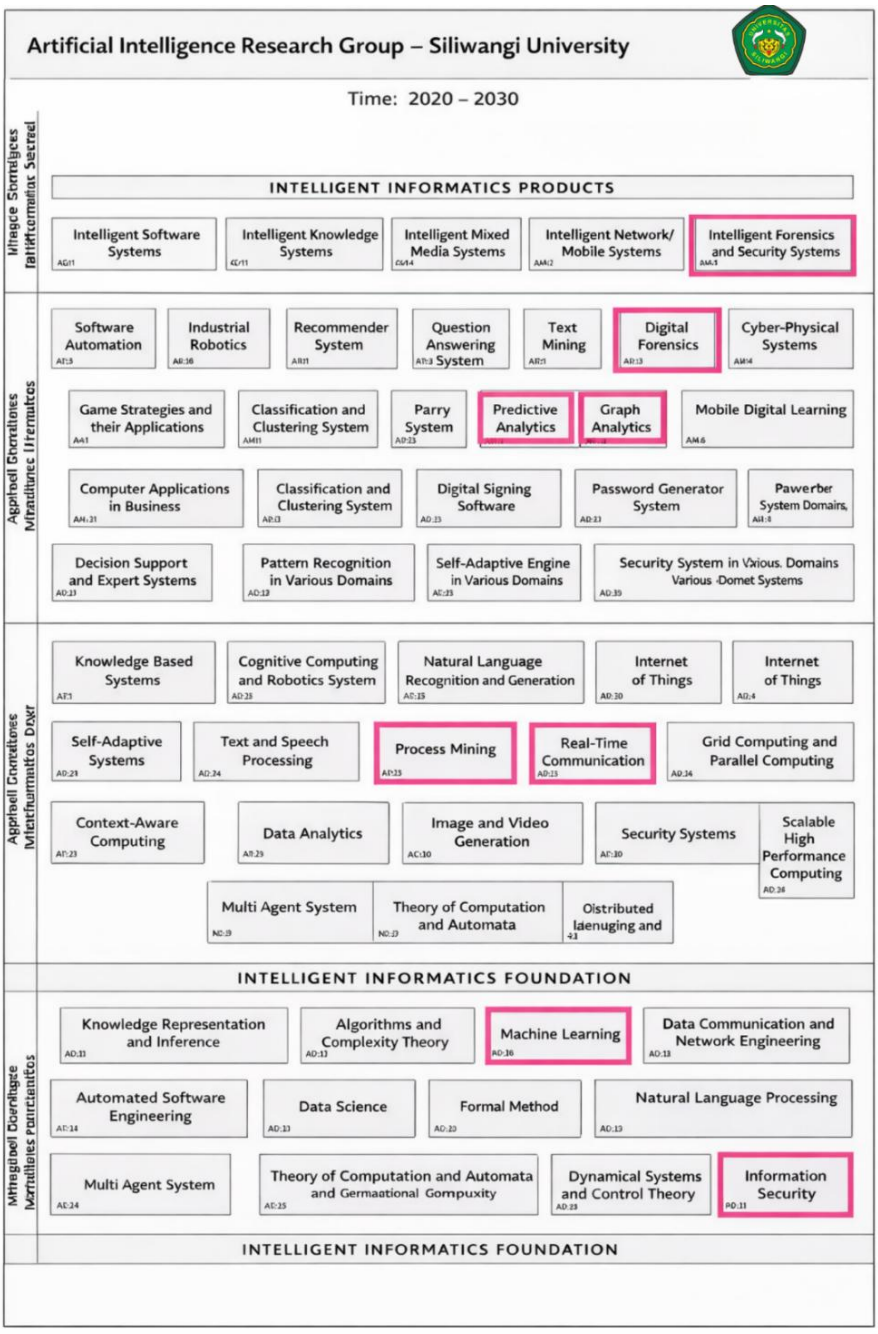
BAB III
METODOLOGI

3.1 *Roadmap* Penelitian

Roadmap penelitian pada Gambar 3.1 diadaptasi dari kelompok keahlian informatika dan sistem inteligen yang berkolaborasi dengan jaringan keamanan forensik pada program studi informatika yang mencerminkan jalan untuk menutup celah secara sistematis. Pada lapisan *basic knowledge*, kotak *machine learning* menjadi landasan pemilihan XGBoost dan pembelajaran terdistribusi, sedangkan *information security* merepresentasikan kebutuhan akan pengelolaan data yang terdistribusi secara aman tanpa pemusatan data mentah. Penelitian *federated learning* pada domain keuangan dan sistem terdistribusi, mendapatkan masalah isu keamanan data sering dikaitkan dengan risiko kebocoran informasi selama proses kolaborasi model (Kairouz dkk., 2021). Penelitian berfokus pada evaluasi penerapan arsitektur pembelajaran *federated learning*, khususnya dalam menilai bagaimana model XGBoost berperilaku ketika dilatih pada data yang terdistribusi, serta implikasinya terhadap kinerja model dan efisiensi operasional.



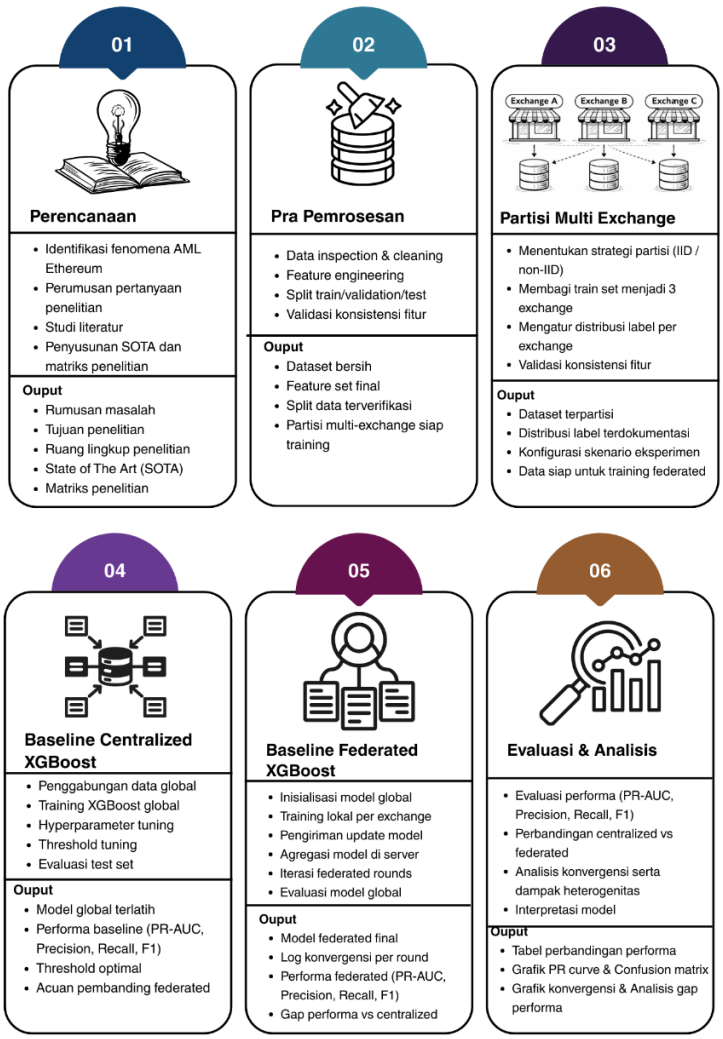
Gambar 3. 1 Roadmap Penelitian dimodifikasi dari (informatika, 2020)

