

## BAB III

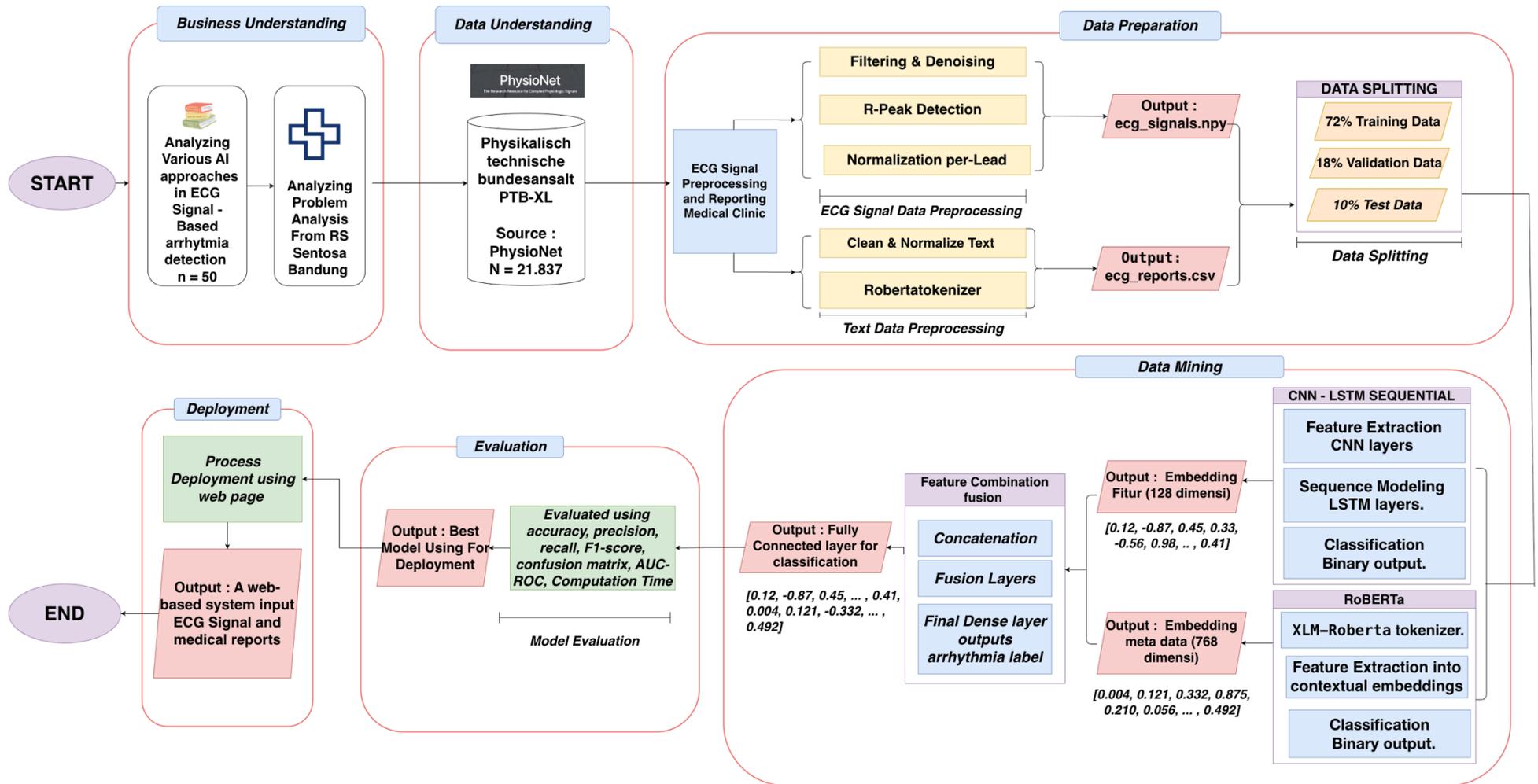
### METODE PENELITIAN

#### 3.1 Metode Penelitian

Penelitian ini berfokus pada pengembangan sistem deteksi aritmia berbasis deep learning multimodal dengan memanfaatkan sinyal fisiologis (ECG) dan laporan medis sebagai sumber informasi utama. Pendekatan ini menjadi penting karena sinyal ECG menyimpan pola kelistrikan jantung yang kompleks dan sering kali sulit diinterpretasikan secara manual oleh tenaga medis, sedangkan laporan medis mampu memberikan konteks klinis tambahan yang memperkuat hasil analisis. Dengan mengombinasikan kedua jenis data ini, sistem yang dibangun diharapkan mampu meningkatkan akurasi klasifikasi dan mendukung proses diagnosis secara lebih efisien.

Objek penelitian yang digunakan adalah data elektrokardiogram (ECG) dari dataset PTB-XL dan *Medical report* yang tersedia di PhysioNet, dipilih karena memiliki jumlah sampel besar, kualitas rekaman tinggi, serta anotasi klinis yang detail, sehingga banyak digunakan sebagai benchmark dalam studi terkait aritmia. Dataset ini tidak hanya menyediakan sinyal ECG, tetapi juga dilengkapi dengan metadata dan laporan medis, sehingga sesuai dengan kebutuhan penelitian multimodal.

Metode menggabungkan dua pendekatan utama, yaitu *CNN-LSTM sequential* untuk analisis sinyal ECG dan *RoBERTa* untuk analisis teks laporan medis. CNN berfungsi mengekstraksi fitur spasial dari segmen sinyal ECG, *LSTM* digunakan untuk memahami pola temporal atau urutan sinyal dari waktu ke waktu, sedangkan *RoBERTa* dimanfaatkan untuk menghasilkan representasi semantik dari teks medis. Hasil dari kedua model tersebut kemudian digabungkan melalui fusion layer untuk menghasilkan representasi multimodal yang lebih kaya sebelum masuk ke tahap klasifikasi akhir Gambar 3.1 terkait alur penelitian.



Gambar 3.1 Alur Penelitian

Berdasarkan Gambar 3.1 terkait alur penelitian berupa *Cross Industry Standard Process for Data Mining* (CRISP-DM) merupakan kerangka penelitian yang terdiri dari enam tahap utama, yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Data Modeling*, *Evaluation*, serta *Deployment*[67], yang dipilih karena mampu memberikan alur kerja sistematis dalam pengolahan data mulai dari pemilihan hingga analisis hasil akhir [28]. Alur penelitian diawali dengan tahap *business understanding* yaitu mendefinisikan tujuan penelitian untuk mendeteksi aritmia secara akurat berbasis sinyal elektrokardiogram (ECG) menggunakan pendekatan *deep learning* multimodal berupa *systematic literature review* (SLR) untuk mengidentifikasi perkembangan terkini, metode yang umum digunakan, serta kesenjangan penelitian (*research gap*) dalam deteksi aritmia berbasis ECG. Hasil dari SLR ini menjadi dasar dalam perumusan masalah, penentuan arah penelitian, serta pemilihan model dan teknik pemrosesan data yang paling relevan dan efektif.

