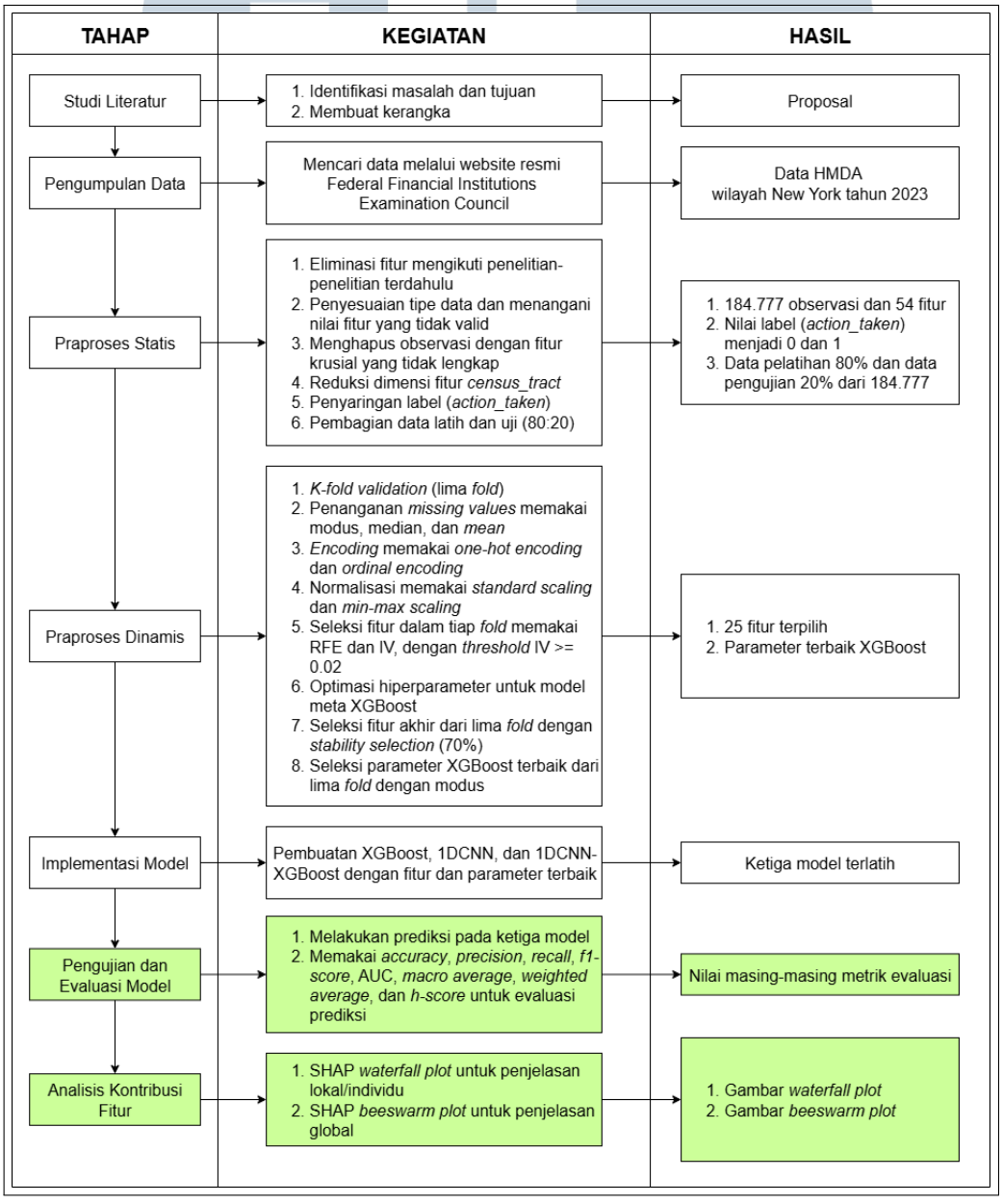
BAB 3
METODOLOGI PENELITIAN

3.1 Alur Penelitian

Gambar 3.1 menunjukkan metodologi penelitian. Metodologi penelitian dimulai dari studi literatur hingga evaluasi model. Metodologi dengan warna hijau menunjukkan langkah yang dipakai untuk menjawab rumusan masalah.



