BAB II
LANDASAN TEORI

2.1 Penelitian Terdahulu

Tabel 2.1 Tabel Penelitian Terdahulu

No	Judul	Penulis	Tahun	Jenis literasi	Kesimpulan
1	Comparative Analysis of Machine Learning and Deep Learning Models for Bitcoin Price Prediction	Omar Ahmed Al-Zakhali & Adnan M. Abdulazeez	2024	Journal Article	Penelitian menggunakan algoritma Random Forest, SVM, GRU, dan LSTM untuk prediksi harga Bitcoin. Hasil terbaik diperoleh oleh Random Forest dengan RMSE ≈ 0,012 dan MAPE ≈ 1,35%. Model GRU menghasilkan RMSE ≈ 0,018 dan MAPE ≈ 1,82%, LSTM memperoleh RMSE ≈ 0,021 dan MAPE ≈ 2,04%, sedangkan SVM memiliki performa terendah dengan RMSE ≈ 0,032 dan MAPE ≈ 3,12%.
2	Bitcoin Price Prediction Using Hybrid LSTM-GRU Models	Nashwan Hussein & Adnan M. Abdulazeez	2024	Journal Article	Penelitian menggunakan model Hybrid LSTM-GRU untuk prediksi harga Bitcoin dan dievaluasi dengan MAE, RMSE, dan MAPE. Hasil terbaik diperoleh model Hybrid LSTM-GRU dengan MAE = 0,0018, RMSE = 0,0031, dan MAPE = 0,52%. Model LSTM tunggal menghasilkan MAE = 0,0024, RMSE = 0,0042, MAPE = 0,68%, sedangkan GRU tunggal memperoleh MAE = 0,0021, RMSE = 0,0038, MAPE = 0,61%.
3	Comparative Performance of Machine Learning Ensemble Algorithms for	V. Derbentsev,V. Babenko,K. Khrustalev,H. Obruch, S.Khrustalova	2021	Journal	Penelitian menggunakan algoritma SGBM dan Random Forest untuk menganalisis harga BTC, ETH, dan XRP dengan metrik MAPE dan RMSE. Pada Bitcoin, SGBM menghasilkan

	Forecasting Cryptocurrency Prices				RMSE = 263.34 dan MAPE = 2.31, sedangkan Random Forest menghasilkan RMSE = 305.95 dan MAPE = 2.61.
4	Short-term bitcoin market prediction via machine learning	PatrickJaquart,David Dann,ChristofWeinhardt	2021	Journal	Penelitian menggunakan LSTM dan GRU untuk prediksi harga kripto dengan optimisasi hyperparameter. Model dievaluasi memakai MAE, RMSE, MAPE, dan hasil terbaik diperoleh LSTM pada Bitcoin dengan MAE = 0.0047, RMSE = 0.0068, MAPE = 0.012, mengungguli GRU.
5	A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms	Mohammad J. Hamaye l, Amani Yousef Owda	2021	Journal Article	Penelitian menggunakan GRU, LSTM, dan bi-LSTM untuk prediksi harga Bitcoin dengan pembagian data 80% pelatihan dan 20% pengujian, dievaluasi menggunakan MAPE dan RMSE. Hasil terbaik diperoleh GRU dengan RMSE = 174.13 dan MAPE = 0.2454%, sedangkan LSTM menghasilkan RMSE = 410.40, MAPE = 1.1234%, dan bi-LSTM menjadi terendah dengan RMSE = 2927.01, MAPE = 5.990%.
6	Deep learning for Bitcoin price direction prediction: models and trading strategies empirically compared	Oluwadamilare Omole & David Enke	2024	Journal	Penelitian memakai CNN–LSTM, LSTNet, TCN, dan ARIMA dengan 87 metrik on-chain serta feature selection Boruta, GA, LightGBM, dan tuning random search. Kombinasi Boruta + CNN–LSTM menjadi terbaik dengan akurasi 82.44%, precision 0.8309, recall 0.8078, F1-score 0.8192, dan backtesting menghasilkan return tahunan 6653%.