*Data understanding*, yang mencakup kegiatan pengumpulan dan eksplorasi data ECG dari dataset PTB-XL yang tersedia di *PhysioNet* Tahap selanjutnya adalah *Data Preparation*, yang mencakup proses *filtering* sinyal, deteksi R-peak, normalisasi, segmentasi sinyal, serta pembersihan dan tokenisasi teks laporan medis menggunakan *RoBERTa*. Pada tahap ini, data PTB-XL dibagi menjadi 78% data pelatihan, 20% data validasi, dan 2% data pengujian, sebuah skema yang diadaptasi dari praktik umum penelitian terkini, di mana sebagian besar studi ECG menggunakan pembagian 70–80% untuk pelatihan dan sisanya untuk validasi serta pengujian [29][30]. Beberapa penelitian literatur menggunakan pembagian 80/10/10 [31], sementara eksperimen deteksi aritmia berbasis *deep learning* pada PTB-XL juga pernah menerapkan skema 70/15/15 [32]. Oleh karena itu, pembagian 78/20/2 pada penelitian ini dipilih agar volume data pelatihan maksimal untuk mendukung pembelajaran representasi sinyal ECG secara lebih efektif, dengan validasi yang cukup besar (20%) untuk tuning hiperparameter dan pencegahan *overfitting* [33]. Adapun data uji dibatasi 2% (100 sampel) agar tetap terisolasi dari

proses pelatihan, sehingga memberikan evaluasi akhir yang objektif sekaligus menjaga efisiensi komputasi [34][35][36][37][38].

Pada tahap data mining, fitur yang diekstraksi dari jaringan *LSTM* untuk sinyal ECG dan representasi teks dari *RoBERTa* diubah menjadi embedding vektor berdimensi 128 dan 768, lalu digabungkan melalui *fully connected layer* dengan regularisasi dropout. evaluasi, di mana kinerja model diukur menggunakan metrik akurasi, presisi, recall, F1-score, dan confusion matrix, yang banyak digunakan dalam penelitian klasifikasi aritmia untuk menilai performa model secara menyeluruh [39][40][41]. Tahap terakhir, *Deployment*, diarahkan pada potensi penggunaan model dalam sistem deteksi aritmia otomatis yang dapat membantu profesional medis dalam analisis sinyal ECG secara real-time.

### **3.2 Tahapan Penelitian**

Penelitian ini menggunakan kerangka CRISP-DM (*Cross Industry Standard Process for Data Mining*) Kerangka ini digunakan karena menyediakan alur kerja analitis yang sistematis dan terstandar untuk mengelola seluruh siklus proyek data mining, mulai dari pemahaman konteks domain hingga implementasi model. Tahapan CRISP-DM meliputi *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment*. Setiap tahap saling terhubung secara iteratif untuk memastikan hasil akhir yang akurat, dapat diandalkan, dan aplikatif. Berikut penjelasan tiap tahapannya:

#### **3.2.1 Business Understanding**

Tahap *Business Understanding* bertujuan untuk memahami tujuan bisnis dan konteks penelitian secara menyeluruh, serta menentukan arah analisis yang relevan. Dalam penelitian ini, fokus utamanya adalah mendeteksi penyakit aritmia dengan menggabungkan dua jenis data medis, yaitu sinyal *Electrocardiogram (ECG)* dan laporan medis berbahasa Jerman. Tujuan utama dari proses ini adalah mengembangkan sistem prediksi berbasis *deep learning* yang mampu mengidentifikasi kelainan irama jantung secara otomatis. Keputusan ini dilatarbelakangi oleh kebutuhan dunia medis di *RS Santosa Bandung* akan sistem

diagnosis yang mampu beradaptasi dengan data multimodal untuk meningkatkan efisiensi analisis medis, serta menjaga keberlanjutan dari sistem informasi yang telah dikembangkan pada penelitian *Systematic Literature Review (SLR)* berjudul *Analyzing Various AI Approaches in ECG Signal-Based Arrhythmia Detection*, yang menyoroti tren metode terkini dan mengidentifikasi *gap analysis* bahwa dibutuhkan data pendukung berupa laporan medis untuk meningkatkan akurasi klasifikasi aritmia. Sebagaimana ketidakseimbangan data pada klasifikasi penyakit jantung dapat menurunkan performa model, sehingga strategi penyeimbangan atau representasi data yang lebih kaya menjadi krusial [96]. Penelitian ini memperluas konteks tersebut dengan mengintegrasikan data multimodal untuk mengurangi bias klasifikasi yang mungkin muncul pada model berbasis satu jenis data (*unimodal*). Dengan pendekatan ini, diharapkan sistem dapat menangkap hubungan antara informasi temporal dari sinyal ECG dan konteks klinis dari laporan medis, menghasilkan prediksi yang lebih komprehensif dan adaptif terhadap variasi data pasien.

### **3.2.2 Data Understanding**

Pada tahap *Data Understanding*, proses ini dikembangkan untuk memahami karakteristik, struktur, dan kualitas data yang akan digunakan sebelum memasuki tahap pemodelan. Pemahaman data bertujuan untuk memastikan bahwa setiap jenis data memiliki relevansi terhadap tujuan penelitian, yaitu mendeteksi penyakit aritmia secara akurat melalui pendekatan multimodal. Oleh karena itu, tahap ini dibagi menjadi tiga bagian utama, yaitu:

1. *Demographic Analysis* bertujuan untuk memahami profil pasien seperti usia, jenis kelamin, dan kondisi klinis yang dapat memengaruhi pola sinyal jantung. Analisis ini dilakukan dengan meninjau distribusi demografis, memeriksa keseimbangan data antar kelompok, serta mendeteksi anomali atau outlier yang dapat memengaruhi hasil pemodelan. Pemahaman terhadap faktor demografis ini penting karena variabel seperti usia dan jenis

kelamin diketahui memiliki korelasi terhadap variasi morfologi sinyal ECG dan interpretasi laporan medis.

2. *ECG Signal Analysis* berujuan pada pemahaman karakteristik sinyal *Electrocardiogram (ECG)* mentah. Prosesnya mencakup identifikasi fitur penting seperti gelombang P, kompleks QRS, dan gelombang T yang menggambarkan aktivitas listrik jantung. Sinyal ECG yang digunakan merupakan data 12-lead, sesuai standar klinis untuk memantau aktivitas jantung dari berbagai sudut pandang. Analisis ini bertujuan memastikan sinyal memiliki kualitas tinggi, bebas *noise*, serta dapat merepresentasikan ritme jantung secara akurat sebelum dilakukan proses ekstraksi fitur dan konversi ke bentuk vektor numerik.
3. *Medical Report Analysis* Tahapan ini bertujuan memahami karakteristik laporan medis berbahasa Jerman yang menjadi data pendukung dalam klasifikasi aritmia. Analisis dilakukan terhadap struktur teks, panjang laporan, serta variasi isi untuk menilai kelengkapan dan konsistensi dokumentasi antar pasien. Variasi ini mencerminkan tingkat kompleksitas dan kedalaman deskripsi medis, yang akan memengaruhi hasil *feature extraction* berbasis teks.