Gambar 3.1. Alur penelitian

3.1.1 Studi Literatur

Studi literatur berfungsi sebagai fondasi teoretis dan konseptual yang mengkaji penelitian terdahulu yang relevan dengan topik XAI untuk mengidentifikasi celah penelitian, merumuskan tujuan, dan menyusun kerangka kerja inovatif. Penyempitan penerapan XAI dalam sektor keuangan dan penekanan perhatian terhadap bias terhadap fitur sensitif dilakukan untuk memfokuskan penelitian pada prinsip keadilan. Sektor keuangan menjadi salah satu sektor penting untuk transparansi keputusan otomatis karena berkaitan dengan kehidupan ekonomi individu. Literatur mengenai model hibrida 1DCNN-XGBoost untuk analisis bias jenis kelamin dalam keputusan pinjaman *peer-to-peer* memakai SHAP menjadi dasar utama penelitian ini.

Keterbatasan model yang bersifat kotak hitam dalam sektor keuangan, terutama keputusan pinjaman otomatis, menjadi pusat permasalahan karena pelanggarannya terhadap transparansi, baik dari segi regulasi yang telah ditetapkan maupun terhadap masyarakat luas. Literatur yang dipelajari mencakup identifikasi bias jenis kelamin dalam proses pengambilan keputusan kredit melalui kontribusi fitur SHAP. Kajian tersebut menyoroti keadilan sistem penilaian risiko bersamaan dengan peningkatan performa model hibrida 1DCNN-XGBoost dalam kasus pinjaman *peer-to-peer* [2]. Selain itu, kajian juga menyarankan eksplorasi model hibrida pada skenario pinjaman lain untuk verifikasi peningkatan performa.

Keputusan untuk memakai praktik pinjaman hipotek perumahan menjadi tujuan yang menjawab celah penelitian bersamaan dengan pengujian performa dan analisis bias jenis kelamin dan ras pada model hibrida. Data pinjaman hipotek yang dipakai merupakan data HMDA dengan kompleksitas dan informasi demografis peminjamnya yang sangat rentan terhadap bias. Pemilihan skenario pinjaman hipotek perumahan memperkuat konteks penelitian karena keputusannya berdampak luas dalam masyarakat dan sering kali diatur secara ketat untuk mencegah diskriminasi. Metode SHAP dipilih sebagai agen XAI untuk menjelaskan kontribusi masing-masing fitur secara transparan dalam mengidentifikasi potensi bias dalam data pinjaman.

Langkah terakhir dalam studi literatur, yakni pembuatan kerangka penelitian disusun dengan menyintesis metodologi dari studi terdahulu dan inovasi yang dibuat berdasarkan identifikasi masalah dan tujuan. Arsitektur model hibrida beserta teknik evaluasi dan validasi model tetap dipertahankan, tetapi langkah pemrosesan dan pembersihan data disesuaikan dengan data yang digunakan. Dengan demikian,

studi literatur tidak hanya mengonfirmasi landasan ilmiah, tetapi juga menegaskan orisinalitas penelitian dalam memperluas aplikasi XAI ke ranah pinjaman hipotek perumahan yang lebih kompleks dan berdampak sosial tinggi.

3.1.2 Pengumpulan Data

Pengumpulan data dalam penelitian ini menggunakan data HMDA yang dipublikasikan secara terbuka melalui situs web resmi yang disediakan oleh Federal Financial Institutions Examination Council Amerika Serikat. Data HMDA memuat informasi komprehensif terkait aplikasi pinjaman hipotek perumahan, termasuk atribut demografis peminjam seperti ras, jenis kelamin, pendapatan, serta hasil keputusan persetujuan atau penolakan pinjaman. Ketersediaan atribut sensitif ini menjadikan HMDA relevan untuk menganalisis bias dalam sistem keputusan otomatis.

Data HMDA yang diambil merupakan data terbaru yang disediakan dalam situs web, yaitu tahun 2023 dan hanya mencakup wilayah New York. Wilayah New York dipilih karena kepadatan penduduknya yang tinggi dan paling memiliki keanekaragaman ras. New York telah menjadi tempat tinggal bagi hampir seluruh orang dengan berbagai kewarganegaraan dari seluruh dunia sehingga rasio penduduk mayoritas Amerika Serikat yang berkulit putih akan lebih sedikit dibandingkan dengan kota-kota lain [24].