7	Comparative Performance of LSTM and ARIMA for the Short-Term Prediction of Bitcoin Prices	Navmeen Latif, Joseph Durai Selvam, Manohar Kapse, Vinod Sharma, dan Vaishali Mahajan	2023	Journal Article	Penelitian membandingkan ARIMA (3,1,3) dan LSTM untuk prediksi harga Bitcoin menggunakan data penutupan 21 Desember 2020–21 Desember 2021 dengan interval 10 menit, dibagi 99.5% latih dan 0.5% uji. ARIMA menggunakan p=3, d=1, q=3, sedangkan LSTM menggunakan 200 historical prices, batch size 1, dan epoch 1. Evaluasi memakai MAE, MAPE, RMSE, dan Accuracy. Hasil menunjukkan LSTM unggul dengan akurasi 99.73%, RMSE 151.95, MAPE 0.27%, MAE 126.97, sementara ARIMA (3,1,3) hanya mencapai akurasi 98.21%, RMSE 940.40, MAPE 1.79%, MAE 837.77.
8	Bitcoin price prediction using LSTM, GRU and hybrid LSTM-GRU with bayesian optimization, random search, and grid search for the next days.	I. Sibel Kervancı, M. Fatih Akay, & Eren Özceylan	2024	Journal	Penelitian menggunakan LSTM, GRU, dan Hybrid LSTM-GRU dengan optimisasi Grid Search, Random Search, dan Bayesian Optimization (BO) untuk prediksi harga Bitcoin per jam. Dari ketiga metode, BO, khususnya BO-GP, memberikan hasil terbaik. Model BO-GP Hybrid LSTM-GRU mencapai performa tertinggi dengan RMSE = 0.003269, MSE = 0.000015, MAE = 0.002302, dan MAPE = 0.005497, serta mengungguli seluruh metode lain dalam akurasi prediksi jangka pendek.

9	Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach	Phumudzo Lloyd Seabe, Claude Rodrigue Bambe Moutsinga, dan Edson Pindza	2023	Jurnal Artikel	Penelitian membandingkan LSTM, GRU, dan Bi-LSTM untuk prediksi harga BTC, ETH, dan LTC dengan optimisasi hyperparameter seperti jumlah neuron, epoch, dan batch size = 120 sebagai nilai terbaik. Evaluasi menggunakan RMSE dan MAPE menunjukkan Bi-LSTM sebagai model terbaik. Pada Bitcoin, Bi-LSTM menghasilkan RMSE = 1029.36 dan MAPE = 0.0356, lebih baik dibandingkan LSTM (RMSE = 1031.34; MAPE = 0.0394) dan GRU (RMSE = 1274.17; MAPE = 0.0572). Hasil serupa pada ETH dan LTC menegaskan bahwa Bi-LSTM paling akurat dan memiliki error terendah di antara ketiga model.
10	Price Prediction of Bitcoin Based on Adaptive Feature Selection and Model Optimization	Yingjie Zhu, Jiageng Ma, Fangqing Gu, Jie Wang, Zhijuan Li, Youyao Zhang, Jiani Xu, Yifan Li, Yiwen Wang, Xiangqun Yang	2023	Journal	Model gabungan menggunakan TWSVR (Twin Support Vector Regression) dikombinasikan dengan optimisasi hyperparameter (WOA dan PSO) dan pemilihan fitur adaptif (XGBoost dan Random Forest) menghasilkan prediksi harga Bitcoin yang terbaik dengan EVS = 0,9547. TWSVR lebih cepat daripada SVR, dan memiliki akurasi yang lebih baik.

Hingga saat ini, belum terdapat kesepakatan yang jelas mengenai apakah LSTM atau GRU yang memberikan hasil lebih baik dalam memprediksi harga Bitcoin. Setiap penelitian cenderung menghasilkan temuan yang berbeda, sehingga sulit menarik kesimpulan model mana yang sebenarnya lebih unggul. Salah satu penyebabnya adalah karena sebagian besar studi terdahulu belum menerapkan proses optimasi hyperparameter yang terstruktur, padahal performa kedua model

tersebut sangat dipengaruhi oleh pengaturan parameter seperti jumlah unit, learning rate, nilai dropout, maupun ukuran batch. Selain itu, penggunaan Bayesian Optimization sebagai metode optimasi yang lebih efisien dan sistematis juga masih terbatas. Akibatnya, perbandingan kinerja antara LSTM dan GRU belum dilakukan secara benar-benar seimbang, sehingga membuka peluang penelitian baru untuk menguji kembali kedua model dengan prosedur optimasi yang lebih tepat.