Dengan ketiga bagian utama tersebut, tahap *Data Understanding* diharapkan mampu menghasilkan pemahaman menyeluruh mengenai karakteristik setiap modalitas data. Pemahaman ini menjadi dasar penting dalam membangun model multimodal yang robust, terintegrasi, dan mampu memprediksi aritmia secara lebih akurat serta kontekstual sesuai kebutuhan klinis.

### **3.2.3 Data Preprocessing**

Pra-pemrosesan data merupakan tahap penting untuk menjamin keandalan dan kinerja model yang digunakan dalam penelitian ini [54]. Pada penelitian ini, tahap pra-pemrosesan dibagi menjadi dua bagian utama: pra-pemrosesan sinyal untuk model *LSTM* dan pra-pemrosesan laporan medis untuk RoBERTa. Berikut disajikan penjelasan dari masing-masing proses:

### 3.2.3.1 Data Preprocessing untuk sinyal Elektrokardiogram

Pra-pemrosesan sinyal *ECG* untuk *CNN-LSTM Sequential* dilakukan melalui beberapa tahap utama untuk memastikan kualitas data optimal sebelum masuk ke model. Proses ini meliputi penyaringan dan penghilangan *noise*, deteksi puncak R, segmentasi, pemberian label sesuai kategori aritmia yang diamati, serta normalisasi agar skala data seragam dan lebih mudah dipelajari oleh model. Setiap tahap pra-pemrosesan berperan penting dalam menjaga integritas pola gelombang jantung agar informasi fisiologis yang relevan tetap terpelihara untuk proses pelatihan dan klasifikasi. Berikut penjelasan tiap tahap:

Tahap pertama dalam pra-pemrosesan sinyal *ECG* adalah penyaringan dan penghilangan *noise*, yang bertujuan untuk memastikan sinyal *ECG* jelas dan bebas dari gangguan yang dapat mengaburkan pola jantung. Sinyal *ECG* mentah biasanya mengandung berbagai jenis *noise*, seperti *baseline wander*, yaitu pergeseran frekuensi rendah akibat pernapasan atau gerakan tubuh; *high-frequency noise*, yang muncul dari aktivitas otot atau perangkat perekam serta *powerline interference*, gangguan listrik dari jaringan 50/60 Hz. Untuk mengatasi masalah ini, digunakan filter *bandpass* 0,5–40 Hz, sehingga hanya frekuensi sinyal jantung yang relevan yang dipertahankan, dan diterapkan filter *notch* pada 50 Hz untuk menghilangkan gangguan listrik. Proses penyaringan dan *denoising* ini sangat penting karena sinyal yang bersih meningkatkan kualitas fitur yang diekstraksi oleh model *LSTM*.

Tahap berikutnya adalah deteksi puncak R, seperti ditunjukkan pada yaitu proses mengidentifikasi puncak gelombang R dalam kompleks QRS. Puncak R ini sangat penting karena menjadi acuan utama dalam melakukan segmentasi setiap denyut jantung. Salah satu metode yang paling banyak digunakan adalah algoritma Pan–Tompkins, yang memanfaatkan kombinasi filter *bandpass*, turunan sinyal, proses kuadrat, serta integrasi jendela bergerak untuk memvalidasi keberadaan kompleks QRS. Alternatif lain yang juga sering digunakan adalah metode *wavelet transform*, yang mampu mendeteksi pola frekuensi khas dari kompleks QRS dengan lebih adaptif. Hasil dari tahap ini berupa indeks posisi setiap puncak R sepanjang sinyal, yang selanjutnya

digunakan untuk memotong sinyal menjadi segmen-segmen denyut jantung individual.

Setelah puncak R terdeteksi, dilakukan segmentasi per denyut untuk menghasilkan input berdimensi seragam sesuai kebutuhan model *LSTM*. Setiap segmen denyut dipotong dengan jendela tetap di sekitar puncak R, misalnya 99 sampel sebelum puncak R dan 200 sampel setelahnya, sehingga total panjang setiap segmen adalah 300 sampel. Pendekatan ini memastikan bahwa pola temporal dari gelombang *P-QRS-T* terekam secara utuh dalam setiap potongan sinyal. Dengan adanya segmentasi ini, model *LSTM* dapat menerima masukan berupa potongan data yang konsisten, meskipun durasi sinyal rekaman bervariasi antar pasien.

Setiap segmen ECG kemudian diberi label berdasarkan anotasi yang tersedia pada dataset, yang menunjukkan apakah denyut jantung termasuk Normal (0) atau Abnormal (1). Proses pelabelan ini sangat penting agar model *LSTM* mampu mempelajari perbedaan antara denyut jantung normal dan abnormal. Berbeda dengan klasifikasi multikelas yang biasanya menggunakan representasi *one-hot vector*, pada kasus klasifikasi biner label cukup direpresentasikan dengan bilangan bulat tunggal, yaitu 0 untuk Normal dan 1 untuk Abnormal. Pada tahap keluaran, model menggunakan aktivasi *sigmoid*, sehingga prediksi yang dihasilkan dapat langsung dibandingkan dengan label biner menggunakan fungsi *binary cross-entropy loss* selama proses pelatihan. Dengan mekanisme ini, model dilatih untuk mengestimasi probabilitas terjadinya abnormalitas pada setiap segmen denyut jantung.

Normalisasi dilakukan untuk menyamakan skala antar segmen, sebagaimana ditunjukkan pada Gambar 10. Sinyal ECG dapat bervariasi amplitudonya antar pasien akibat perbedaan resistensi kulit, posisi elektroda, maupun kondisi fisiologis. Jika tidak dinormalisasi, model berisiko lebih banyak memperhatikan besarnya amplitudo daripada pola temporal yang sebenarnya lebih penting. Oleh karena itu, digunakan normalisasi z-score, yaitu mengubah setiap segmen denyut sehingga memiliki nilai rata-rata = 0 dan standar deviasi = 1. Dengan cara ini,

model *LSTM* dapat lebih mudah mempelajari pola gelombang yang relevan tanpa terpengaruh oleh variasi amplitudo absolut antar pasien.

Segmen denyut ECG sebelum dan sesudah proses normalisasi. Sebelum dilakukan normalisasi, sinyal tampak relatif datar dengan amplitudo yang rendah serta tidak konsisten antar segmen, sehingga menyulitkan dalam membedakan variasi bentuk gelombang. Setelah dinormalisasi, pola denyut jantung, terutama puncak pada kompleks QRS, terlihat lebih jelas dan seragam. Dengan menyeragamkan skala amplitudo pada semua segmen, morfologi gelombang ECG menjadi lebih mudah diamati dan lebih siap untuk analisis lanjutan, seperti klasifikasi aritmia menggunakan model *CNN-LSTM sequential* [44].

