

BAB 2

TINJAUAN PUSTAKA

2.1 Gitar

Gitar adalah alat musik yang sangat populer, digunakan dalam berbagai genre musik. Ini merupakan instrumen senar yang dimainkan dengan memetik, umumnya menggunakan jari. Struktur gitar terdiri dari badan, leher, dan kepala, dengan senar yang direntangkan di antaranya. Gitar memiliki beragam jenis dan ukuran, mulai dari gitar klasik dengan senar nilon hingga gitar elektrik dengan senar baja.

Gitar berfungsi tidak hanya sebagai alat musik untuk melodi, tetapi juga sering digunakan untuk harmoni dan ritme. Pemahaman dasar tentang chord, skala, dan teknik bermain merupakan inti dari pembelajaran gitar. Gitar telah menjadi bagian penting dalam musik populer, berperan dalam pembentukan dan ekspresi musikal. Dengan bermain gitar dapat lebih meningkatkan personal branding di era digital saat ini banyak para masyarakat umum yang mendadak terkenal dengan memainkan alat musik seperti gitar [4].

2.2 Chord

Chord adalah harmonisasi tiga nada atau lebih yang dibunyikan secara bersamaan [8]. Dalam konteks pembelajaran gitar, memahami chord adalah fundamental karena hampir semua komposisi musik melibatkan penggunaan berbagai chord untuk menciptakan harmoni. Chord, yang merupakan kumpulan dari tiga atau lebih nada yang dibunyikan secara bersamaan, memiliki variasi seperti mayor dan minor, masing-masing dengan karakter suara yang unik. Pembelajaran chord melibatkan pengenalan posisi jari yang tepat di fret gitar dan transisi antar chord yang efisien. Penggunaan aplikasi pembelajaran gitar berbasis Android dapat memanfaatkan teknologi seperti feedback audio visual untuk membantu pemula mempelajari dan memperbaiki penggunaan chord secara interaktif.

2.3 Realtime Audio Analysis

Realtime Audio Analysis merupakan teknologi yang memungkinkan pengolahan dan analisis data audio secara langsung. Persyaratan dasar dalam aplikasi realtime adalah waktu respons harus sekecil mungkin dan keamanan harus berada pada tingkat yang lebih tinggi [9]. Teknologi ini berperan penting dalam aplikasi yang memerlukan respon cepat terhadap input audio, seperti dalam aplikasi pembelajaran musik atau pengolahan suara. Dalam konteks pembelajaran gitar, analisis audio realtime dapat digunakan untuk memberikan umpan balik instan pada akurasi penyetelan dan permainan chord.

Teknologi ini mengandalkan algoritma canggih untuk mengidentifikasi dan membedakan berbagai aspek suara. Realtime Audio Analysis memungkinkan aplikasi untuk menangkap dan menanggapi nuansa kecil dalam permainan musik, membantu pemula memperbaiki kesalahan secara tepat waktu. Penggunaan teknologi ini dalam aplikasi pembelajaran musik meningkatkan kualitas pembelajaran, memungkinkan pengalaman yang lebih interaktif dan mendalam.

2.4 Audio Processing

Audio Processing adalah proses manipulasi atau pengolahan sinyal audio untuk meningkatkan kualitas atau mengekstrak informasi tertentu. Proses ini termasuk pengurangan noise, ekualisasi, dan penguatan suara. Ada banyak aplikasi pengolah audio dan jumlahnya terus bertambah [10]. Dalam konteks aplikasi pembelajaran gitar, audio processing digunakan untuk menyaring dan memperjelas suara gitar, memudahkan identifikasi chord.

Teknik audio processing memainkan peran penting dalam mengembangkan aplikasi yang dapat secara akurat mengenali dan membedakan suara gitar. Pengolahan suara canggih membantu dalam memberikan umpan balik yang jelas dan tepat kepada pengguna. Aplikasi yang memanfaatkan teknologi ini memberikan pengalaman pembelajaran yang lebih efektif, membantu pemula dalam mengenali dan memperbaiki kesalahan secara real-time.

2.5 Aplikasi

Aplikasi adalah program perangkat lunak yang dirancang untuk membantu pengguna melakukan satu set tugas tertentu. Aplikasi modern dibuat dengan berbagai tujuan, mulai dari produktivitas hingga hiburan. Dalam konteks pembelajaran musik, aplikasi digunakan untuk menyediakan alat bantu yang interaktif dan mudah diakses.

Pembangunan aplikasi untuk pembelajaran gitar melibatkan integrasi berbagai teknologi, termasuk analisis audio dan visualisasi. Aplikasi ini dirancang untuk memudahkan proses pembelajaran, memberikan materi instruksional, latihan, dan umpan balik instan. Aplikasi pembelajaran gitar menjadi semakin populer karena kemudahannya dalam memberikan akses ke sumber belajar kapan saja dan di mana saja. Aplikasi bertema musik banyak ditemukan di smartphone, termasuk yang berbasis Android [3].