Sejauh ini bukti empiris mengenai model mana yang benar-benar mampu menghasilkan tingkat kesalahan prediksi paling rendah masih belum konsisten, karena setiap studi menggunakan parameter berbeda tanpa proses optimasi yang seragam. Kebanyakan penelitian sebelumnya masih mengandalkan Grid Search atau Random Search, dua teknik yang cenderung memakan waktu dan sering kali tidak mampu menjelajahi ruang parameter secara maksimal. Di sisi lain, karakteristik Bitcoin yang sangat volatil menuntut model yang dapat mempelajari pola ketergantungan jangka panjang secara stabil. Namun hingga kini, belum ada kajian yang menegaskan apakah LSTM atau GRU lebih mampu mempertahankan stabilitas performa ketika diterapkan pada data dengan fluktuasi ekstrim tersebut.

Pendekatan yang digunakan dalam penelitian ini berbeda dengan sebagian besar studi serupa yang menetapkan nilai hyperparameter secara manual. Pada banyak penelitian, parameter seperti jumlah unit, laju pembelajaran, maupun tingkat dropout biasanya langsung ditentukan oleh peneliti berdasarkan rekomendasi umum atau praktik sebelumnya. Pola ini membuat hasil model berpotensi bias dan tidak dapat dipastikan berada pada konfigurasi terbaiknya. Dalam penelitian ini, seluruh parameter utama tidak ditetapkan sejak awal, tetapi dicarikan secara otomatis melalui Bayesian Optimization. Pendekatan ini membuat proses pemilihan parameter lebih objektif dan memungkinkan perbandingan LSTM dan GRU dilakukan dalam kondisi yang lebih adil.

Metode yang digunakan pada penelitian terdahulu yang berjudul *"Comparative Performance of LSTM and ARIMA for the Short-Term Prediction of Bitcoin*

Prices" menggunakan algoritma ARIMA dan LSTM dengan hasil akurasi LSTM lebih tinggi mencapai 99.73%, dan ARIMA 98.21%.