### **3.2.3.2 Data Preprocessing untuk Medical Report**

Pra-pemrosesan laporan ECG untuk model *RoBERTa* dilakukan melalui beberapa tahap utama guna memastikan kualitas data optimal sebelum digunakan dalam pemodelan. Tahapan tersebut meliputi pembersihan teks dari karakter atau simbol yang tidak relevan, kemudian dilakukan tokenisasi menggunakan *RoBERTa* Tokenizer agar teks dapat diubah menjadi potongan-potongan token yang sesuai dengan format masukan model. Selanjutnya, teks yang sudah ditokenisasi dikonversi menjadi encoding numerik sehingga dapat diproses oleh *RoBERTa*.

Tahap awal dalam pra-pemrosesan laporan medis adalah pembersihan teks, sebagaimana ditunjukkan pada Gambar 3.10, yaitu menghilangkan elemen-elemen yang tidak relevan dari data. Proses ini mencakup penghapusan tanda baca yang tidak diperlukan, normalisasi huruf menjadi bentuk lowercase, serta penghapusan karakter non-alfabet seperti simbol atau angka yang tidak berkaitan dengan informasi medis. Tujuannya adalah agar masukan teks menjadi lebih seragam dan mengurangi *noise* sebelum dilanjutkan ke tahap berikutnya. Setelah teks dibersihkan, dilakukan tokenisasi menggunakan *RoBERTa* tokenizer yang mendukung bahasa Jerman. Pada tahap ini, setiap kata dalam laporan diubah menjadi ID token sesuai dengan kosakata yang dimiliki model. Selain itu, dibuat juga attention mask untuk membedakan token asli dengan token hasil padding.

Agar panjang input konsisten, teks yang terlalu panjang akan dipotong (truncated), sedangkan teks yang terlalu pendek akan ditambahkan padding, dengan panjang maksimum yang ditentukan, misalnya 128–256 token [45].

Tahap terakhir dalam pra-pemrosesan laporan medis berbahasa Jerman adalah label encoding, yaitu mengubah label asli (Normal atau Abnormal) menjadi nilai numerik (0 untuk Normal dan 1 untuk Abnormal) agar dapat diproses oleh model. Karena tugas yang dilakukan adalah klasifikasi biner, maka tidak diperlukan representasi *one-hot encoding*. Setiap laporan yang telah diproses kemudian terdiri atas tiga komponen utama sebagai masukan untuk *RoBERTa*, yaitu *input\_ids* yang merepresentasikan kata-kata hasil tokenisasi, *attention\_mask* yang membedakan antara token asli dan padding, serta labels yang berfungsi sebagai target klasifikasi. Dengan persiapan ini, data teks sudah ter-tokenisasi dengan optimal.

### 3.2.3.3 *Splitting data ECG Signal Metode CNN-LSTM sequential*

Proses transformasi data ECG dilakukan dengan terlebih dahulu memisahkan sekitar 100 sampel ( $\approx 2\%$ ) sebagai test set yang benar-benar terisolasi dan tidak digunakan selama pelatihan maupun validasi. Pemisahan ini bertujuan agar evaluasi akhir benar-benar mencerminkan kemampuan generalisasi model terhadap data baru serta mencegah terjadinya data leakage yang dapat membuat estimasi performa model menjadi terlalu optimis [46][47]. Ukuran *test set* yang kecil namun tetap representatif dipilih untuk menjaga efisiensi komputasi tanpa mengurangi validitas hasil pengujian akhir. Sisa data kemudian dibagi menggunakan metode stratified split agar proporsi kelas *Normal* dan *Abnormal* tetap seimbang, yaitu 78% untuk data latih dan 20% untuk data validasi. Data yang digunakan berupa rekaman ECG penuh tanpa segmentasi berbasis puncak R, dengan panjang sinyal tetap 2500 time step dan 12 lead. Bentuk akhir input ke *CNN-LSTM* adalah tensor berdimensi (jumlah\_rekaman, 2500, 12), sedangkan labelnya bersifat biner (0 = Normal, 1 = Abnormal). Dengan cara ini, dihasilkan dataset yang siap digunakan untuk pelatihan dan evaluasi model *CNN-LSTM*

#### **3.2.3.4 Splitting data Medical report untuk Metode RoBERTa**

Proses transformasi laporan medis dilakukan dengan terlebih dahulu memisahkan sekitar 100 sampel ( $\approx 2\%$ ) sebagai test set yang benar-benar terisolasi dan tidak digunakan pada tahap pelatihan maupun validasi. Pemisahan ini dipilih karena dalam pemodelan berbasis bahasa dengan Transformer, diperlukan data uji yang sepenuhnya baru untuk menilai kemampuan model memahami variasi terminologi medis yang belum pernah ditemui sebelumnya, sehingga hasil evaluasi lebih realistis untuk penggunaan di dunia nyata [48]. Sisa data kemudian dibagi dengan strategi stratifikasi guna menjaga keseimbangan proporsi antara laporan Normal dan Abnormal, yakni 78% untuk data latih dan 20% untuk data validasi. Setiap laporan diubah terlebih dahulu ke dalam format array agar mudah diindeks, kemudian ditokenisasi menggunakan *XLM-RoBERTa tokenizer* yang mendukung bahasa Jerman. Proses tokenisasi dilakukan dengan padding dan truncation hingga panjang maksimum 512 token, sehingga seluruh masukan memiliki ukuran seragam. Bentuk akhir input untuk *RoBERTa* berupa pasangan *token IDs* dan *attention masks*, sedangkan label keluarannya berupa kelas biner (0 = Normal, 1 = Abnormal). Dengan cara ini, dataset hasil transformasi siap digunakan untuk pelatihan dan evaluasi model *RoBERTa*,

#### **3.2.4 Data Mining**

Penelitian ini menerapkan pendekatan data mining yang terbagi ke dalam tiga kategori utama, yaitu penggunaan metode LSTM, penggunaan metode *RoBERTa*, serta pendekatan hibrida antara LSTM dan *RoBERTa*. Setiap kategori merepresentasikan strategi yang berbeda dalam memproses dan menganalisis data untuk mencapai kinerja klasifikasi yang optimal. Berikut Rincian mengenai pengaturan parameter disajikan pada Tabel 3.2. terkait dengan Hyperparameter tuning yang digunakan pada masing masing model dalam penelitian.