3.1.3 Praproses Statis

Praproses statis merupakan tahap awal penyiapan data yang dilakukan secara terpusat di luar skema pembagian data dalam *k-fold validation*. Tahap ini bertujuan untuk membersihkan data-data dengan langkah-langkah yang tidak bergantung pada pembagian *fold* maupun potensi kebocoran informasi dari data yang seharusnya tidak terlihat oleh model selama pelatihan. Proses ini bersifat deterministik dan menjamin konsistensi struktur data dasar.

A Eliminasi Fitur Mengikuti Penelitian-penelitian Terdahulu

Pada tahap awal praproses statis, dilakukan penghapusan fitur berdasarkan penelitian sebelumnya. Fitur-fitur yang disingkirkan merupakan fitur yang dianggap tidak relevan dengan tujuan penelitian, misalnya *denial_reason*. Selain itu, fitur yang merupakan turunan dari fitur lain, yakni yang dimulai dengan *derived_*

atau fitur berulang, seperti *applicant_race-2* hingga *applicant_race-5* yang sudah memiliki fitur utama *applicant_race-1* juga dihapus [4]. Dengan demikian, fokus analisis dapat diarahkan pada informasi yang benar-benar penting dan relevan untuk tujuan penelitian.

B Penyesuaian Tipe Data dan Menangani Nilai Fitur dengan Kode Khusus

Langkah ini bertujuan memastikan konsistensi antara tipe data yang tercantum dalam dokumentasi resmi situs penyedia data HMDA dengan tipe data yang dimuat dalam tahap praproses statis. Selain itu, kode khusus yang digunakan sebagai representasi data yang hilang atau tidak tersedia (seperti kode numerik 1111 atau ”8888”) dalam suatu fitur juga ditangani dengan menggantinya menjadi *NaN*. Kedua hal ini bermanfaat untuk mencegah interpretasi yang keliru dalam proses analisis selanjutnya.

C Menghapus observasi dengan Fitur Krusial yang Tidak Lengkap

Demi menjaga integritas dan keandalan data, dilakukan penghapusan observasi data yang memiliki kekosongan pada fitur-fitur penting seperti *debt_to_income_ratio*, *loan_amount*, *income*, *applicant_credit_score_type*, dan masih banyak lagi. Langkah ini diambil untuk menghindari kekeliruan dalam analisis yang dapat timbul akibat imputasi nilai pada fitur-fitur krusial tersebut, mengingat bahwa *debt_to_income_ratio* dan skor kredit, misalnya, menjadi bagian dari beberapa faktor penentu dalam keputusan pemberian pinjaman [25]. Dengan demikian, penghapusan dilakukan secara selektif terhadap observasi yang dinilai dapat mengganggu validitas model apabila tetap diikutsertakan tanpa nilai yang memadai.

D Reduksi Dimensi Fitur census_tract

Fitur *census_tract* memiliki jumlah nilai unik yang sangat banyak, mencapai puluhan ribu, yang berpotensi menyebabkan dimensi data menjadi sangat tinggi nantinya setelah proses *one-hot encoding*. Untuk mengatasi hal tersebut, dilakukan penyederhanaan terlebih dahulu dalam praproses statis ini dengan mengambil digit ke-3 hingga ke-5 dari nilai *census_tract*, yang merepresentasikan kode *county* dan hanya memiliki puluhan kategori unik. Pendekatan ini membantu mengurangi kompleksitas data tanpa menghilangkan informasi geografis yang relevan.

E Penyaringan Label

Dalam langkah terakhir praproses statis, observasi data dipilih hanya dengan label yang jelas, yakni keputusannya mutlak ditolak atau diterima oleh institusi pemberi pinjaman, bukan karena hal lain. Nilai label yang dipilih, antara lain: 1 (*loan originated*), 2 (*application approved but not accepted* yang tetap dihitung sebagai penolakan), dan 3 (*application denied*) [4]. Tujuannya memastikan data yang dipakai memiliki status persetujuan yang jelas. Nilai label selain 1, 2, dan 3 memberikan kesan informasi yang tidak jelas dan hanya muncul dalam sebagian kecil dari keseluruhan data [5].