2.2 Teori yang berkaitan

2.2.1 Bitcoin

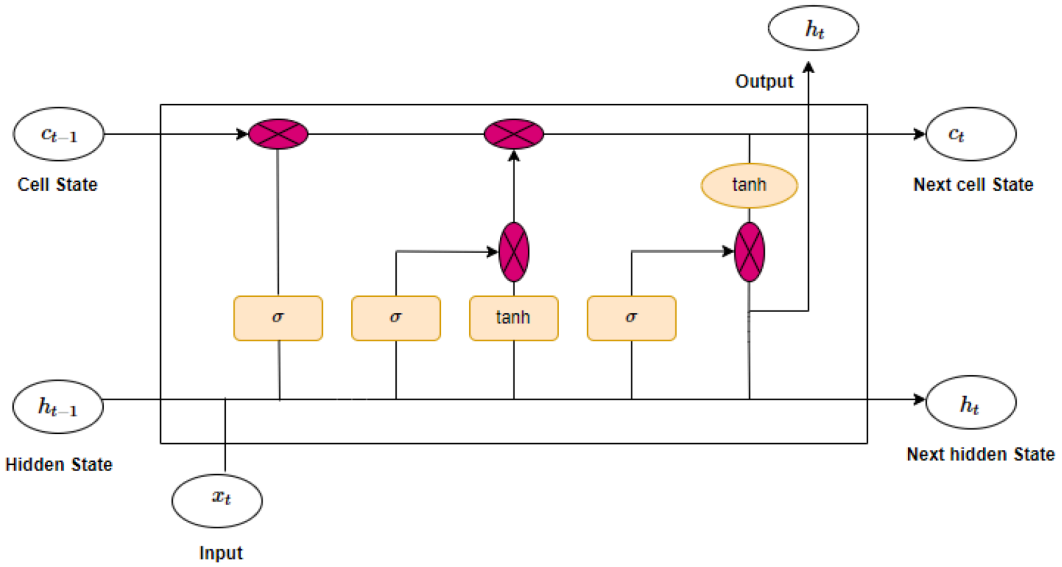
Bitcoin merupakan mata uang digital yang berdiri pada tahun 2008 oleh pendiri yang bernama Satoshi Nakamoto. Bitcoin sendiri merupakan aset *decentralized digital cryptocurrency* pertama yang tidak memiliki *central authority* dalam membuat uang baru dan dalam proses verifikasi transfer dalam sistem bitcoin [8]. Intensi dari pembuatan bitcoin adalah untuk menciptakan suatu transaksi yang bebas dari pihak sentral atau otoritas moneter, yang dibuat berbasis algoritma matematika daripada "pihak ketiga" [17]. Menurut data dari coinmarketcap.com pada Oktober 2025 bitcoin telah mencapai kapitalisasi pasar sekitar $2 *trillion*. Dan telah menjadi aset *cryptocurrency* terbesar dari sisi kapitalisasi pasar dibandingkan dengan *cryptocurrency* lainnya. Bitcoin memiliki batas ketersediaan dengan total 21 juta koin [18]. Pasokan bitcoin yang sudah ditetapkan sebanyak 21 juta koin ini berfungsi untuk penegasan ekonomi yang kuat dan membuat valuasi menjadi kerangka yang kokoh [18]. Bitcoin didistribusikan dengan proses yang bernama *mining* [18].. Penambangan untuk seluruh bitcoin yang berjumlah 21 juta koin diperkirakan akan sepenuhnya tertambang pada tahun 2140 [18].

2.3 Deep Learning

2.3.1 Long Short Term Memory (LSTM)

LSTM merupakan salah satu algoritma pembelajaran mesin yang berbasis jaringan syaraf buatan yang dirancang khusus untuk menangani data berurutan dengan memori jangka pendek yang terukur. Jaringan ini berfungsi untuk menangkap hubungan temporal jangka panjang secara efektif dan sering dianggap memiliki fleksibilitas yang tinggi. Algoritma ini memiliki arsitektur seperti gaya RNN dengan gerbang yang mengatur aliran informasi antar sel. Struktur input gate dan forget gate dapat menyesuaikan informasi yang diproses di sepanjang

status sel yang menghasilkan keluaran akhir berupa hasil yang dari sel berdasarkan konteks dari yang di input. Berikut merupakan ilustrasi algoritma LSTM. [5]



Gambar 2.1 Struktur dari LSTM

Berikut Rumus dari proses pelatihan LSTM:

$$ft = \sigma(Wf \cdot [ht - 1, xt] + bf)$$

$$it = \sigma(Wi \cdot [ht - 1, xt] + bi)$$

$$Ct = ft * Ct - 1 + it * tanh(Wc \cdot [ht - 1, xt] + bc)$$

$$Ot = \sigma(Wo \cdot [ht - 1, xt] + bo)$$

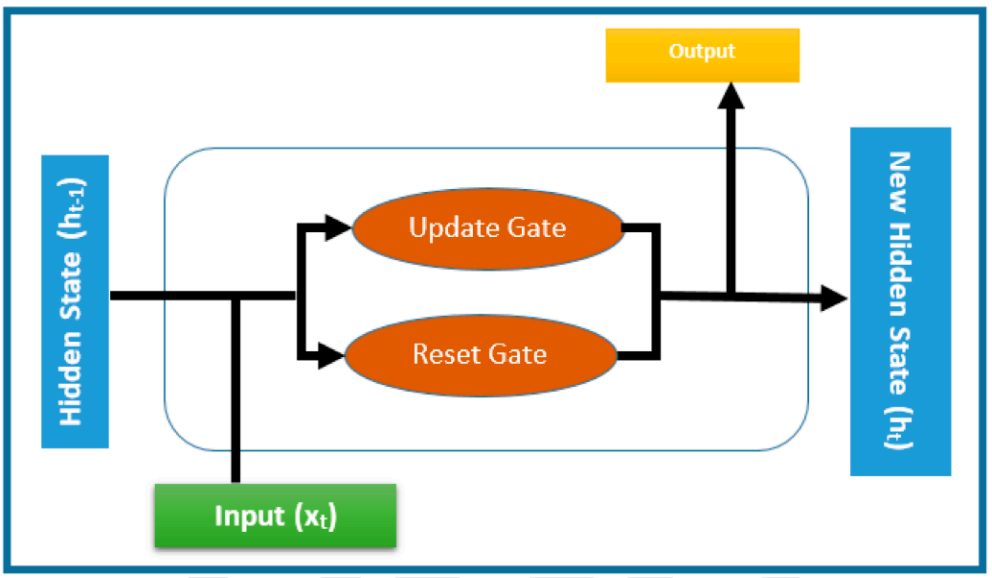
$$ht = ot * tanh(Ct)$$

LSTM memiliki kelebihan dalam menangkap hubungan jangka panjang dalam data yang bersifat sekuensial dengan lebih baik [19]. Selain itu LSTM memiliki struktur *gate* berjumlah tiga yaitu *input, forget, output gate* yang membuat kontrol dalam informasi yang lebih detail [19]. Namun memiliki kekurangan dalam waktu pelatihan yang lama karena struktur yang lebih kompleks [19]. Serta konsumsi

memori yang tinggi, sehingga kurang efisien untuk memproses data yang besar atau perangkat dengan hardware yang terbatas [19].

2.3.2 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) adalah algoritma yang dapat menunjukan efektivitasnya dalam berbagai aplikasi yang memerlukan data sekuensial. Berikut merupakan tampilan gambar yang menggambarkan fungsi transisi dalam unit GRU yang tersembunyi. [5]



Gambar 2.2 Struktur dari GRU

Berikut rumus fungsi transisi dalam unit tersembunyi dalam GRU:

$$zt = \sigma(Wzxt + Vzht - 1 + bz)$$

$$rt = \sigma(Wrxt + Vrht - 1 + br)$$

$$h\sim t = tanh(Wcxt + Vc(rt \cdot ht - 1))$$

$$ht = (1 - zt) \cdot ht - 1 + zt \cdot h\sim t$$

Parameter dari model mencangkup seluruh W $\epsilon RdXd$, V $\epsilon RdXd$, dan b $\epsilon RdXd$. Selama tahap pelatihan semua W $\epsilon RdXd$, V $\epsilon RdXd$, dan b $\epsilon RdXd$ dipelajari dan dimiliki bersama oleh semua langkah waktu. Produk per lemen ditunjukan (.), dan k menunjukan hiperparameter yang menunjukan dimensionalitas vektor tersembunyi seperti rumus diatas.

GRU memiliki kelebihan dalam struktur yang lebih sederhana karena memiliki dua *gate* saja yang berupa *update* dan *reset gate*, sehingga proses pelatihan lebih cepat [19]. Efisiensi terhadap komputasi lebih baik dan memiliki sumber daya lebih kecil dibandingkan dengan LSTM [19]. Namun GRU memiliki kekurangan dalam menangkap dependensi jangka panjang dibandingkan LSTM [19]. Selain itu dengan pola yang sederhana, hal ini mengakibatkan kurangnya fleksibilitas dalam mengontrol aliran informasi [19].

2.4 Matrik Evaluasi

2.4.1 Mean Absolute Error (MAE)

Metrik MAE berfungsi untuk mengukur rata-rata error yang absolut antara nilai aktual dan nilai prediksi [9]. Kelebihan matrix MAE adalah tidak menambahkan bobot apapun terhadap kesalahan besar yang mengakibatkan metrik ini lebih tahan terhadap outliers [9].

$$MAE = (1 / n) \Sigma\_(t = 1)\^n |A\_t - P\_t|$$

Nilai terbaik dari metrik MAE adalah 0 yang merupakan kesempurnaan sifat prediksi [9]. Semakin rendah nilai MAE maka nilai prediksi dianggap semakin baik [9].

