dari prediksi .....	66
Gambar 4. 14 Grafik hasil prediksi .....	66
Gambar 4. 15 Hasil Prediksi Gabungan .....	67
Gambar 4. 16 Data dari API .....	69

## DAFTAR TABEL

Tabel 4. 1 Dataset konsumsi energi .....	46
Tabel 4. 2 Handling Missing values.....	47
Tabel 4. 3 Hasil dari feature engineering .....	48
Tabel 4. 4 Hasil dari feature engineering .....	49
Tabel 4. 5 Hasil Split data .....	50
Tabel 4. 6 Evaluasi regresi linier.....	51
Tabel 4. 7 Daftar fitur.....	51
Tabel 4. 10 Hasil evaluasi model Regresi Linier .....	61
Tabel 4. 11 Hasil evaluasi model MLP .....	61



## ABSTRAK

Penelitian ini bertujuan untuk membangun model prediksi konsumsi energi yang akurat pada jaringan blockchain Bitcoin yang menggunakan algoritma konsensus *Proof of Work* (PoW). Permasalahan utama yang diangkat adalah tingginya konsumsi energi PoW yang bersifat dinamis dan non-linier, sehingga sulit diprediksi menggunakan metode linier sederhana. Untuk mengatasi hal ini, penelitian ini menerapkan metode *deep learning* menggunakan *Multilayer Perceptron* (MLP) untuk memprediksi selisih konsumsi energi (*gap\_twh*). Kinerja model MLP kemudian dievaluasi dan dibandingkan dengan model *Regresi Linier* yang digunakan sebagai *baseline*. Data historis dari *Digiconomist* (2017–2025) digunakan setelah melalui tahap *preprocessing* dan *feature engineering*. Hasil evaluasi menunjukkan bahwa model MLP memiliki kinerja yang jauh lebih unggul, dengan nilai MAE 3.4620, MSE 46.9281, dan  $R^2$  0.9390, dibandingkan dengan *Regresi Linier*. Temuan ini membuktikan bahwa MLP sangat efektif dalam memodelkan kompleksitas konsumsi energi PoW dan dapat menjadi acuan bagi para pemangku kepentingan untuk mengoptimalkan efisiensi energi dalam aktivitas penambangan kripto.

**Kata kunci:** Proof of Work, Bitcoin, Konsumsi Energi, Regresi Linier, Multilayer Perceptron

## ABSTRACT

This research aims to develop an accurate energy consumption prediction model for the Bitcoin blockchain network using the *Proof of Work* (PoW) algorithm. The main issue raised is the high energy consumption of PoW, which is dynamic and non-linear, making it difficult to predict using simple linear methods. To address this issue, this research applies a deep learning method using a *Multilayer Perceptron* (MLP) to predict the energy consumption gap (*gap\_twh*). The performance of the MLP model is then evaluated and compared with a *Linear Regression* model used as a *baseline*. Historical data from *Digiconomist* (2017–2025) is used after *preprocessing* and *feature engineering*. The evaluation results show that the MLP model has significantly superior performance, with MAE values of 3.4620, MSE of 46.9281, and  $R^2$  of 0.9390, compared to Linear Regression. These findings demonstrate that the MLP model is highly effective in modeling the complexity of PoW energy consumption and can serve as a reference for stakeholders to optimize energy efficiency in crypto mining activities..

**Keywords:** Proof of Work, Bitcoin, Energy Consumption, Linear Regression, Multilayer Perceptron

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Permasalahan konsumsi energi pada sistem digital modern menjadi isu global yang semakin krusial, terutama dalam konteks teknologi blockchain. Salah satu sumber utama permasalahan ini adalah algoritma konsensus *Proof of Work* (PoW), yang digunakan oleh mata uang kripto besar seperti Bitcoin. PoW mengharuskan penambang menyelesaikan teka-teki komputasi kompleks untuk memvalidasi transaksi, sehingga menuntut daya komputasi yang sangat besar dan berdampak langsung pada konsumsi energi. Konsumsi energi yang masif ini bukan hanya berdampak pada aspek teknis, tetapi juga menimbulkan konsekuensi lingkungan, sosial, dan ekonomi dalam skala global. Teknologi *blockchain* menawarkan revolusi dalam pencatatan transaksi yang aman, transparan, dan terdesentralisasi. Mekanisme ini tidak tergantung pada otoritas pusat dan telah meluas penggunaannya dari sektor keuangan, kesehatan, hingga logistik (Poongodi dkk., 2020). Inti dari keamanannya terletak pada algoritma konsensus, sebuah mekanisme yang memastikan validitas data di seluruh jaringan (Xiong dkk., 2022).

Di antara berbagai algoritma konsensus, salah satunya *Proof of Work* (PoW) merupakan yang paling fundamental dan teruji. Algoritma ini mengharuskan penambang menyelesaikan teka-teki matematika kompleks menggunakan daya komputasi tinggi untuk memvalidasi transaksi dan menambahkan blok baru ke rantai *blockchain* (Afrian dkk., 2024). Mekanisme PoW dikenal sangat tangguh terhadap serangan, karena membutuhkan daya komputasi yang sangat besar untuk memanipulasi data, sebuah konsep yang dikenal sebagai 51% *attack* (Lee & Kim, 2020).

Kekuatan komputasi yang menjadi dasar keamanan algoritma *Proof of Work* (PoW) juga menjadi sumber utama permasalahan terkait keberlanjutan energi. Penelitian oleh (de Vries, 2020) menyoroti bahwa seringkali konsumsi energi Bitcoin tidak diestimasi dengan tepat. Menurut analisisnya, setiap

kenaikan 1 dolar AS pada harga Bitcoin diperkirakan meningkatkan konsumsi energi tahunan jaringan sebesar 5,4 Terawatt-hour (TWh). Dengan mempertimbangkan dinamika pasar, estimasi konsumsi energi tahunan jaringan pada akhir tahun 2020 bisa mencapai 128 TWh, sebuah angka yang setara dengan konsumsi listrik negara Argentina atau Uni Emirat Arab pada tahun yang sama. Angka ini menunjukkan bahwa dampak energi dari setiap fluktuasi harga aset kripto sangatlah signifikan. Dampak konsumsi ini bersifat multidimensi, termasuk emisi karbon yang mencapai lebih dari 90 juta ton per tahun (Sapra dkk., 2023). Serta peningkatan limbah elektronik hingga 44.400 ton per tahun akibat masa pakai perangkat keras penambang yang singkat. Proyeksi jangka panjang menunjukkan potensi dampak lingkungan yang signifikan, termasuk jaminan kesehatan masyarakat akibat polusi udara (Truby dkk., 2022). Secara sosial-ekonomi, konsumsi listrik masif ini telah terbukti memicu krisis energi lokal dan mengancam stabilitas jaringan listrik nasional yang mendorong negara seperti Tiongkok, Argentina, dan Iran telah memberlakukan kebijakan terkait pembatasan bahkan pelarangan total terhadap aktivitas *mining*. Sebagai bentuk tanggapan terhadap krisis energi yang ditimbulkan (Sang dkk., 2022).

Sebagai respons terhadap permasalahan konsumsi energi yang ditimbulkan oleh *Proof of Work* (PoW), industri *blockchain* telah mengembangkan mekanisme alternatif yang lebih efisien seperti *Proof of Stake* (PoS), yang telah diadopsi oleh *Ethereum* dengan pendekatan yang lebih hemat energi. Melalui pembaharuan “*The Merge*” pada tahun 2022 (Tsuyuguchi & Wang, 2024), transisi ini berhasil memangkas konsumsi energi hingga 99,95%. Meskipun demikian, *Proof of Work* (PoW) tetap menjadi mekanisme utama yang digunakan oleh mata uang kripto salah satunya *Bitcoin* dengan kapitalisasi pasar terbesar didunia. Oleh karena itu, mengabaikan dampak PoW bukanlah sebuah pilihan yang tepat dalam konteks ekosistem energi global.

Masalah ini semakin kompleks karena sistem Bitcoin memiliki siklus tetap yang disebut Bitcoin Halving yaitu pengurangan imbalan blok setiap



empat tahun sekali. Dampak langsung dari halving adalah berkurangnya pendapatan para miner hingga 50%, sementara biaya listrik dan investasi perangkat keras cenderung tetap tinggi. Dalam kondisi seperti ini, kalkulasi konsumsi energi dan efisiensi mining menjadi faktor penentu profitabilitas.

Hingga tahun 2025, lebih dari 93% total Bitcoin (dari total suplai 21 juta BTC yang ditetapkan Satoshi Nakamoto) telah ditambang. Artinya, sisa Bitcoin yang bisa ditambang semakin sedikit, sementara persaingan dan kesulitan jaringan terus meningkat. Kondisi ini menciptakan tekanan ekonomi yang nyata bagi para miner dan investor yang tidak memiliki kemampuan analisis energi yang presisi.

Permasalahannya, konsumsi energi jaringan PoW sangat bersifat sangat dinamis dan non-linier, sehingga sulit dipahami dan diprediksi secara akurat. Beberapa penelitian sebelumnya mencoba menggunakan pendekatan sederhana seperti *regresi linier* namun hasilnya kurang memuaskan. Penelitian sebelumnya oleh (Rukhiran dkk., 2024), menyimpulkan *regresi linier* memiliki keterbatasan dalam menangani pola konsumsi energi yang non-linier. Namun, hingga saat ini belum banyak penelitian yang mengembangkan model prediktif berbasis MLP dengan validasi metrik komparatif yang jelas terhadap regresi linier untuk konteks konsumsi energi PoW, khususnya pada jaringan *bitcoin*.

Untuk mengatasi keterbatasan ini, penelitian ini mengusulkan penerapan *MultiLayer Perceptron* (MLP) sebagai penyempurnaan atas pendekatan regresi linier. MLP ini dipilih karena kemampuannya dalam menangkap hubungan non-linier serta memproses data dengan dimensi dan kompleksitas yang tinggi, sehingga diharapkan dapat memberikan hasil prediksi yang lebih presisi dan representatif terhadap realitas fluktuatif konsumsi energi PoW (Gulo & Lubis, 2024).

Metode analisis lama seperti *regresi linier* ternyata belum cukup untuk memahami pola konsumsi energi pada sistem Proof of Work (PoW) yang sering berubah-ubah dan tidak beraturan. Meskipun regresi linier memiliki keterbatasan dalam memodelkan pola non-linier, model ini tetap digunakan

sebagai baseline untuk mengukur peningkatan kinerja yang diberikan oleh model Multilayer Perceptron (MLP) dalam menangani pola non-linier. Di sisi lain, masalah ini semakin sulit karena kebijakan yang ada juga seringkali hanya bersifat reaktif baru bergerak setelah krisis energi muncul. Oleh karena itu, diperlukan model prediksi yang dapat menangkap pola konsumsi energi PoW dengan lebih baik dan lebih akurat. Penelitian ini bertujuan untuk membuat model prediksi konsumsi energi PoW menggunakan dua pendekatan, yaitu *Regresi Linier* dan *Multilayer Perceptron* (MLP). Dengan menggunakan data historis konsumsi energi Bitcoin, diharapkan model ini dapat membantu memberikan gambaran yang lebih jelas tentang tren energi, dan dapat digunakan untuk menyusun strategi penghematan energi yang lebih tepat di masa depan.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah disampaikan, rumusan masalah yang akan dibahas dalam penelitian ini adalah sebagai berikut:

1. Bagaimana karakteristik dan pola kuantitatif konsumsi energi pada jaringan *blockchain* berbasis *Proof of Work* (PoW) berdasarkan analisis data historis?
2. Seberapa akurat model *Regresi Linier* dan *Multilayer Perceptron* (MLP) dalam memprediksi konsumsi energi *Proof of Work* (PoW), dan sejauh mana MLP mampu meningkatkan akurasi prediksi tersebut?
3. Bagaimana hasil prediksi model dapat diinterpretasikan sebagai alat bantu mengambil keputusan bagi investor ritel dan startup green mining untuk mendukung efisiensi energi di sektor mining *blockchain*?

## 1.3 Pembatasan Masalah

1. Penelitian ini hanya terfokus pada algoritma konsensus *Proof of Work* (PoW) tidak mencakup atau membandingkan dengan algoritma lainnya.
2. Data yang digunakan bersumber dari publikasi resmi dan *dataset* terbuka dari Digiconomist dengan cakupan data historis dari periode 2017-2024

serta perkiraan tahun 2025 berdasarkan proyeksi dataset terbuka, yang mencakup data waktu operasional, perkiraan total jumlah konsumsi energi per-tahun dan jumlah minimum penggunaan energi.

3. Metode pemodelan yang digunakan terdiri dari dua tahap. *Regresi Linier* sebagai model awal prediksi, kemudian disempurnakan dengan penggunaan *Artificial Neural Network* (ANN) yaitu *Multilayer Perceptron* (MLP).
4. Evaluasi performa model dibatasi pada aspek akurasi prediksi menggunakan metrik statistik seperti *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), dan *R-squared* ( $R^2$ ).

#### 1.4 Tujuan Penelitian

1. Membangun model prediksi konsumsi energi algoritma PoW menggunakan *Regresi Linier* dan *Multilayer Perceptron* (MLP).
2. Membandingkan tingkat akurasi kedua model untuk menunjukkan efektivitas pendekatan non-linier.
3. Mengembangkan aplikasi berbasis web untuk membantu investor dan pelaku mining menganalisis tren konsumsi energi serta memperkirakan potensi risiko kerugian menjelang periode *halving*.

#### 1.5 Manfaat Penelitian

1. Penelitian ini memberikan kontribusi terhadap pengembangan model prediksi konsumsi energi berbasis *Artificial neural Network* (ANN) pada sistem blockchain dengan konteks *Proof of Work* (PoW). Model ini dapat menjadi referensi dalam studi pemodelan sistem non-linier dan fluktuatif di bidang teknologi informasi dan energi digital.
2. Menjadi alat bantu pengambilan keputusan bagi pelaku industri kripto untuk mengoptimalkan waktu dan biaya mining, serta menekan risiko kerugian akibat fluktuasi pasokan energi dan perubahan reward Bitcoin

## 1.6 Sistematika Penulisan

### BAB I : PENDAHULUAN

Pada BAB I menjelaskan tentang latar belakang, pemilihan judul, rumusan masalah, Batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penelitian.

### BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Pada Bab II ini memuat tinjauan penelitian-penelitian sebelumnya yang relevan serta landasan teori yang digunakan, mencakup konsep *blockchain*, algoritma *Proof of Work*, Regresi Linier, dan *Multilayer Perceptron* (MLP).

### BAB III : METODE PENELITIAN

Pada BAB III menjelaskan proses penelitian yang dimulai dari pengumpulan data hingga evaluasi hasil prediksi.

### BAB IV : HASIL DAN ANALISIS PENELITIAN

Pada BAB IV berisi tentang pemaparan hasil penelitian yang dimulai dari pembuatan sistem sampai dengan proses deployment.

### BAB V : KESIMPULAN DAN SARAN

Pada BAB V merangkum keseluruhan proses penelitian dari awal sampai akhir.



## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Dalam beberapa tahun terakhir, cryptocurrency dengan algoritma konsensus menjadi salah satu fokus utama penelitian di bidang blockchain, mengingat perannya yang krusial dalam menjamin integritas, keamanan, dan keberlangsungan jaringan terdistribusi. Kajian pustaka ini bertujuan untuk menggali dan memetakan berbagai pendekatan yang telah digunakan sebelumnya dalam mengevaluasi algoritma konsensus yaitu *Proof of Work* (PoW), dari perspektif efisiensi energi, dampak ekosistem lingkungan, serta potensi penerapan pendekatan *Artificial Intelligence* dalam pengembangan model prediksi yang lebih adaptif dan akurat .

Studi oleh (Sapra dkk., 2023) mengkaji tingkat konsumsi energi *bitcoin* dan menemukan bahwa total energi yang dikonsumsi oleh proses *mining* tahunan setara dengan konsumsi energi nasional beberapa negara maju. Sementara itu, (Truby dkk., 2022) menekankan bahwa emisi karbon dari proses mining berbasis PoW berpotensi menyebabkan kerugian lingkungan jangka panjang, termasuk peningkatan risiko kesehatan dan degradasi lingkungan.

Studi yang dilakukan oleh (de Vries, 2020) menunjukkan adanya korelasi kuat antara lonjakan harga aset seperti *bitcoin* dengan peningkatan *hash rate* jaringan, yang secara langsung mendorong konsumsi energi tingkat yang semakin ekstrem. Ini menciptakan siklus yang membuat kebutuhan energi PoW terus bertumbuh secara eksponensial, menjadikannya sebuah masalah yang semakin mendesak setiap harinya.

Berbagai upaya pemodelan prediktif konsumsi energi yang ada saat ini secara konsisten menyoroti keterbatasan fundamental dari pendekatan linier dalam kasus konsumsi energi PoW, studi oleh (Rukhiran dkk., 2024) memang berhasil mengidentifikasi hubungan signifikan antara parameter teknis dan penggunaan energi, namun model *regresi linier* yang dilakukan masih dirasa



kurang untuk menangkap variasi tren data yang lebih kompleks, meskipun berhasil mengidentifikasi hubungan antar parameter teknis. Keterbatasan ini dipertegas lebih lanjut oleh (Iqbal dkk., 2023), yang melakukan perbandingan tingkat konsumsi energi antara algoritma *Proof of Work* (PoW) dan *Proof of Stake* (PoS) dengan membangun model prediksi menggunakan regresi linier. Hasil penelitian menunjukkan bahwa meskipun PoS mampu menurunkan konsumsi energi secara signifikan. Akan tetapi, model regresi linier yang digunakan kurang akurat dalam memodelkan data historis yang bersifat non-linier dan dinamis, kedua penelitian ini mengonfirmasi bahwa metode linier tidak cukup andal, sehingga membuka jalan bagi eksplorasi metode yang lebih canggih.

Keterbatasan model linier mengarahkan fokus pada pendekatan *machine learning* model *Multilayer Perceptron* (MLP) sebagai solusi untuk data kompleks, telah ditegaskan dalam berbagai survei. Survei yang dilakukan oleh (Kayikci & Khoshgoftaar, 2024) menegaskan bahwa *Multilayer Perceptron* (MLP) dan model *deep learning* lainnya memiliki potensi besar dalam menangani data numerik dan prediksi kompleks, termasuk konsumsi energi. Meski begitu, mayoritas penerapan *Multilayer Perceptron* (MLP) dalam blockchain masih difokuskan pada prediksi harga atau klasifikasi transaksi, bukan pada isu energi. Keunggulan teknis MLP dalam menangani data yang dinamis juga telah dibuktikan oleh (Hisham dkk., 2022) , yang menunjukkan bahwa MLP mampu mendeteksi anomali maupun klasifikasi dalam jaringan *blockchain* yang fluktuatif dan non-linier dibandingkan metode yang lain seperti *Random Forest* dan *Regresi Logistik*. Kemampuan fundamental MLP dalam memodelkan pola non-linier yang kompleks ini bersifat general dan tidak terbatas pada satu domain. Temuan ini sejalan dengan studi yang dilakukan oleh (Gulo & Lubis, 2024), yang menunjukkan bahwa MLP sangat efektif dalam klasifikasi sosial dengan akurasi mencapai 96,9%. Hal ini menguatkan bahwa kapasitas MLP tidak hanya unggul dalam menangani data non-linier tetapi juga fleksibel untuk berbagai jenis data yang serupa..

Meskipun keterbatasan model linier telah diidentifikasi dan potensi MLP telah diakui secara luas, penerapan MLP secara spesifik untuk memprediksi konsumsi energi PoW dengan memanfaatkan *lag-features* dan komponen waktu sebagai prediktor utama masih belum banyak dieksplorasi. Oleh karena itu penelitian ini bertujuan untuk mengisi celah tersebut dengan mengembangkan dan mengevaluasi model MLP yang adaptif untuk kasus ini.

## 2.2 Dasar Teori

### 2.2.1 Blockchain dan Algoritma Konsensus

*Blockchain* merupakan teknologi inovatif yang mendasari berbagai sistem terdesentralisasi modern, khususnya dalam bidang keuangan digital seperti *cryptocurrency* (Nanda Sari & Gelar, 2024). Teknologi ini berfungsi sebagai buku besar digital (*digital ledger*) yang mencatat setiap transaksi dalam bentuk blok-blok data yang saling terhubung (Paul, 2021). Setiap blok memiliki informasi *hash* unik dari blok sebelumnya, menciptakan rantai yang tidak dimodifikasi tanpa mempengaruhi seluruh struktur. Sifat inilah yang menjadikan *blockchain* sebagai sistem yang tahan terhadap manipulasi data (Sarwar dkk., 2023).

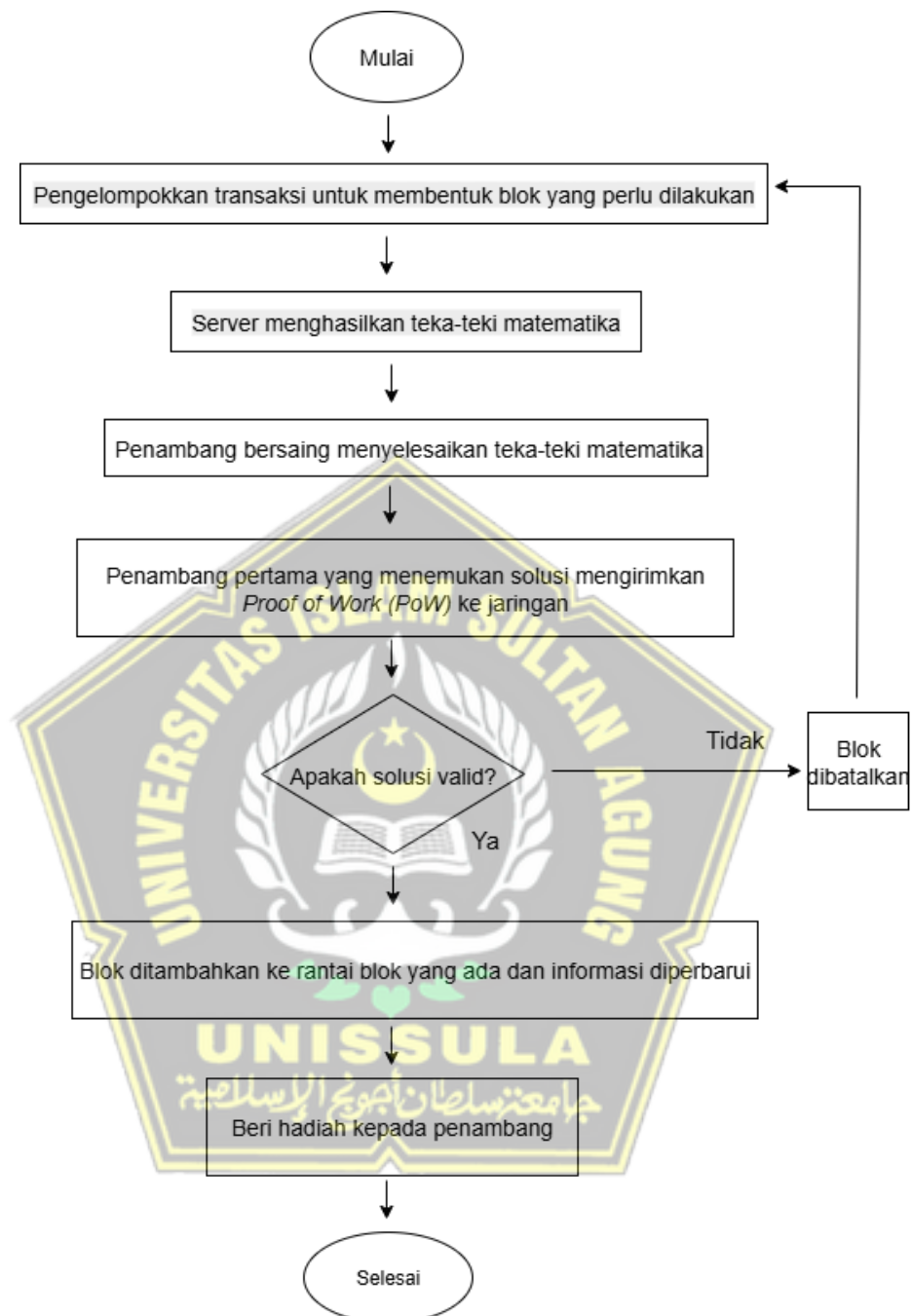
Salah satu karakteristik utama dari *blockchain* adalah desentralisasinya. Tidak seperti sistem konvensional yang tergantung pada otoritas pusat seperti bank atau institusi pemerintah, *blockchain* memungkinkan semua partisipan jaringan (*nodes*) memiliki salinan data yang sama dan turut serta dalam proses validasi transaksi. Untuk menyepakati transaksi mana yang valid dan dapat ditambahkan ke dalam rantai blok, diperlukan mekanisme yang disebut algoritma konsensus (Afrian dkk., 2024).