Lapisan *basic application*, kotak *security systems* dan *process mining* mencerminkan karakter AML sebagai sistem keamanan yang menganalisis alur transaksi. Selanjutnya, pada lapisan *applied application*, penekanan pada *digital forensics, predictive analytics*, dan *graph mining* sejalan dengan penelitian AML Ethereum sebelumnya oleh Zhou dkk. (2022) yang menegaskan bahwa pendekatan

graf lebih efektif dibanding analisis transaksi individual. Sebagian besar penelitian tersebut masih mengandalkan data *on chain* dan pendekatan terpusat, sehingga belum sepenuhnya mencerminkan kondisi data terdistribusi.

Lapisan *market product*, posisi *intelligent forensics and security systems* menggambarkan arah kontribusi jangka panjang penelitian ini, yaitu menyediakan fondasi metodologis untuk sistem deteksi alamat berisiko terkait aktivitas ilegal dalam konteks AML Ethereum *multi exchange* yang kolaboratif dan aman secara privasi dengan penggunaan *machine learning*.

3.2 Metode Penelitian



Gambar 3. 2 Tahapan Penelitian diadopsi dari (Schröer dkk., 2021)

Metode eksperimental kuantitatif dengan desain penelitian *improvement based evaluation* pada Gambar 3.2 digunakan pada penelitian terinspirasi oleh penelitian (Khan dkk., 2024) untuk menganalisis penerapan model XGBoost dalam dua skenario pelatihan, yaitu *centralized learning* dan *federated learning*. Eksperimen dirancang untuk mengevaluasi bagaimana perubahan mekanisme pelatihan memengaruhi kinerja model dan karakteristik proses pembelajaran dalam konteks data terdistribusi.

Tahapan penelitian disusun secara sistematis mengikuti kerangka CRISP DM (Schröer dkk., 2021), dimulai dari perumusan masalah hingga evaluasi model. Perencanaan mencakup identifikasi permasalahan deteksi AML pada *blockchain* Ethereum, perumusan pertanyaan penelitian, studi literatur, serta penetapan ruang lingkup dan metrik evaluasi sebagai dasar penelitian.

3.2.1 Perencanaan

Permasalahan deteksi *Anti Money Laundering* (AML) pada transaksi *blockchain* Ethereum, secara umum telah diteliti menggunakan pendekatan berbasis analisis fitur transaksi maupun struktur graf. Penelitian sebelumnya menunjukkan bahwa pola *laundering* seperti *fan-in, fan-out,* dan *layering* dapat dimodelkan melalui interaksi alamat dan aliran dana dalam graf transaksi (Zhou dkk., 2022); (Liu dkk., 2023). Namun, mayoritas Penelitian mengasumsikan ketersediaan data dalam lingkungan terpusat, sehingga proses pelatihan model dilakukan pada satu kumpulan data global. Asumsi ini tidak sepenuhnya merepresentasikan kondisi operasional pada ekosistem *multi exchange*, karena data transaksi tersebar di berbagai *exchange* dan tidak secara langsung dapat digabungkan untuk keperluan analitik.

Aktivitas ilegal melibatkan pergerakan dana lintas *exchange*, sehingga deteksi berbasis satu sumber data berpotensi kehilangan konteks global. Literatur regulasi AML internasional, menekankan pentingnya keterlacakan dan kolaborasi lintas penyedia layanan aset virtual, tetapi pada saat yang sama menunjukkan kompleksitas implementasi lintas yurisdiksi dan keterbatasan pertukaran data operasional (FATF, 2019). Kondisi ini menimbulkan kebutuhan untuk

mengeksplorasi pendekatan pembelajaran kolaboratif yang tidak bergantung pada penggabungan data mentah, namun tetap mampu mempertahankan utilitas model deteksi.

Federated learning diposisikan bukan sekadar solusi desentralisasi data, melainkan sebagai kerangka pembelajaran yang sensitif terhadap heterogenitas distribusi dan dinamika agregasi model (Zhang dkk., 2021). Survei komprehensif pada penelitian optimisasi *federated* pada jaringan heterogen menunjukkan bahwa performa model *federated* sangat dipengaruhi oleh perbedaan distribusi lokal (non IID) dan variasi ukuran data antar klien (Li dkk., 2020). Model global dapat mengalami penurunan utilitas dibandingkan *centralized learning* yang memiliki akses terhadap distribusi penuh (Garst dkk., 2025). Penelitian ini menekankan perlunya evaluasi empiris terhadap kinerja *federated* XGBoost dalam konteks deteksi alamat berisiko pada transaksi *blockchain* Ethereum, khususnya dalam skenario simulasi *multi exchange*. Evaluasi tidak hanya difokuskan pada utilitas model, tetapi pada dampak distribusi data terhadap stabilitas pelatihan serta *overhead* operasional. Pendekatan *federated* tidak diasumsikan secara apriori memiliki performa yang sebanding dengan pendekatan *centralized*, melainkan diuji melalui analisis perbandingan berbasis simulasi.