2.4.2 Root Mean Squared Error (RMSE)

Metrik RMSE merupakan metrik yang paling banyak digunakan untuk mengevaluasi teknik regresi [9]. RMSE berfungsi untuk menghitung rata-rata kuadrat selisih antara nilai aktual dan nilai prediksi [9]. Kuadrat akan diberikan bobot lebih besar untuk kesalahan yang lebih besar atau berat [9]. Berbeda seperti MAE, RMSE merupakan metrik yang sangat sensitif terhadap outliers [9].

$$RMSE = \sqrt{((\Sigma\_(t = 1)\^n (A\_t - P\_t)\^2) / n)}$$

Nilai terbaik dari metrik RMSE adalah 0 yang merupakan kesempurnaan sifat prediksi [9]. Semakin rendah nilai RMSE maka nilai prediksi dianggap semakin baik [9].

2.4.3 R-Squared (R^2)

Metrik R2 dikenal sebagai coefficient of determination atau R-squared yang berfungsi untuk mengukur kesempurnaan proporsi variabilitas data target [9]. Kelebihan metrik ini adalah dapat mengukur prediksi secara menyeluruh yang memberikan wawasan seberapa baik pola model prediksi [9].

$$R^2 = 1 - (\Sigma\_(t = 1)\hat{}n\,(A\_t - P\_t)\hat{}2\,) / (\Sigma\_(t = 1)\hat{}n\,(A\_t - \bar{A})\hat{}2\,)$$

Nilai R-squared hanya dalam range 0 sampai 1, dengan penilaian semakin mendekati angka 1 maka menunjukan model dapat menjelaskan sebagian besar variasi dalam data yang menunjukan positif [9]. Nilai negatif dianggap model yang memiliki kinerja yang buruk [9].

2.4.4 Confusion Matrix

Confusion matrix adalah alat evaluasi yang digunakan untuk mengukur kinerja model klasifikasi [12]. Matriks ini berisikan kelas aktual dan kolom yang mewakili kelas hasil prediksi yang berbentuk persegi [12]. Confusion matrix berukuran 2x2 yang mencangkup komponen sebagai berikut [12]:

Table 2.1 Komponen confusion matrix

Actual Value	Positive	Negative
positive	True Positive	False Negative
Negative	False Positive	True Negative

Komponen tersebut meliputi: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) [12]. Model yang baik akan menghasilkan nilai yang terdapat di area diagonal, dan kolom sisanya akan menghasilkan angka

0 [12]. Sedangkan model yang tidak baik akan mendapatkan nilai yang terdistribusi pada semua kolom [12]. *Error matrix* akan memberi informasi seberapa buruk suatu model ketika model itu memang buruk [12]. Nilai yang terdapat pada setiap sel dapat mengidentifikasi pola yang salah diklasifikasikan [12]. Metode yang digunakan untuk melakukan pengukuran adalah *accuracy, precision*, dan *recall* [12].

2.5 Model Optimization

2.5.1 Bayesian Optimization

Optimisasi ini adalah langkah yang cukup penting untuk hasil efektivitas dari algoritma. Pemilihan hyperparameter yang tepat dapat mempengaruhi kinerja dari suatu model dengan lebih akurat [9]. Optimisasi bayesian dapat mengoptimalkan hyperparameter dengan kombinasi yang berbeda pada setiap *epoch* [8]. *Surrogate model* perlu dibuat untuk memetakan nilai pada *Mean Absolute Error* atau MAE pada seluruh ruang pencarian atau *search domain* [11]. Hal ini kemudian disempurnakan sesuai tahapan sampai menemui titik konvergensi. Optimisasi bayesian memiliki batas ruang yang ditentukan oleh interval [11]. Setelah dibentuknya sebuah *surrogate model* dari MAE, efeknya yang dihasilkan adalah MAE dapat diminimalkan secara efisien untuk mencari kombinasi hyperparameter yang optimal [11]. Sehingga hal ini dapat menghasilkan nilai kesalahan global yang paling kecil [11].