Algoritma konsensus adalah protokol yang memastikan seluruh *node* dalam jaringan mencapai kesepakatan atau konsensus mengenai status data. Tanpa adanya otoritas pusat, algoritma ini menjadi fondasi dalam menjaga integritas, konsistensi, dan kepercayaan antar pengguna jaringan *blockchain* (Afrian dkk., 2024). Beberapa algoritma konsensus yang populer di antaranya adalah *Proof of work* (PoW), *Proof of Stake* (PoS), *Delegated Proof of Stake*

(DPoS), dan *Practical Byzantine Fault Tolerance* (PBFT), meskipun terdapat beberapa algoritma konsensus yang cukup populer, penelitian ini difokuskan pada algoritma *Proof of Work* (PoW) mengingat dampaknya yang cukup signifikan terhadap konsumsi energi dan relevansinya pada jaringan *blockchain* utama.

### 2.2.2 *Proof of Work* (PoW)

*Proof of Work* adalah algoritma konsensus pertama yang digunakan secara luas melalui implementasinya dalam bitcoin oleh Satoshi Nakamoto pada tahun 2008 (Pavlov dkk., 2024). Dalam sistem ini, validasi blok dilakukan oleh penambang (*miners*) yang harus menyelesaikan teka-teki kriptografi kompleks. Proses ini membutuhkan daya komputasi yang tinggi karena teka-teki tersebut didesain agar sulit diselesaikan namun mudah diverifikasi (Maulana, 2024). Namun, PoW memiliki kelemahan signifikan dari segi efisiensi energi, konsumsi energi yang dibutuhkan sangat besar karena jutaan perangkat di seluruh dunia bersaing untuk menemukan solusi secara simultan. Selain itu, PoW juga memerlukan perangkat keras khusus (*ASIC miner*) yang mahal, dan rentan terhadap sentralisasi karena hanya entitas besar yang mampu bersaing dalam *mining*.



Gambar 2 1 Alur algoritma *Proof of Work* (PoW)

Untuk sistem kerja algoritma konsensus *Proof of Work* (PoW) dimulai dengan pengumpulan seluruh transaksi yang belum terkonfirmasi (Yadav & Singh, 2021), yang kemudian di kelompokkan menjadi satu blok. Setelah blok terbentuk, sistem secara otomatis menghasilkan sebuah teka-teki kriptografi yang kompleks, yang membutuhkan proses pencarian nilai *nonce* yaitu angka acak yang harus dicari sedemikian rupa sehingga menghasilkan nilai *hash*

blok yang sesuai dengan target jaringan. Setelah itu, setiap *node* atau *miners* akan saling berlomba untuk menemukan nilai *nonce* yang valid. Setelah salah satu *node* berhasil menyelesaikan teka-teki tersebut, solusi tersebut akan dikirimkan keseluruhan jaringan. Kemudian *node-node* yang lain dalam jaringan akan melakukan proses verifikasi untuk memastikan bahwa solusi yang dikirim benar-benar valid. Jika solusi dinyatakan valid oleh *node*, maka blok tersebut akan ditambahkan ke dalam rantai *blockchain* sebagai data transaksi secara permanen. Namun, jika kemungkinan solusi tidak valid maka blok akan membatalkan proses transaksi.

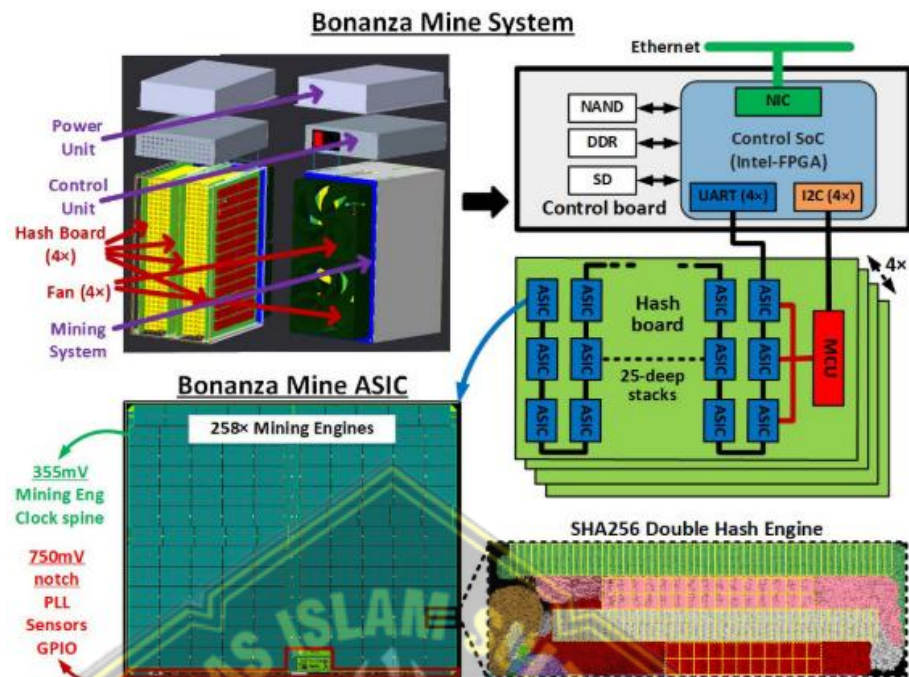
### 2.2.3 Penambangan *Bitcoin* (Mining)

Penambangan *Bitcoin* (Mining) adalah proses kompetitif di mana para penambang (*miners*) menggunakan perangkat keras khusus untuk memecahkan teka-teki matematika kompleks (Zahedi dkk., 2024). Tujuannya untuk memverifikasi transaksi dan menambahkannya sebagai blok baru ke dalam *blockchain*. Penambang pertama yang berhasil menemukan solusi *hash* valid sesuai kesulitan jaringan (*difficulty*) akan mendapatkan hak untuk menambahkan blok dan menerima rewards berupa bitcoin serta *transaction fees*. *Transaction fees* adalah sejumlah kecil imbalan yang diberikan oleh pengguna kepada *miner* agar transaksi mereka diproses dan dimasukkan ke dalam blok baru di *blockchain*.

### 2.2.4 *Application-Specific Integrated Circuit* (ASIC)

*Application-Specific Integrated Circuit* (ASIC) merupakan sirkuit terpadu yang dirancang secara khusus untuk menjalankan satu fungsi komputasi secara optimal. Berbeda dengan prosesor serbaguna seperti CPU dan GPU, ASIC memfokuskan seluruh arsitektur chip untuk satu tugas spesifik, sehingga mampu mencapai efisiensi komputasi dan konsumsi energi yang sangat tinggi. Dalam konteks *blockchain* yang menggunakan algoritma konsensus *Proof of Work* (PoW), ASIC berperan sebagai perangkat keras utama dalam proses penambangan, khususnya untuk menjalankan algoritma *hashing* seperti SHA-256 pada jaringan *Bitcoin*.





Gambar 2.2 Bonanza Mine System

Pada gambar 2.2 Bonanza Mine System memberikan ilustrasi menyeluruh mengenai bagaimana ASIC diimplementasikan dalam sistem penambangan modern. Sistem ini terdiri dari beberapa komponen utama yang saling terintegrasi, yaitu power unit, *control board*, *hash board*, dan fan pendingin. Control board dilengkapi dengan NIC, DDR, NAND, dan SoC berbasis Intel-FPGA yang mengatur distribusi data dan komunikasi antar komponen. Hash board berisi 25 stack ASIC yang masing-masing terhubung ke microcontroller unit (MCU), berfungsi sebagai pusat komputasi untuk proses *hashing* (Yaish & Zohar, 2023).

Pada bagian inti sistem, ditampilkan ASIC Bonanza yang memiliki 258 *mining engines* yang beroperasi pada tegangan rendah 355mV dan frekuensi 750MHz. Efisiensi ini memungkinkan perangkat untuk menghasilkan *hash rate* tinggi dengan konsumsi daya yang relatif rendah. Di dalam ASIC, terdapat SHA256 Double Hash Engine yang menjadi pusat eksekusi algoritma PoW, di mana proses *hashing* dilakukan secara paralel dan berulang untuk menemukan nilai *nonce* yang valid.

Keunggulan ASIC dalam sistem ini terletak pada kemampuannya untuk mengoptimalkan performa penambangan sekaligus menekan konsumsi

energi. Namun, spesialisasi ini juga menimbulkan tantangan, seperti biaya produksi yang tinggi, siklus hidup produk yang pendek, dan potensi peningkatan limbah elektronik. Selain itu, dominasi entitas besar dalam penguasaan perangkat ASIC dapat memicu sentralisasi kekuatan komputasi, yang bertentangan dengan prinsip desentralisasi dalam *blockchain*.

Dengan demikian, gambar Bonanza Mine System tidak hanya memperlihatkan struktur fisik dari sistem penambangan berbasis ASIC, tetapi juga merepresentasikan dinamika teknologi yang menjadi fondasi sekaligus tantangan dalam ekosistem *blockchain* modern. Arsitektur ini menjadi bukti nyata bahwa efisiensi komputasi dan desain perangkat keras memainkan peran krusial dalam keberlanjutan dan keamanan jaringan PoW.

#### 2.2.5 Deep Learning

*Deep Learning* merupakan sub-bidang dari *machine learning* yang berfokus pada penggunaan *Artificial Neural Network* (ANN) dengan banyak lapisan, yang dikenal sebagai deep neural network (Alzubaidi dkk., 2021). Pendekatan ini terinspirasi dari cara kerja otak manusia dalam memproses informasi, di mana setiap neuron buatan menerima input, melakukan transformasi melalui bobot dan bias, lalu meneruskan hasilnya ke neuron berikutnya. Proses ini berlangsung secara berlapis, memungkinkan jaringan untuk membentuk representasi data yang semakin abstrak dan kompleks di setiap levelnya.

Keunggulan utama dari *deep learning* terletak pada kemampuannya untuk mempelajari representasi fitur secara otomatis, tanpa memerlukan proses rekayasa fitur manual yang ekstensif. Hal ini menjadikannya sangat efektif dalam menangani data berukuran besar dan berdimensi tinggi, seperti citra digital, sinyal suara, teks alami, maupun data deret waktu (time series). Dalam konteks penelitian ini, pendekatan deep learning digunakan untuk memodelkan konsumsi energi *blockchain* yang bersifat dinamis dan non-linier (Kaur dkk., 2024). Arsitektur Umum *Deep Learning* secara umum, arsitektur deep learning terdiri dari tiga komponen utama:

### 1. Input Layer

Lapisan pertama yang menerima data mentah atau hasil preprocessing. Data ini dapat berupa nilai numerik, vektor fitur, atau representasi temporal yang telah dinormalisasi.

### 2. Hidden Layers

Terdiri dari beberapa lapisan tersembunyi yang memproses informasi melalui operasi matematis berbasis bobot dan bias. Setiap neuron dalam lapisan ini menerapkan fungsi aktivasi non-linear seperti *ReLU (Rectified Linear Unit)*, sigmoid, atau tanh, yang memungkinkan jaringan menangkap pola kompleks dan non-linier dalam data.

### 3. Output Layer

Lapisan terakhir yang menghasilkan *output* sesuai dengan tujuan model, baik berupa nilai kontinu (regresi) maupun kelas diskrit (klasifikasi).

Selama proses pelatihan, *Artificial neural Network* (ANN) melakukan forward propagation untuk menghasilkan prediksi, kemudian menghitung *loss function* sebagai ukuran kesalahan antara prediksi dan nilai aktual. Berdasarkan nilai loss tersebut, dilakukan *backpropagation* untuk memperbarui bobot dan bias menggunakan algoritma optimisasi seperti, *Stochastic Gradient Descent* (SGD) mengupdate parameter secara bertahap berdasarkan subset data (mini-batch), cocok untuk dataset besar dan adam *optimizer* merupakan algoritma adaptif yang menggabungkan momentum dan pembelajaran berbasis rata-rata kuadrat gradien, sehingga lebih cepat konvergen dan stabil (Reyad dkk., 2023).

Dengan iterasi yang berulang, jaringan belajar untuk meminimalkan kesalahan dan meningkatkan akurasi prediksi. Proses ini memungkinkan model *deep learning* untuk membentuk pemahaman yang mendalam terhadap struktur data, bahkan tanpa eksplisit diberi aturan atau pola.

### 2.2.6 Regresi Linier

*Regresi linier* adalah metode statistik yang digunakan untuk memodelkan hubungan antara satu variabel dependen (Y) dengan satu atau beberapa variabel independen (X). Secara konseptual, metode ini bekerja dengan menemukan garis paling pas (*best-fit line*) yang dapat merepresentasikan sebaran data antara variabel-variabel tersebut. Ketika hubungan yang analisis hanya melibatkan satu variabel independen, metode ini dikenal sebagai *Regresi Linier* sederhana, model matematisnya yang merepresentasikan sebuah garis lurus (Rachmawati & Aini, 2018), dirumuskan sebagai berikut:

$$Y = \beta_0 + \beta_1 X + \varepsilon \quad (1)$$

- Y = Nilai variabel dependen (variabel hasil prediksi)
- $\beta_0$  = konstanta regresi, yaitu nilai Y saat semua  $X_i = 0$
- $\beta_1$  = Koefisien regresi untuk masing – masing variabel independen
- X = Variabel independen (prediktor)
- $\varepsilon$  = Error term, mewakili factor-faktor lain yang tidak dimasukkan dalam model tetapi mempengaruhi Y

Namun, dalam banyak kasus nyata, sebuah variabel dependen dipengaruhi oleh lebih dari satu faktor. Untuk mengakomodasi hal ini, digunakan regresi linier berganda, yang merupakan perluasan dari model sederhana untuk melibatkan dua atau lebih variabel independen. Oleh karena penelitian ini menguji lebih dari satu variabel independen yaitu hash rate, difficulty\_est dan ordinal\_time, maka model yang relevan adalah regresi linier berganda dengan rumus umum sebagai berikut :

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon \quad (2)$$

- Y = Nilai variabel dependen (variabel hasil prediksi)
- $\beta_0$  = konstanta regresi, yaitu nilai Y saat semua  $X_i = 0$
- $\beta_1, \beta_2, \dots, \beta_n$  = Koefisien regresi untuk masing – masing variabel independen
- $X_1, X_2, \dots, X_n$  = Variabel independen (prediktor)



- $\varepsilon$  = Error term, mewakili factor-faktor lain yang tidak dimasukkan dalam model tetapi mempengaruhi Y.

Model *regresi linier* ini bekerja dengan prinsip *Ordinary Least Squares* (OLS), yakni meminimalkan jumlah kuadrat dari selisih antara nilai aktual dan nilai yang diprediksi. Dalam *regresi linier*, asumsi-asumsi penting seperti *linearitas* hubungan antara variabel, normalitas eror, dan varian eror konstanta, untuk penelitian ini menggunakan *regresi linier* berganda (*Multiple Liniear Regression*) dikarenakan terdapat lebih dari satu variabel independen yang akan diuji, yaitu Hash Rate dan Jumlah Transaksi. Regresi linier digunakan sebagai model dasar (*baseline model*) untuk melihat pola hubungan antara faktor-faktor teknis dan konsumsi energi dari dua algoritma konsensus.

Penggunaan *regresi linier* juga berfungsi sebagai tahap awal untuk melakukan analisis eksploratif terhadap perilaku data historis energi *blockchain*, di mana hasil dari analisisnya digunakan untuk proses selanjutnya penggunaan MLP sebagai proses prediksi. Dimana *Artificial neural Network* (ANN) mampu menangkap relasi non-linier dalam data dengan kompleksitas tinggi. Dengan hasil prediksi regresi linier yang selanjutnya di sempunakan menggunakan MLP, diharapkan diperoleh gambaran yang lebih jelas tentang sejauh mana model berbasis machine learning mampu meningkatkan akurasi prediksi konsumsi energi jaringan *blockchain*.

### 2.2.7 Multilayer Perceptron

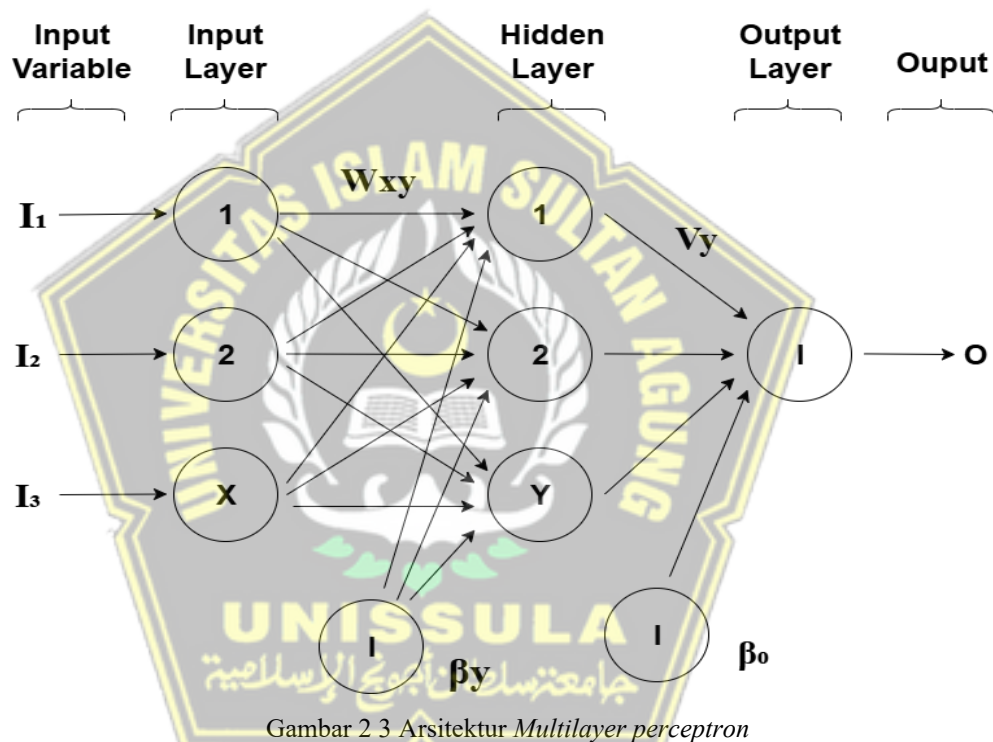
*Multilayer Perceptron* (MLP) merupakan salah satu arsitektur paling dasar dari *Artificial Neural Network* (ANN) yang banyak digunakan dalam tugas prediksi, klasifikasi, serta pemodelan hubungan *non-linier* dalam suatu data (Pawar dkk., 2025). Kemampuannya untuk mempelajari pola yang kompleks menjadikannya pilihan yang kuat sebagai model penyempurna setelah analisis awal dengan regresi linier. *Multilayer perceptron* (MLP) sendiri terdiri dari tiga jenis lapisan utama, yaitu:

- Input layer : Menerima data masukan awal jumlah neuron di lapisan ini sama dengan jumlah fitur pada data.



- Hidden layer : Terdiri dari satu atau lebih lapisan neuron yang berada di antara *input* dan *output layer*. Disinilah proses komputasi utama terjadi. Di mana data ditransformasi melalui bobot, bias, dan fungsi aktivasi untuk mempelajari pola non-linier.
- Output layer : Menghasilkan nilai prediksi akhir. Jumlah *neuron* di lapisan ini bergantung pada tugas prediksi.

Untuk setiap *neuron* (node) dalam suatu lapisan terhubung dengan semua *neuron* di lapisan berikutnya, membentuk fully connected layer.



Gambar 2.3 Arsitektur *Multilayer perceptron*

Dapat dilihat pada gambar 2.3 yang menunjukkan bahwa informasi dari *dataset*, alur prediksi dalam model ini dimulai dengan serangkaian masukan (input), yang terdiri dari beberapa fitur data seperti  $I_1$ ,  $I_2$ , hingga  $I_3$ , atau secara umum dinotasikan sebagai  $I_x$ . Setiap input  $I_x$  ini kemudian dihitung nilainya menggunakan bobot  $W_{xy}$  yang menghubungkannya ke setiap neuron di lapisan tersembunyi yang diindeks oleh  $y$ . Di dalam setiap neuron tersembunyi  $y$  tersebut, ditambahkan sebuah nilai bias  $\beta_y$  untuk penyesuaian. Selanjutnya, sinyal yang sudah diolah dari semua neuron tersembunyi dikumpulkan dan dihubungkan ke lapisan output menggunakan set bobot kedua, yaitu  $V_y$ . Pada tahap akhir ini, ditambahkan lagi sebuah bias output  $\beta_0$

sebelum jaringan menghasilkan satu nilai keluaran (output) akhir yang dinotasikan sebagai  $O$ . Secara matematis, model MLP dapat dinyatakan sebagai berikut:

$$O = AF_1 (\beta_0 + \sum_{y=1}^m V_y [\sum_{x=1}^n AF_2 (W_{xy} I_x + \beta_y)]) \quad (3)$$

- $O$  = Nilai output atau hasil prediksi akhir dari jaringan.
- $AF_1$  = Fungsi aktivasi untuk lapisan output (output layer).
- $AF_2$  = Fungsi aktivasi untuk lapisan tersembunyi (hidden layer)
- $I_x$  = Nilai dari fitur input ke- $x$  (*hash rate, difficulty*).
- $W_{xy}$  = *Weight* yang menghubungkan input ke- $x$  dengan *neuron* tersembunyi ke- $y$
- $\beta_y$  = Bias untuk *neuron* tersembunyi ke- $y$
- $V_y$  = *weight* yang menghubungkan *neuron* tersembunyi ke- $y$  dengan *neuron* output
- $\beta_0$  = Bias untuk *neuron* output.
- $n$  = Jumlah total fitur input.
- $m$  = Jumlah total *neuron* di lapisan tersembunyi.