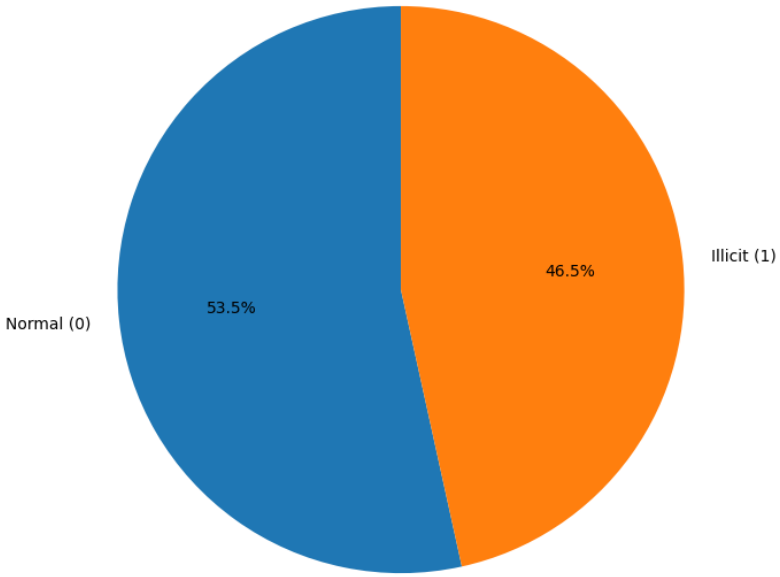
Penelitian deteksi alamat berisiko pada *blockchain* Ethereum telah mengeksplorasi berbagai model *machine learning*, termasuk teknik *boosting* berbasis pohon menunjukkan bahwa XGBoost dapat mengidentifikasi alamat berisiko di jaringan Ethereum dengan akurasi tinggi, menegaskan efektivitasnya dalam memodelkan pola perilaku transaksi yang kompleks (Farrugia dkk., 2020). Dataset Ethereum yang digunakan berasal dari *Kaggle* yang telah digunakan di beberapa penelitian sebelumnya, menemukan bahwa model XGBoost unggul dibandingkan *Random Forest* dan regresi logistik dalam klasifikasi aktivitas ilegal, menghasilkan akurasi dan metrik evaluasi lainnya yang kompetitif (Farrugia dkk., 2020); (Pahuja & Kamal, 2023). Secara umum, kajian komprehensif dalam integrasi *machine learning* untuk deteksi alamat berisiko terkait aktivitas ilegal dalam konteks AML kripto mencatat bahwa algoritma XGBoost konsisten muncul

dalam penelitian terkini sebagai salah satu teknik performa tinggi dalam klasifikasi aktivitas ilegal (El-Attar dkk., 2025). Merujuk pada penelitian mengenai pendekatan *ensemble learning* yang melibatkan teknik *boosting* juga menunjukkan performa kuat dalam tugas klasifikasi transaksi Ethereum (Gu & Dib, 2025).

Penelitian terdahulu umumnya mengkaji deteksi alamat berisiko pada transaksi Ethereum menggunakan pendekatan *centralized*, sementara penerapan *federated learning* masih bersifat umum dan belum mempertimbangkan karakteristik spesifik data *blockchain*. Berdasarkan celah tersebut, penelitian ini berfokus pada penerapan dan evaluasi empiris *federated* XGBoost dalam deteksi alamat berisiko pada transaksi Ethereum, dengan menggunakan pendekatan *centralized* sebagai *baseline* pembanding dalam skenario simulasi *multi exchange*. Penerapan dilakukan dalam lingkungan simulasi terkontrol untuk menghasilkan baseline awal yang menilai kelayakan pendekatan federated, khususnya dalam konteks data terdistribusi tanpa berbagi data mentah. Penelitian ini tidak bertujuan mengembangkan algoritma baru, melainkan mengadaptasi mekanisme federated learning melalui pendekatan *sequential federated boosting* agar sesuai dengan karakteristik XGBoost sebagai model berbasis pohon. Fokus utama analisis terletak pada *trade off* antara utilitas model dan overhead operasional, serta stabilitas pelatihan dalam skenario distribusi data terdesentralisasi.

3.2.2 Pra pemrosesan

Tahap pra pemrosesan diposisikan sebagai fondasi validitas eksperimen yang memastikan bahwa perbandingan kinerja antara pendekatan *centralized* dan *federated learning* dilakukan secara adil dan terkontrol. Dataset yang digunakan merupakan data transaksi Ethereum yang telah dilengkapi label biner (FLAG) yang menunjukkan status alamat sebagai normal (0) atau *illicit* (1) (Farrugia dkk., 2020).



Gambar 3. 3 Visualisasi Distribusi Kelas

Dataset awal memiliki distribusi kelas sebesar 53,5% normal dan 46,5% *illicit*, Gambar 3.3 menunjukkan kondisi relatif seimbang tanpa ketimpangan ekstrem. Selanjutnya, dilakukan proses pembersihan data yang mencakup penghapusan duplikasi, validasi tipe data, serta eliminasi fitur yang berpotensi menyebabkan data leakage, seperti identifier dan fitur yang secara langsung merepresentasikan label.

Proses deduplikasi mengurangi jumlah data dari 4.681 menjadi 4.153 sampel. Perubahan ini berdampak pada distribusi kelas, namun tidak mengubah karakteristik fundamental dataset, melainkan merupakan konsekuensi dari pembersihan data. Penghapusan duplikasi bertujuan untuk memastikan bahwa model tidak dilatih pada observasi yang identik, sehingga evaluasi yang dihasilkan mencerminkan kemampuan generalisasi terhadap data baru.

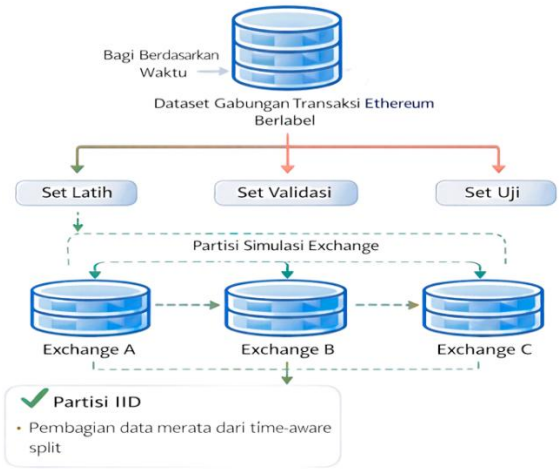
Penanganan *missing value* dilakukan dengan mempertimbangkan karakteristik data. Nilai kosong yang muncul pada fitur transaksi diinterpretasikan sebagai *structural missing*, yaitu kondisi yang merepresentasikan tidak adanya aktivitas transaksi. Oleh karena itu, tidak dilakukan imputasi agresif, dan penanganan diserahkan pada mekanisme bawaan XGBoost yang mampu mengelola nilai kosong secara langsung.