F Pembagian Data menjadi Data Latih dan Data Uji

Pembagian data menggunakan rasio 80% untuk pelatihan dan 20% untuk pengujian yang sering dipakai dalam penelitian [14] [26] [27]. Pertimbangan ini mengharapkan keseimbangan antara data latih yang memadai untuk secara efektif melatih model dan memiliki cukup data untuk diuji. Tidak ada perbedaan performa yang signifikan antara rasio 70:30 dan 80:20, yang tertera melalui Lampiran 3, sehingga rasio 80:20 dipilih untuk lebih memaksimalkan jumlah data latih yang cukup untuk menangkap variabilitas dan distribusi fitur agar mencakup sebagian besar skenario yang diharapkan dalam hal mengantisipasi *unseen* data [2].

3.1.4 Praproses Dinamis

Berbeda dengan praproses statis, praproses dinamis merupakan tahap transformasi data yang dilakukan secara terpisah dalam setiap iterasi *k-fold validation*, sehingga prosesnya disesuaikan dengan partisi data latih pada tiap *fold* untuk mencegah kebocoran data. Pendekatan ini meniru kondisi *unseen* data dan memastikan bahwa model hanya belajar dari data yang tersedia dalam konteks pelatihan. Dengan demikian, estimasi performa model menjadi lebih representatif terhadap kondisi nyata saat model dihadapkan pada data baru yang belum pernah dilihat sebelumnya.

A K-Fold Validation

K-fold validation dengan $k = 5$ sebagai kerangka praproses dinamis diterapkan untuk optimalisasi data latih sekaligus mencegah *overfitting* dan

kebocoran data. Metode ini membagi data secara acak menjadi lima *fold* dengan kombinasi data yang unik, di mana tiap iterasi menggunakan 4/5 data (80% dari proporsi data latih) untuk pelatihan dan 1/5 (20% dari proporsi data latih) untuk validasi yang akan berperan sebagai penilai performa model sementara [28] [29] [30]. Setiap *subset* data akan menjadi validasi tepat satu kali, menghasilkan estimasi performa yang lebih stabil. Seluruh proses kritis (optimasi hiperparameter, seleksi fitur, transformasi data) dilakukan secara independen per *fold* untuk memastikan validitas evaluasi.

B Penanganan Data yang Hilang dengan Modus, Median, dan Mean

Penanganan data yang hilang dilakukan dengan pendekatan imputasi menggunakan median dan rata-rata untuk fitur numerik, serta modus untuk fitur kategorikal. Pengisian pada fitur numerik bergantung pada nilai *skewness* masing-masing fitur. Jika suatu fitur memiliki distribusi yang mendekati distribusi normal, maka penanganan dilakukan dengan imputasi rata-rata atau *mean*, sedangkan untuk fitur yang memiliki distribusi miring, diisi dengan imputasi nilai tengah atau median.

Pendekatan ini bertujuan untuk mengurangi distorsi data akibat imputasi nilai yang tidak sesuai dengan karakteristik distribusinya. Metode ini menggabungkan teknik yang digunakan dalam studi sebelumnya, yaitu pemberian nilai rata-rata untuk fitur dengan data yang tersedia [4] serta pemilihan metode imputasi berdasarkan tingkat *skewness* [24]. Dengan penerapan metode ini, nilai yang diisi dipastikan tetap mencerminkan distribusi asli dari data yang hilang.

C Encoding memakai One-Hot Encoding dan Ordinal Encoding

Proses *encoding* dilakukan untuk mengubah fitur kategorikal menjadi representasi numerik agar dapat digunakan dalam model pembelajaran mesin. Dua jenis teknik *encoding* yang digunakan adalah *one-hot encoding* dan *ordinal encoding*. *One-hot encoding* diterapkan pada fitur kategorikal nominal yang tidak memiliki urutan alami antar kategorinya. Sebaliknya, *ordinal encoding* digunakan pada fitur kategorikal *ordinal*, yakni adanya urutan antar kategori dengan memberikan nilai numerik bertingkat. Pemilihan teknik disesuaikan dengan karakteristik masing-masing fitur untuk menjaga relevansi semantik semua data tiap fitur.