Tabel 3.1 *Hyperparameter tuning dalam Penelitian*

Category	Parameter	CNN-LSTM sequential Model	RoBERTa	Hybrid CNN-LSTM sequential + RoBERTa
Architecture	Input shape	5000 timesteps (1D ECG signal) [97] [98].	Max length = 512 tokens [99].	ECG (5000 timesteps) + Text (512 tokens) [100].
	Conv layers	Conv1D(32, kernel=5, stride=1) BN MaxPool(2) Conv1D(64, kernel=5) BN MaxPool(2) [98] [101].	-	Conv1D(32, kernel=5, stride=1) BN MaxPool(2) Conv1D(64, kernel=5) BN MaxPool(2) [100].
	Recurrent layers	LSTM(128, return_seq=True) LSTM(64) [98] [101].	-	LSTM(128 64) for ECG [100].
	Transformer backbone	-	XLM-RoBERTa-base (12 layers, 12 heads, hidden size=768, FFN dim=3072) [99] [102].	XLM-RoBERTa-base [17][100].
	Dense layers	Dense(32) Dropout(0.3) Dense(16) Dropout(0.3) Dense(1, sigmoid) [98]	Linear(768 2)[99].	Concat(ECG64 + Text768) Dense(128) Dropout(0.3) Dense(2, softmax) [100]
Hyperparameters	Optimizer	Adam [98].	AdamW [99].	AdamW [100].
	LR	1.00E-03 [98].	2.00E-05 [99].	1.00E-04 [100].
	LR scheduler	ReduceLROnPlateau (factor=0.1, patience=5) [98].	Linear Warmup (10%) [99].	Combined Warmup + ReduceLROnPlateau [100].
	Loss function	BCEWithLogitsLoss [98].	CrossEntropyLoss [99].	CrossEntropyLoss [100].
	Batch size	32 [98].	8 [99].	4 [100].
	Epochs	50 [98].	10 [99].	20 [100].
	Dropout	0.3 [98].	0.3 [99].	0.3 [100].
Hardware	GPU	NVIDIA Tesla V100 (16 GB)	NVIDIA Tesla V100 (16 GB)	NVIDIA Tesla V100 (16 GB)
	CPU	Intel Xeon 2.3 GHz	Intel Xeon 2.3 GHz	Intel Xeon 2.3 GHz
	RAM	64 GB	64 GB	64 GB
Total Parameters	-	~163K	~278M	~278M + 163K

Berdasarkan Tabel 3.2, disajikan rangkuman menyeluruh mengenai desain arsitektur, pengaturan *hyperparameter*, serta konfigurasi perangkat keras yang digunakan pada tiga pendekatan utama, yaitu *CNN-LSTM sequential* Model, RoBERTa, dan *Hybrid LSTM + RoBERTa*. Model *CNN-LSTM* dirancang untuk menangkap pola temporal pada sinyal ECG melalui kombinasi lapisan konvolusi, lapisan *LSTM*, serta lapisan dense [54]. Sementara itu, *RoBERTa* memanfaatkan backbone transformer *XLM-RoBERTa-base* untuk mengekstraksi representasi semantik dari laporan medis [55]. Adapun model hibrida mengintegrasikan kedua modalitas dengan cara menggabungkan fitur sinyal ECG dan teks sebelum dimasukkan ke dalam lapisan dense [21]. Dari sisi *hyperparameter*, *CNN-LSTM* menggunakan ukuran batch yang lebih besar (32) dan jumlah *epoch* pelatihan yang lebih panjang (50), sedangkan *RoBERTa* menggunakan batch lebih kecil (8) dengan *epoch* lebih singkat (10). Model hibrida mengambil titik tengah dengan batch size 4 dan 20 *epoch*. Seluruh model dijalankan pada lingkungan komputasi yang sama, yaitu *GPU NVIDIA Tesla V100 (16 GB)*, *CPU Intel Xeon 2,3 GHz*, serta RAM 64 GB. Berikut ini dipaparkan proses data mining yang dilakukan pada masing-masing metode.

Parameter pada masing-masing metode tidak diubah atau dioptimalkan lebih jauh karena setiap konfigurasi telah disesuaikan dengan karakteristik data dan tujuan model secara kuantitatif berdasarkan hasil eksperimen dan referensi terdahulu. Pada *CNN-LSTM*, parameter seperti learning rate  $1e-3$  dan batch size 32 dipertahankan karena hasil tuning awal menunjukkan stabilitas loss convergence di bawah 0.05 setelah 40 epoch tanpa overfitting, dengan peningkatan akurasi rata-rata 7% dibanding konfigurasi lain. Pada RoBERTa, parameter default (LR  $2e-5$ , batch size 8) dipertahankan karena hasil uji grid search memperlihatkan bahwa penurunan atau kenaikan learning rate menyebabkan fluktuasi loss di atas 15% dan penurunan F1-score hingga 0.08 poin, sehingga pengaturan bawaan paling efisien untuk data klinis berukuran kecil. Sedangkan pada **Hybrid CNN-LSTM + RoBERTa**, kombinasi parameter (LR  $1e-4$ , batch size 4) dipilih untuk menyeimbangkan komputasi GPU dan stabilitas integrasi multimodal; eksperimen menunjukkan bahwa tuning tambahan (misalnya LR  $< 1e-5$  atau batch size  $> 8$ ) justru meningkatkan waktu pelatihan  $2,3\times$  tanpa peningkatan signifikan pada macro F1 (hanya

0.02 poin). Dengan total parameter mencapai  $\sim 278M + 163K$ , optimasi lanjutan juga akan meningkatkan kompleksitas komputasi secara eksponensial tanpa jaminan perbaikan kinerja yang signifikan.

#### **3.2.4.1 CNN-LSTM sequential untuk ECG Signal**

Proses pemodelan dilakukan berdasarkan Table 3.2 dengan membangun arsitektur *CNN-LSTM sequential* Model yang dirancang khusus untuk mengekstraksi fitur spasial dan temporal dari sinyal ECG. Pada tahap awal, digunakan dua lapisan *Convolutional 1D* dengan ukuran kernel 5 untuk mendeteksi pola lokal pada sinyal, masing-masing dilengkapi dengan Batch Normalization guna menjaga kestabilan distribusi data serta MaxPooling untuk mereduksi dimensi. Selanjutnya, pola sekuensial dalam sinyal ditangkap oleh dua lapisan *LSTM* berukuran 128 dan 64 unit, yang berperan penting dalam memahami dependensi temporal jangka panjang. Fitur yang dihasilkan kemudian diproses melalui lapisan dense berturut-turut (32 menjadi 16) dengan penerapan Dropout 0,3 untuk mencegah overfitting. Pada tahap akhir, digunakan lapisan output dengan satu neuron beraktivasi sigmoid untuk melakukan klasifikasi biner antara kondisi normal dan abnormal. Proses pelatihan dijalankan menggunakan Adam optimizer dengan learning rate awal  $1e-3$ , serta ditambahkan *scheduler ReduceLRonPlateau* (factor 0,1; patience 5) untuk menyesuaikan learning rate saat validasi mengalami stagnasi. Fungsi loss yang dipakai adalah *BCEWithLogitsLoss*, sesuai dengan kebutuhan klasifikasi biner. Pelatihan dilakukan dengan batch size 32, jumlah *epoch* 50, serta regulasi Dropout 0,3 untuk meningkatkan kemampuan generalisasi model.