Dataset kemudian dibagi menggunakan *stratified split* ke dalam data pelatihan, validasi, dan pengujian dengan rasio 70:15:15 (Haseeb dkk., 2023). Pendekatan ini dipilih karena penelitian ini berfokus pada evaluasi komparatif arsitektur pembelajaran, bukan prediksi berbasis waktu sehingga konsistensi distribusi kelas antar subset menjadi lebih penting dibandingkan urutan temporal data.

3.2.3 Partisi Simulasi *Multi Exchange*

Partisi *multi exchange* dalam penelitian ini bertujuan untuk mensimulasikan kondisi distribusi data lintas *exchange*, karena dataset yang digunakan tidak berasal dari beberapa exchange. Partisi dilakukan secara horizontal pada data latih (train set) menjadi 3 subset yang merepresentasikan 3 *exchange* dalam sistem *federated learning*. Setiap subset data memiliki struktur fitur yang identik dan dibentuk melalui partisi data secara horizontal dengan asumsi distribusi *Independent and Identically Distributed* (IID).

Data dianggap independen karena setiap alamat Ethereum direpresentasikan sebagai entitas yang berdiri sendiri, dan fitur yang digunakan merupakan agregasi aktivitas transaksi tanpa ketergantungan langsung antar alamat. Asumsi independensi dipenuhi karena setiap sampel data merepresentasikan alamat Ethereum yang dianalisis secara individual. Fitur yang digunakan merupakan hasil agregasi aktivitas masing-masing alamat, sehingga tidak terdapat ketergantungan langsung antar sampel dalam proses pembelajaran model. Asumsi *identically distributed* dipenuhi dengan memastikan bahwa proses pembagian data ke dalam masing-masing exchange dilakukan secara acak, sehingga setiap subset data memiliki karakteristik distribusi fitur dan label yang relatif serupa dengan dataset awal. Jadi, IID dalam penelitian ini bukan hanya asumsi pembagian data, tetapi didasarkan pada karakteristik data dan proses sampling yang memastikan independensi dan keseragaman distribusi secara statistik.



Gambar 3. 4 Skema Simulasi Multi Exchange melalui Partisi Horizontal IID pada Data Latih diadopsi dari (Le dkk., 2021)

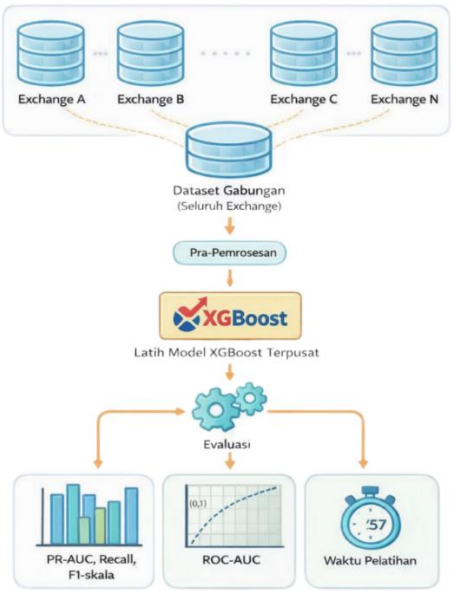
Penelitian ini memastikan bahwa perbedaan performa antara *federated* dan *centralized* merefleksikan dampak mekanisme pelatihan, bukan bias akibat distribusi data yang tidak seragam. Literatur *federated learning* menunjukkan bahwa non IID dapat menjadi sumber utama degradasi performa dan instabilitas konvergensi (Kairouz dkk., 2021) ; (Li dkk., 2020). Oleh karena itu, penggunaan IID pada tahap awal memberikan *baseline federated* yang lebih terkontrol sehingga gap performa yang muncul dapat diinterpretasikan sebagai dampak arsitektur pelatihan, bukan akibat distribusi yang timpang.

Merujuk pada Gambar 3.4 data dipartisi secara horizontal sehingga setiap *exchange* memperoleh subset berbeda dengan ruang fitur yang identik, guna menjamin konsistensi agregasi pada *federated* XGBoost. Proporsi data antar *exchange* juga dikontrol untuk menghindari bias kontribusi dalam pembaruan model global (Kairouz dkk., 2021). Partisi dirancang sebagai kontrol eksperimental yang memastikan bahwa hasil evaluasi mencerminkan dampak arsitektur *federated*, bukan artefak dari ketimpangan distribusi data.

Partisi *multi exchange* digunakan untuk membangun skenario *federated* XGBoost, data latih didistribusikan ke beberapa *exchange* dan diproses secara lokal. Sebaliknya, pendekatan *centralized* menggunakan data latih (train set) yang sama dalam bentuk teragregasi penuh sebagai *baseline* pembanding, tanpa partisi.

Validation dan *test set* dipertahankan untuk menjaga konsistensi evaluasi. Perbedaan kinerja diamati dapat dikaitkan secara langsung pada mekanisme pelatihan terdistribusi, bukan pada perbedaan komposisi data sehingga validitas komparatif antar pendekatan tetap terjaga.

3.2.4 *Baseline Centralized* XGBoost



Gambar 3. 5 Arsitektur Centralized XGBoost dimodifikasi dari (Naik, D., Naik, 2024)

Baseline *centralized* XGBoost digunakan sebagai baseline pembanding dan diposisikan sebagai *upper bound* kinerja, gambar 3.4 merepresentasikan kondisi ideal tanpa pembatasan distribusi data, di mana seluruh data latih digabungkan sebelum pelatihan. Model pada konfigurasi ini memiliki akses penuh terhadap distribusi global, sehingga menjadi referensi utama untuk menilai degradasi utilitas pada *federated* XGBoost (Naik, D., Naik, 2024). *Baseline* ini berfungsi sebagai titik pembanding empiris untuk mengukur sejauh mana kinerja *federated* mendekati kondisi optimal ketika data tidak terfragmentasi.