D Normalisasi memakai Standard Scaling dan Min-Max Scaling

Normalisasi bertujuan untuk menyelaraskan skala data numerik agar proses pelatihan model tidak terpengaruh oleh perbedaan rentang antar fitur. Dua teknik normalisasi yang digunakan adalah *min-max* dan *standard scaling*. *Min-max scaling* diterapkan pada fitur numerik yang memiliki distribusi miring (*skewed*) dengan cara mengubah nilai ke dalam rentang [0, 1], sehingga mengurangi dominasi fitur dengan rentang besar. Sementara itu, *standard scaling* digunakan untuk fitur numerik yang memiliki distribusi mendekati normal, dengan menstandarkan nilai agar memiliki rata-rata nol dan standar deviasi satu. Pemilihan teknik normalisasi ini dilakukan berdasarkan distribusi masing-masing fitur untuk memastikan efektivitas proses pelatihan model.

E Seleksi Fitur dalam Tiap Fold dengan Recursive Feature Elimination dan Information Value

Seleksi fitur bertujuan untuk memperoleh fitur-fitur yang paling relevan dan representatif terhadap data secara keseluruhan, sehingga hanya fitur yang benar-benar berkontribusi terhadap prediksi target yang digunakan dalam proses pelatihan dan pengujian model. Dengan memanfaatkan kekuatan RFE dalam melakukan eliminasi fitur berbasis model, serta IV yang mampu mengukur kekuatan prediktif suatu fitur terhadap target biner berdasarkan distribusi *log odds* antar kelas, jumlah fitur dapat dikurangi guna menghindari permasalahan dimensi tinggi serta menurunkan beban komputasi. Fitur yang akhirnya dipilih adalah fitur-fitur yang memenuhi ambang signifikansi yang telah ditetapkan.

Dalam penelitian ini, penggabungan RFE dan IV dilakukan secara bertahap, dimulai dengan eliminasi menggunakan RFE, yang kemudian dilanjutkan dengan seleksi berdasarkan nilai IV. Pada pendekatan RFE, model *estimator* yang digunakan adalah Logistic Regression [14]. Setelah melalui proses RFE, jumlah fitur dalam setiap *fold* tersebar acak, dalam rentang 143 hingga 164 fitur.

Pada tahap seleksi fitur menggunakan IV, dilakukan perhitungan nilai IV untuk setiap fitur hasil RFE. Fitur dengan nilai IV kurang dari 0,02 dianggap tidak memiliki informasi yang cukup relevan terhadap target dan dieliminasi dari *subset* fitur. Proses ini secara signifikan mengurangi jumlah fitur, dengan hasil yang lebih terpusat pada 25 hingga 29 fitur yang lolos seleksi pada masing-masing *fold*.

F Optimasi Hiperparameter Model XGBoost

Optimasi hiperparameter dilakukan dengan menerapkan pencarian *grid* pada model XGBoost dalam tiap *fold*. Pencarian *grid* berguna untuk mengeksplorasi kombinasi parameter yang telah ditentukan untuk menemukan konfigurasi optimal yang dapat memaksimalkan akurasi model. Pendekatan ini memastikan bahwa model yang dihasilkan memiliki generalisasi yang kuat dan tidak hanya mengandalkan nilai ambang tertentu. Setelah proses pencarian *grid*, konfigurasi terbaik yang menghasilkan skor tertinggi disimpan dalam variabel rekaman untuk setiap *fold*.

G Seleksi Fitur Akhir dari Lima Fold dengan Stability Selection

Setelah mendapatkan rekaman fitur yang dipakai dalam lima *fold*, pemilihan fitur final dilakukan dengan pendekatan *stability selection*. Pendekatan ini bertujuan untuk memilih fitur-fitur yang muncul lebih dari persentase tertentu dalam rekaman [31], yang dipakai dalam penelitian ini adalah 70% dan mendapatkan 25 fitur akhir. Dengan demikian, fitur final yang dipakai dipastikan konsisten dan dapat diandalkan dalam representasi data secara keseluruhan saat pelatihan dan pengujian model akhir.