2.6 Android

Seiring dengan berkembangnya teknologi, penggunaan smartphone semakin sering kita jumpai [2]. Android adalah sistem operasi yang paling banyak digunakan di perangkat mobile. Sistem operasi ini dikembangkan oleh Google dan berbasis pada kernel Linux. Android dikenal karena keflexibelannya, mendukung berbagai perangkat dengan berbagai spesifikasi dan ukuran layar.

Pembangunan aplikasi untuk Android memerlukan pemahaman mendalam tentang platform ini, termasuk cara kerjanya dan fitur-fitur yang disediakan. Android menyediakan lingkungan pengembangan yang kaya, dengan dukungan untuk berbagai alat dan bahasa pemrograman. Ini memungkinkan pengembang untuk menciptakan aplikasi yang responsif dan dapat diakses oleh jutaan pengguna di seluruh dunia.

2.7 Fast Fourier Transform (FFT)

Untuk mengubah domain waktu ke domain frekuensi membutuhkan algoritma FFT (Fast Fourier Transform) [11]. FFT memungkinkan identifikasi dan analisis komponen frekuensi dalam sinyal audio, yang berguna untuk berbagai aplikasi, termasuk pengenalan chord gitar. Algoritma ini dikenal karena efisiensinya dalam melakukan transformasi frekuensi, membuatnya ideal untuk aplikasi real-time. Dalam konteks aplikasi pembelajaran musik, FFT membantu mengubah sinyal audio yang kompleks menjadi data yang lebih mudah dianalisis. Ini memungkinkan aplikasi untuk secara akurat mengenali dan memproses berbagai nada dan chord yang dimainkan.

Proses penggunaan FFT dalam aplikasi ini dimulai dengan menangkap suara gitar melalui mikrofon perangkat Android. Sinyal audio yang diterima kemudian diubah ke domain frekuensi menggunakan algoritma FFT. Hasil transformasi ini adalah spektrum frekuensi yang menunjukkan berbagai frekuensi dalam sinyal audio. Penggunaan FFT memungkinkan deteksi chord secara cepat dan akurat, memberikan pengalaman belajar yang interaktif dan responsif bagi pemula.

2.8 Librosa

Librosa adalah pustaka Python yang berguna untuk analisis suara dan musik [12]. Pustaka ini populer dalam komunitas pengolahan audio karena kemudahannya dalam penggunaan dan kemampuannya untuk menangani berbagai format audio. Librosa memungkinkan ekstraksi fitur-fitur audio penting seperti pitch dan chroma. Fitur-fitur ini sangat berguna dalam aplikasi pengenalan musik dan audio. Dengan menggunakan Librosa, pengembang dapat dengan mudah mengimplementasikan berbagai teknik analisis audio yang kompleks dalam aplikasi mereka.

Untuk memulai penggunaan Librosa, pustaka Python untuk analisis suara dan musik, langkah pertama adalah menginstalnya. Instalasi Librosa dapat dilakukan dengan mudah melalui pip, manajer paket Python. Pastikan pip sudah terinstal di

sistem. Jika belum, panduan instalasi dapat ditemukan di situs resmi pip. Setelah pip tersedia, buka terminal atau command prompt dan jalankan perintah `pip install librosa`. Perintah ini akan mengunduh dan menginstal Librosa beserta semua dependensinya secara otomatis.

Setelah Librosa terinstal, pustaka ini dapat digunakan untuk menganalisis file audio. Langkah pertama adalah memuat file audio menggunakan fungsi `librosa.load(file_path)`, di mana `file_path` adalah path ke file audio yang dimaksud. Fungsi ini akan mengembalikan dua variabel `y`, yang berisi data audio dalam bentuk array NumPy, dan sample rate dari file audio tersebut. Untuk mengekstrak fitur chroma dari sinyal audio, fungsi `librosa.feature.chroma_stft(y=y, sr=sr)` dapat digunakan. Terakhir, untuk menghitung rata-rata fitur chroma, dapat digunakan `np.mean(chroma, axis=1)`. Hasilnya adalah vektor rata-rata fitur chroma yang akan digunakan dalam proses klasifikasi chord pada analisis musik.

2.9 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah seperangkat metode pembelajaran terbimbing yang menganalisis data dan mengenali pola [13]. SVM dikenal karena kemampuannya yang baik dalam menangani data dengan dimensi tinggi dan menghasilkan model klasifikasi yang kuat dan akurat. Dalam konteks pengenalan chord gitar, SVM digunakan untuk mengklasifikasikan fitur-fitur audio yang diekstraksi dari sinyal suara menjadi chord-chord tertentu. Algoritma ini sangat efektif dalam mengenali pola dan membuat prediksi berdasarkan data yang ada. Dengan menggunakan SVM, aplikasi dapat memberikan hasil klasifikasi chord yang akurat dan andal berdasarkan data audio yang diterima.