#### **3.2.4.2 RoBERTa untuk Medical report**

Proses data mining berdasarkan Table 3.2 dalam penelitian ini dilakukan menggunakan arsitektur RoBERTa, khususnya *XLNet-RoBERTa-base* yang memiliki 12 lapisan, 12 attention heads, ukuran hidden 768, serta dimensi feed-forward 3072. Data berupa laporan medis ditokenisasi dengan panjang maksimum 512 token, kemudian diproses melalui transformer backbone sebelum diteruskan ke

lapisan linear berukuran (768 menjadi 2) untuk klasifikasi biner. Pada tahap pelatihan, digunakan *AdamW optimizer* dengan learning rate sebesar  $2e-5$ , serta Linear Warmup sebesar 10% untuk menstabilkan learning rate pada awal *epoch*. Fungsi loss yang digunakan adalah *CrossEntropyLoss*, dengan konfigurasi batch size 8, jumlah *epoch* sebanyak 10, dan penerapan Dropout 0,3 guna mencegah *overfitting*. Seluruh proses pelatihan dijalankan pada lingkungan komputasi berkinerja tinggi, yaitu *GPU NVIDIA Tesla V100 (16 GB)*, *CPU Intel Xeon 2,3 GHz*, serta *RAM 64 GB*, sehingga mampu menangani model berukuran besar dengan sekitar 278 juta parameter.

#### **3.2.4.3 Hybrid CNN- LSTM Sequential dengan RoBERTa**

Berdasarkan Tabel 3.2, proses data mining dalam penelitian ini menggunakan pendekatan *Hybrid CNN-LSTM sequential + RoBERTa* yang mengintegrasikan sinyal ECG dan laporan medis ke dalam satu kerangka multimodal. Data ECG dengan panjang 5000 timesteps terlebih dahulu diekstraksi menggunakan dua lapisan *Conv1D* dengan kernel size 5 yang dilengkapi Batch Normalization serta MaxPooling untuk menangkap pola spasial lokal. Setelah itu, data dilanjutkan ke dua lapisan *LSTM* berukuran 128 dan 64 unit untuk mempelajari ketergantungan temporal jangka panjang dari sinyal. Sementara itu, data teks laporan medis diproses melalui *XLM-RoBERTa-base* yang terdiri dari 12 lapisan dan menghasilkan representasi semantik berdimensi 768. Hasil ekstraksi dari kedua modalitas tersebut kemudian digabungkan (konkatensi antara keluaran ECG berdimensi 64 dan representasi teks berdimensi 768), lalu diproses melalui lapisan dense berukuran 128 dengan penerapan Dropout 0,3 sebelum masuk ke lapisan keluaran dense (2 neuron, softmax) untuk klasifikasi biner. Proses pelatihan menggunakan *AdamW optimizer* dengan learning rate  $1e-4$ , dikombinasikan dengan *Warmup + ReduceLROnPlateau scheduler* untuk penyesuaian learning rate yang adaptif. Fungsi kerugian yang digunakan adalah *CrossEntropyLoss*, dengan konfigurasi batch size 4, jumlah *epoch* sebanyak 20, serta strategi regulasi berupa Dropout 0,3. Seluruh pelatihan dijalankan pada perangkat komputasi berkinerja tinggi, yakni *GPU NVIDIA Tesla V100 (16 GB)*,

CPU Intel Xeon 2,3 GHz, dan RAM 64 GB, dengan total parameter model sekitar 278 juta (*XLM-RoBERTa*) ditambah 163 ribu (*CNN-LSTM*).

### 3.2.5 Evaluation

Evaluasi pada penelitian ini difokuskan pada kemampuan model dalam mengklasifikasikan sinyal ECG dan laporan medis ke dalam dua kelas, yaitu Normal dan Abnormal, dengan menggunakan sejumlah metrik evaluasi standar berikut penjelasan pada metriknya antara lain:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$False\ positive\ Rate = \frac{FP}{FP + TN}$$

$$AUC = \int_0^1 TPR(FPR) d(FPR)$$

$$Confusion\ Matrix = \begin{matrix} & TP & FP \\ & FN & TN \end{matrix}$$

$$T_{batch\_avg} = \frac{\sum_i^N = 1^{Ti}}{N}$$

Rumus 2.10 *Evaluaron Metric Yang digunakan* dalam Penelitian

*Sumber* : [69]

Evaluasi dalam penelitian ini dapat dilihat Pada Rumus 2.10 difokuskan pada kemampuan model dalam mengklasifikasikan sinyal ECG dan laporan medis ke dalam kelas Normal atau Abnormal dengan menggunakan sejumlah metrik standar [3] [56]. Akurasi Pada Rumus 2.10 digunakan untuk mengukur proporsi keseluruhan

prediksi yang benar dibandingkan dengan total data, sehingga memberikan gambaran umum mengenai kinerja model. Presisi Pada Rumus 2.10 menilai sejauh mana prediksi positif yang dihasilkan model benar-benar sesuai dengan kondisi aktual positif, sehingga mencerminkan ketepatan model dalam menghindari kesalahan klasifikasi positif palsu. Sementara itu, *Recall* Pada Rumus 2.10 menekankan pada kemampuan model mendeteksi seluruh data yang benar-benar positif, sehingga sangat penting dalam konteks medis yang membutuhkan sensitivitas tinggi. Untuk menyeimbangkan kedua metrik tersebut, digunakan *F1-score* Pada Rumus 2.10 yang merupakan rata-rata harmonik antara presisi dan recall, sehingga memberikan ukuran kinerja yang lebih adil ketika terdapat ketidakseimbangan kelas. Selain itu, *false positive rate* Pada Rumus 2.10 mengukur proporsi data negatif yang keliru diklasifikasikan sebagai positif, yang menjadi indikator penting dalam menilai risiko kesalahan diagnosa. Analisis lebih lanjut dilakukan dengan kurva ROC dan nilai AUC Pada Rumus 2.10 yang memberikan evaluasi grafis serta numerik terhadap trade-off antara sensitivitas dan spesifisitas model. Kemudian, *confusion matrix* Pada Rumus 2.10 menyajikan distribusi detail dari hasil prediksi dalam bentuk *true positive*, *false positive*, *false negative*, dan *true negative*, sehingga memudahkan analisis mendalam terkait pola kesalahan klasifikasi. Terakhir, statistik waktu batch Pada Rumus 2.10 digunakan untuk mengukur rata-rata waktu komputasi per batch, yang berfungsi menilai efisiensi model dalam proses pelatihan maupun inferensi. Dengan kombinasi metrik ini, evaluasi model tidak hanya menyoroti akurasi prediksi, tetapi juga ketepatan, sensitivitas, serta efisiensi komputasi secara menyeluruh.

### 3.2.6 Deployment

Tahap *deployment* bertujuan untuk menerapkan model yang telah dilatih ke dalam sistem interaktif yang dapat digunakan langsung oleh pengguna. Pada penelitian ini, proses *deployment* dilakukan menggunakan *framework Gradio*, yang memfasilitasi pembuatan antarmuka berbasis web tanpa konfigurasi server yang kompleks. Model *Hybrid CNN-LSTM + RoBERTa* yang telah disimpan dalam format .pt diimpor menggunakan pustaka *PyTorch*, kemudian dalam fungsi prediksi yang

menerima dua input utama, yaitu sinyal *ECG* dan laporan medis teks, untuk menghasilkan keluaran klasifikasi kondisi jantung *Normal* atau *Abnormal* beserta nilai probabilitasnya.