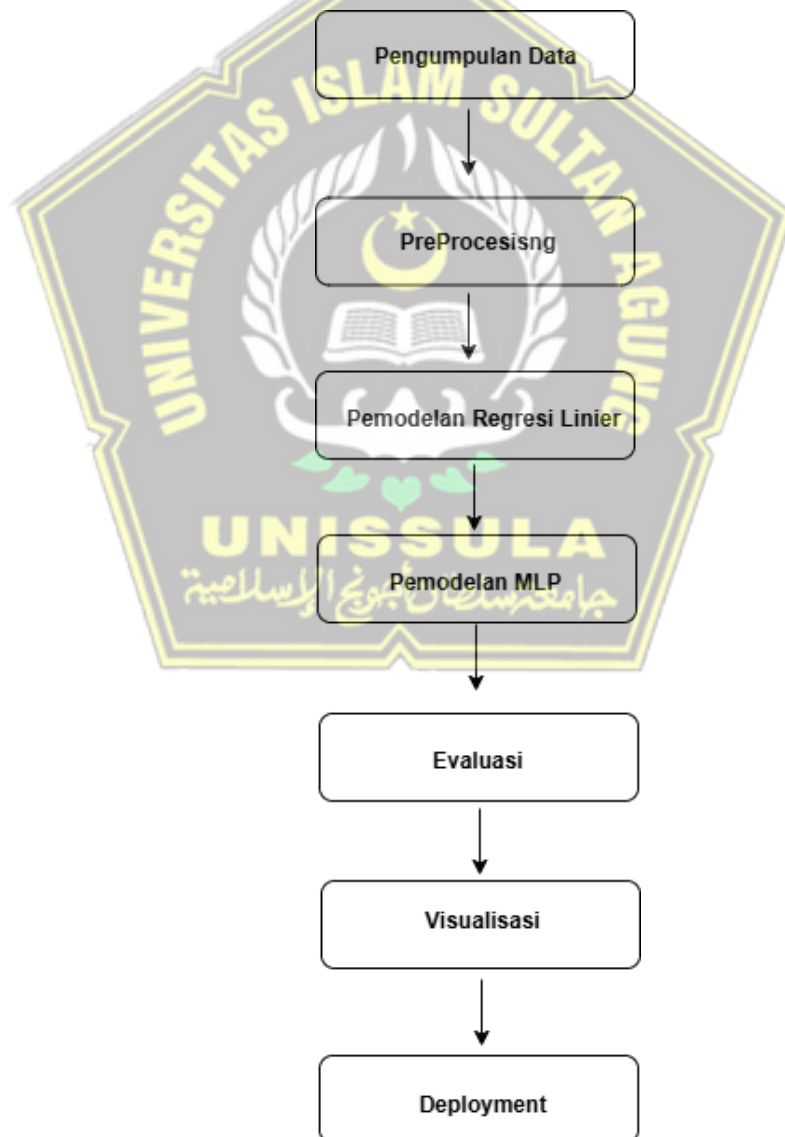
Dalam rumus tersebut, variabel keluaran dan masukan diwakili oleh  $O$  dan  $I_x$ , nilai bias untuk neuron tersembunyi dan untuk neuron keluaran diwakili oleh  $\beta_y$  dan  $\beta_0$ ; fungsi aktivasi diwakili oleh  $AF_1$  dan  $AF_2$  bobot antara masukan  $x$  dan neuron tersembunyi  $y$  diwakili oleh  $W_{xy}$  dan, bobot antara neuron tersembunyi dan neuron keluaran diwakili oleh  $V_y$ .

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 METODE PENELITIAN

Pada tahap ini menguraikan arsitektur dan alur kerja keseluruhan penelitian yang akan dilakukan, mulai dari pengumpulan data hingga evaluasi hasil prediksi. Perancangan sistem ini dirancang untuk memastikan setiap tahapan penelitian dilakukan secara sistematis dan efisien, menghasilkan analisis tren konsumsi energi yang akurat.

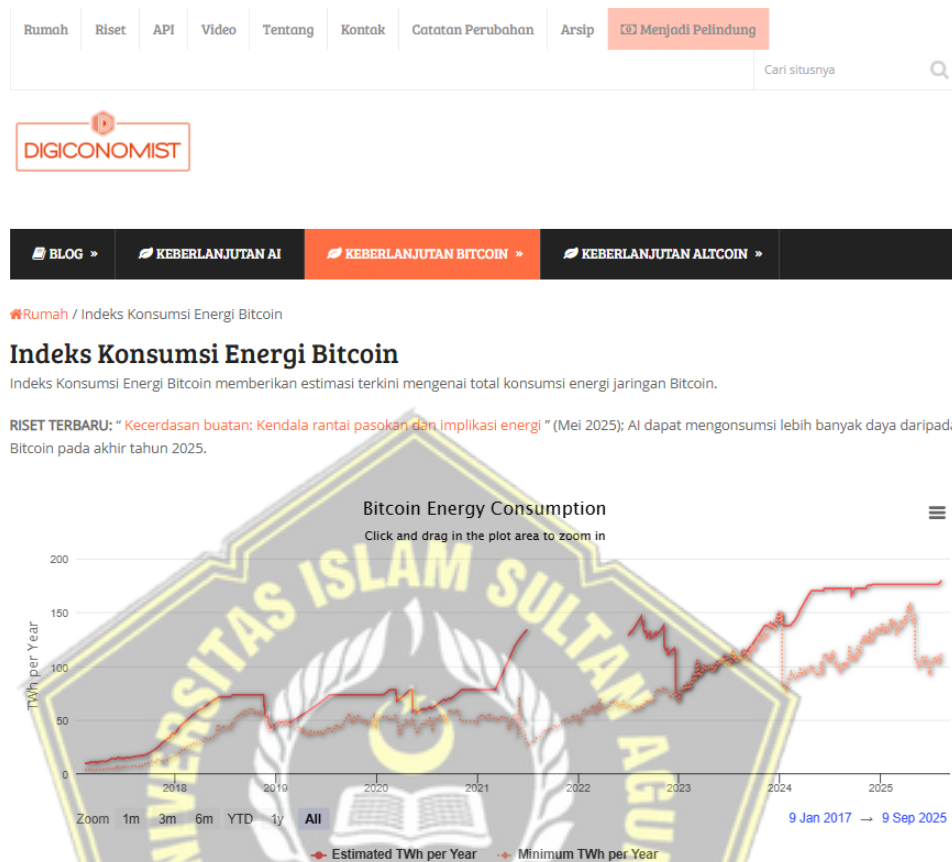


Gambar 3. 1 Alur tahapan penelitian

Pada gambar 3.1 merupakan langkah-langkah yang akan dilakukan secara komprehensif dengan tahapan awal mengambil dataset konsumsi energi Bitcoin dari sumber *Digiconomist*, yang memuat informasi datetime, estimasi konsumsi energi tahunan (TWh), dan konsumsi minimum tahunan. Dataset ini digunakan sebagai dasar analisis tren dan pelatihan model prediksi. Proses selanjutnya *Preprocessing*, yang mencakup cleaning data, mengubah nilai string ke float, menangani *missing values* untuk memastikan kualitas data. Proses ini dilanjutkan dengan *feature engineering* yaitu mengubah variabel dari data mentah agar lebih informatif dan relevan untuk model prediksi dengan ekstraksi fitur waktu dari kolom DateTime untuk membantu model mengenali pola musim serta tren jangka panjang, serta dibuat *lag features* yang bertujuan mempresentasikan nilai konsumsi energi pada hari-hari sebelumnya. Setelah data siap, dilakukan pemodelan *regresi linier* untuk mengidentifikasi tren konsumsi energi dari waktu ke waktu serta memahami pola dasar pertumbuhan energi pada jaringan blockchain *Proof of Work* (PoW). Hasil dari proses analisis regresi linier digunakan sebagai dasar untuk membangun model prediksi berbasis *Artificial Neural Network* (ANN), yaitu *Multilayer Perceptron* (MLP). Model ini dirancang untuk menangkap pola non-linier dan fluktuatif dalam data.

Model dievaluasi menggunakan metrik seperti *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE), dan *R-squared* ( $R^2$ ), untuk mengukur akurasi dan performa prediksi. Selanjutnya, dilakukan visualisasi hasil prediksi dan residual guna mempermudah interpretasi dan analisis performa model. Tahapan akhir adalah *deployment*, yaitu penerapan model dalam antarmuka interaktif berbasis Streamlit agar pengguna dapat melihat prediksi konsumsi energi secara real-time dan eksploratif.

### 3.2 Pengumpulan Data



Gambar 3. 2 Halaman beranda sumber data historis Digiconomist

Pada gambar 3.2 merupakan tahap awal pengumpulan data yang diperlukan untuk penelitian dengan menggunakan *dataset* yang berasal dari sumber terbuka. Dataset yang digunakan diambil dari platform *Digiconomist* atau bisa di akses melalui link berikut <https://digiconomist.net/>, yang secara rutin menyediakan estimasi konsumsi energi jaringan *blockchain*. Dataset tersebut kemudian diunduh dan disimpan secara lokal untuk kebutuhan pengolahan data dan pemodelan.

Dataset yang digunakan secara khusus berfokus pada jaringan Bitcoin dan memuat tiga kolom utama, yaitu *DateTime*, *Estimated TWh per Year*, dan *Minimum TWh per Year*. Data ini dipilih karena menyajikan informasi konsumsi energi tahunan yang relevan, lengkap, dan terstruktur, sehingga dapat digunakan untuk menganalisis efisiensi algoritma konsensus *Proof of Work* (PoW). Selain itu, *dataset* ini juga memiliki keunggulan karena mencakup data historis yang cukup panjang, mulai dari tahun 2017 hingga



2025, yang sangat berguna untuk membangun model prediksi berbasis tren jangka panjang.

Nilai Estimated TWh per Year dan Minimum TWh per Year pada dataset ini nantinya akan digunakan untuk menghitung selisih konsumsi energi (*gap\_twh*) sebagai variabel target prediksi. Sementara itu, kolom DateTime akan diolah menjadi berbagai fitur berbasis waktu seperti tahun, bulan, dan *ordinal time* untuk membantu model mengenali pola musiman. Data yang telah dikumpulkan ini menjadi fondasi utama dalam proses analisis dan pelatihan model, baik untuk metode *Regresi Linier* sebagai *baseline* maupun *Multilayer Perceptron* (MLP) untuk prediksi dengan pendekatan non-linear.

### 3.3 Preprocessing Data

Tahap *preprocessing* dilakukan untuk mempersiapkan dataset sebelum masuk ke proses pelatihan model, sehingga data yang digunakan bersih, konsisten, dan siap diolah. Data preprocessing meliputi tahapan sebagai berikut:

#### 3.3.1 Data Cleaning

*Data cleaning* adalah proses mengidentifikasi dan memperbaiki kesalahan atau inkonsistensi dalam dataset agar siap digunakan untuk analisis atau pemodelan. Ini adalah tahap krusial dalam preprocessing karena kualitas data sangat memengaruhi akurasi model.

##### 1. Penanganan nilai yang hilang (*Missing Value*)

Proses ini diawali dengan pemeriksaan terhadap keberadaan data duplikat serta nilai kosong (*missing values*) pada setiap kolom. Hasil pemeriksaan menunjukkan adanya data kosong pada kolom *Estimated TWh per Year* 365 row yang terbagi menjadi 2, untuk tahun 2021 terdapat 184 row dan 181 row pada tahun 2022. Nilai yang bernilai NaN ini diisi menggunakan metode *interpolasi linear* agar tren konsumsi energi tetap konsisten dan tidak terputus. Dengan menggunakan rumus berikut:

$$y = y_0 + (x - x_0) \times \frac{y_1 - y_0}{x_1 - x_0} \quad (4)$$

- $y$  = Nilai hasil dari interpolasi pada titik  $x$  yang dicari
- $x$  = Nilai variabel independen
- $x_0$  = Nilai variabel independen pada titik data pertama
- $y_0$  = Nilai variabel dependen pada titik pertama
- $y_1$  = Nilai variabel dependen pada titik data kedua
- $x_1$  = Nilai variabel independen pada titik data kedua

### 3.3.2 Feature Engineering

*Feature engineering* merupakan proses penting dalam melakukan analisis data yang bertujuan untuk melakukan ekstraksi, transformasi serta membentuk fitur-fitur baru yang lebih informatif dan relevan dari data mentah. Dalam penelitian ini, *Feature engineering* dilakukan terhadap data konsumsi energi pada sistem *blockchain*. Proses ini terbagi menjadi beberapa langkah utama:

#### 1. Ekstraksi komponen waktu

Langkah awal melakukan ekstraksi komponen waktu dari kolom *datetime* untuk menciptakan fitur-fitur baru seperti tahun, bulan, dan hari. Ekstraksi ini sendiri bertujuan untuk mendukung model dalam mengenali pola musiman atau periodik dalam konsumsi energi penambangan *bitcoin*. Misalnya, tren penggunaan energi dapat berbeda antara awal dan akhir tahun karena berbagai faktor seperti penyesuaian algoritma, aktivitas penambangan, dan harga pasar.

Komponen waktu ini juga membentuk fondasi untuk membangun *lag features* dan *ordinal time*. Tanpa adanya batasan waktu model akan mengalami kesulitan dalam mengenali pengaruh temporal secara eksplisit, karena hanya melihat waktu sebagai nilai tunggal yang terus meningkat. Oleh karena itu, fitur waktu ini tidak hanya berdiri sendiri, tetapi menjadi landasan penting dalam membangun fitur-fitur turunan lanjutan lainnya.

#### 2. Lag Features

Setelah ekstraksi waktu dilakukan, dibuat juga fitur-fitur historis

(*lag features*), yaitu fitur yang mengambil nilai masa lalu dari variabel utama *estimated\_twh* dan *minimum\_twh*. Fitur ini mencakup *TWh\_Lag1*, *TWh\_MA7*, *TWh\_MA30*, dan *TWh\_MA90*, yang masing-masing merepresentasikan konsumsi energi pada hari sebelumnya, rata-rata 7 hari, 30 hari, dan 90 hari terakhir. Penambahan fitur *lag features* ini memberikan konteks historis yang penting bagi model pembelajaran *machine learning*, khususnya MLP dalam mengenali ketergantungan waktu dan pola musiman yang terdapat dalam data historis dalam data. Tujuan utama dari tahapan ini adalah meningkatkan kualitas prediksi data, sehingga model dapat melakukan prediksi secara akurat dan kontekstual terhadap dinamika konsumsi energi berbasis *blockchain*, dengan menyediakan masukan yang lebih kontekstual dan representatif terhadap dinamika data konsumsi energi *blockchain*.

### 3. *Ordinal time*

Fitur *ordinal\_time* dibentuk dengan mengubah *datetime* menjadi angka berupa jumlah hari sejak tanggal awal dataset. Transformasi ini dilakukan agar model MLP yang berbasis numerik dapat mengenali urutan waktu secara linier. Meskipun fitur ini tampak sederhana, fungsinya sangat penting karena menjadi semacam penanda kronologi yang menjaga urutan logistik antar *lag*, tren musiman, dan perubahan nilai *hash rate*.

Misalnya, saat *hash rate* meningkat drastis pada suatu titik waktu, fitur *ordinal\_time* memastikan bahwa model memahami bahwa peristiwa tersebut berada pada posisi kronologi tertentu dalam waktu deret. Dengan kata lain, fitur ini mengikat seluruh dimensi waktu lainnya baik secara eksplisit seperti bulan dan hari, maupun implisit seperti *lag* ke dalam satu garis urut yang konsisten.

### 4. *Hash-Rate*

Fitur *Hash-rate* adalah indikator langsung dari kekuatan komputasi jaringan *bitcoin*. Nilai ini menunjukkan seberapa banyak daya komputasi yang digunakan untuk menyelesaikan proses penambangan. Tingkat hash

yang tinggi biasanya terhambat dengan peningkatan konsumsi energi. Oleh karena itu, memasukkan *hash rate* sebagai fitur yang sangat penting agar model mampu menangkap hubungan teknis antara kekuatan jaringan dan kebutuhan energinya.

Fitur ini juga erat kaitannya dengan waktu dan *lag*. Seperti contoh peningkatan *hash rate* yang konsisten dalam beberapa hari yang tercermin dalam *lag* dan ordinal time akan mendorong peningkatan energi, sehingga model dapat mengenali tren tersebut sebagai bagian dari pola. Dengan *hash rate* sebagai *input*, model tidak hanya mengenali pola dari data historis energi semata, tetapi juga mempertimbangkan kondisi jaringan yang sebenarnya.

#### 5. *Difficulty*

*Difficulty* adalah fitur teknis lain yang menunjukkan tingkat kesulitan algoritma *Proof of Work* (PoW) dalam menyelesaikan satu blok transaksi. Nilai ini disesuaikan secara periodik oleh protokol *bitcoin* untuk menjaga waktu blok yang konstan. Semakin tinggi tingkat kesulitan, semakin besar pula energi dan *hash rate* yang dibutuhkan. Oleh karena itu, kesulitan sangat relevan sebagai penyebab melemahkan konsumsi energi yang selanjutnya tercermin dalam fitur *gap\_twh*.

Fitur ini berperan untuk melengkapi tingkat hash, karena dalam beberapa kondisi tingkat hash dapat meningkat, tetapi tingkat kesulitannya tetap, atau sebaliknya. Oleh karena itu, kombinasi *difficulty* dan *hash rate* memberikan konteks teknis yang saling melengkapi. Bersama dengan ordinal time dan *lag*, fitur ini membantu model mengenali perubahan signifikan dalam sistem PoW secara berurutan.

#### 6. *Gap per TWh*

Pada fitur ini dihitung sebagai selisih antara *estimated\_twh* dan *minimum\_twh*. Nilai ini digunakan sebagai variabel target pada model regresi linier dan MLP, yang menggambarkan seberapa besar fluktuasi atau kelebihan konsumsi energi yang terjadi dalam periode tertentu. Dengan menargetkan gap, model ini di arahkan untuk mempelajari faktor-

faktor yang menyebabkan konsumsi energi melebihi batas minimumnya, baik karena faktor waktu, faktor teknis, maupun faktor musiman.

Gap ini tidak dapat dipahami dengan baik tanpa dukungan dari fitur lain yang menjelaskannya. Oleh karena itu, seluruh fitur sebelumnya disusun secara terstruktur untuk memberikan konteks terhadap pembentukan dan prediksi gap\_twh.

#### 7. *MinMaxScaler*

Sebelum seluruh fitur numerik digunakan dalam penelitian model MLP, dilakukan proses normalisasi menggunakan *MinMaxScaler*, yang mengubah skala nilai setiap fitur ke rentang 0 hingga 1. Proses ini dilakukan dengan persamaan:

$$X_{\text{Scaled}} = \frac{X - X_{\text{Min}}}{X_{\text{Max}} - X_{\text{Min}}} \quad (5)$$

Dengan:

- $X$  = Nilai asli
- $X_{\text{Min}}$  = Nilai minimum pada fitur
- $X_{\text{Max}}$  = Nilai maksimum pada fitur
- $X_{\text{Scaled}}$  = Nilai hasil normalisasi

Normalisasi ini sangat penting untuk mencegah dominasi fitur *hash\_rate* atau *ordinal\_time* terhadap fitur lain dengan rentang lebih kecil. Dengan skala yang seragam, model dapat belajar secara optimal dari interaksi antara fitur yang telah dirancang sebelumnya dan semuanya di normalisasikan, model dapat lebih mudah menangkap pola gabungan yang menyebabkan gesekan gap\_twh.

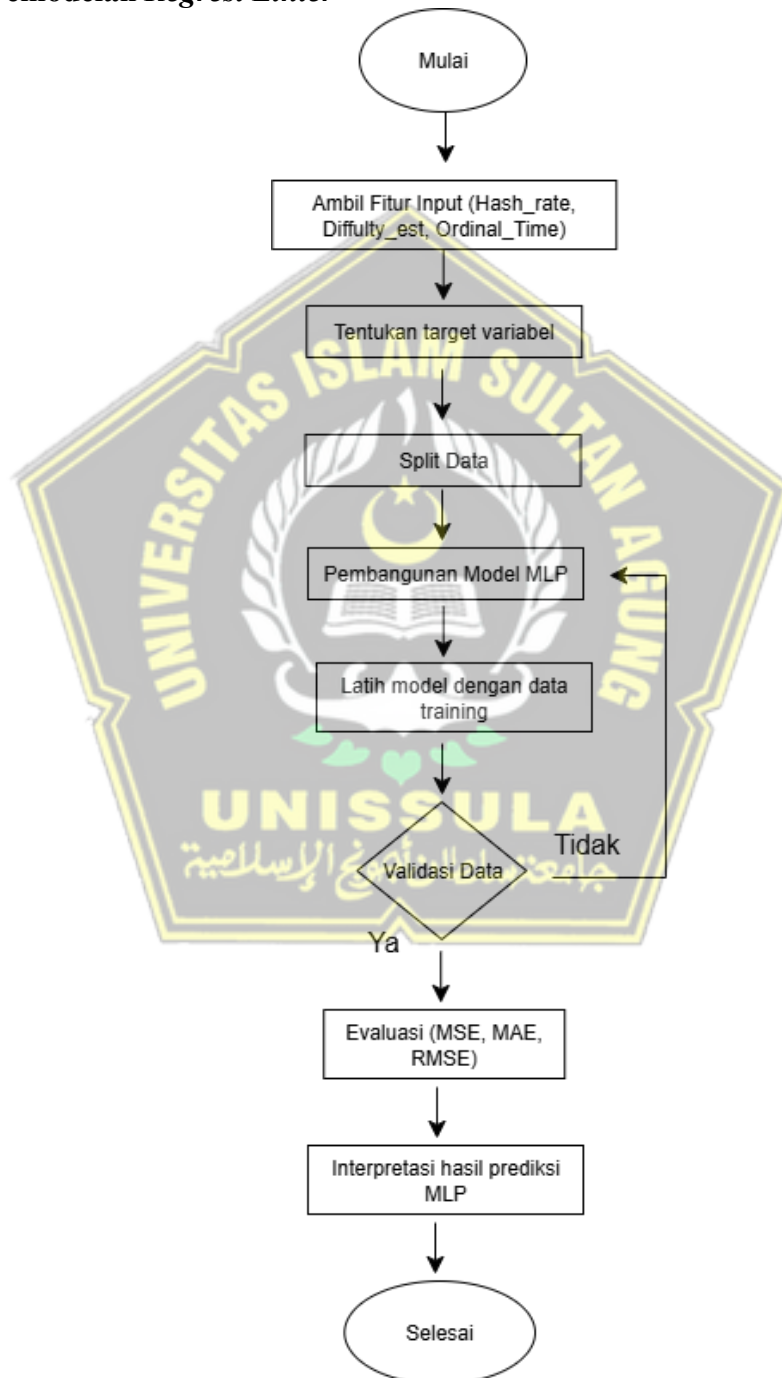
### 3.4 Pelatihan Model

Tahap pelatihan model merupakan inti penelitian ini, dengan menguji dua pendekatan pemodelan: Regresi Linier sebagai baseline konvensional dan Multilayer Perceptron (MLP) sebagai algoritma deep learning. Regresi Linier dipilih karena kesederhanaannya dalam menggambarkan hubungan linier antara variabel prediktor (*ordinal time*, *hash rate*, dan *difficulty\_est*)



dengan target (gap\_twh). Model ini berfungsi sebagai acuan awal untuk menilai efektivitas MLP. Sebaliknya, MLP digunakan karena kemampuannya menangkap pola non-linier yang kompleks serta interaksi antar variabel yang tidak dapat dimodelkan oleh regresi linier.