Baseline centralized XGBoost dikonstruksi identik dengan skenario *federated* pada ruang fitur, konfigurasi model, dan protokol evaluasi, dengan satu-satunya perbedaan pada mekanisme pelatihan. Data latih digabungkan secara global

dan dibagi secara kronologis menjadi *train, validation*, dan *test set* untuk mencegah bias temporal. Kesetaraan desain ini memastikan bahwa *baseline centralized* tidak diuntungkan oleh konfigurasi evaluasi yang lebih longgar, sehingga perbedaan performa yang diamati dapat dikaitkan secara langsung pada arsitektur pelatihan, bukan pada perbedaan *pipeline* data (Qi dkk., 2025).

Tabel 3. 1 Pseudocode Baseline Centralized XGBoost

```
INPUT:
    D_train      = dataset pelatihan
    D_val        = dataset validasi
    D_test       = dataset pengujian
    θ            = himpunan hyperparameter XGBoost
    T            = jumlah maksimum boosting round
    ES           = early stopping rounds

OUTPUT:
    M_best       = model centralized terbaik
    τ*           = threshold optimal
    H_final      = hasil evaluasi akhir pada data test

BEGIN
    1. Inisialisasi model XGBoost dengan hyperparameter θ
       M ← Initialize_XGBoost(θ)

    2. Latih model menggunakan D_train dengan evaluasi pada D_val
       M_best ← Train_XGBoost(M, D_train, D_val, T, ES)

    3. Prediksi probabilitas pada data validasi menggunakan model
terbaik
       P_val ← Predict_Probability(M_best, D_val)

    4. Tentukan threshold optimal τ* berdasarkan data validasi
       τ* ← Threshold_Tuning(P_val, D_val.label)

    5. Lakukan evaluasi akhir pada data pengujian
```

```
       P_test ← Predict_Probability(M_best, D_test)
       Y_pred ← Apply_Threshold(P_test, τ*)

    6. Hitung metrik evaluasi akhir
       H_final ← Evaluate(Y_pred, P_test, D_test.label)
       // Accuracy, Precision, Recall, F1-score, PR-AUC

    7. Kembalikan model terbaik, threshold optimal, dan hasil
evaluasi
       RETURN M_best, τ*, H_final
END
```

Pseudocode pada Tabel 3.1 menggambarkan alur pelatihan *centralized* XGBoost sebagai baseline pembanding. Pada pendekatan ini, seluruh data latih digabungkan dalam satu lingkungan komputasi sehingga model dapat mempelajari distribusi data secara terpusat. Proses pembelajaran dilakukan melalui mekanisme *boosting*, yaitu penambahan pohon keputusan secara bertahap untuk memperbaiki kesalahan prediksi sebelumnya.

Konfigurasi *hyperparameter* θ ditentukan melalui eksperimen awal pada *baseline centralized* XGBoost. *Hyperparameter* inti seperti max_depth, learning_rate, subsample, dan colsample_bytree dipilih berdasarkan performa dan stabilitas pada data validasi. Konfigurasi tersebut kemudian digunakan dalam proses pelatihan model final, sebagaimana ditunjukkan pada Tabel 3.1. Penyajian *pseudocode* dimulai dari tahap inisialisasi model dengan *hyperparameter* terpilih karena proses tuning diposisikan sebagai tahap penentuan konfigurasi awal, bukan bagian dari evaluasi akhir model.

Konvergensi tidak dianalisis secara eksplisit pada *centralized* sebagaimana pada *federated learning*, melainkan diakomodasi secara implisit melalui mekanisme *early stopping*. Proses pelatihan dihentikan ketika performa pada data validasi tidak lagi mengalami peningkatan signifikan dalam sejumlah iterasi berturut-turut, sehingga kondisi tersebut mencerminkan bahwa model telah mencapai titik stabil dalam proses pembelajaran. Oleh karena itu, konvergensi pada pendekatan *centralized* dalam penelitian ini dipahami sebagai stabilisasi performa

yang dikendalikan oleh early stopping, bukan sebagai hasil analisis perubahan metrik secara eksplisit antar iterasi seperti pada *federated rounds*.

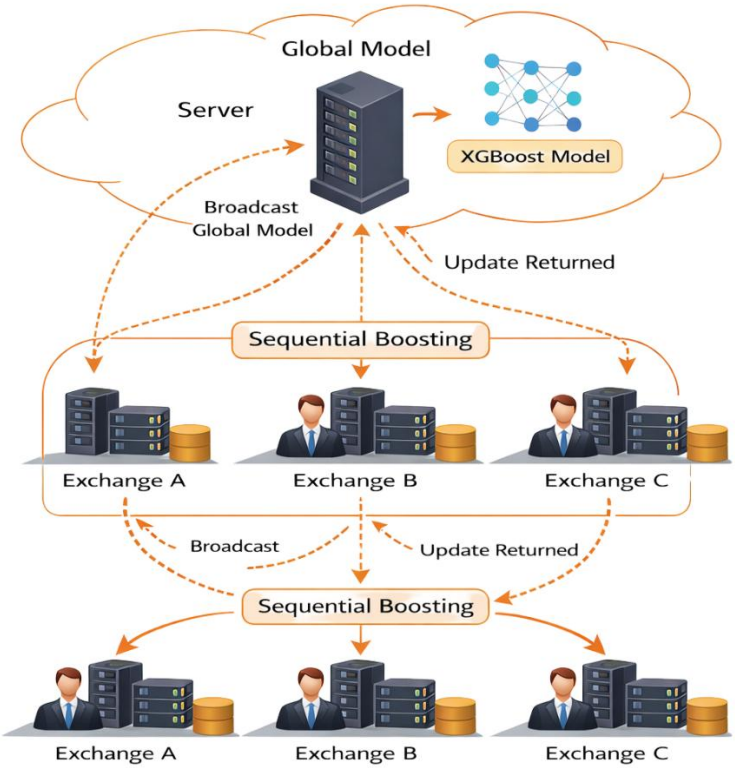
Threshold tuning pada data validasi untuk menentukan ambang probabilitas optimal dalam mengubah output probabilitas menjadi label klasifikasi akhir. *Threshold tuning* tidak mengubah parameter model, tetapi mengatur batas keputusan klasifikasi agar hasil prediksi dapat mengontrol keseimbangan antara *false positive* dan *false negative*. Output utama dari tahap ini mencakup model centralized terbaik, *threshold* optimal, serta hasil evaluasi akhir pada data uji yang digunakan sebagai baseline untuk membandingkan kinerja *federated* XGBoost.