Bayesian optimization memiliki kelebihan dalam menemukan kombinasi parameter terbaik tanpa perlu memproses seluruh kemungkinan yang dapat memberikan efisiensi dalam optimasi fungsi yang kompleks [20]. Selain itu optimasi ini juga mampu bekerja pada struktur fungsi tujuan yang belum diketahui atau sulit diturunkan (*non-differentiable*), sehingga cocok untuk fungsi "*black box*". *BO* juga memiliki kelemahan dalam ketergantungannya pada *surrogate model* misalnya terhadap *gaussian process* [20]. Hal ini membuat kurangnya efisiensi dalam ruang dimensi tinggi [20]. Selain itu optimasi ini juga memiliki sisi kurang terhadap kestabilan ketika pencarian sumber daya terbatas, sehingga hasil dapat menjadi *sub-optimal* atau tidak maksimal [20].

2.6 Software yang digunakan

2.6.1 Google Colab

Google Colab merupakan sebuah produk yang dimiliki oleh Google Research yang memungkinkan pengguna untuk menulis dan mengeksekusi kode dari python melewati browser atau mesin pencari [10]. Google colab sangat cocok dipakai untuk masalah machine learning ataupun data analis [10]. Google colab merupakan aplikasi tidak berbayar berbasis layanan cloud untuk AI ataupun machine learning [10]. Google colab juga sudah support dengan modul modul yang dibutuhkan untuk analisa data science seperti module numpy, scipy, pandas, tensorflow, keras, dan lainnya [10]. Kelebihan dari google colab adalah seperti akselerasi GPU gratis, telah dibangun diatas jupyter notebook, fitur kolaborasi dengan tim secara online, pre-installed library [10].

2.6.2 Python

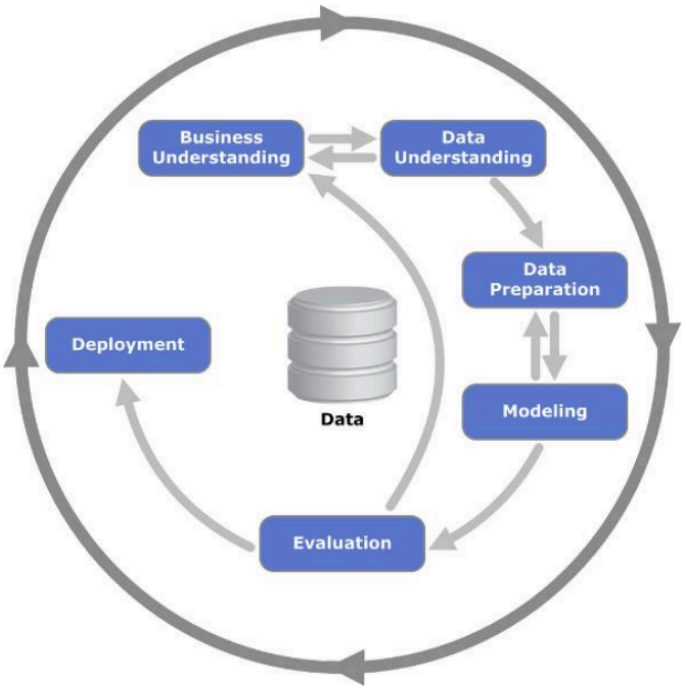
Python merupakan bahasa object oriented untuk pemrograman yang didesain untuk menulis program yang sederhana dalam skala yang besar maupun kecil [11]. Python dapat mendukung banyak paradigma pemrograman, termasuk object oriented, wajib, fungsional, atau prosedural [11]. Python juga memiliki memori otomatis sistem manajemen yang baik dan pustaka yang luas [11].