### 3.4.1 Pemodelan *Regresi Linier*



Gambar 3. 3 Flowchart pembangunan model menggunakan *regresi linier*

Pada gambar 3.3 *Flowchart* di atas menggambarkan tahapan prediksi konsumsi energi menggunakan pendekatan *Regresi Linier*, yg berperan sebagai *baseline* awal sebelum pemodelan MLP. Proses dimulai dari tahap awal pengolahan data hingga interpretasi hasil prediksi, dengan validasi pada proses normalisasi untuk memastikan integritas data input. Dengan langkah-langkah sebagai berikut:

3.4.1.1 Pada tahap pemodelan regresi linier, proses dimulai dengan melakukan persiapan data yang digunakan sebagai masukan ke dalam model. Fitur *input* yang digunakan terdiri dari tiga variabel, yaitu *Hash\_rate*, *Difficulty\_est*, dan *Ordinal\_time*. Variabel *Hash\_rate* merepresentasikan kecepatan komputasi jaringan dalam melakukan proses penambangan, *Difficulty\_est* merupakan tingkat kesulitan algoritma konsensus *Proof of Work* yang terus menyesuaikan berdasarkan kondisi jaringan, sedangkan *Ordinal\_time* adalah representasi waktu yang telah diubah menjadi nilai ordinal agar dapat digunakan sebagai variabel numerik. Ketiga variabel ini berperan sebagai variabel independen dalam model, sedangkan variabel target yang diprediksi adalah *Estimated TWh per Year*, yaitu estimasi konsumsi energi tahunan dalam satuan terawatt-hour (TWh).

#### 3.4.1.2 Preprocessing Data

Sebelum proses pemodelan dilakukan, data terlebih dahulu melalui tahap pra-pemrosesan. Pada tahap ini dilakukan pemeriksaan terhadap nilai yang hilang (*missing values*) baik pada kolom fitur maupun target. Apabila ditemukan nilai yang tidak valid, maka data diperbaiki atau dihilangkan agar tidak menimbulkan bias dalam proses pelatihan model. Selain itu, data juga dinormalisasi untuk menyamakan skala antar variabel, sehingga model dapat berfungsi lebih stabil. Setelah data dinyatakan bersih, *dataset* dibagi menjadi dua bagian, yaitu data latih (*training data*) dan data uji (*testing data*) dengan rasio tertentu, misalnya 80% untuk pelatihan dan 20% untuk pengujian.

#### 3.4.1.3 Pembangunan Model Regresi Linier

Proses pemodelan dilakukan dengan membangun persamaan *Multiple*

*Linear Regression* karena melibatkan lebih dari satu variabel independen sebagai *prediktor*. Fitur input yang digunakan terdiri dari tiga variabel, yaitu *Hash\_rate*, *Difficulty\_est*, dan *Ordinal\_time*. Variabel *Hash\_rate* menggambarkan kekuatan komputasi jaringan Bitcoin, *Difficulty\_est* mencerminkan tingkat kesulitan proses penambangan, sedangkan *Ordinal\_time* merepresentasikan waktu dalam bentuk numerik. Ketiga variabel tersebut berfungsi sebagai variabel independen, sedangkan variabel target yang diprediksi adalah *Estimated TWh per Year*, yaitu estimasi konsumsi energi tahunan dalam satuan *terawatt-hour* (TWh).

Secara matematis, model regresi linier berganda dapat dinyatakan dalam persamaan berikut:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon \quad (6)$$

- $Y$  = Nilai variabel dependen (variabel hasil prediksi)
- $\beta_0$  = konstanta regresi, yaitu nilai  $Y$  saat semua  $X_i = 0$
- $\beta_1, \beta_2, \dots, \beta_n$  = Koefisien regresi untuk masing – masing variabel independen
- $X_1, X_2, \dots, X_n$  = Variabel independen (prediktor)
- $\varepsilon$  (*epsilon*) = Error term, mewakili factor-faktor lain yang tidak dimasukkan dalam model tetapi mempengaruhi  $Y$ .

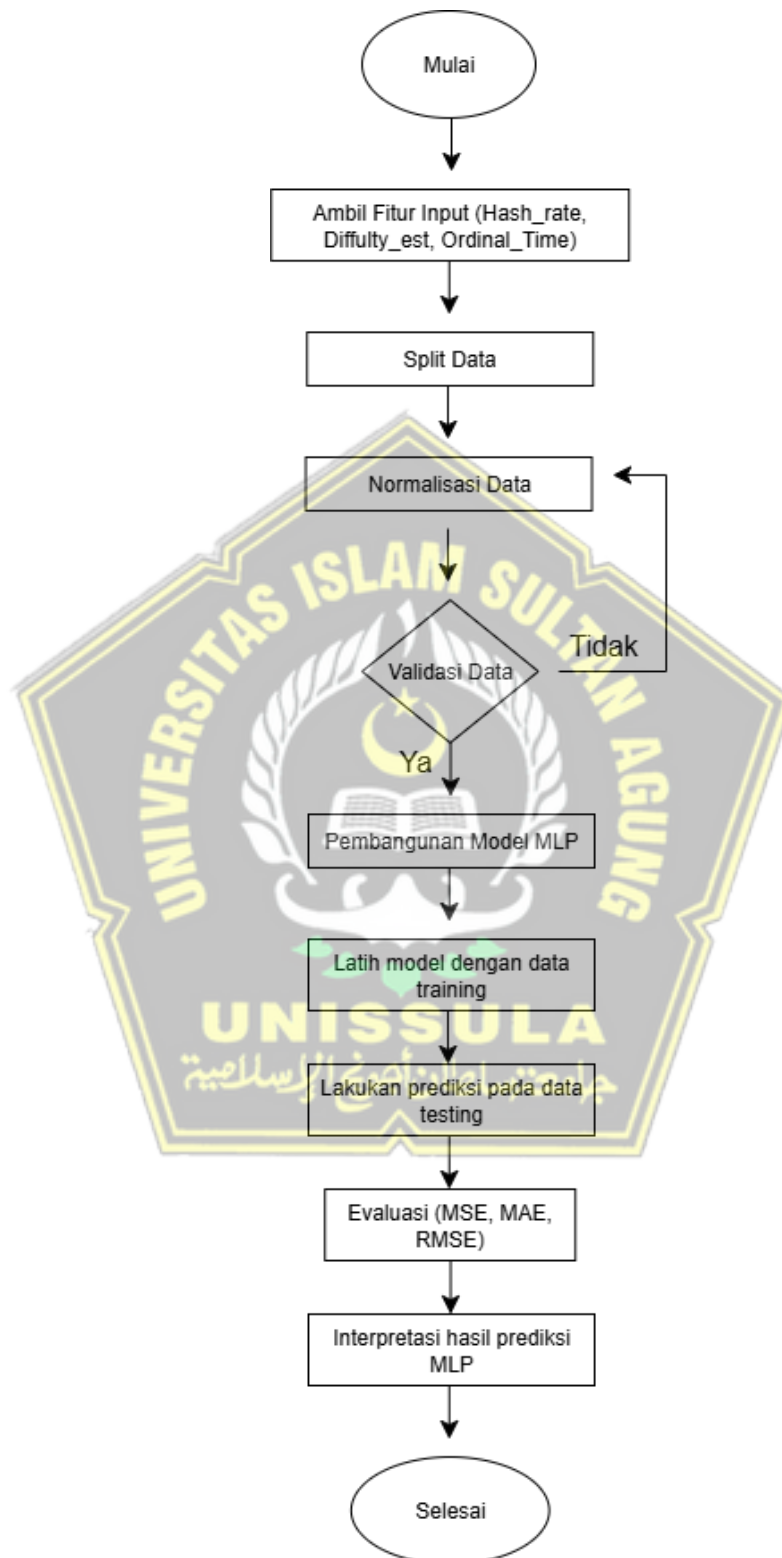
Dalam model tersebut,  $Y$  merupakan variabel dependen atau target, yaitu nilai yang ingin diprediksi berdasarkan jumlah variabel independen. nilai  $\beta_0$  adalah konstanta regresi (intersep), yang merepresentasikan nilai  $Y$  ketika seluruh variabel prediktor  $X_i$  bernilai nol. Koefisien  $\beta_1, \beta_2, \dots, \beta_n$  adalah parameter regresi yang menunjukkan seberapa besar pengaruh dari masing-masing variabel independen  $X_1, X_2, \dots, X_n$  terhadap variabel dependen  $Y$ .

Setiap  $X_i$  adalah variabel independen atau prediktor yang diasumsikan memiliki hubungan linier dengan  $Y$ . Sementara itu,  $\varepsilon$  (*epsilon*) merupakan error term atau komponen galat yang menggambarkan variasi dari  $Y$  yang

tidak dapat dijelaskan oleh model, seperti faktor luar yang tidak dimasukkan ke dalam variabel prediktor. Keberadaan  $\varepsilon$  (epsilon) penting dalam model statistik karena mewakili faktor acak dan ketidakpastian. Dengan model ini memungkinkan peneliti untuk mengukur dan menganalisis pengaruh kolektif dari banyak variabel terhadap satu outcome utama, sehingga sangat efektif digunakan dalam penelitian prediktif berbasis data historis seperti pada studi konsumsi energi bitcoin.

- 3.4.1.4 Tahap berikutnya adalah pelatihan model menggunakan data latih untuk memperoleh parameter regresi yang optimal. Setelah model terbentuk, dilakukan pengujian menggunakan data uji dengan cara memasukkan nilai *Hash\_rate*, *Difficulty\_est*, dan *Ordinal\_time* ke dalam persamaan regresi. Hasil prediksi kemudian dibandingkan dengan nilai aktual dari Estimated TWh per Year untuk mengukur kinerja model.
- 3.4.1.5 Evaluasi dilakukan dengan menghitung metrik seperti *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), dan  $R^2$  (koefisien determinasi). Nilai MAE, MSE, dan RMSE menunjukkan besarnya rata-rata kesalahan prediksi, sedangkan  $R^2$  menunjukkan proporsi variasi data target yang dapat dijelaskan oleh variabel input dalam model. Nilai  $R^2$  yang mendekati 1 menandakan bahwa model memiliki kemampuan yang baik dalam menjelaskan hubungan antara input dan output.
- 3.4.1.6 Tahap terakhir adalah interpretasi hasil, yaitu memberikan penjelasan atas makna nilai koefisien regresi maupun metrik evaluasi yang diperoleh. Misalnya, koefisien  $\beta_1$  pada variabel *Hash\_rate* dapat menunjukkan seberapa besar peningkatan konsumsi energi dipengaruhi oleh peningkatan kekuatan komputasi jaringan. Dengan demikian, pemodelan regresi linier ini tidak hanya memberikan prediksi kuantitatif, tetapi juga mampu menjelaskan hubungan antara variabel-variabel penting dalam sistem konsensus Proof of Work dengan tingkat konsumsi energi Bitcoin.

### 3.4.2 Pemodelan *Multilayer Perceptron* (MLP)



Gambar 3. 4 Alur prediksi menggunakan MLP  
 Pada gambar 3.4 *Flowchart* di atas menggambarkan tahapan prediksi



konsumsi energi menggunakan pendekatan *Artificial Neural Network* (ANN), yaitu *Multilayer Perceptron* (MLP). Proses dimulai dari tahap awal pengolahan data hingga interpretasi hasil prediksi, dengan validasi pada proses normalisasi untuk memastikan integritas data input. Dengan langkah-langkah sebagai berikut:

3.4.2.1 Langkah pertama adalah mengambil fitur input yang relevan dari dataset yang telah melalui tahap feature engineering dengan fitur yang digunakan meliputi:

1. *Ordinal time* (dibentuk dengan mengubah datetime menjadi angka berupa jumlah hari sejak tanggal awal dataset)
2. *Hash-Rate* (indikator langsung dari kekuatan komputasi jaringan bitcoin)
3. *Difficulty\_est* (fitur teknis lain yang menunjukkan tingkat kesulitan algoritma *Proof of Work* (PoW) dalam menyelesaikan satu blok transaksi.)

Dari fitur-fitur ini nantinya disusun menjadi vector input yang digunakan sebagai masukan dalam *Artificial Neural Network* (ANN).

#### 3.4.2.2 Data Splitting

Proses *data splitting* ini bertujuan untuk memisahkan dataset menjadi dua bagian independen: satu untuk *training set* dan satu lagi untuk menguji kinerjanya secara objektif (*testing set*). Langkah pertama yaitu mendefinisikan variabel independen atau X dan variabel target variabel dependen atau y yang akan diprediksi. Setelah itu, fungsi `train_test_split` dari pustaka Scikit-learn digunakan untuk membagi data.

1. `test_size=0.2`: Menetapkan bahwa 20% dari total dataset akan dialokasikan sebagai data uji (`X_test`, `y_test`), sementara 80% sisanya menjadi data latih (`X_train`, `y_train`).
2. `random_state=42`: Digunakan untuk memastikan bahwa pembagian data akan selalu menghasilkan urutan acak yang sama setiap kali kode dijalankan. Ini penting untuk reproduksibilitas hasil penelitian.

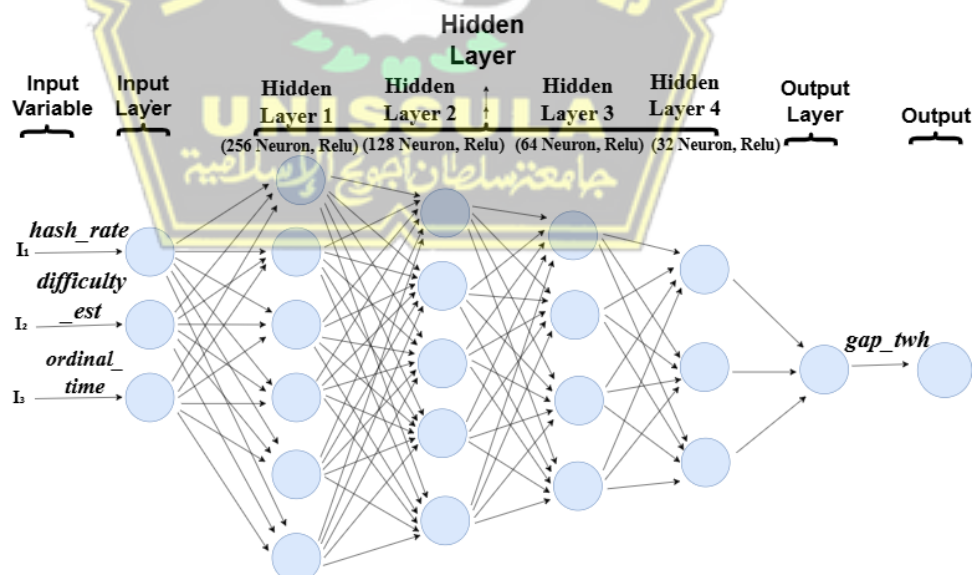
3.4.2.3 Normalisasi data dengan seluruh fitur numerik dinormalisasikan menggunakan metode *minmaxscaler* untuk mengubah nilai ke dalam rentang (0, 1) Normalisasi diperlukan agar semua fitur berada dalam skala yang seimbang, dan mencegah dominasi fitur bernilai besar terhadap bobot jaringan. Menggunakan persamaan:

$$\hat{X} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (7)$$

- $\hat{X}$  : Nilai hasil normalisasi (dalam rentang 0-1)
- $X$  : Nilai asli dari fitur yang ingin dinormalisasikan
- $X_{\min}$  : Nilai minimum dari seluruh nilai pada fitur  $x$
- $X_{\max}$  : Nilai maksimum dari seluruh nilai pada fitur  $x$

#### 3.4.2.4 Pembangunan model

Untuk pembangunan model MLP, Proses prediksi output dilakukan melalui kombinasi linier bobot dan bias yang diaktifkan oleh fungsi non-linier pada lapisan tersembunyi (hidden layer). Model ini dirancang menggunakan library Keras dari TensorFlow dengan beberapa komponen.



Gambar 3. 5

Pada gambar 3.5 arsitektur model *Multilayer Perceptron* (MLP) terdiri dari enam lapisan utama yang dirancang untuk menangkap

kompleksitas hubungan antar fitur dalam data konsumsi energi *blockchain*. Lapisan pertama adalah *Input Layer* yang terdiri dari tiga neuron, masing-masing merepresentasikan fitur *ordinal\_time*, *hash\_rate*, dan *difficulty\_est*. Karena lapisan ini hanya berfungsi sebagai penerima data tanpa melakukan transformasi, maka tidak digunakan fungsi aktivasi (None).

Selanjutnya, *Hidden Layer 1* memiliki 256 neuron dengan fungsi aktivasi *ReLU*. Lapisan ini berperan dalam menangkap pola non-linear awal dari ketiga fitur input, memperluas kapasitas representasi model sejak tahap awal. *Hidden Layer 2* terdiri dari 128 neuron dengan aktivasi *ReLU*, yang memperkuat kemampuan model dalam mengenali pola lanjutan yang lebih kompleks. *Hidden Layer 3* dan *Hidden Layer 4*, masing-masing memiliki 64 dan 32 neuron dengan aktivasi *ReLU*, berfungsi untuk memperdalam abstraksi fitur dan menyaring informasi yang relevan sebelum proses prediksi.

Terakhir, *Output Layer* menggunakan satu neuron dengan fungsi aktivasi *linear*, yang sesuai untuk tugas regresi karena menghasilkan *output* berupa nilai kontinu. *Output* yang dihasilkan adalah prediksi *gap\_twh*, yaitu selisih konsumsi energi *blockchain* dalam satuan *terawatt-hour*. Struktur arsitektur ini dirancang untuk menggabungkan kapasitas representasi yang dalam dengan efisiensi komputasi, sehingga model dapat memberikan prediksi yang akurat dan relevan terhadap konteks data yang digunakan.

Sebelum menetapkan arsitektur final, dilakukan penyetelan *hyperparameter* menggunakan *GridSearchCV* untuk menguji berbagai kombinasi parameter secara sistematis. Parameter yang diuji meliputi *hidden\_layer\_sizes*, fungsi activation *relu*, *tanh*, *solver*, *adam* dan *alpha* (parameter regularisasi). Berdasarkan hasil pencarian, konfigurasi terbaik yang diperoleh adalah *hidden\_layer\_sizes* = 100, 50, *activation* = *relu*, *solver* = *adam*, dan *alpha* = 0.0005. Kombinasi ini menunjukkan performa terbaik pada data validasi dan digunakan sebagai acuan dalam

menetapkan arsitektur dan parameter pelatihan akhir model MLP

#### 3.4.2.5 Proses pelatihan model *Multilayer Perceptron* (MLP)

Proses ini bersifat *iteratif* menyesuaikan parameter internal model untuk meminimalkan kesalahan prediksi. Secara umum, siklus pelatihan MLP terdiri dari serangkaian langkah yang terkoordinasi. Pertama, *inisialisasi* bobot dilakukan dengan memberikan nilai acak pada setiap bobot ( $w$ ) dan bias ( $b$ ) di seluruh jaringan. Selanjutnya, pada tahap *forward Propagation*, data input dialirkan maju dari *input layer* menuju *output layer*, melewati setiap *hidden layer*. Di setiap neuron, sinyal masukan diolah melalui perkalian dengan bobot, penambahan bias, dan dilewatkan pada fungsi aktivasi non-linier seperti *ReLU* hingga menghasilkan nilai prediksi akhir ( $\hat{y}$ ) yang akurat pada data baru. Proses pelatihan ini melibatkan tiga komponen utama:

##### 1. Fungsi *Loss*

Fungsi *Loss* adalah metrik yang mengukur seberapa besar kesalahan (error) antara nilai yang diprediksi oleh model dengan nilai aktual. Fungsi ini menjadi panduan bagi model untuk mengetahui seberapa baik performanya. Untuk tugas regresi seperti memprediksi konsumsi energi, fungsi *Loss* yang digunakan adalah MSE (*Mean Squared Error*). Tujuan utama dari proses pelatihan ini untuk menemukan bobot dan bias yang dapat meminimalkan nilai.

##### 2. *Backpropagation*

Setelah kesalahan dihitung, algoritma *backpropagation* berfungsi dengan mengirimkan informasi kesalahan itu secara terbalik, dari lapisan output menuju lapisan *input*. Selama proses ini, *backpropagation* menentukan kontribusi atau gradien setiap bobot dan bias terhadap total kesalahan. Gradien ini mengindikasikan arah dan besar penyesuaian yang harus dilakukan pada parameter model untuk meminimalkan kesalahan di iterasi selanjutnya.

##### 3. Optimizer

Optimizer ialah algoritma yang memanfaatkan gradien dari

backpropagation untuk memperbarui nilai bobot dan bias di dalam jaringan. *Optimizer* bertugas memastikan bahwa pembaruan ini dilaksanakan dengan efisien sehingga nilai fungsi kerugian dapat mencapai titik minimumnya. Proses latihan ini dilakukan kembali dalam beberapa siklus (*epoch*) sampai model mencapai tingkat akurasi yang diinginkan dengan menggunakan 200 *epoch*.