3.2.5 *Federated* XGBoost

XGBoost dipilih dalam penelitian ini karena kemampuannya yang unggul dalam menangani data tabular kompleks seperti transaksi blockchain yang bersifat tidak seimbang, *noisy*, dan memiliki hubungan non linear. Namun, pendekatan XGBoost secara konvensional bergantung pada skenario *centralized* yang tidak sesuai dengan lingkungan *multi exchange* yang terdistribusi dan memiliki keterbatasan berbagi data. Oleh karena itu, *federated learning* digunakan sebagai pendekatan untuk memungkinkan pelatihan model secara kolaboratif tanpa memindahkan data mentah. Metode federated konvensional seperti FedAvg tidak cocok untuk model berbasis pohon karena tidak dapat melakukan agregasi parameter secara langsung. Untuk mengatasi hal tersebut, penelitian ini mengadopsi pendekatan *sequential federated boosting* sebagai adaptasi yang mempertahankan mekanisme *boosting* XGBoost dalam skenario terdistribusi, sekaligus menjadi dasar untuk mengevaluasi kelayakan penerapannya dalam konteks deteksi alamat berisiko.

Federated XGBoost diimplementasikan dengan mempertahankan data latih pada masing-masing *exchange* (client), setiap *exchange* memperbarui model secara lokal dan meneruskan model hasil pembaruan ke *exchange* berikutnya, sehingga model global terbentuk melalui akumulasi pembaruan berurutan dalam mekanisme *sequential federated boosting*. Pelatihan dilakukan dalam distribusi data IID untuk

mengisolasi pengaruh arsitektur *federated* terhadap kinerja model. Proses ini menghasilkan model global, log konvergensi antar round, serta metrik evaluasi yang digunakan untuk menilai degradasi performa dibandingkan *centralized baseline.*



Gambar 3. 6 Arsitektur Federated XGBoost dimodifikasi dari (Le dkk., 2021)

Federated XGBoost diimplementasikan melalui skema *sequential federated boosting*, pada Gambar 3.6 skema *sequential federated boosting*, model global tidak dibentuk melalui agregasi paralel parameter seperti FedAvg, tetapi diperbarui secara berurutan melalui proses pelatihan lokal pada setiap *exchange*. Model yang telah diperbarui pada *exchange* pertama diteruskan ke *exchange* berikutnya, sehingga model global pada akhir satu *round* merepresentasikan akumulasi pembaruan bertahap dari seluruh *exchange*. Berbeda dengan agregasi parameter pada model lain, proses ini bergantung pada integrasi informasi gradien untuk mempertahankan konsistensi struktur model (Le dkk., 2021). Eksperimen ini secara eksplisit mengisolasi mekanisme tersebut tanpa melibatkan komponen *privacy*

preserving, sehingga kinerja yang dihasilkan merefleksikan dampak arsitektur federated terhadap utilitas model.

Implementasi *federated* XGBoost diawali dengan sinkronisasi konfigurasi global di server, mencakup ruang fitur, *hyperparameter*, dan jumlah *federated rounds*. Model global awal kemudian didistribusikan ke setiap *exchange* sebagai titik awal pelatihan. Pelatihan dilakukan melalui *sequential federated boosting*, setiap *exchange* secara bergantian memperbarui model global dengan menghitung statistik *gradien* dan *hessian* berdasarkan data lokalnya. Pembaruan ini dikirim ke server untuk diagregasi, sehingga model global diperbarui secara iteratif hingga jumlah *round* tercapai atau konvergensi terpenuhi.

Tabel 3. 2 Pseudocode federated XGBoost pada mekanisme sequential federated boosting

```
INPUT:
    {D1, D2, D3} = dataset pelatihan lokal pada masing-masing
exchange
    D_val        = dataset validasi global
    D_test       = dataset pengujian global
    θ            = himpunan hyperparameter XGBoost
    R            = jumlah federated rounds
    B            = jumlah local boosting step per exchange

OUTPUT:
    M_best       = model federated terbaik
    r_best       = round terbaik
    τ*           = threshold optimal
    H_final      = hasil evaluasi akhir pada data test

BEGIN
    1. Inisialisasi model global awal
       M_global ← Initialize_XGBoost(θ)
    2. Inisialisasi variabel performa terbaik
       best_score ← -∞
       M_best ← NULL
```

```
       r_best ← 0
    3. FOR setiap round r = 1 sampai R DO
          a. Inisialisasi model kerja round ke-r
             M_round ← Copy(M_global)
          b. Exchange 1 memperbarui model secara lokal
             M_round ← Local_Train(M_round, D1, B)

          c. Exchange 2 memperbarui model hasil Exchange 1
             M_round ← Local_Train(M_round, D2, B)
          d. Exchange 3 memperbarui model hasil Exchange 2
             M_round ← Local_Train(M_round, D3, B)
          e. Tetapkan model global baru sebagai hasil pembaruan
berurutan
             M_global ← Copy(M_round)
          f. Evaluasi model global pada data validasi
             P_val_r ← Predict_Probability(M_global, D_val)
             score_r ← Compute_PR_AUC(P_val_r, D_val.label)
          g. Jika score_r lebih baik dari best_score
                best_score ← score_r
                M_best ← Copy(M_global)
                r_best ← r
             ENDIF

       END FOR

    4. Lakukan threshold tuning menggunakan model terbaik pada
data validasi
       P_val_best ← Predict_Probability(M_best, D_val)
       τ* ← Threshold_Tuning(P_val_best, D_val.label)
    5. Lakukan evaluasi akhir pada data pengujian
       P_test ← Predict_Probability(M_best, D_test)
       Y_pred ← Apply_Threshold(P_test, τ*)
    6. Hitung metrik evaluasi akhir
       H_final ← Evaluate(Y_pred, P_test, D_test.label)
```

```
        // Accuracy, Precision, Recall, F1-score, PR-AUC
    7.  Kembalikan  model  terbaik,  round  terbaik,  threshold
optimal, dan hasil evaluasi
        RETURN M_best, r_best, τ*, H_final
END
```