2.7 Metode yang digunakan

2.7.1 CRISP-DM

Metodologi Cross-Industry Standard Process for Data Mining (CRISP-DM) merupakan kerangka kerja proses standar yang banyak diadopsi dalam proyek *data science* dan *data mining* karena memberikan panduan yang sistematis dan terstruktur dari awal hingga akhir proyek [41]. Berdasarkan ulasan sistematis terhadap penelitian terbaru di berbagai publikasi ilmiah, CRISP-DM masih dipandang sebagai metode standar *de-facto* dalam pelaksanaan penelitian dan pengembangan model berbasis data, meskipun telah lebih dari dua dekade sejak pertama kali diperkenalkan [41]. Studi tersebut mengidentifikasi bahwa CRISP-DM mencakup enam fase inti mulai dari *business understanding*, *data*

understanding, *data preparation*, *modeling*, *evaluation*, hingga *deployment* dan fase-fase ini saling terkait sehingga mendukung analisis yang konsisten serta hasil yang dapat diulang ulang oleh peneliti atau praktisi data mining [41]. Selain itu, penelitian ini juga menemukan *best practices* dan tantangan yang dihadapi dalam penerapan setiap fase, termasuk bahwa banyak studi yang tidak memasukkan fase *deployment* secara eksplisit, sehingga memberikan wawasan tentang area yang masih perlu diperkuat dalam implementasi kerangka kerja tersebut [41].



Gambar 2.3 Metode CRISP-DM

1. **Business Understanding**: Pada tahap ini dilakukan penilaian terhadap kondisi bisnis untuk memperoleh gambaran umum mengenai sumber daya yang tersedia serta yang dibutuhkan. Salah satu aspek paling penting dalam fase ini adalah penentuan tujuan dari proses data mining. Jenis analisis yang akan dilakukan, misalnya klasifikasi atau prediksi, harus dijelaskan secara jelas, begitu pula dengan kriteria keberhasilan seperti tingkat akurasi atau presisi hasil model. Selain itu, perlu dibuat rencana

proyek yang terstruktur sebagai pedoman selama proses penelitian berlangsung.

2. **Data Understanding**: Tahap ini melibatkan proses pengumpulan data dari berbagai sumber, eksplorasi awal, serta pemeriksaan kualitas data. Kegiatan utama yang dilakukan meliputi penjelasan karakteristik data melalui analisis statistik dan identifikasi atribut yang akan digunakan serta hubungannya. Tujuan dari tahap ini adalah memastikan bahwa data yang akan diolah memiliki kualitas baik dan sesuai dengan kebutuhan analisis pada tahap berikutnya.
3. **Data Preparation**: Pada fase ini dilakukan pemilihan data dengan menetapkan kriteria inklusi dan eksklusi agar hanya data yang relevan yang digunakan. Data yang memiliki kualitas buruk diperbaiki melalui proses *data cleaning*. Bergantung pada model yang akan digunakan, atribut turunan (*derived attributes*) dapat dibuat untuk meningkatkan performa model. Setiap langkah dalam tahap ini dapat dilakukan dengan berbagai metode tergantung pada karakteristik model dan kebutuhan analisis.
4. **Modeling**: Tahapan pemodelan mencakup pemilihan teknik analisis yang sesuai, pembangunan model uji (*test case*), serta pengembangan model utama. Semua teknik *data mining* dapat diterapkan, tergantung pada permasalahan yang dihadapi dan karakteristik data yang digunakan. Yang terpenting bukan hanya pemilihan metode, tetapi juga penjelasan alasan pemilihan tersebut. Dalam membangun model, parameter tertentu perlu diatur agar hasil yang diperoleh optimal. Selanjutnya, model dievaluasi berdasarkan kriteria penilaian untuk menentukan performa terbaik.
5. **Evaluation**: Pada tahap evaluasi, hasil yang diperoleh dibandingkan dengan tujuan bisnis atau penelitian yang telah ditetapkan sebelumnya. Hasil model perlu diinterpretasikan untuk memastikan kesesuaiannya dengan sasaran awal, sekaligus menentukan langkah lanjutan yang perlu dilakukan. Selain itu, seluruh proses penelitian juga direviu untuk

memastikan setiap tahapan telah berjalan dengan baik dan tidak ada bagian yang terlewat.

6. **Deployment**: Tahap terakhir dalam model CRISP-DM dijelaskan secara umum sebagai proses penerapan hasil penelitian. Bentuk implementasi dapat berupa laporan akhir, sistem prediksi, atau komponen perangkat lunak. Pada tahap ini juga dilakukan perencanaan penerapan, pemantauan kinerja model, serta kegiatan pemeliharaan agar hasil yang diperoleh dapat digunakan secara berkelanjutan.