#### 3.4.2.6 Evaluasi Hasil Prediksi *Multilayer Perceptron* (MLP)

Setelah model selesai dilatih menggunakan data *training*, tahap selanjutnya adalah mengevaluasi performanya pada data uji (*test data*). Evaluasi ini bertujuan untuk mengukur akurasi prediksi model secara objektif dengan membandingkan nilai prediksi dengan nilai aktual menggunakan metrik statistik. Sesuai dengan batasan masalah dalam penelitian ini, metrik evaluasi yang digunakan adalah:

- 1 *Mean Absolute Error* (MAE): Metrik ini mengukur rata-rata kesalahan absolut antara nilai prediksi dan aktual. Keunggulannya adalah mudah diinterpretasikan karena memiliki satuan yang sama dengan variabel target.
- 2 *Mean Squared Error* (MSE): Metrik ini menghitung rata-rata dari kuadrat selisih antara nilai prediksi dan aktual. MSE sangat berguna untuk memberi penalti yang lebih besar pada kesalahan prediksi yang signifikan (*outlier*).
- 3 *R-Squared* ( $R^2$ ): Dikenal juga sebagai Koefisien Determinasi, metrik ini mengukur seberapa baik model dapat menjelaskan variasi data. Nilainya berkisar dari 0 hingga 1, di mana nilai yang mendekati 1 menandakan performa model yang sangat baik dalam memetakan hubungan antara fitur dan target.

### 3.5 Evaluasi Model

Setelah model selesai melalui proses *forward propagation*, *backpropagation*, dan *optimizer*, maka tahap selanjutnya dilakukan evaluasi untuk mengukur seberapa akurat model dalam melakukan prediksi. Evaluasi



ini penting untuk mengetahui kualitas hasil prediksi terhadap nilai konsumsi energi yang sebenarnya. Untuk menilai performa dan akurasi model prediksi yang telah dibangun, diperlukan evaluasi kuantitatif menggunakan metrik statistik. Metrik ini membandingkan nilai yang diprediksi oleh model dengan nilai aktual dari data. Sesuai dengan batasan masalah dalam penelitian ini, metrik yang digunakan adalah *Mean Absolute Error* (MAE), *Mean Squared* (MSE), dan *R-squared* ( $R^2$ ).

#### 1 *Mean Absolute Error* (MAE)

Digunakan untuk mengukur rata-rata dari kesalahan absolut antara nilai prediksi dan nilai aktual. Metrik ini mudah karena memiliki satuan yang sama dengan variabel target.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (8)$$

- $n$  : Jumlah total data atau observasi dalam dataset
- $y$  : Nilai aktual atau nilai sebenarnya dari data ke- $i$
- $\hat{y}_i$  : Nilai prediksi yang dihasilkan oleh model untuk data ke- $i$
- $i$  : Simbol untuk nilai absolut, dimana nilai didalamnya baik berupa negatif maupun positif tetap akan di ubah ke nilai positif.

#### 2 *Mean Squared Error* (MSE)

Menghitung rata-rata dari kuadrat selisih antara nilai prediksi dan aktual. Dengan mengkuadratkan selisihnya, MSE memberikan penalti yang lebih besar pada kesalahan yang besar, sehingga sangat sensitif terhadap outlier.

$$MSE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|^2 \quad (9)$$

- $n$  : Jumlah total data atau observasi dalam dataset
- $y$  : Nilai aktual atau nilai sebenarnya dari data ke- $i$
- $\hat{y}_i$  : Nilai prediksi yang dihasilkan oleh model untuk data ke- $i$
- $i$  : Simbol untuk nilai absolut, dimana nilai didalamnya baik berupa negatif maupun positif tetap akan di ubah ke nilai positif.

### 3 *R-Squared* ( $R^2$ )

*R-squared* mengukur seberapa baik variabel independen dapat menjelaskan variabilitas dari variabel dependen. Nilainya berkisar antara 0 hingga 1, di mana nilai yang mendekati 1 menunjukkan bahwa model mampu menjelaskan sebagian besar variabilitas data dengan baik.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (10)$$

- $R^2$ : Hasil akhir dari metrik Koefisien Determinasi, yang mengukur seberapa baik model menjelaskan variasi data.
- $n$  : Jumlah total data atau observasi.
- $y_i$  : Nilai aktual atau nilai sebenarnya dari data ke- $i$
- $\hat{y}_i$  : Nilai prediksi yang dihasilkan oleh model untuk data ke- $i$
- $\bar{y}$  : Nilai rata-rata dari seluruh nilai aktual ( $y$ )

### 3.6 Visualisasi Model

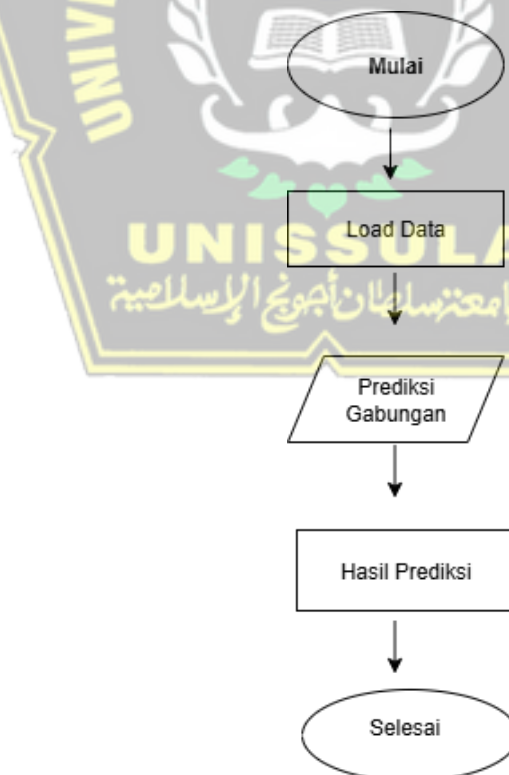
Tahapan visualisasi model berfungsi untuk memberikan gambaran kinerja prediksi secara jelas dan mudah dipahami, baik dari sisi analisis kualitatif maupun kuantitatif. Visualisasi ini membantu menginterpretasikan sejauh mana model mampu memprediksi tren konsumsi energi Bitcoin, sekaligus memvalidasi hasil yang diperoleh. Implementasinya dilakukan dengan menampilkan grafik perbandingan antara data historis dan hasil prediksi model, sehingga perbedaan maupun kesesuaian tren dapat diamati secara langsung.

Grafik yang ditampilkan mencakup tren estimasi konsumsi energi (TWh) terhadap waktu, serta rentang prediksi yang dihasilkan model untuk periode mendatang. Selain itu, ditambahkan pula indikator seperti selisih energi (*gap\_twh*) untuk memberikan konteks tambahan terkait volatilitas atau fluktuasi nilai prediksi. Tampilan visual ini memanfaatkan pustaka interaktif sehingga pengguna dapat menjelajah data secara detail, memperbesar area tertentu, dan mengamati pola musiman atau anomali yang muncul. Dengan

visualisasi ini, proses evaluasi model menjadi lebih intuitif, memudahkan identifikasi keakuratan prediksi, dan membantu dalam pengambilan keputusan berbasis data.

### 3.7 Perancangan User Interface

Pada tahap ini peneliti akan menentukan alur kerja sistem berupa *flowchart* yang akan memberikan gambaran dari alur sistemnya. Dimana model yang dipilih nantinya akan melakukan prediksi konsumsi energi pada jaringan Bitcoin. Alur kerja dirancang secara sederhana namun efektif, sehingga proses dapat dipahami dan dijalankan secara berurutan. Tahapan dimulai dari pemuatan data historis yang telah dipersiapkan, kemudian dilanjutkan pada proses prediksi gabungan menggunakan model time series Prophet dan model regresi MLP. Hasil prediksi ditampilkan dalam bentuk visualisasi grafik dan metrik, sehingga memudahkan interpretasi terhadap tren konsumsi energi yang diproyeksikan.

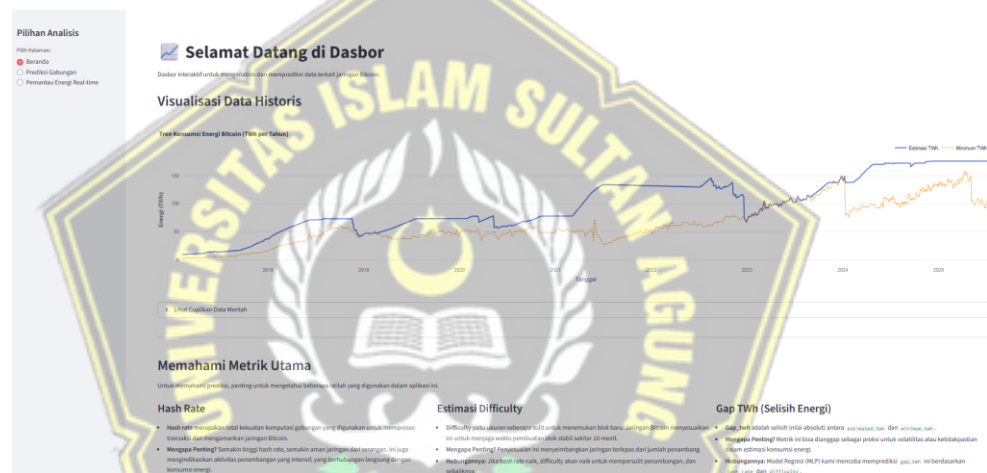


Gambar 3. 6 *Flowchart* perancangan system

Berdasarkan gambar 3.6 *flowchart* di atas, proses perancangan sistem

berjalan secara linear dari tahap awal hingga akhir. Setelah data berhasil dimuat, sistem langsung melakukan proses prediksi gabungan yang mengombinasikan tren jangka panjang dan variasi selisih energi. Tahap akhir menampilkan hasil prediksi secara interaktif, yang dapat digunakan sebagai dasar pengambilan keputusan maupun kajian lanjutan terkait efisiensi energi pada mekanisme *Proof of Work* (PoW). Alur yang sederhana ini diharapkan dapat meminimalkan kompleksitas proses dan memaksimalkan kecepatan analisis.

### 3.7.1 Halaman Awal Sistem



Gambar 3. 7 Tampilan dashboard

Pada gambar 3.7 merupakan halaman awal sistem prediksi konsumsi energi jaringan *bitcoin*. Pada halaman ini, pengguna disambut dengan judul “Selamat Datang di Dasbor” yang diikuti dengan deskripsi singkat mengenai fungsi sistem.

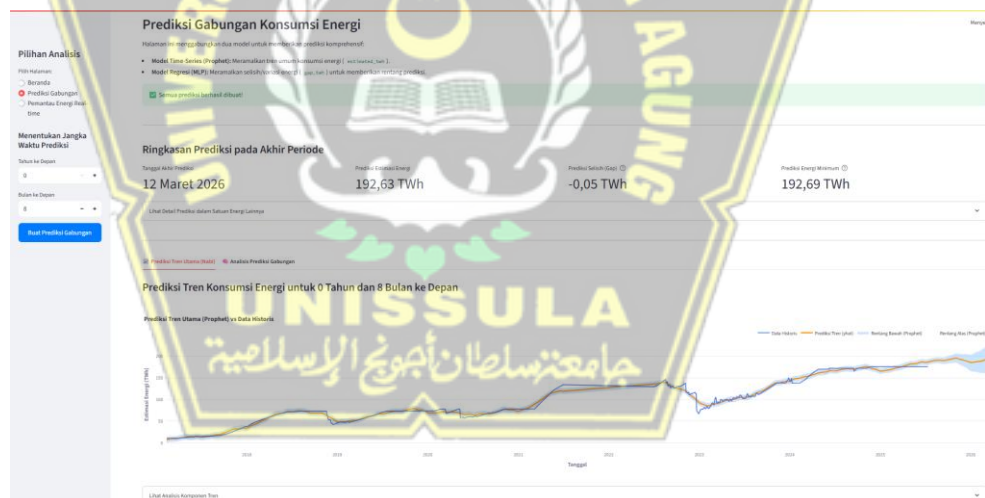
Bagian utama dari halaman ini adalah visualisasi data historis konsumsi energi *bitcoin*, yang ditampilkan dalam bentuk grafik interaktif menggunakan Plotly. Grafik ini memperlihatkan tren estimasi energi dan energi minimum sejak tahun 2017 hingga data terakhir yang tersedia. Sumbu horizontal mewakili waktu, sementara sumbu vertikal menunjukkan total konsumsi energi dalam satuan Terawatt-jam (TWh).

Selain grafik, halaman ini juga menampilkan penjelasan tiga metrik utama yang digunakan dalam model prediksi, yaitu:

- Tingkat *Hash Rate*– Menggambarkan kekuatan komputasi jaringan.
- *Difficulty Estimate*– Mengukur tingkat kesulitan menemukan blok baru di jaringan Bitcoin.
- Selisih Energi (Gap TWh) – Merupakan nilai absolut selisih antara estimasi dan minimum energi, yang dijadikan target prediksi dalam model MLP.

Setiap penjelasan dilengkapi dengan informasi mengenai pentingnya metrik tersebut dan bagaimana keterkaitannya dengan konsumsi energi secara keseluruhan. Antarmuka ini dirancang secara interaktif, informatif, dan responsif, dengan memanfaatkan *framework* Streamlit. Tujuannya adalah untuk memberikan pengalaman pengguna yang mudah diakses oleh pengguna umum maupun akademisi yang tertarik terhadap analisis energi dalam jaringan *blockchain*.

### 3.7.2 Tampilan Hasil Sistem



Gambar 3. 8 Tampilan halaman prediksi

Gambar 3.8 menampilkan antarmuka sistem pada halaman Prediksi Gabungan Konsumsi Energi, setelah pengguna memilih periode waktu prediksi dan menekan tombol Buat Prediksi Gabungan. Pada bagian atas halaman, sistem menampilkan ringkasan hasil prediksi yang mencakup tanggal akhir prediksi, nilai estimasi konsumsi energi, nilai selisih energi (*Gap*), dan estimasi konsumsi energi minimum. Informasi ini membantu pengguna memahami proyeksi kebutuhan energi jaringan *bitcoin* di masa mendatang, termasuk margin ketidakpastian yang ditunjukkan oleh nilai *gap*.



Nilai gap dapat bernilai positif atau negatif. Gap positif menandakan bahwa estimasi energi berada di atas ambang minimum, yang menunjukkan adanya aktivitas penambangan yang lebih tinggi dari yang dibutuhkan. Sebaliknya, gap negatif mengindikasikan bahwa estimasi energi lebih rendah dari nilai minimum, yang bisa mencerminkan efisiensi energi atau potensi penurunan aktivitas penambangan, meskipun secara logika nilai gap absolut seharusnya tidak negatif—ini menjadi catatan penting dalam menafsirkan keluaran model regresi.

Pada bagian bawah halaman, ditampilkan grafik tren prediksi konsumsi energi dari model time series Prophet. Visualisasi ini menunjukkan nilai estimasi ( $\hat{y}$ ) serta batas atas dan bawah prediksi berdasarkan hasil regresi dari model MLP terhadap nilai gap. Rentang ini divisualisasikan dalam bentuk area bayangan (*uncertainty band*) yang memberikan gambaran variasi estimasi yang mungkin terjadi. Dengan visualisasi ini, pengguna dapat mengevaluasi potensi fluktuasi energi yang diperkirakan, sekaligus mempertimbangkan margin error yang mungkin timbul dalam prediksi.

Antarmuka ini dirancang agar informatif, mudah dipahami, dan mendukung proses analisis berbasis data untuk perencanaan atau studi dampak energi dari sistem *blockchain* seperti *bitcoin*.

### 3.8 Deployment

Tahap *deployment* merupakan proses penerapan model prediksi yang telah dikembangkan ke dalam lingkungan aplikasi yang dapat diakses dan digunakan oleh pengguna akhir. Pada penelitian ini, model prediksi konsumsi energi *Bitcoin* diimplementasikan dalam bentuk dashboard interaktif berbasis web menggunakan framework Streamlit. Dashboard ini dirancang agar dapat menampilkan visualisasi data historis, hasil prediksi, serta analisis tren secara real-time maupun prediksi untuk periode mendatang.

Seluruh komponen model, termasuk data historis, algoritma prediksi (Prophet dan MLP), serta proses normalisasi dan rekayasa fitur, telah diintegrasikan ke dalam aplikasi sehingga pengguna tidak memerlukan

instalasi atau konfigurasi tambahan di sisi mereka. Proses *deployment* juga mencakup optimasi performa aplikasi, seperti pemanggilan data secara efisien dan penggunaan caching untuk mempercepat waktu respon.

Dengan adanya *deployment* ini, hasil penelitian tidak hanya berhenti pada tahap analisis, tetapi juga memberikan manfaat praktis dalam bentuk alat prediksi yang siap digunakan, memudahkan pemantauan konsumsi energi *bitcoin*, serta mendukung pengambilan keputusan berbasis data secara cepat dan akurat.

### 3.9 Analisa Kebutuhan

Pada tahap ini dilakukan analisis terhadap kebutuhan perangkat lunak, bahasa pemrograman, dan *library* yang diperlukan selama proses pengembangan sistem. Analisis ini bertujuan untuk memastikan seluruh komponen yang digunakan dapat mendukung pembangunan aplikasi prediksi konsumsi energi Bitcoin secara optimal. Adapun rincian kebutuhannya adalah sebagai berikut:

#### 1. Bahasa Pemrograman

##### a. Python

Python digunakan sebagai bahasa pemrograman utama karena memiliki sintaks yang sederhana, mudah dibaca, dan kaya akan pustaka yang mendukung pengolahan data, analisis statistik, hingga pembelajaran mesin. Kemampuannya untuk mengintegrasikan berbagai *framework* serta mendukung analisis data skala besar menjadikannya pilihan ideal dalam pengembangan sistem ini.

#### 2. Software yang digunakan

##### a. Jupyter Notebook

Digunakan untuk eksplorasi data, eksperimen model, dan visualisasi hasil secara interaktif. Jupyter memudahkan proses dokumentasi kode, sehingga setiap tahapan penelitian dapat tercatat secara sistematis.

b. Streamlit

Framework berbasis Python yang digunakan untuk membuat dashboard interaktif. Streamlit memudahkan implementasi hasil penelitian ke dalam aplikasi web tanpa memerlukan penguasaan HTML, CSS, atau JavaScript.

c. Visual Studio Code

Digunakan sebagai text editor utama dalam pengembangan aplikasi, karena mendukung multi-bahasa pemrograman, memiliki banyak ekstensi, dan performa yang ringan.

3. Library dan Framework

a. Pandas

Digunakan untuk manipulasi dan pembersihan data, seperti membaca file CSV, mengatur tipe data, dan melakukan transformasi dataset

b. NumPy

Digunakan untuk perhitungan numerik yang efisien, terutama dalam pengolahan array dan matriks.

c. Scikit-learn

Dimanfaatkan untuk proses *preprocessing* data, pembagian dataset, serta normalisasi menggunakan *MinMaxScaler*.

d. Prophet

Library *time series forecasting* yang digunakan untuk memodelkan dan memprediksi tren konsumsi energi berdasarkan data historis..

e. Plotly

Digunakan untuk membuat visualisasi interaktif seperti grafik tren, hasil prediksi, dan perbandingan data historis dengan prediksi.

f. TensorFlow

Framework pembelajaran mesin yang digunakan untuk membangun dan menjalankan model Multilayer Perceptron (MLP) dalam memprediksi *gap* konsumsi energi.

## BAB IV

### HASIL DAN ANALISIS PENELITIAN

#### 4.1 Hasil penelitian

##### 4.1.1 Pengumpulan Data

Tahap pengumpulan data pada penelitian ini dilakukan dengan memanfaatkan sumber data yang relevan dan dapat dipercaya, yaitu *Digiconomist*, sebuah platform yang menyediakan estimasi konsumsi energi jaringan *blockchain* secara berkala.

Tabel 4. 1 *Dataset* konsumsi energi

DateTime	Estimated TWh per Year	Minimum TWh per Year
02/10/2017 00:00	9.585659451	3.331026429
02/11/2017 00:00	9.547335368	3.316227361
02/12/2017 00:00	9.578774259	3.198969144
02/13/2017 00:00	9.476506126	3.045178216
02/14/2017 00:00	9.544807316	3.15006245
02/15/2017 00:00	9.569005513	3.337439903
02/16/2017 00:00	9.392170135	3.330369055
02/17/2017 00:00	9.323758505	3.238448041

*Dataset* yang digunakan secara khusus berfokus pada jaringan Bitcoin dan terdiri dari tiga kolom utama, yaitu Estimated TWh per Year, DateTime, dan Minimum TWh per Year. Data ini diperoleh dalam format CSV dan memiliki rentang waktu historis yang cukup panjang, sehingga dapat digunakan untuk menganalisis tren konsumsi energi *Bitcoin* dari waktu ke waktu. Proses pengumpulan data dilakukan melalui dua metode. Pertama, data historis diunduh secara langsung dari hasil pemrosesan sebelumnya yang telah disimpan dalam file CSV bernama *hasil\_filter.csv*. Kedua, untuk keperluan pembaruan data, penelitian ini juga memanfaatkan API *Digiconomist* guna mengambil data real-time konsumsi energi harian *Bitcoin*. Hasil dari kedua sumber data ini kemudian digabungkan dan disimpan untuk digunakan pada tahap preprocessing.

Pemilihan *dataset* ini didasarkan pada beberapa pertimbangan. *Digiconomist* merupakan salah satu sumber data yang kredibel dalam

mempublikasikan estimasi konsumsi energi *blockchain* dengan metodologi yang jelas dan teruji. Selain itu, *dataset* ini memiliki format terstruktur dan memuat informasi yang relevan bagi analisis efisiensi algoritma konsensus *Proof of Work* (PoW), termasuk estimasi konsumsi energi minimum serta estimasi konsumsi aktual yang dapat digunakan untuk menghitung selisih konsumsi energi (gap). Dengan karakteristik tersebut, *dataset* ini sangat sesuai digunakan untuk membangun model prediksi dan analisis tren konsumsi energi berbasis machine learning dan time series.

#### 4.1.2 Preprocessing data

Tahap preprocessing dalam penelitian ini dilakukan untuk memastikan data yang digunakan bersih, konsisten, dan siap diproses lebih lanjut oleh model prediksi. Proses ini mencakup beberapa langkah yang saling terkait:

##### 1. Mengatasi kolom yang bernilai *missing value*

Tahap awal proses *Preprocessing* dimulai dengan melakukan pengecekan terhadap nilai yang hilang (*missing values*) pada dataset konsumsi energi yang telah diunduh. Langkah ini bertujuan untuk memastikan kelengkapan data pada setiap kolom fitur yang digunakan, karena data yang tidak lengkap berpotensi mengganggu proses pelatihan model dan menghasilkan prediksi yang kurang akurat.