H Seleksi Parameter XGBoost Terbaik dari Lima Fold dengan Modus

Selain melakukan pemilihan fitur akhir, dilakukan juga pemilihan parameter XGBoost terbaik dari rekaman lima *fold* saat proses *k-fold validation* sebelumnya. Pendekatan yang dipakai adalah dengan modus, di mana memilih kombinasi parameter yang paling banyak muncul dari lima *fold* tersebut. Dengan demikian, diperoleh kombinasi parameter yang paling konsisten memberikan performa optimal di berbagai *fold*, sekaligus mengurangi risiko *overfitting* terhadap satu *subset* data tertentu.

3.1.5 Implementasi Model

Setelah diperoleh kombinasi fitur dan parameter terbaik melalui proses *k-fold validation*, tahap selanjutnya adalah membangun model akhir untuk proses evaluasi lebih lanjut. Tiga model akhir dikonstruksi berdasarkan arsitektur yang telah ditentukan, yaitu model tunggal XGBoost, model tunggal 1DCNN, dan

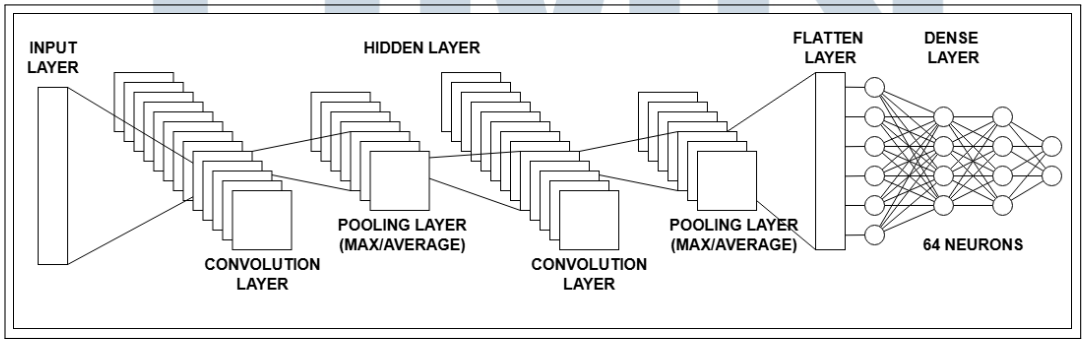
model hibrida 1DCNN–XGBoost. Masing-masing model dilatih menggunakan fitur terpilih dan parameter optimal untuk memastikan hasil evaluasi mencerminkan performa terbaik dari setiap pendekatan yang diusulkan.

A Model Tunggal XGBoost

Model tunggal XGBoost yang diimplementasikan dalam penelitian ini menggunakan pendekatan pemrosesan berbasis GPU untuk mempercepat waktu pelatihan, dengan mengatur parameter *tree_method* menjadi histogram yang dikenal efisien dalam membangun pohon keputusan secara histogram. Evaluasi model dilakukan menggunakan metrik *log loss* sebagai fungsi kerugian utama selama pelatihan. Selain parameter inti yang telah ditetapkan, parameter lain seperti *max_depth*, *learning_rate*, dan sebagainya diambil dari parameter terbaik saat optimasi hiperparameter.

B Model Tunggal 1DCNN

Implementasi model tunggal 1DCNN dipilih berdasarkan salah satu dari dua model 1DCNN yang dipakai dalam model dasar hibrida 1DCNN-XGBoost pada penelitian sebelumnya [2]. Keputusan ini dipilih karena pada penelitian tersebut tidak disebutkan secara spesifik arsitektur tunggal 1DCNN yang dipakai, melainkan hanya berupa visualisasi arsitektur yang dapat dilihat pada Gambar 3.2. Setelah pelatihan dan pengujian, didapat performa dengan kombinasi lapisan *max pooling* dan *dense*.