Pseudocode pada Tabel 3.2 meliputi inisialisasi model global, distribusi ke *client*, pelatihan lokal berurutan, agregasi gradien, dan pembaruan model global hingga konvergensi. Model akhir dievaluasi pada test set menggunakan PR-AUC dan metrik pendukung. Output utama mencakup model global, log konvergensi antar *round*, threshold optimal, biaya komunikasi, waktu pelatihan dan metrik kinerja yang menjadi dasar analisis degradasi performa dibanding *centralized baseline.*

Pseudocode pada Tabel 3.2 menggambarkan mekanisme *sequential federated boosting* yang, model global tidak dilatih secara paralel pada masing-masing client seperti pada *federated averaging* (FedAvg), melainkan diperbarui secara berurutan dari satu exchange ke exchange lainnya dalam setiap federated round. Model global awal dikirim ke Exchange 1 untuk dilakukan pelatihan lokal, kemudian hasil pembaruannya diteruskan ke Exchange 2 dan selanjutnya ke Exchange 3. Setelah seluruh exchange selesai melakukan pembaruan dalam satu siklus, model hasil pembaruan berurutan tersebut ditetapkan sebagai model global baru. Pendekatan ini dipilih karena XGBoost merupakan metode berbasis *boosting* yang bersifat *sequential*, sehingga lebih sesuai menggunakan mekanisme pembaruan bertahap dibandingkan agregasi paralel berbasis rata-rata seperti pada FedAvg. Model global dalam penelitian ini merepresentasikan akumulasi pembaruan berurutan dari seluruh *exchange*, bukan hasil agregasi parameter secara paralel.

Konvergensi dalam penelitian ini dianalisis untuk mengevaluasi apakah proses pelatihan telah mencapai kondisi stabil, yang ditunjukkan melalui peningkatan dan stabilisasi metrik evaluasi khususnya PR-AUC dan accuracy, antar *federated rounds*. Model dikatakan menunjukkan kecenderungan konvergen ketika nilai metrik meningkat pada fase awal pelatihan dan kemudian mengalami perubahan yang semakin kecil hingga stabil pada round berikutnya. Setelah model

terbaik diperoleh berdasarkan performa validasi, dilakukan *threshold tuning* untuk menentukan ambang probabilitas optimal dalam mengubah output probabilitas menjadi keputusan klasifikasi akhir. Proses ini tidak mengubah parameter model, tetapi mengatur batas keputusan untuk mengontrol keseimbangan antara fa*lse positive* dan *false negative* sesuai kebutuhan deteksi. Output akhir mencakup model terbaik, *round* terbaik, *threshold* optimal, serta metrik evaluasi yang digunakan sebagai dasar analisis performa, stabilitas, dan kelayakan penerapan *federated* XGBoost dibandingkan dengan *centralized baseline*.

3.2.6 Evaluasi dan Analisis

Tahap evaluasi dirancang sebagai *controlled comparative experiment* antara *centralized* XGBoost dan *federated* XGBoost, untuk memastikan bahwa *pipeline* pra pemrosesan, ruang fitur, serta pembagian *train, validation, test* identik pada kedua skenario (Kairouz dkk., 2021). Desain ini bertujuan menjaga validitas komparatif, sehingga setiap perbedaan performa dapat dikaitkan secara langsung pada mekanisme pelatihan terdistribusi, bukan pada variasi data atau konfigurasi model. Untuk itu, seluruh skenario dievaluasi menggunakan *test set* global yang sama guna menjamin *fairness*.

Evaluasi *federated* dilakukan pada tiga aspek utama yaitu utilitas model, efisiensi pelatihan, dan dinamika konvergensi. Utilitas model diukur menggunakan PR-AUC sebagai metrik utama karena metrik ini mampu merepresentasikan *trade off* antara *precision* dan *recall* dalam mendeteksi kelas positif, yaitu alamat berisiko, serta dilengkapi *Precision, Recall,* F1-*score*, dan *Accuracy* sebagai metrik pendukung (Li dkk., 2020). Pemilihan PR-AUC didasarkan pada karakteristik deteksi AML yang menitikberatkan pada kemampuan identifikasi kelas positif (illicit), sehingga evaluasi tidak bias terhadap distribusi kelas (Khan dkk., 2024).

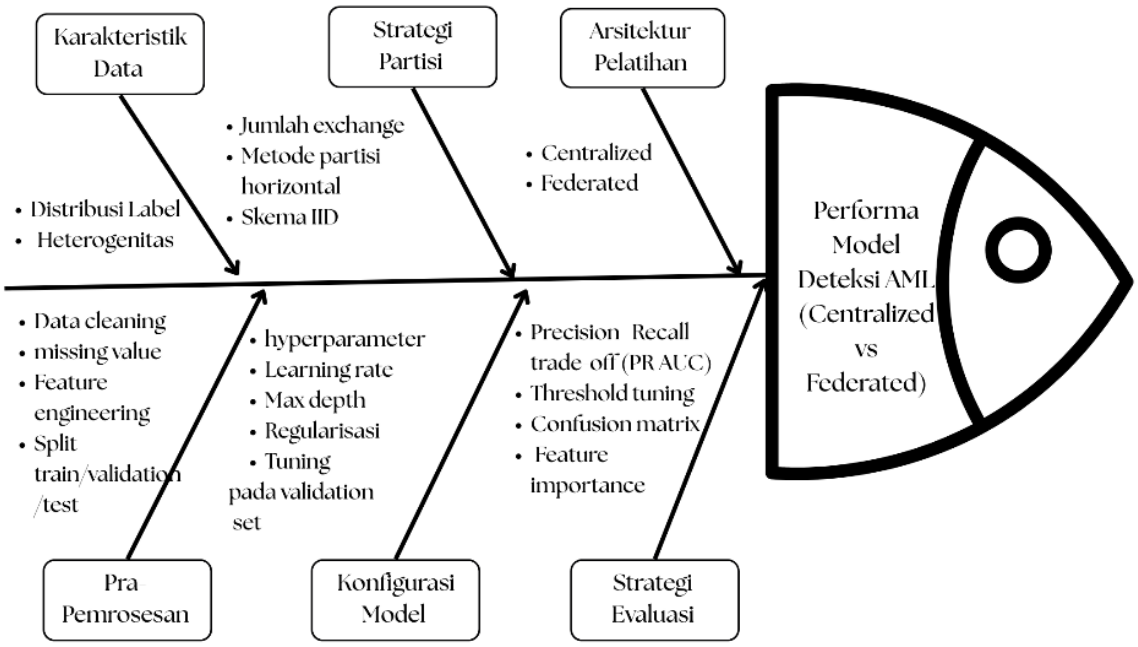
Efisiensi pelatihan dianalisis melalui *training runtime*, baik secara total maupun per *federated round*, untuk mengukur biaya komputasi yang dihasilkan oleh masing-masing pendekatan (Carrascosa, 2025). Sementara itu, dinamika pelatihan diamati melalui analisis konvergensi, yang bertujuan mengevaluasi

stabilitas performa model selama proses iteratif, khususnya pada skenario *federated* (Naik, D., Naik, 2024). Penentuan *threshold* dilakukan secara eksplisit pada *validation set* dengan aturan yang konsisten antar *baseline*, seperti memaksimalkan F1-*score* atau memenuhi batas minimum *recall*, untuk mencegah bias akibat optimasi terhadap *test set* (Khan dkk., 2024).