Pengecekan dilakukan dengan menghitung jumlah nilai kosong (NaN) pada setiap kolom fitur, yaitu *DateTime*, *Estimated TWh per Year*, dan *Minimum TWh per Year*. Hasil pemeriksaan tersebut dapat dilihat pada Tabel berikut:

Tabel 4. 2 Handling Missing values

Fitur Colom	Handling Missing Values
DateTime	0
Estimated TWh per year	365
Minimum TWh per year	0

Berdasarkan Tabel 4.2, kolom *Estimated TWh per Year* yang berisi estimasi konsumsi energi memiliki 365 nilai kosong (*missing values*). Nilai kosong tersebut terbagi menjadi dua periode, yaitu 184 data pada



tahun 2021 dan 181 data pada tahun 2022. Keberadaan nilai *NaN* pada kedua tahun tersebut dapat dilihat pada Gambar 4.3.

```
Jumlah NaN per tahun pada kolom 'Estimated TWh per Year ':
year
2021    184
2022    181
dtype: int64
```

Gambar 4. 1 Hasil pengecekan NaN

Pada gambar 4.1 menunjukkan adanya nilai NaN. Pada tahun 2021 dan 2022 untuk menangani masalah ini, dilakukan pengisian nilai kosong menggunakan metode *interpolasi linear*. Metode ini dipilih karena didasarkan pada asumsi bahwa perubahan konsumsi energi cenderung berlangsung secara bertahap dan berkelanjutan dari waktu ke waktu, sehingga nilai di antara titik data yang diketahui dapat diperkirakan secara proporsional

## 2. Mengecek duplikasi data

```
DateTime Estimated TWh per Year Minimum TWh per Year
1133 2020-03-19 00:00:00 67.884338 46.202545
1251 2020-07-14 00:00:00 61.898143 51.119417
```

Gambar 4. 2 Mengecek duplikasi data

Berdasarkan Gambar 4.2, pada tahap ini dilakukan proses untuk memeriksa apakah terdapat data yang terduplikasi pada dataset yang digunakan. Hasil pemeriksaan menunjukkan bahwa terdapat 2 baris data yang terduplikasi. Untuk mengatasi hal tersebut, digunakan metode `drop_duplicates()`, yang secara otomatis menghapus baris dengan nilai identik dan hanya menyisakan satu entri yang bersifat unik.

## 3. Hasil *Feature Engineering*

Tahapan *feature engineering* dilakukan untuk meningkatkan kualitas data sebelum dimasukkan ke dalam model prediksi.

Tabel 4. 3 Hasil dari feature engineering

year	month	day	TWh_Lag1	TWh_MA7	TWh_MA30	TWh_MA90	dinal_tin
2017	2	10		9.5277512	9.8673701	10.59931	736370
2017	2	11	9.5856595	9.5277512	9.8673701	10.59931	736371
2017	2	12	9.5473354	9.5277512	9.8673701	10.59931	736372
2017	2	13	9.5787743	9.5277512	9.8673701	10.59931	736373
2017	2	14	9.4765061	9.5277512	9.8673701	10.59931	736374

Tabel 4. 4 Hasil dari feature engineering

energy_per_has	hash_rate	hash_rate_Hs	difficulty_est	gap_twh
3.00E-08	1.0132E+13	1.01E+31	1.42E+24	6.254633022
3.00E-08	1.00915E+13	1.01E+31	1.41E+24	6.231108007
3.00E-08	1.01247E+13	1.01E+31	1.41E+24	6.379805115
3.00E-08	1.00166E+13	1.00E+31	1.40E+24	6.43132791
3.00E-08	1.00888E+13	1.01E+31	1.41E+24	6.394744866
3.00E-08	1.01144E+13	1.01E+31	1.41E+24	6.23156561
3.00E-08	9.92746E+12	9.93E+30	1.39E+24	6.061801079
3.00E-08	9.85515E+12	9.86E+30	1.38E+24	6.085310464
3.00E-08	9.85855E+12	9.86E+30	1.38E+24	6.153652947

. Tabel 4.3 dan 4.4 merupakan hasil dari *feature engineering*. Langkah awal dimulai dengan ekstraksi komponen waktu dari kolom *datetime* untuk menghasilkan fitur seperti tahun, bulan, dan hari, yang membantu model mengenali pola musiman dalam konsumsi energi *blockchain*. Selanjutnya, dibentuk *lag features* seperti TWh\_Lag1, TWh\_MA7, TWh\_MA30, dan TWh\_MA90 yang merepresentasikan nilai historis konsumsi energi, memberikan konteks temporal yang penting bagi model. Fitur *ordinal\_time* ditambahkan dengan mengubah waktu menjadi urutan numerik agar model MLP dapat memahami kronologi data secara linier. Selain itu, fitur teknis seperti *hash\_rate* dan *difficulty* dimasukkan untuk menangkap hubungan antara kekuatan komputasi jaringan dan kebutuhan energi, di mana keduanya saling melengkapi dalam menjelaskan dinamika sistem *Proof of Work*. Fitur *gap\_twh* dihitung sebagai selisih antara *estimated\_twh* dan *minimum\_twh*, dan digunakan sebagai target prediksi untuk memahami fluktuasi konsumsi energi yang melebihi batas minimum. Terakhir, seluruh fitur numerik dinormalisasi menggunakan *MinMaxScaler* agar model dapat belajar secara optimal tanpa dipengaruhi oleh perbedaan skala antar fitur.

#### 4. Pembagian data (*split data*)

Tahapan pembagian data dilakukan untuk memisahkan dataset menjadi dua bagian utama, yaitu data pelatihan (*training*) dan data pengujian (*testing*), guna memastikan bahwa model yang dibangun dapat melakukan generalisasi terhadap data yang belum pernah dilihat sebelumnya.

Tabel 4. 5 Hasil Split data

Dataset	Jumlah data	Split data	
		Data Training	Data Testing
Konsumsi_energi	3074	2457	615

Berdasarkan Tabel 4.5, total jumlah data konsumsi energi yang digunakan dalam penelitian ini adalah sebanyak 3.074 baris. Dari jumlah tersebut, sebanyak 2.457 baris dialokasikan sebagai data pelatihan, sementara 615 baris sisanya digunakan sebagai data pengujian. Proporsi ini mengikuti rasio umum 80:20 yang sering digunakan dalam praktik *machine learning*, di mana sebagian besar data digunakan untuk melatih model agar dapat mengenali pola, dan sisanya digunakan untuk menguji akurasi serta kemampuan prediksi model terhadap data baru. Pembagian ini menjadi fondasi penting dalam evaluasi performa model regresi linier maupun MLP yang digunakan dalam penelitian.

#### 4.1.3 Pembuatan model Regresi Linier dan Multilayer Perceptron (MLP)

Pada tahap ini, dilakukan implementasi dua pendekatan pemodelan untuk memprediksi *gap\_twh*. Model *Regresi Linier* digunakan sebagai *baseline* untuk mengukur hubungan linier dasar, sementara *Multilayer Perceptron* (MLP) dikembangkan untuk menangkap pola non-linier yang lebih kompleks dalam data.

##### 4.1.3.1 Pembangunan dan Pelatihan Model Regresi Linier

Model regresi linier yang digunakan pada penelitian ini dibangun menggunakan pustaka *scikit-learn* dengan memanfaatkan kelas Linear Regression. Fitur input distandarisasi menggunakan Standard Scaler agar setiap variabel memiliki skala yang sama, yaitu dengan rata-rata nol dan standar deviasi satu. Proses pelatihan dilakukan menggunakan algoritma *Ordinary Least Squares* (OLS) yang bertujuan untuk mencari parameter regresi terbaik dengan meminimalkan jumlah kuadrat selisih antara nilai aktual dan nilai prediksi. Target prediksi pada model ini adalah *gap\_twh*, yaitu selisih antara *Estimated TWh per Year* dan *Minimum TWh per Year*.

Tabel 4. 6 Evaluasi regresi linier

Model	Skala fitur	Algoritma	Target
LinearRegression (scikit-learn)	StandardScaler	Ordinary Least Squares (OLS)	gap_twh

Tabel 4.6 menunjukkan konfigurasi inti model *regresi linier* yang digunakan pada penelitian ini. Model dibangun menggunakan Linear Regression dari pustaka *scikit-learn*, yang menerapkan algoritma *Ordinary Least Squares* (OLS) untuk menghitung koefisien regresi. Fitur input diskalakan menggunakan Standard Scaler, yang mengubah setiap variabel menjadi distribusi dengan rata-rata nol dan standar deviasi satu. Skema ini membantu menghindari dominasi fitur dengan skala besar terhadap pembentukan parameter model. Target prediksi adalah gap\_twh, yaitu selisih konsumsi energi dalam satuan TWh yang dihitung dari perbedaan antara *Estimated TWh per Year* dan *Minimum TWh per Year*.

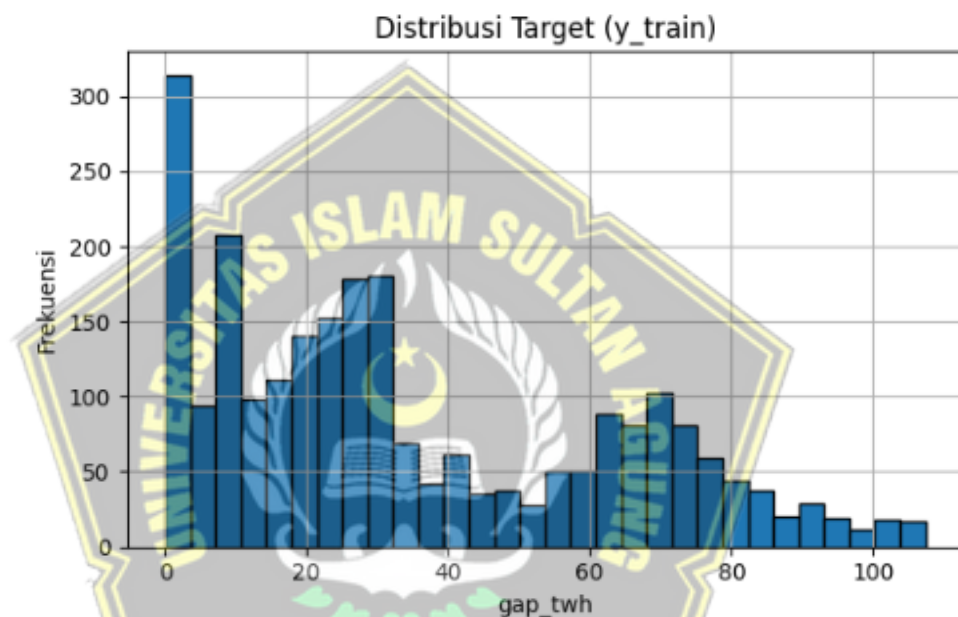
Tabel 4. 7 Daftar fitur

Name kolom	Jenis data	Jenis fitur
hash_rate	Numerik	Input
difficulty_est	Numerik	Input
ordinal_time	Numerik	Input
gap_twh	Numerik	Target

Tabel 4.7 menampilkan daftar fitur yang digunakan dalam penelitian beserta jenis data dan perannya dalam pemodelan. Seluruh fitur memiliki tipe data numerik, yang berarti nilainya dapat direpresentasikan dalam bentuk angka dan diolah secara langsung oleh model. Tiga fitur pertama, yaitu *hash\_rate*, *difficulty\_est*, dan *ordinal\_time*, berperan sebagai fitur input yang digunakan oleh model untuk mempelajari hubungan terhadap variabel target. Sementara itu, gap\_twh merupakan fitur target, yaitu nilai yang ingin diprediksi oleh model.

Fitur *hash\_rate* merepresentasikan tingkat kekuatan komputasi jaringan blockchain, *difficulty\_est* menunjukkan tingkat kesulitan penambangan, dan *ordinal\_time* adalah representasi urutan waktu secara

numerik. Sementara itu, variabel *gap\_twh* dihitung sebagai selisih antara *Estimated TWh per Year* dan *Minimum TWh per Year*, yang menggambarkan perbedaan konsumsi energi tahunan pada jaringan blockchain. Model regresi linier dibangun untuk memprediksi variabel target *gap\_twh* tersebut. Sebelum proses pelatihan model dilakukan, ditampilkan terlebih dahulu visualisasi distribusi dari data pelatihan (*y\_train*) untuk memahami karakteristik awal target yang akan diprediksi.



Gambar 4. 3 Distribusi *y\_train*

Pada Gambar 4.3 terlihat bahwa distribusi nilai *gap\_twh* cukup bervariasi dan tidak mengikuti distribusi normal sempurna, dengan konsentrasi data pada rentang 0 hingga 40 TWh serta beberapa outlier pada nilai yang lebih tinggi. Informasi ini penting karena regresi linier cenderung bekerja lebih optimal ketika distribusi data mendekati normal, sehingga adanya deviasi dari pola normal ini perlu diperhatikan pada tahap evaluasi hasil model.

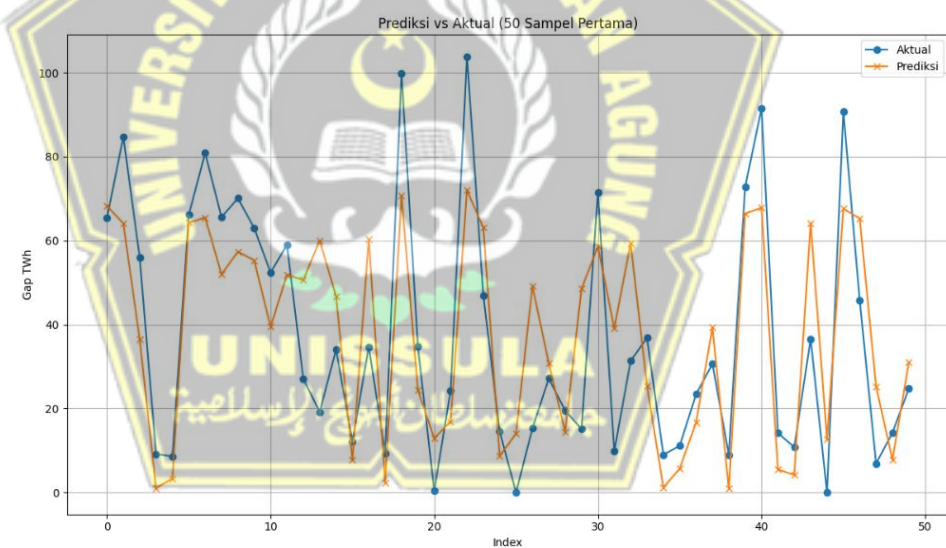
```
Intercept: 1.8274
          Fitur Koefisien
0  ordinal_time -134.187812
1    hash_rate   94.136008
2 difficulty_est  94.136008
```

Gambar 4. 4 Nilai *Intercept* dan Koefisien Regresi Linier



Gambar 4.4 menampilkan nilai *intercept* dan koefisien yang dihasilkan oleh model regresi linier setelah proses pelatihan. Model memperoleh nilai *intercept* sebesar 1.8274, dengan tiga koefisien utama masing-masing sebesar -134.187812, 94.136008, dan 94.136008. Ketiga koefisien ini merepresentasikan pengaruh dari fitur *ordinal\_time*, *hash\_rate*, dan *difficulty\_est* terhadap prediksi nilai *gap\_twh*.

Dari hasil ini, terlihat bahwa *hash\_rate* dan *difficulty\_est* memiliki koefisien positif, yang berarti peningkatan pada salah satu dari kedua fitur ini akan cenderung meningkatkan nilai prediksi *gap\_twh*. Sebaliknya, *ordinal\_time* memiliki koefisien negatif, yang mengindikasikan adanya kecenderungan penurunan *gap\_twh* seiring bertambahnya waktu. Pola ini dapat mencerminkan efek musiman atau penurunan konsumsi energi relatif terhadap waktu yang terekam dalam data historis.

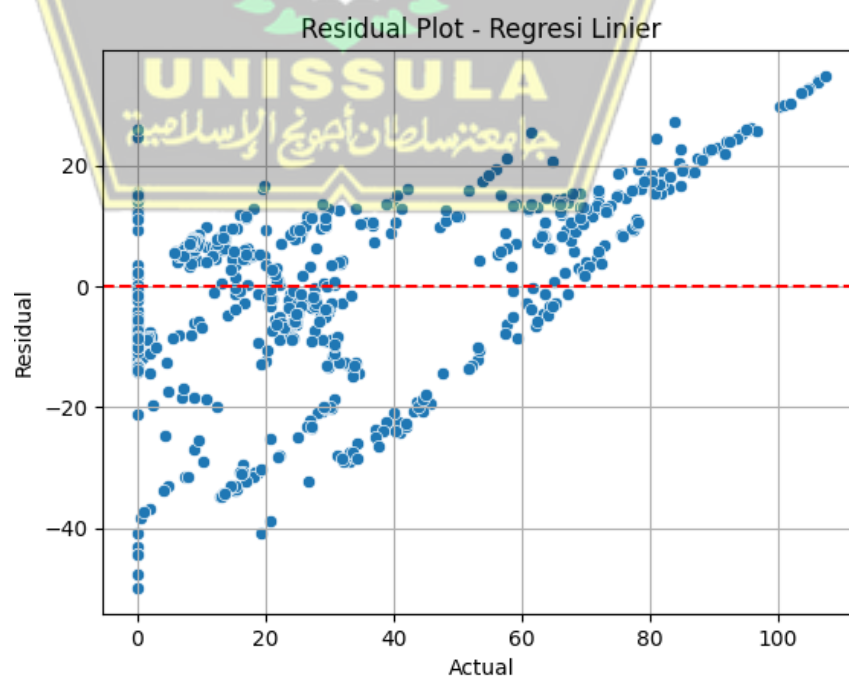


Gambar 4. 5 Prediksi vs aktual *gap\_twh* menggunakan *regresi linier*

Gambar 4.5 menampilkan hasil evaluasi model regresi linier melalui visualisasi Prediksi vs Aktual untuk variabel *gap\_twh*. Grafik ini menunjukkan perbandingan antara nilai aktual dan hasil prediksi model pada 50 sampel acak dari data uji. Secara umum, prediksi model mampu mengikuti pola fluktuasi nilai aktual dengan cukup baik, yang mengindikasikan bahwa model berhasil menangkap tren utama dalam data.

Namun demikian, terlihat adanya beberapa perbedaan yang cukup mencolok pada titik-titik tertentu, terutama ketika nilai aktual berada pada kondisi ekstrem. Hal ini menunjukkan bahwa meskipun model cukup efektif dalam menggambarkan pola umum, masih terdapat keterbatasan dalam menangani variasi data yang bersifat ekstrem. Kondisi ini kemungkinan dipengaruhi oleh adanya *outlier* atau kompleksitas pola data yang tidak sepenuhnya dapat ditangkap oleh regresi linier. Oleh karena itu, meskipun *feature engineering* telah dilakukan, pemanfaatan model yang lebih kompleks misalnya model non-linear atau berbasis pembelajaran mesin yang dapat menjadi alternatif untuk meningkatkan akurasi pada kasus-kasus ekstrem.

Untuk menganalisis kualitas prediksi lebih dalam, dilakukan analisis residu. Residu adalah selisih antara nilai observasi aktual dengan nilai yang diprediksi oleh model ( $\text{Residu} = Y_{\text{aktual}} - Y_{\text{prediksi}}$ ). Sebuah model yang baik idealnya memiliki residu yang tersebar secara acak di sekitar garis nol, yang menandakan bahwa model tidak memiliki bias sistematis dalam prediksinya. Berikut adalah plot residu untuk model Regresi Linier



Gambar 4. 6 Residu plot

Gambar 4.6 menampilkan residu plot, yaitu grafik selisih antara nilai aktual dan prediksi terhadap nilai aktual itu sendiri. Hasil plot menunjukkan bahwa residu tersebar relatif merata di sekitar garis horizontal nol, meskipun terdapat beberapa pola *cluster* residu negatif pada nilai rendah. Ini menunjukkan bahwa meskipun model tidak sepenuhnya bias terhadap nilai tertentu, masih terdapat *error* sistematis yang mungkin dapat diperbaiki dengan model non-linier seperti MLP.

Secara keseluruhan, hasil pelatihan model regresi linier menunjukkan bahwa model dapat menjelaskan sebagian besar variasi dalam data, namun memiliki keterbatasan dalam menangkap kompleksitas hubungan non-linier yang mungkin ada antara fitur teknis *blockchain* dan konsumsi energi. Oleh karena itu, pada bagian selanjutnya akan dibahas pendekatan non-linier menggunakan model MLP sebagai pembandingan.

#### 4.1.3.2 Pembangunan dan Pelatihan Model Multilayer Perceptron (MLP)

Untuk mengatasi keterbatasan model *Regresi Linier* dalam menangkap pola non-linier, digunakan pendekatan *Multilayer Perceptron* (MLP) yang dirancang untuk memodelkan hubungan kompleks antara fitur input dan target *gap\_twh*. MLP merupakan jenis *Artificial neural Network* (ANN) yang mampu belajar dari data dengan struktur yang tidak linier, sehingga lebih fleksibel dalam menangkap dinamika konsumsi energi blockchain.

Setelah data siap digunakan, langkah selanjutnya adalah membangun arsitektur model *Multilayer Perceptron* (MLP) untuk melakukan prediksi konsumsi energi *bitcoin*. Model ini dirancang menggunakan library Keras dari TensorFlow.

Tabel 4. 8 Arsitektur *Multilayer perceptron* (MLP)

Komponen	Jumlah Neuron	Fungsi Aktivitas	Keterangan
Input Layer	3	None	ordinal_time, hash_rate, difficulty_est
Hidden	256	ReLU	Menangkap pola non-

Layer 1			linier awal dengan menerima 3 fitur input
Hidden Layer 2	128	ReLU	Menangkap pola non-linier lanjutan
Hidden Layer 3	64	ReLU	Memperdalam representasi fitur
Hidden Layer 4	32	ReLU	Abstraksi akhir sebelum prediksi
Output Layer	1	Linear	Memprediksi nilai kontinu gap_twh

Pada Tabel 4.8, arsitektur model *Multilayer Perceptron* (MLP) terdiri dari enam lapisan utama yang dirancang untuk menangkap kompleksitas hubungan antar fitur dalam data konsumsi energi *blockchain*. Lapisan pertama adalah *Input Layer* yang terdiri dari tiga neuron, masing-masing merepresentasikan fitur *ordinal\_time*, *hash\_rate*, dan *difficulty\_est*. Karena lapisan ini hanya berfungsi sebagai penerima data tanpa melakukan transformasi, maka tidak digunakan fungsi aktivasi (*None*).

Selanjutnya, Hidden Layer 1 memiliki 256 neuron dengan fungsi aktivasi *ReLU*. Lapisan ini berperan dalam menangkap pola non-linear awal dari ketiga fitur input, memperluas kapasitas representasi model sejak tahap awal. Hidden Layer 2 terdiri dari 128 neuron dengan aktivasi *ReLU*, yang memperkuat kemampuan model dalam mengenali pola lanjutan yang lebih kompleks. Hidden Layer 3 dan Hidden Layer 4, masing-masing memiliki 64 dan 32 neuron dengan aktivasi *ReLU*, berfungsi untuk memperdalam abstraksi fitur dan menyaring informasi yang relevan sebelum proses prediksi.