Gambar 3.2. Arsitektur model tunggal 1DCNN
Sumber: [2]

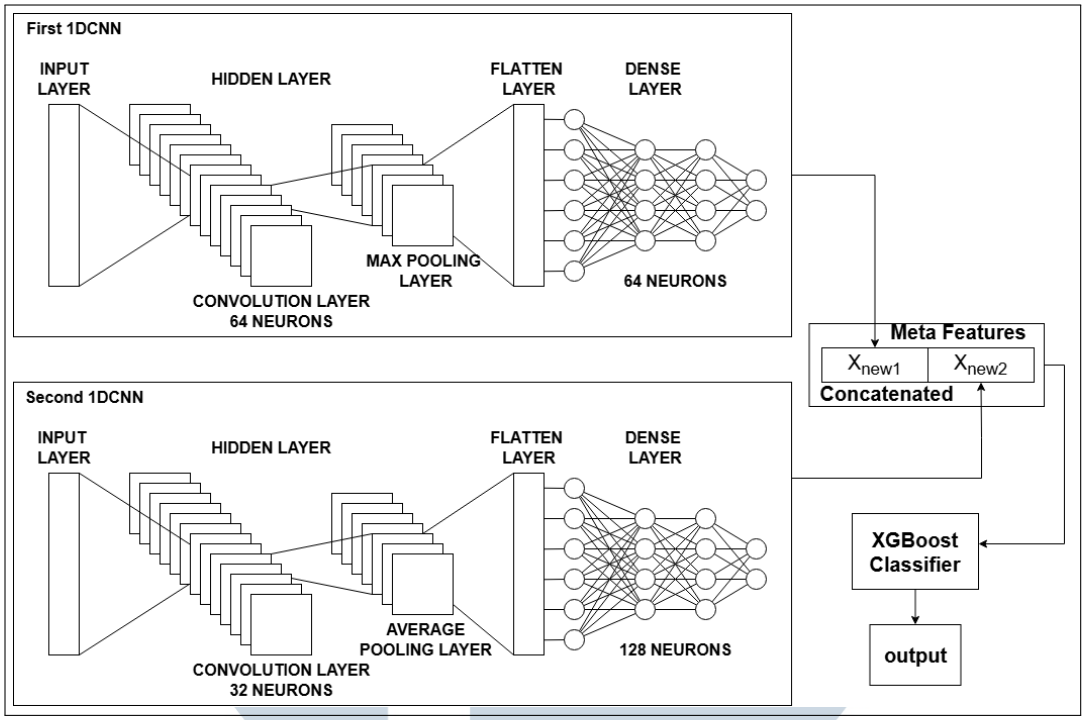
Dalam tahap awal proses ekstrak fitur, lapisan *convolutional 1D* dan *max pooling* dipakai dengan ukuran filter 64 dan ukuran *pool* 2 guna mengurangi

dimensi data dan mencegah *overfitting*. Pada lapisan *convolutional 1D*, ukuran *kernel* diatur nilainya sebesar 8 dengan menerapkan fungsi aktivasi *ReLU* untuk menangani nonlinieritas [14]. Dengan demikian, berdasarkan konfigurasi yang spesifik ini, model dapat mengekstraksi fitur secara efisien dari data.

Setelah tahap ekstraksi fitur selesai, model dilanjutkan dengan penyusunan lapisan *dense* untuk melakukan klasifikasi. Pada tahap ini, digunakan lapisan *dense* dengan 64 unit dengan fungsi aktivasi *ReLU* untuk mengoptimalkan pembelajaran fitur yang telah diekstrak. Lapisan ini diakhiri dengan lapisan *output* tunggal dengan fungsi aktivasi *sigmoid* untuk menghasilkan probabilitas yang sesuai untuk klasifikasi biner. Konfigurasi model 1DCNN dengan lapisan yang berurutan inilah yang memberikan fondasi untuk prediksi yang akurat. Terakhir, pada saat pelatihan model, *epochs* diatur sebanyak 150 kali pengulangan dan ukuran *batch* sebesar 16 [14].