Output dari tahap evaluasi ini berupa metrik kinerja model, nilai *threshold* optimal, runtime pelatihan, serta pola konvergensi, yang secara kolektif digunakan untuk menganalisis *trade off* antara utilitas, serta operasional terkait efisiensi dan stabilitas (Jimenez-Gutierrez dkk., 2026). Hasil ini menjadi dasar utama dalam menilai sejauh mana *federated* XGBoost mampu mendekati kinerja *centralized baseline* serta implikasi praktisnya dalam skenario simulasi *multi exchange*.

3.3 *Fishbone* Diagram

Fishbone pada Gambar 3.7 disusun menggunakan *pendekatan cause and effect analysis* yang diperkenalkan oleh (Ishikawa, 1985) sebagai metode sistematis untuk mengidentifikasi faktor-faktor yang memengaruhi suatu outcome. Struktur faktor dalam diagram kemudian diturunkan dari sintesis literatur *federated learning* dan *machine learning* terapan. Dengan merepresentasikan faktor-faktor utama yang secara sistematis memengaruhi performa model deteksi alamat berisiko terkait aktivitas ilegal dalam konteks AML dalam skenario perbandingan *centralized* dan *federated learning*. Performa model tidak semata-mata ditentukan oleh algoritma yang digunakan, tetapi merupakan bagian dari hasil interaksi antara karakteristik data, desain eksperimen, konfigurasi teknis, serta strategi evaluasi yang diterapkan (Kairouz dkk., 2021).



Gambar 3. 7 Fishbone Diagram dimodifikasi dari (Ishikawa, 1985)

Sisi hulu, karakteristik data menjadi fondasi utama yang membentuk dinamika pelatihan. Distribusi label awal (53:47), lalu setelah mengalami pra pemrosesan di dataset (60:40) menunjukkan dataset relatif seimbang, sehingga isu utama bukan ketimpangan kelas ekstrem, melainkan potensi heterogenitas distribusi antar *exchange*. Heterogenitas ini menjadi faktor krusial dalam *federated learning* karena (Li dkk., 2020) dapat memengaruhi stabilitas agregasi dan konvergensi model global. Dengan demikian, data tidak hanya berfungsi sebagai input, tetapi sebagai determinan struktur kesulitan pembelajaran.

Strategi partisi menentukan bagaimana dataset direpresentasikan dalam skenario simulasi *multi exchange* (Jimenez-Gutierrez dkk., 2026). Pemilihan metode partisi horizontal dan distribusi data IID secara langsung membentuk kondisi simulasi terkontrol pada *federated*. Dalam konteks ini, partisi bukan sekadar pembagian data, melainkan instrumen untuk mensimulasikan realitas distribusi terdesentralisasi. Perbedaan distribusi label antar *exchange* dapat menghasilkan bias lokal yang berdampak pada kualitas agregasi model global.

Arsitektur pelatihan, menjadi pembeda inti antara *centralized* dan *federated learning*. Pada *centralized learning*, seluruh data digabung sehingga model memperoleh representasi global secara langsung. *Federated learning* melatih model secara terdistribusi dengan mekanisme agregasi parameter (Karimireddy dkk., 2020). Perbedaan ini menciptakan potensi *trade off* antara utilitas model dan stabilitas konvergensi. Oleh karena itu, arsitektur pelatihan merupakan faktor yang paling dekat secara kausal dengan performa akhir model.

Sisi operasional, pada pra pemrosesan berfungsi menjamin validitas eksperimen (Xu dkk., 2025). Proses pembersihan data, penanganan *missing value*, rekayasa fitur, serta pembagian *train/validation/test* menentukan *fairness* perbandingan antara *centralized* dan *federated*. Kesalahan pada tahap ini dapat menghasilkan data *leakage* atau bias evaluasi, yang pada akhirnya merusak integritas hasil penelitian.

Penelitian ini tidak dilakukan proses *hyperparameter tuning* secara terpisah pada masing-masing pendekatan. Konfigurasi *hyperparameter* ditentukan berdasarkan eksperimen awal dan diseragamkan antara skenario *centralized* dan *federated*. Pendekatan ini bertujuan untuk memastikan bahwa perbedaan performa yang diamati berasal dari mekanisme pembelajaran, bukan dari perbedaan konfigurasi model (Cheng dkk., 2021). Hal ini penting untuk menjaga validitas komparasi dalam mengevaluasi dampak arsitektur *federated* terhadap kinerja model.

Strategi evaluasi menentukan bagaimana performa diukur dan ditafsirkan. Penggunaan PR-AUC menekankan kemampuan model dalam mendeteksi kelas alamat berisiko terkait aktivitas ilegal dalam konteks AML secara presisi dan sensitif. Threshold tuning memperjelas *trade off* antara *precision* dan *recall* (Karami, 2025), sedangkan *confusion matrix* memberikan gambaran konkret atas hasil dan kesalahan klasifikasi. Analisis konvergensi dan *feature importance* memperkaya interpretasi hasil, sehingga evaluasi tidak berhenti pada angka agregat semata.

Fishbone menunjukkan bahwa performa model merupakan konsekuensi dari rangkaian keputusan metodologis yang saling berkaitan. Perbedaan antara *centralized* dan *federated learning* tidak dapat dianalisis secara parsial, melainkan harus dipahami sebagai hasil interaksi antara struktur data, desain partisi, konfigurasi model, serta strategi evaluasi yang diterapkan secara sistematis.