Terakhir, Output Layer menggunakan satu neuron dengan fungsi aktivasi linear, yang sesuai untuk tugas regresi karena menghasilkan *output* berupa nilai kontinu. *Output* yang dihasilkan adalah prediksi gap\_twh, yaitu selisih konsumsi energi *blockchain* dalam satuan *terawatt-hour*. Struktur arsitektur ini dirancang untuk menggabungkan kapasitas representasi yang

dalam dengan efisiensi komputasi, sehingga model dapat memberikan prediksi yang akurat dan relevan terhadap konteks data yang digunakan.

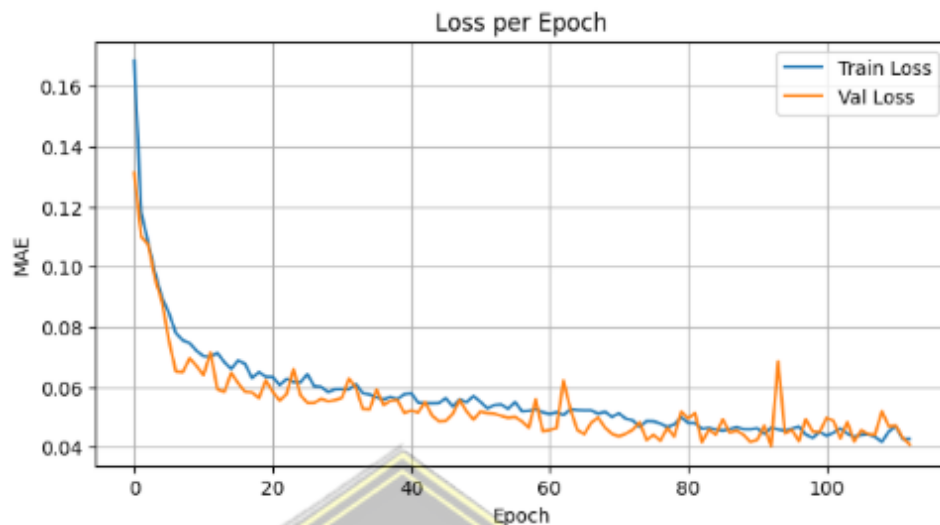
Tabel 4. 9 *Hyperparameter*

Parameter	Opsi Diuji	Konfigurasi
Hidden layer sizes	(50), (100),	(100, 50)
Activation	Relu, Tanh	Relu
Solver	Adam, Sgd	Adam
Alpa	0.0001, 0.001, 0.01	0,0005

Sebelum menetapkan arsitektur final, dilakukan penyetelan *hyperparameter* menggunakan *GridSearchCV* untuk menguji berbagai kombinasi parameter secara sistematis. Parameter yang diuji meliputi *hidden\_layer\_sizes*, fungsi activation *relu*, *tanh*, *solver*, *adam* dan *alpha* (parameter regularisasi). Berdasarkan hasil pencarian, konfigurasi terbaik yang diperoleh adalah *hidden\_layer\_sizes* = 100, 50, *activation* = *relu*, *solver* = *adam*, dan *alpha* = 0.0005. Kombinasi ini menunjukkan performa terbaik pada data validasi dan digunakan sebagai acuan dalam menetapkan arsitektur dan parameter pelatihan akhir model MLP.

Seluruh fitur *input* telah dinormalisasi menggunakan *MinMaxScaler* ke dalam rentang [0,1] sebelum dimasukkan ke dalam model. Langkah ini bertujuan untuk memastikan proses pelatihan berjalan stabil dan efisien, menghindari dominasi fitur dengan skala besar terhadap bobot model, serta mempercepat konvergensi algoritma *optimisasi*.

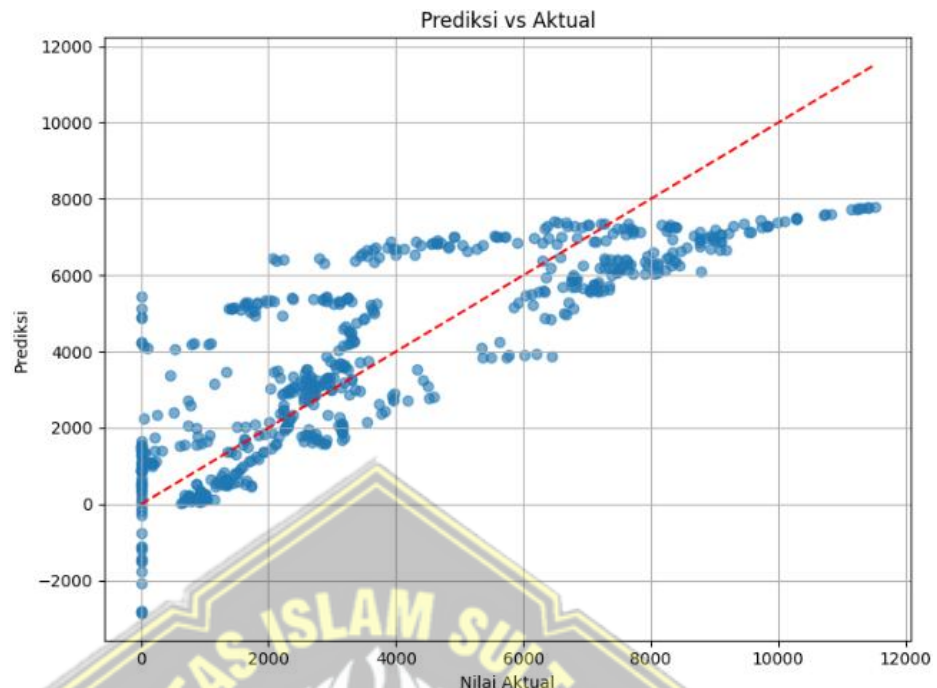




Gambar 4. 7 Grafik *training loss*

Gambar 4.7 memperlihatkan grafik *training loss* dan *validation loss* selama proses pelatihan model MLP. Pada fase awal pelatihan, kedua kurva mengalami penurunan yang cukup tajam, menandakan bahwa model dengan cepat mempelajari pola dasar hubungan antar fitur dan target. Setelah beberapa *epoch*, laju penurunan *loss* mulai melambat dan akhirnya mencapai kondisi relatif stabil (*plateau*), yang mengindikasikan bahwa model telah mendekati titik konvergensi dan tidak lagi memperoleh peningkatan signifikan dalam meminimalkan error.

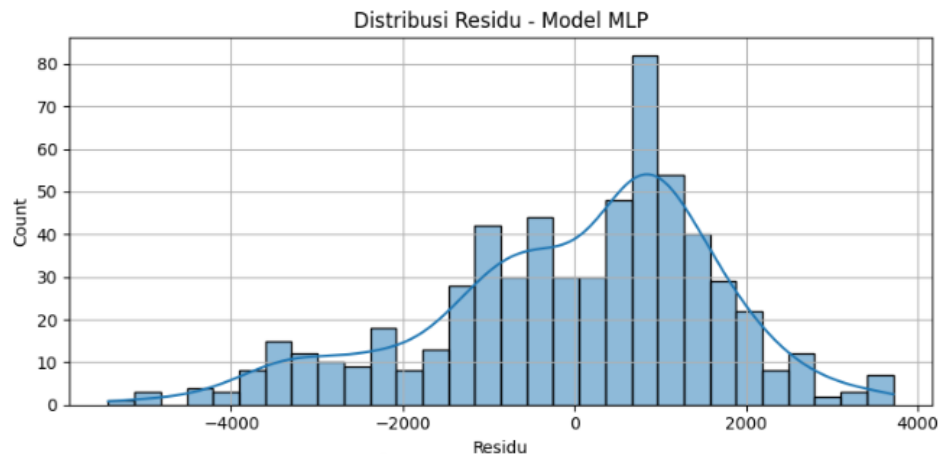
Kedua kurva *loss* menunjukkan tren yang serupa, dengan jarak yang relatif kecil antara *training loss* dan *validation loss*. Hal ini mengindikasikan bahwa model memiliki generalisasi yang baik terhadap data yang belum pernah dilihat sebelumnya, serta tidak menunjukkan gejala *overfitting* yang berarti. Stabilitas *validation loss* hingga akhir pelatihan mengonfirmasi bahwa penggunaan strategi *early stopping* berhasil mencegah model belajar terlalu lama pada pola spesifik data pelatihan yang tidak relevan bagi data uji.



Gambar 4. 8 Residual Plot

Gambar 4.8 menampilkan *Residual Plot* model MLP, yang memvisualisasikan selisih antara nilai aktual dan prediksi (residual) terhadap nilai aktual *gap\_twh*. Sebagian besar titik residu terkonsentrasi di sekitar garis nol, menandakan bahwa model tidak memiliki bias sistematis yang konsisten (misalnya selalu terlalu tinggi atau terlalu rendah).

Namun, terdapat pola penyebaran yang semakin melebar pada nilai *gap\_twh* yang lebih besar, yang menunjukkan heteroskedastisitas kesalahan prediksi cenderung meningkat pada nilai target yang tinggi. Selain itu, terdapat kelompok titik residu yang cukup padat di rentang nilai aktual rendah ( $< 2000$ ), yang mengindikasikan bahwa pada kisaran ini model lebih stabil dalam memprediksi. Sebaliknya, pada rentang nilai aktual tinggi ( $> 8000$ ), residu menjadi lebih variatif, yang bisa disebabkan oleh keterbatasan model dalam menangkap pola kompleks pada data ekstrem.



Gambar 4. 9 Distribusi Residu

Gambar 4.9 memperlihatkan distribusi residu model MLP dalam bentuk histogram dengan kurva *kernel density estimation* (KDE). Sebagian besar residu berada pada rentang -2000 hingga 2000, dengan puncak distribusi di sekitar nol, yang berarti sebagian besar prediksi cukup dekat dengan nilai aktual. Distribusi terlihat relatif simetris, namun memiliki *long tail* di kedua sisi, menandakan adanya sejumlah kecil kasus *outlier* dengan kesalahan prediksi besar, baik positif maupun negatif. Simetri distribusi ini mengindikasikan bahwa kesalahan tidak terfokus pada satu arah tertentu, sementara keberadaan *long tail* menunjukkan bahwa meskipun model memiliki kinerja baik pada mayoritas data, performanya menurun pada kasus-kasus ekstrem.

#### 4.1.4 Evaluasi Model

Setelah kedua model, yaitu *Regresi Linier* dan *Multilayer Perceptron* (MLP), berhasil dilatih, tahap selanjutnya adalah melakukan evaluasi untuk mengukur kinerja prediksi dari masing-masing model. Tujuan utama tahap ini adalah untuk mengukur seberapa dekat nilai prediksi yang dihasilkan oleh model dengan nilai aktual *gap\_twh*. Proses evaluasi dilakukan menggunakan data uji, yaitu data yang sama sekali belum pernah dilihat oleh model selama proses pelatihan. Dengan demikian, hasil evaluasi dapat memberikan gambaran yang objektif mengenai kemampuan generalisasi model pada data baru.

Dalam penelitian ini, metrik evaluasi yang digunakan adalah *Mean*

*Squared Error* (MSE), *Mean Absolute Error* (MAE), dan *R-squared* ( $R^2$ ). MSE mengukur rata-rata kuadrat selisih antara nilai aktual dan nilai prediksi, di mana nilai yang lebih kecil menunjukkan kinerja prediksi yang lebih baik. MAE mengukur rata-rata nilai absolut selisih antara prediksi dan nilai aktual, yang memberikan interpretasi langsung dalam satuan yang sama dengan target.  $R^2$  atau Koefisien Determinasi digunakan untuk mengukur seberapa baik model mampu menjelaskan variasi data aktual. Nilai  $R^2$  mendekati 1 menandakan model semakin baik dalam menangkap pola data.

#### 4.1.4.1 Hasil Evaluasi Model *Regresi Linier*

Tabel 4. 10 Hasil evaluasi model Regresi Linier

Model	MAE	MSE	$R^2$
<i>Regresi Linier</i>	12.3331	237.5541	0.6914

Berdasarkan hasil evaluasi, model *Regresi Linier* memiliki nilai MAE sebesar 12.3331 TWh, yang berarti rata-rata kesalahan prediksi sebesar  $\pm 12,33$  TWh. Nilai MSE sebesar 237.5541 TWh<sup>2</sup> menunjukkan adanya beberapa prediksi dengan kesalahan besar, sedangkan skor  $R^2$  sebesar 0.6914 mengindikasikan bahwa model hanya mampu menjelaskan sekitar 69,14% variasi pada data aktual. Hal ini mengindikasikan bahwa, meskipun Regresi Linier dapat menangkap sebagian besar tren umum pada data, kemampuannya untuk memodelkan pola non-linier masih terbatas.

#### 4.1.4.2 Hasil Evaluasi Model *Multilayer Perceptron* (MLP)

Tabel 4. 11 Hasil evaluasi model MLP

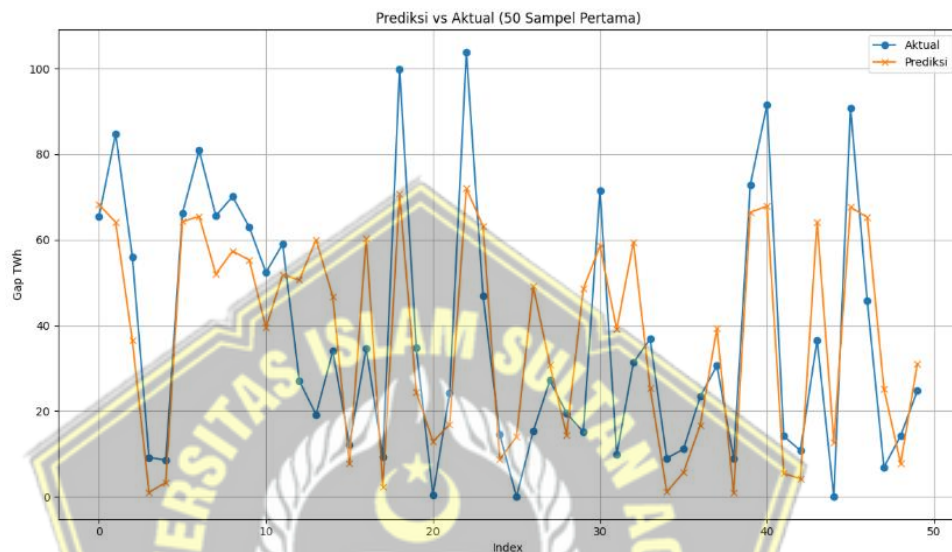
Model	MAE	MSE	$R^2$
<i>Multilayer Perceptron</i>	3.4620	46.9281	0.9390

Hasil evaluasi menunjukkan bahwa model *Multilayer Perceptron* (MLP) unggul secara signifikan dibandingkan *Regresi Linier*. Nilai MAE sebesar 3.4620 TWh menunjukkan rata-rata kesalahan prediksi yang jauh lebih rendah, sedangkan MSE sebesar 46.9281 TWh<sup>2</sup> mengindikasikan kesalahan kuadrat yang jauh lebih kecil dibandingkan regresi linier. Skor  $R^2$  yang tinggi, yaitu 0.9390, memperlihatkan bahwa model mampu menjelaskan sekitar 93,90% variasi pada data aktual. Hal ini membuktikan bahwa arsitektur *Artificial neural Network* (ANN) yang digunakan berhasil menangkap pola

non-linier pada data dengan sangat baik, sehingga meningkatkan akurasi dan ketepatan prediksi.

## 4.2 Visualisasi Hasil

### 4.2.1 Visualisasi hasil model *Regresi Linier*



Gambar 4. 10 Hasil grafik prediksi Regresi Linier

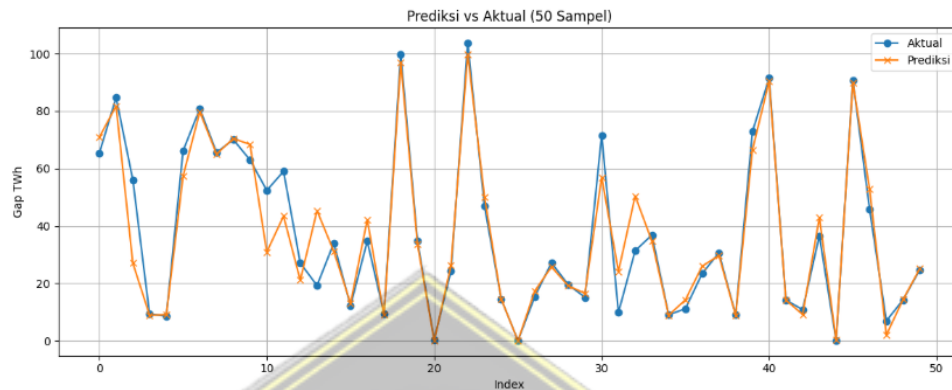
Gambar 4.10 menyajikan perbandingan visual antara nilai aktual *gap\_twh* (garis biru) dengan nilai yang diprediksi oleh model *Regresi Linier* (garis oranye) pada 50 sampel pertama dari data uji. Grafik ini memberikan gambaran awal mengenai kemampuan model dalam memetakan hubungan antara fitur *input* dan target konsumsi energi *blockchain*. Secara umum, garis prediksi mengikuti arah tren dari data aktual, meskipun terdapat beberapa deviasi yang menunjukkan keterbatasan model linier dalam menangkap fluktuasi yang lebih kompleks.

Pada beberapa titik, prediksi terlihat cukup akurat dan mendekati nilai aktual, namun pada titik-titik dengan perubahan tajam, model cenderung mengalami kesalahan prediksi yang lebih besar. Hal ini mengindikasikan bahwa model *Regresi Linier* memiliki keterbatasan dalam menangkap pola non-linier atau interaksi antar fitur yang bersifat dinamis. Meskipun demikian, visualisasi ini tetap menunjukkan bahwa model mampu memberikan estimasi awal yang cukup representatif, dan dapat digunakan



sebagai baseline untuk membandingkan performa dengan model yang lebih kompleks seperti MLP atau *Gradient Boosting*.

#### 4.2.2 Visualisasi hasil model *Multilayer Perceptron*



Gambar 4. 11 Hasil grafik prediksi Multilayer Perceptron

Gambar 4.11 menampilkan perbandingan visual antara nilai aktual *gap\_twh* (garis biru) dan nilai prediksi dari model *Multilayer Perceptron* (MLP) (garis oranye) pada 50 sampel pertama dari data uji. Grafik ini menunjukkan bahwa model MLP memiliki kemampuan yang sangat baik dalam mengikuti pola konsumsi energi *blockchain*, dengan garis prediksi yang bergerak sangat dekat dengan garis aktual. Pola fluktuasi yang kompleks, termasuk lonjakan dan penurunan tajam, berhasil ditangkap oleh model secara konsisten.

Kedekatan antara kedua garis ini mengindikasikan bahwa model MLP mampu mempelajari hubungan non-linier antara fitur *input* dan target dengan akurasi tinggi. Bahkan pada titik-titik ekstrem, seperti puncak dan lembah, model tetap menunjukkan respons yang adaptif, meskipun terdapat sedikit deviasi yang masih dalam batas wajar. Visualisasi ini secara efektif mengonfirmasi bahwa arsitektur MLP yang digunakan telah berhasil menggeneralisasi pola dalam data historis, sehingga mampu menghasilkan prediksi yang sangat mendekati nilai aktual.

### 4.3 Hasil Deployment

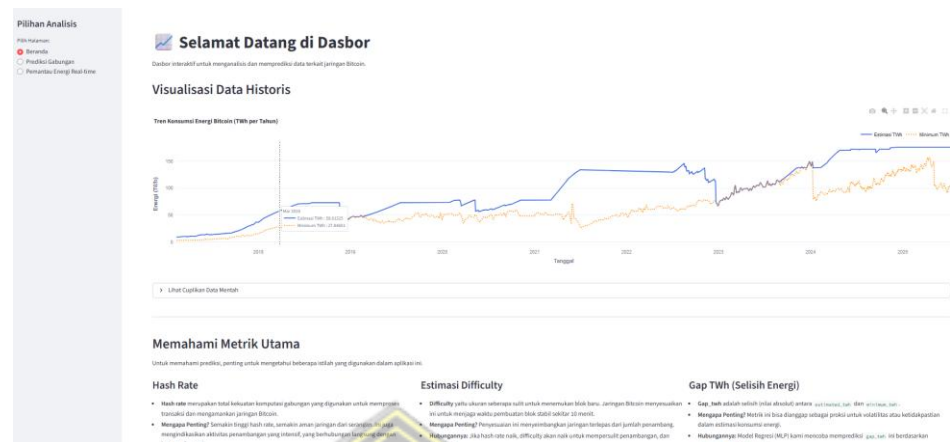
Sebagai tahap akhir dari implementasi, model prediksi *Multilayer Perceptron* (MLP) dan *Linear Regression* yang telah dilatih dan dievaluasi sebelumnya di-deploy ke dalam sebuah aplikasi web interaktif. Tujuan dari proses ini adalah untuk mentransformasikan model dari bentuk skrip analisis menjadi sistem fungsional yang dapat diakses dan digunakan secara praktis oleh pengguna akhir. *Deployment* dilakukan menggunakan *framework Streamlit*, sebuah pustaka Python yang dirancang khusus untuk membangun antarmuka web dalam konteks data sains secara cepat, ringan, dan efisien.

Langkah pertama dalam proses *deployment* adalah menyimpan kedua model final ke dalam file .pkl menggunakan pustaka *joblib*, sehingga model dapat dimuat ulang tanpa perlu pelatihan ulang. Setelah itu, dikembangkan sebuah skrip utama dengan nama *app.py* yang berfungsi sebagai inti dari antarmuka pengguna. Skrip ini mencakup logika untuk memuat model pilihan pengguna, melakukan *preprocessing* terhadap data input, menjalankan fungsi prediksi, dan menyajikan hasil prediksi dalam bentuk nilai numerik serta visualisasi grafik yang interaktif.

Antarmuka aplikasi dirancang agar pengguna dapat memilih model, mengunggah data input, melihat hasil prediksi, serta mengevaluasi performa model melalui metrik seperti MAE, MSE, dan  $R^2$ . Visualisasi yang ditampilkan mencakup grafik Prediksi vs Aktual, *Residual Plot*, dan distribusi error, yang semuanya disajikan secara dinamis menggunakan *plotly* agar dapat di-zoom dan di-hover.

Aplikasi dijalankan secara lokal melalui perintah terminal *streamlit run app.py*, yang secara otomatis membuka antarmuka web pada peramban melalui alamat *localhost*. Hasil dari proses *deployment* ini adalah sebuah Sistem Prediksi Konsumsi Energi Blockchain berbasis web yang tidak hanya mendemonstrasikan kemampuan teknis model, tetapi juga memberikan aksesibilitas dan kemudahan penggunaan bagi pengguna non-teknis.