C Model Hibrida 1DCNN-XGBoost

Model hibrida yang dikembangkan dalam penelitian ini terdiri dari dua model dasar berbasis 1DCNN yang dirancang untuk mengekstraksi fitur-fitur penting dari data input secara berurutan. Implementasi model hibrida ini dapat dilihat pada Gambar 3.3. Model dasar pertama menggunakan filter *convolutional 1D* dengan ukuran 64, diikuti oleh lapisan *dense* dengan 64 unit, sedangkan model dasar kedua menggunakan filter dengan ukuran 32 dan lapisan *dense* dengan 128 unit. Kedua arsitektur 1DCNN ini dilatih secara terpisah untuk menangkap pola lokal dan temporal yang mungkin tidak terdeteksi oleh metode konvensional. Pendekatan ini memungkinkan masing-masing model memperoleh representasi fitur yang mendalam dan berbeda, yang kemudian digunakan sebagai input fitur meta untuk model selanjutnya.



Gambar 3.3. Arsitektur model hibrida 1DCNN-XGBoost

Sumber: [2]

Setelah proses pelatihan model dasar, dilakukan penggabungan hasil prediksi atau representasi fitur melalui pendekatan *generalized stacking*. Pada tahap ini, fitur meta yang diperoleh dari kedua 1DCNN disusun menjadi data baru yang kemudian digunakan untuk melatih model meta menggunakan XGBoost. Metode *stacking* ini memungkinkan model meta XGBoost untuk belajar dari kesalahan yang dilakukan oleh model dasar, sehingga mampu mengoreksi prediksi yang kurang tepat dan meningkatkan akurasi keseluruhan.

Optimasi hiperparameter diterapkan dalam model meta XGBoost yang dibuat. Optimasi dilakukan oleh *GridSearchCV*, dengan parameter yang coba diatur adalah *n_estimators*, *learning_rate*, *max_depth*, *subsample*, *colsample_bytree*, *min_child_weight*, dan *gamma* untuk mengoptimalkan akurasi klasifikasi model meta XGBoost dalam pengambilan keputusan akhir. Dengan demikian, model hibrida ini tidak hanya memanfaatkan kekuatan *deep learning* untuk ekstraksi fitur, tetapi juga kemampuan XGBoost dalam menangani interaksi fitur yang kompleks dan mengoptimalkan kinerja prediksi.

3.1.6 Pengujian dan Evaluasi Model

Evaluasi model dilakukan dengan membagi data HMDA menjadi 80% data latih dan 20% data uji. Setelah proses pelatihan selesai, performa model hibrida 1DCNN-XGBoost serta dua model tunggal, yaitu XGBoost dan 1DCNN, diukur menggunakan metrik evaluasi seperti *accuracy*, *precision*, *recall*, *f1-score*, *macro average*, *weighted average*, *AUC*, dan *h-score*. Untuk model hibrida, evaluasi didasarkan pada prediksi akhir dari XGBoost sebagai model meta. Seluruh hasil dinyatakan dalam rentang 0 hingga 1 dengan empat angka desimal untuk memungkinkan perbandingan performa yang lebih spesifik.

3.1.7 Analisis Kontribusi Fitur

Analisis kontribusi fitur dalam penelitian ini dilakukan menggunakan SHAP untuk menginterpretasikan pengaruh fitur pada model hibrida 1DCNN-XGBoost dalam dua tingkat, yakni lokal dan global. Penentuan kontribusi fitur akan melibatkan ekstraksi nilai SHAP dari seluruh proses pelatihan, mencakup dua model dasar (1DCNN untuk ekstraksi fitur) dan XGBoost sebagai model meta. Proses ini menggunakan *KernelExplainer*, yang cocok untuk model bersifat kotak hitam seperti metode *ensemble*, untuk menghitung kontribusi setiap fitur terhadap prediksi akhir.

Pada level individu, *waterfall plot* diterapkan untuk menjelaskan kontribusi fitur pada sampel spesifik, yakni sampel pertama. Fitur seperti ras atau jenis kelamin dapat dijelaskan bagaimana nilainya meningkatkan atau mengurangi probabilitas prediksi. Sementara itu, *beeswarm plot* digunakan untuk analisis global, menggambarkan distribusi nilai SHAP seluruh fitur pada data. Plot ini mengungkap pola fitur dominan yang paling berpengaruh terhadap keputusan model dan interaksi antar-fitur yang memengaruhi prediksi, yang ditunjukkan oleh gradien warna SHAP. Kedua visualisasi ini membantu mengidentifikasi potensi bias jenis kelamin dan ras dalam model.