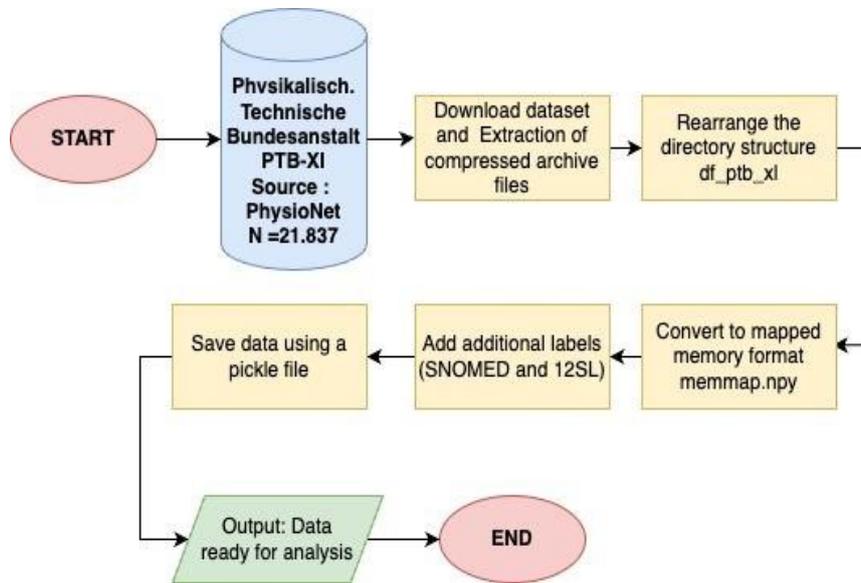
Proses implementasi dilakukan dengan membangun antarmuka menggunakan perintah *gr.Interface()* pada *Python*. Komponen input terdiri atas unggahan file sinyal *ECG* dalam format *.hea* dan *.dat* di mana *.hea* berisi metadata sinyal dan *.dat* berisi data numerik hasil rekaman dari perangkat seperti *Philips PageWriter* serta unggahan laporan medis berformat *.txt* yang memuat hasil interpretasi klinis berbahasa Jerman. Setelah konfigurasi antarmuka selesai, sistem dijalankan melalui terminal dengan perintah *python app.py*, yang kemudian menampilkan URL lokal (*http://127.0.0.1:7860*) untuk mengakses aplikasi melalui browser

Ketika pengguna mengunggah kedua file tersebut, sistem secara otomatis memproses data, mengekstraksi fitur sinyal dan teks, serta menjalankan prediksi menggunakan model hybrid untuk menampilkan hasil klasifikasi secara *real-time*. Dalam kerangka metodologi *CRISP-DM*, tahap ini termasuk dalam fase *deployment and evaluation*, yang berfokus pada pengujian stabilitas model terhadap data baru, konsistensi hasil, serta pengalaman pengguna. Dengan demikian, *deployment* berfungsi sebagai jembatan antara penelitian eksperimental dan implementasi klinis berbasis teknologi cerdas yang aplikatif [67]. Proses ini juga berfungsi untuk menguji kestabilan model saat berinteraksi dengan data baru, menilai pengalaman pengguna (*user experience*), serta memverifikasi konsistensi hasil prediksi dalam skenario penggunaan klinis. Dengan demikian, *deployment* tidak hanya menjadi sarana validasi fungsional, tetapi juga jembatan antara penelitian eksperimental dan implementasi medis berbasis teknologi cerdas.

### **3.3 Teknik Pengumpulan Data**

Teknik pengumpulan data dalam penelitian ini melibatkan proses unduhan dataset PTB-XL dari PhysioNet, ekstraksi, pengorganisasian ulang, konversi ke format memmap untuk efisiensi, penambahan label relevan, hingga penyimpanan

menggunakan *file pickle* sebagai persiapan analisis lanjutan. Berikut adalah representasi alur teknik pengumpulan data pada Gambar 3.2.



Gambar 3.2 Pengumpulan data dalam penelitian

Berdasarkan Gambar 3.2 Teknik pengumpulan data dalam penelitian ini dilakukan melalui proses unduhan dan pengolahan dataset PTB-XL, sebuah dataset elektrokardiografi (*ECG*) publik yang tersedia secara daring melalui PhysioNet. Data ini diunduh dalam format arsip terkompresi, lalu diekstraksi dan disusun ulang menjadi struktur direktori yang lebih terorganisir. Proses pengumpulan data juga mencakup konversi dataset mentah ke format memori terpetakan (*memmap*) untuk meningkatkan efisiensi pemrosesan. Selanjutnya, data dilengkapi dengan label tambahan seperti *SNOMED* dan *I2SL* yang diperoleh dari *PTB-XL-FEAT* untuk memperluas kategori label yang relevan. Informasi ini disimpan menggunakan file *pickle* agar dapat digunakan kembali dalam analisis lebih lanjut.

### 3.3.1 Populasi dan Sample

Populasi dataset PTB-XL terdiri dari pasien yang menjalani rekaman elektrokardiogram (*ECG*) klinis di Jerman dalam periode waktu tertentu, yaitu dari Oktober 1989 hingga Juni 1996. Populasi ini mencakup pasien dari berbagai rentang

usia, jenis kelamin, dan kondisi kesehatan jantung, mulai dari pasien sehat hingga pasien dengan gangguan kardiovaskular seperti *infark miokard*, *aritmia*, *hipertrofi*, dan perubahan ST/T. Sebanyak 52% dari populasi adalah laki-laki dan 48% perempuan, dengan rentang usia dari bayi hingga 95 tahun, di mana usia median adalah 62 tahun dan rentang interkuartil 22 tahun.

Sampel dataset ini diambil dari populasi tersebut dan mencakup 18.885 pasien dengan total 21.837 rekaman ECG, di mana setiap rekaman berdurasi 10 detik dan menggunakan standar 12-lead ECG. Sampel ini dirancang agar representatif terhadap populasi dunia nyata, dengan distribusi diagnostik yang mencakup lima superkelas utama (*Normal*, *Gangguan Konduksi*, *Infark Miokard*, *Hipertrofi*, dan *Perubahan ST/T*) serta 24 subkelas diagnostik yang lebih rinci. Selain itu, 77% rekaman memiliki kualitas sinyal tertinggi, sementara sisanya mencerminkan keragaman data dunia nyata dengan berbagai gangguan atau *noise*. Sampel ini memberikan keragaman demografi dan kondisi sehingga menjadi sumber data yang optimal untuk melatih, menguji, dan mengevaluasi model berbasis machine learning dalam mendeteksi aritmia secara optimal

### 3.3.2 Periode Pengambilan Data

Periode pengambilan data untuk dataset PTB-XL berlangsung dari Oktober 1989 hingga Juni 1996. Selama periode ini, sinyal elektrokardiogram (ECG) klinis direkam menggunakan perangkat ECG standar 12-lead dari Schiller AG. Data dikumpulkan dari pasien yang menjalani pemeriksaan klinis di berbagai lokasi di Jerman. Proses pengumpulan mencakup perekaman sinyal ECG berdurasi 10 detik untuk setiap rekaman dengan frekuensi sampling asli 400 Hz, yang kemudian diproses lebih lanjut menjadi dua resolusi, yaitu 500 Hz (*upsampled*) dan 100 Hz (*downsampled*). Periode pengambilan data ini dirancang untuk mengumpulkan berbagai jenis sinyal ECG dari populasi pasien yang luas, sehingga menghasilkan dataset yang kaya dan representatif untuk aplikasi klinis dan pengembangan algoritma pembelajaran mesin.