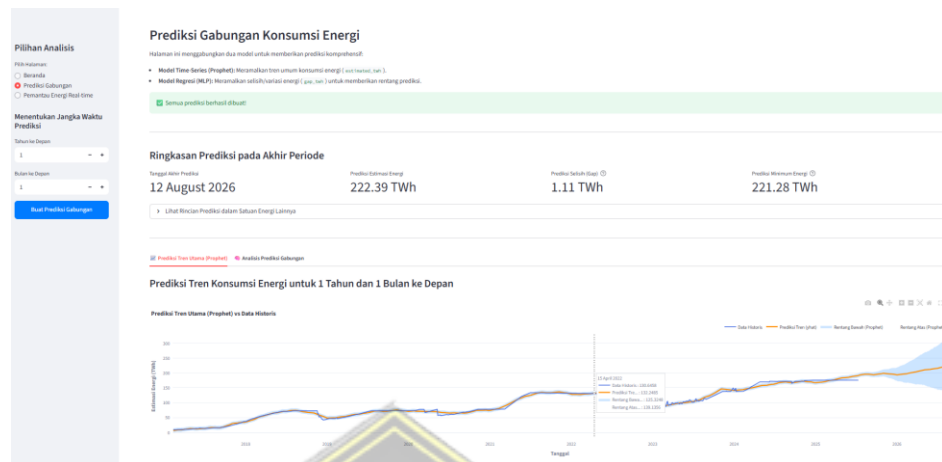
## 1. Halaman Awal



Gambar 4. 12 Halaman awal

Gambar 4.12 menampilkan halaman utama dari sistem analisis konsumsi energi *Bitcoin* yang telah berhasil *di-deploy* menjadi sebuah aplikasi web interaktif menggunakan framework Streamlit. Sistem ini dirancang dengan antarmuka dua panel untuk memastikan fungsionalitas yang terstruktur dan mudah digunakan. Pada bagian atas halaman, terdapat judul sistem yang memberikan identitas visual terhadap aplikasi. Di sisi kiri, panel navigasi bertajuk "Pilihan Analisis" berfungsi sebagai pusat interaksi utama, di mana pengguna dapat memilih halaman yang ingin ditampilkan, seperti Beranda, Prediksi Gabungan, atau Pemantauan Energi Real-time. Panel utama di sisi kanan menyajikan informasi secara komprehensif, dimulai dengan visualisasi data historis berupa grafik "Tren Konsumsi Energi Bitcoin (TWh per Tahun)" yang menampilkan kurva estimasi dan minimum konsumsi energi dari tahun 2017 hingga 2025. Di bawah grafik tersebut, terdapat bagian "Memahami Metrik Utama" yang menjelaskan konsep-konsep penting seperti *hash rate*, *Estimasi Difficulty*, dan *Gap TWh*. Penjelasan ini bertujuan untuk memberikan pemahaman awal kepada pengguna mengenai indikator teknis yang digunakan dalam sistem prediksi, sehingga pengguna dapat menavigasi dan menginterpretasi hasil analisis dengan lebih baik.

## 2. Halaman hasil prediksi



Gambar 4. 13 Hasil dari prediksi



Gambar 4. 14 Grafik hasil prediksi

Gambar 4.13 dan Gambar 4.14 menampilkan fitur prediksi gabungan konsumsi energi *blockchain* yang diimplementasikan dalam sistem web interaktif berbasis *Streamlit*. Kedua gambar ini merepresentasikan hasil integrasi dua pendekatan model, yaitu *Prophet* untuk proyeksi tren jangka panjang dan MLP untuk estimasi selisih konsumsi energi (*gap\_twh*). Pada Gambar 4.13, pengguna diberikan fleksibilitas untuk menentukan periode prediksi melalui *input* tahun dan bulan, yang kemudian digunakan untuk menghasilkan estimasi konsumsi energi pada tanggal akhir periode yang dipilih. Hasil prediksi ditampilkan dalam bentuk nilai estimasi, selisih, dan minimum energi, lengkap dengan konversi ke satuan Wh, kWh, dan MWh. Notifikasi keberhasilan dan visualisasi prediksi disajikan secara interaktif, memperkuat pengalaman pengguna dalam memahami hasil analisis.

Sementara itu, Gambar 4.14 memperluas fitur prediksi dengan menyajikan grafik tren konsumsi energi untuk periode 1 tahun dan 1 bulan ke depan. Grafik utama berjudul *Prediksi Tren Utama vs Data Historis*

menampilkan kurva konsumsi energi dari tahun 2017 hingga 2026, dengan garis biru sebagai data historis dan garis oranye sebagai prediksi tren harian. Rentang prediksi ditampilkan sebagai bayangan berwarna biru muda, yang mengindikasikan batas atas dan bawah estimasi model. Di bawah grafik tersebut, sistem menyajikan analisis komponen tren yang mencakup pola tahunan, mingguan, dan harian, memungkinkan pengguna untuk mengidentifikasi siklus musiman dan fluktuasi jangka pendek yang relevan terhadap dinamika konsumsi energi.

Kedua fitur ini menunjukkan bahwa sistem prediksi gabungan tidak hanya mampu menghasilkan estimasi kuantitatif yang presisi, tetapi juga menyajikan konteks temporal yang kaya melalui visualisasi tren dan dekomposisi komponen. Integrasi antara model statistik dan antarmuka pengguna yang interaktif menjadikan sistem ini sebagai alat analisis yang komprehensif, fleksibel, dan siap digunakan untuk eksplorasi data maupun pengambilan keputusan berbasis prediksi.

### 3. Hasil prediksi gabungan



Gambar 4. 15 Hasil Prediksi Gabungan

Gambar 4.15 menyajikan hasil prediksi gabungan konsumsi energi blockchain untuk periode Juli 2025, yang dihasilkan melalui integrasi dua pendekatan model, yaitu Prophet dan MLP. Grafik utama berjudul *Hasil Prediksi Gabungan* menampilkan kurva estimasi konsumsi energi dari tahun 2017 hingga 2026. Garis biru menunjukkan data historis, sementara



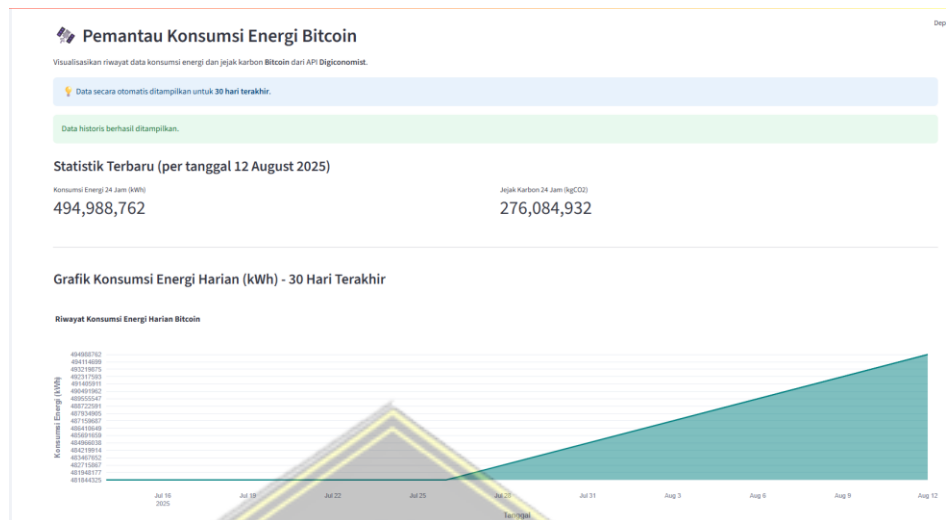
garis oranye merepresentasikan prediksi tren utama yang dihasilkan oleh model *Prophet*. Di sekeliling garis prediksi, terdapat bayangan berwarna oranye muda yang menunjukkan rentang prediksi atas dan bawah yang dihitung berdasarkan output model MLP, memberikan visualisasi terhadap ketidakpastian dan fluktuasi prediksi.

Grafik ini menunjukkan bahwa sistem mampu menggabungkan proyeksi jangka panjang dengan estimasi granular harian, sehingga menghasilkan prediksi yang tidak hanya akurat secara statistik, tetapi juga informatif secara operasional. Rentang prediksi yang sempit dan konsisten mengindikasikan stabilitas model dalam menangkap pola konsumsi energi yang berkelanjutan.

Di bawah grafik, ditampilkan tabel prediksi selisih energi (*gap\_twh*) untuk tanggal 13 hingga 22 Juli 2025. Tabel ini menyajikan nilai selisih dalam berbagai satuan energi, mulai dari TWh hingga Wh, yang memungkinkan pengguna untuk melakukan interpretasi dan konversi sesuai kebutuhan analisis. Nilai selisih yang ditampilkan berkisar antara 0.1678 TWh hingga 0.1573 TWh, menunjukkan penurunan bertahap yang konsisten selama periode tersebut. Konversi ke satuan MWh, kWh, dan Wh dilakukan secara otomatis oleh sistem, memperkuat fleksibilitas dan aksesibilitas informasi.

Visualisasi dan tabel ini secara keseluruhan menunjukkan bahwa sistem prediksi gabungan mampu memberikan estimasi yang presisi dan dapat diakses dalam berbagai skala, menjadikannya alat yang relevan untuk analisis teknis, monitoring energi, maupun pengambilan keputusan berbasis data.

#### 4. Data dari API



Gambar 4. 16 Data dari API

Gambar 4.16 menampilkan fitur ketiga dari sistem, yaitu halaman Pemantauan Konsumsi Energi Bitcoin secara Real-time yang terintegrasi dengan API eksternal dari *Digiconomist*. Fitur ini dirancang untuk menyajikan data konsumsi energi dan jejak karbon jaringan Bitcoin secara aktual, dengan pembaruan otomatis setiap hari. Pada bagian atas halaman, sistem menampilkan notifikasi bahwa data yang ditampilkan mencakup 30 hari terakhir, serta konfirmasi bahwa data historis berhasil dimuat.

Panel utama menyajikan statistik terkini per tanggal 12 Agustus 2025, yaitu konsumsi energi selama 24 jam sebesar 494.988.762 kWh dan jejak karbon sebesar 276.084.932 kg CO<sub>2</sub>. Nilai-nilai ini diambil langsung dari *endpoint* API dan ditampilkan secara dinamis di antarmuka pengguna. Di bawah statistik tersebut, terdapat grafik Grafik Konsumsi Energi Harian (kWh) - 30 Hari Terakhir yang menunjukkan tren konsumsi energi dari tanggal 16 Juli hingga 12 Agustus 2025. Grafik ini memperlihatkan pola kenaikan bertahap, yang mengindikasikan peningkatan aktivitas penambangan dan intensitas komputasi jaringan *Bitcoin* selama periode tersebut. Antarmuka sistem dirancang dengan dua panel: panel navigasi di sisi kiri yang memungkinkan pengguna berpindah antar fitur, dan panel utama di sisi kanan yang menyajikan data dan visualisasi.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Dari hasil penelitian ini dapat memberikan beberapa kesimpulan yaitu:

1. Kedua model, *Regresi Linier* dan *Multilayer Perceptron* (MLP), berhasil diimplementasikan untuk memprediksi konsumsi energi pada jaringan *blockchain* berbasis algoritma *Proof of Work* (PoW). Namun, hasil menunjukkan bahwa model *Regresi Linier* memiliki keterbatasan dalam menangkap pola non-linier yang kompleks pada data konsumsi energi.
2. Model MLP terbukti memberikan hasil prediksi yang jauh lebih akurat dengan nilai  $R^2$  sebesar 0,9390 dibandingkan *Regresi Linier* sebesar 0,6914. Hal ini menunjukkan bahwa pendekatan berbasis *deep learning* lebih mampu merepresentasikan fluktuasi dan dinamika konsumsi energi yang terjadi pada sistem PoW secara lebih realistis.
3. Penelitian ini memberikan kontribusi praktis bagi pelaku industri kripto dan investor ritel, khususnya dalam mengantisipasi risiko kerugian akibat siklus *Bitcoin Halving*. Model MLP yang dikembangkan dapat menjadi alat bantu dalam merencanakan strategi efisiensi energi dan memprediksi kebutuhan daya secara akurat untuk menjaga profitabilitas di tengah peningkatan biaya operasional *mining*.

#### 5.2 Saran

1. Perluasan Variabel Input

Penelitian ini menggunakan tiga variabel utama sebagai *input* model, yaitu *ordinal time*, *hash rate*, dan *difficulty*. Untuk meningkatkan akurasi dan kedalaman analisis, diharapkan penelitian selanjutnya mempertimbangkan penambahan variabel eksternal seperti harga Bitcoin, volume transaksi, emisi karbon, atau data geografis lokasi *mining pool* yang dapat memengaruhi konsumsi energi secara signifikan.

2. Eksplorasi Arsitektur Model Lain

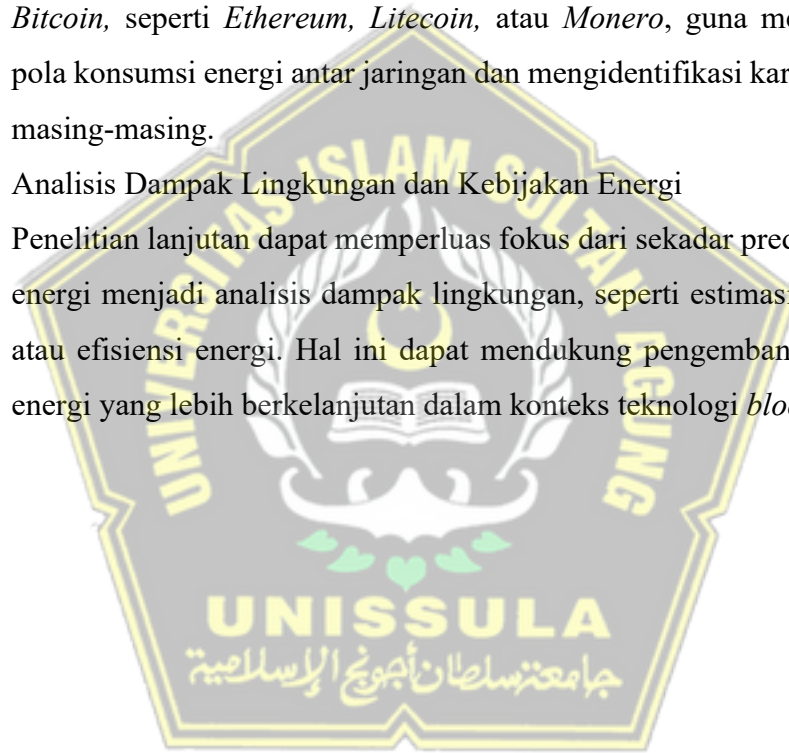
Meskipun *Multilayer Perceptron* (MLP) menunjukkan performa yang baik, pendekatan lain seperti Long Short-Term Memory (LSTM), *Convolutional Neural Network* (CNN), atau *Transformer-based models* dapat dieksplorasi untuk menangkap pola temporal dan spasial yang lebih kompleks dalam data konsumsi energi *blockchain*..

3. Validasi dengan Data *Multiplatform*

Untuk meningkatkan generalisasi model, disarankan agar penelitian selanjutnya menggunakan data dari berbagai platform *blockchain* selain *Bitcoin*, seperti *Ethereum*, *Litecoin*, atau *Monero*, guna membandingkan pola konsumsi energi antar jaringan dan mengidentifikasi karakteristik unik masing-masing.

4. Analisis Dampak Lingkungan dan Kebijakan Energi

Penelitian lanjutan dapat memperluas fokus dari sekadar prediksi konsumsi energi menjadi analisis dampak lingkungan, seperti estimasi emisi karbon atau efisiensi energi. Hal ini dapat mendukung pengembangan kebijakan energi yang lebih berkelanjutan dalam konteks teknologi *blockchain*.



## DAFTAR PUSTAKA

- Afrian, F., Akbar, N. I., Khalifatun, S., Latifah, N., Mutiara, D., Informatika, J., Teknik, F., Semarang, U. P., Lantai, G. B., Jl, K., & Timur, S. (2024). ANALISIS SISTEMATIK ALGORITMA KONSENSUS PADA BLOCKCHAIN pendekatan yang merujuk pada metodologi riset yang dirancang untuk mengidentifikasi ., 9(Sens 9), 519–527.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Dalam *Journal of Big Data* (Vol. 8, Nomor 1). Springer International Publishing. <https://doi.org/10.1186/s40537-021-00444-8>
- de Vries, A. (2020). Bitcoin's energy consumption is underestimated: A market dynamics approach. *Energy Research and Social Science*, 70(February), 101721. <https://doi.org/10.1016/j.erss.2020.101721>
- Gulo, S. H., & Lubis, A. H. (2024). Penerapan Multi-Layer Perceptron untuk Mengklasifikasi Penduduk Kurang Mampu. *Explorer*, 4(2), 51–59.
- Kaur, H., Rani, V., Kumar, M., Sachdeva, M., Mittal, A., & Kumar, K. (2024). Federated learning: a comprehensive review of recent advances and applications. *Multimedia Tools and Applications*, 83(18), 54165–54188. <https://doi.org/10.1007/s11042-023-17737-0>
- Lee, S., & Kim, S. (2020). Proof-of-stake at stake: Predatory, destructive attack on PoS cryptocurrencies. *CRYBLOCK 2020 - Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, Part of *MobiCom 2020*, 7–11. <https://doi.org/10.1145/3410699.3413791>
- Maarif, M. R., Saleh, A. R., Habibi, M., Fitriyani, N. L., & Syafrudin, M. (2023). Energy Usage Forecasting Model Based on Long Short-Term Memory (LSTM) and eXplainable Artificial Intelligence (XAI). *Information (Switzerland)*, 14(5). <https://doi.org/10.3390/info14050265>
- Maulana, M. R. (2024). BITCOIN DAN KONSEP UANG DIGITAL : TINJAUAN HISTORIS DAN TEORITIS. 1(2), 69–79.
- Nanda Sari, A., & Gelar, T. (2024). Blockchain: Teknologi Dan Implementasinya. *Jurnal Mnemonic*, 7(1), 63–70. <https://doi.org/10.36040/mnemonic.v7i1.6961>
- Paul, P. K. (2021). Blockchain Technology and its Types—A Short Review. *International Journal of Applied Science and Engineering*, 9(2), 1–13. <https://doi.org/10.30954/2322-0465.2.2021.7>
- Pavlov, R., Zarutskaya, O., Pavlova, T., Grynko, T., Levkovich, O., & Hordieieva-Herasymova, L. (2024). Blockchain As a Management Technology: Institutionalization of Crypto-Assets and Transformation of Entrepreneurial Models Using the Example of Ethereum. *Financial and Credit Activity: Problems of Theory and Practice*, 6(59), 151–166. <https://doi.org/10.55643/fcaptop.6.59.2024.4529>
- Pawar, M. K., Patil, P., Narayan, D. G., Pandey, V., Jain, S., & Kumar, P. (2025). Efficient Miner Selection in Blockchain Based on Predicted Transaction Time. *Procedia Computer Science*, 252, 202–211. <https://doi.org/10.1016/j.procs.2024.12.022>



- Poongodi, M., Sharma, A., Vijayakumar, V., Bhardwaj, V., Sharma, A. P., Iqbal, R., & Kumar, R. (2020). Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system. *Computers and Electrical Engineering*, 81, 106527. <https://doi.org/10.1016/j.compeleceng.2019.106527>
- Rachmawati, B. D., & Aini, Q. (2018). Estimasi Parameter Regresi Linear Menggunakan Regresi Kuantil. *Eigen Mathematics Journal*, 37–42. <https://doi.org/10.29303/emj.v2i2.15>
- Reyad, M., Sarhan, A. M., & Arafa, M. (2023). A modified Adam algorithm for deep neural network optimization. *Neural Computing and Applications*, 35(23), 17095–17112. <https://doi.org/10.1007/s00521-023-08568-z>
- Rukhiran, M., Boonsong, S., & Netinant, P. (2024). Sustainable Optimizing Performance and Energy Efficiency in Proof of Work Blockchain: A Multilinear Regression Approach. *Sustainability (Switzerland)*, 16(4). <https://doi.org/10.3390/su16041519>
- Sang, X., Leng, X., Xue, L., & Ran, X. (2022). Based on the Time-Spatial Power-Based Cryptocurrency Miner Driving Force Model, Establish a Global CO2 Emission Prediction Framework after China Bans Cryptocurrency. *Sustainability (Switzerland)*, 14(9). <https://doi.org/10.3390/su14095332>
- Sapra, N., Shaikh, I., & Dash, A. (2023). Impact of Proof of Work (PoW)-Based Blockchain Applications on the Environment: A Systematic Review and Research Agenda. *Journal of Risk and Financial Management*, 16(4). <https://doi.org/10.3390/jrfm16040218>
- Sarwar, M. I., Maghrabi, L. A., Khan, I., Naith, Q. H., & Nisar, K. (2023). Blockchain: A Crypto-Intensive Technology - A Comprehensive Review. *IEEE Access*, 11(December), 141926–141955. <https://doi.org/10.1109/ACCESS.2023.3342079>
- Truby, J., Brown, R. D., Dahdal, A., & Ibrahim, I. (2022). Blockchain, climate damage, and death: Policy interventions to reduce the carbon emissions, mortality, and net-zero implications of non-fungible tokens and Bitcoin. *Energy Research and Social Science*, 88(August 2021), 102499. <https://doi.org/10.1016/j.erss.2022.102499>
- Tsuyuguchi, T., & Wang, H. (2024). Quasi-experimental research and spillover effects on Ethereum Merge. *Applied Economics Letters*, DiD. <https://doi.org/10.1080/13504851.2024.2308587>
- Xiong, H., Chen, M., Wu, C., Zhao, Y., & Yi, W. (2022). Research on Progress of Blockchain Consensus Algorithm: A Review on Recent Progress of Blockchain Consensus Algorithms. *Future Internet*, 14(2). <https://doi.org/10.3390/fi14020047>
- Yadav, A. K., & Singh, K. (2021). Comparative analysis of consensus algorithms and issues in integration of blockchain with iot. *Advances in Intelligent Systems and Computing*, 1168(April), 25–46. [https://doi.org/10.1007/978-981-15-5345-5\\_3](https://doi.org/10.1007/978-981-15-5345-5_3)
- Yaish, A., & Zohar, A. (2023). Correct Cryptocurrency ASIC Pricing: Are Miners Overpaying? *Leibniz International Proceedings in Informatics, LIPIcs*, 282(2), 1–2. <https://doi.org/10.4230/LIPIcs.AFT.2023.2>

Zahedi, R., Aslani, A., Ali, M., & Seraji, N. (t.t.). Life cycle assessment of Bitcoin mining in the top ten miner countries. <https://doi.org/10.18282/i-es.v>