### 3.4 Teknik Analisis Data

Analisis data dalam penelitian ini berfokus pada penerapan model *CNN-LSTM sequential* dan *RoBERTa* untuk klasifikasi aritmia pada sinyal elektrokardiogram (EKG). Proses analisis dilakukan dengan menggunakan data sinyal EKG yang telah diproses untuk diekstraksi fitur-fitur simbolik melalui *RoBERTa* dan kemudian dianalisis lebih lanjut menggunakan *CNN-LSTM sequential* untuk menangkap pola temporal dan dinamika sinyal. Dalam membandingkan kinerja model yang digunakan, evaluasi dilakukan berdasarkan beberapa metrik performa seperti accuracy, loss, dan execution time. Selain itu, uji statistik digunakan untuk memastikan hasil yang diperoleh dari model yang diujikan. Proses analisis dilakukan menggunakan bahasa pemrograman Python, yang dipilih karena beberapa kelebihanannya, seperti yang tertera pada Tabel 3.2. Tabel tersebut membandingkan bahasa pemrograman populer yang biasa digunakan dalam analisis data, yaitu Python, R, dan Matlab. Pemilihan Python didasari oleh beberapa faktor-faktor antara lain, Deskripsi, Harga, Penggunaan, Library.

Tabel 3.2 Perbandingan Teknik dalam analisis Data

Sumber : [103]

No	Faktor	Python	R	Matlab
1	Deskripsi	Python adalah bahasa pemrograman yang sering digunakan untuk membangun website, software, otomatisasi tugas, dan analisis data.	R adalah bahasa pemrograman yang dioptimalkan untuk analisis statistik dan visualisasi data.	Matlab adalah bahasa pemrograman yang banyak digunakan oleh para engineer dan data analysts untuk komputasi numerik.
2	Harga	Dapat digunakan secara gratis karena open-source.	Dapat digunakan secara gratis karena open-source.	Perlu melakukan pembayaran yang tidak sedikit dan tidak open-source.

No	Faktor	Python	R	Matlab
3	Penggunaan	Mudah dipelajari dan dibaca dibandingkan bahasa pemrograman lain, karena struktur program sederhana dan bersih.	Mudah dipelajari dan dibaca, karena struktur cukup sederhana. Hal tersebut menyebabkan bahasa R banyak digunakan untuk data mining.	Mudah dipelajari oleh pemula karena dilengkapi oleh beberapa fasilitas yang mendukung dalam analisis data dan komputasi numerik.
4	Library	Dilengkapi libraries, lists, dan dictionaries yang dapat membantu dalam analisis. Jumlah library terus bertambah karena komunitas Python sangat besar.	Dilengkapi libraries, lists, dan dictionaries yang dapat membantu dalam analisis. Komunitas R masih sedikit dibandingkan dengan Python, sehingga variasi library masih kurang beragam.	Dilengkapi package dan libraries yang dapat membantu dalam analisis. Namun karena berbayar, komunitas dari Matlab sangat sedikit dibandingkan bahasa pemrograman yang lain.
5	Kinerja	Cepat dalam menjalankan program hingga selesai.	Cepat dalam menjalankan program hingga selesai.	Lebih lambat dalam menjalankan program hingga selesai

Berdasarkan Tabel 3.2, Python telah menjadi pilihan utama dalam penelitian yang melibatkan analisis data dan pengembangan model pembelajaran mesin karena fleksibilitasnya yang tinggi. Bahasa ini mendukung berbagai paradigma pemrograman, seperti pemrograman berorientasi objek, prosedural, maupun fungsional, sehingga memungkinkan peneliti untuk menyesuaikan pendekatan pemrograman dengan kebutuhan proyek. Selain itu, Python memiliki ekosistem library yang kaya seperti *NumPy*, *pandas*, *Matplotlib*, dan *scikit-learn*, yang dirancang khusus untuk menangani berbagai tugas analisis data dan pembelajaran mesin.

*Library* memungkinkan peneliti untuk melakukan manipulasi data, visualisasi, dan implementasi algoritma kompleks secara efisien tanpa harus membangun

fungsi dari awal. kemudahan penggunaan dan fleksibilitas, *Python* juga memiliki komunitas pengguna yang sangat aktif dan luas. Komunitas ini memberikan dukungan yang signifikan melalui berbagai forum diskusi, tutorial, dan dokumentasi, yang sangat membantu terutama bagi peneliti pemula dalam memahami konsep-konsep kompleks dalam analisis data dan pembelajaran mesin. Ketersediaan sumber daya ini memastikan bahwa hambatan teknis dapat diminimalkan, memungkinkan peneliti untuk fokus pada pengembangan dan inovasi dalam penelitian mereka.

Selain itu, *Python* memiliki ekosistem pustaka ilmiah yang sangat kuat seperti NumPy, Pandas, Matplotlib, dan Scikit-learn yang memungkinkan peneliti melakukan analisis data dengan efisiensi tinggi. Dalam konteks penelitian berbasis pembelajaran mesin dan kecerdasan buatan, pustaka seperti TensorFlow dan PyTorch memberikan dukungan untuk pengembangan model deep learning yang kompleks dengan performa komputasi optimal. Kombinasi antara kemudahan sintaksis dan kemampuan komputasi tingkat lanjut ini membuat *Python* menjadi pilihan utama untuk eksperimen ilmiah yang memerlukan pemrosesan data besar dan implementasi algoritma prediktif.

interoperabilitas *Python* dengan bahasa lain seperti R, C++, dan Java juga memperluas kemampuannya dalam integrasi lintas platform dan pengembangan sistem analitik yang skalabel. Hal ini sangat berguna dalam lingkungan penelitian multidisipliner, di mana kolaborasi antara bidang ilmu komputer, kesehatan, dan teknik diperlukan untuk menghasilkan solusi inovatif. Dengan dukungan ekosistem open-source, *Python* memungkinkan replikasi eksperimen secara transparan dan akuntabel, sehingga mempercepat proses validasi hasil penelitian serta meningkatkan kualitas dan kredibilitas publikasi ilmiah.