Untuk menerapkan Support Vector Machine (SVM) dalam pengenalan chord gitar, langkah pertama adalah menginstal pustaka yang diperlukan. Pustaka `scikit-learn`, yang menyediakan implementasi SVM, dapat diinstal melalui pip dengan perintah `pip install scikit-learn`. Setelah semua pustaka terinstal, proses pelatihan

model dimulai dengan menyiapkan data fitur dan label. Fitur chroma dari sinyal audio dapat diekstraksi menggunakan fungsi `librosa.feature.chroma_stft(y=y, sr=sr)`. Model SVM kemudian dilatih dengan perintah `make_pipeline(StandardScaler(), svm.SVC(kernel='linear'))`. Model yang telah dilatih disimpan dalam file dengan perintah `joblib.dump(svm_model, 'svm_model.pkl')`.

Dalam implementasi kode, setelah fitur chroma rata-rata dari sinyal audio, klasifikasi dilakukan dengan memanggil fungsi `self.recognize_chord_svm(chroma_mean)`. Fungsi ini memanfaatkan model SVM yang telah dilatih untuk memprediksi chord berdasarkan fitur chroma yang diberikan, menghasilkan hasil klasifikasi yang akurat dan dapat diandalkan.

2.10 Kuesioner

Kuesioner adalah alat pengumpul data yang berbentuk pertanyaan yang akan diisi atau dijawab oleh responden [14]. Kuesioner sering digunakan dalam penelitian karena kemampuannya untuk mengumpulkan data dari sejumlah besar responden dengan efisien. Berdasarkan jenis pertanyaannya, kuesioner dapat dibagi menjadi tiga kategori utama yaitu kuesioner tertutup, kuesioner terbuka, dan kuesioner semi-terbuka. Kuesioner tertutup berisi pertanyaan dengan pilihan jawaban yang telah ditentukan sebelumnya, memudahkan proses analisis data karena jawaban sudah terstruktur. Kuesioner terbuka memberikan kesempatan kepada responden untuk memberikan jawaban dalam bentuk uraian bebas, memungkinkan jawaban yang lebih mendalam dan rinci. Kuesioner semi-terbuka adalah kombinasi dari keduanya, menggabungkan pilihan jawaban yang telah ditentukan dengan kesempatan untuk memberikan jawaban bebas.

2.11 Uji Hipotesis

Uji Hipotesis adalah salah satu cabang Ilmu Statistika Inferensial yang digunakan untuk menguji kebenaran atas suatu pernyataan secara statistik serta

menarik kesimpulan akan diterima atau ditolaknya pernyataan tersebut [15]. Proses uji hipotesis dimulai dengan merumuskan dua hipotesis yaitu Hipotesis Nol (H_0) dan Hipotesis Alternatif (H_1). Hipotesis Nol (H_0) yang menyatakan tidak adanya efek atau perbedaan, dan Hipotesis Alternatif (H_1) yang menyatakan adanya efek atau perbedaan. Uji hipotesis melibatkan pengumpulan data, perhitungan statistik uji, dan penentuan nilai p . Jika nilai p lebih kecil dari tingkat signifikansi yang telah ditetapkan (misalnya 0.05), maka Hipotesis Nol ditolak, menunjukkan bahwa data memberikan cukup bukti untuk mendukung Hipotesis Alternatif.

Beberapa jenis uji hipotesis yang umum digunakan meliputi uji rata-rata, uji normalitas, dan uji homogenitas. Uji rata-rata digunakan untuk menguji apakah rata-rata dari dua kelompok data berbeda secara signifikan. Uji normalitas dilakukan untuk memeriksa apakah data mengikuti distribusi normal, yang merupakan salah satu asumsi dasar dalam banyak uji statistik. Uji homogenitas digunakan untuk memeriksa kesamaan varians antar kelompok data, memastikan bahwa varians yang ada tidak berbeda secara signifikan. Melalui serangkaian uji ini, validitas dan reliabilitas hasil analisis statistik dapat ditingkatkan, memastikan bahwa kesimpulan yang ditarik dari data akurat dan dapat dipercaya.

Alur pengujian hipotesis dimulai dengan uji normalitas, dengan hipotesis H_0 : Data berdistribusi normal dan H_1 : Data tidak berdistribusi normal. Jika H_0 diterima dan H_1 ditolak, pengujian dilanjutkan dengan uji homogenitas. Namun, jika H_0 ditolak dan H_1 diterima, digunakan uji nonparametrik. Selanjutnya, uji homogenitas dilakukan dengan hipotesis H_0 : Data homogen dan H_1 : Data tidak homogen. Jika H_0 diterima dan H_1 ditolak, pengujian dilanjutkan ke uji hipotesis. Jika H_0 ditolak dan H_1 diterima, maka digunakan uji hipotesis nonparametrik karena asumsi homogenitas tidak terpenuhi. Dalam uji hipotesis, hipotesis yang diuji adalah $H_0: \mu_{\text{pretest}} = \mu_{\text{posttest}}$ dan $H_1: \mu_{\text{pretest}} < \mu_{\text{posttest}}$. Jika H_0 ditolak dan H_1 diterima maka terdapat perbedaan signifikan antara nilai rata-rata pretest dan posttest, sedangkan jika H_0 diterima dan H_1 ditolak maka tidak terdapat perbedaan signifikan antara nilai rata-rata pretest dan posttest.

2.12 Unified Modeling Language (UML)

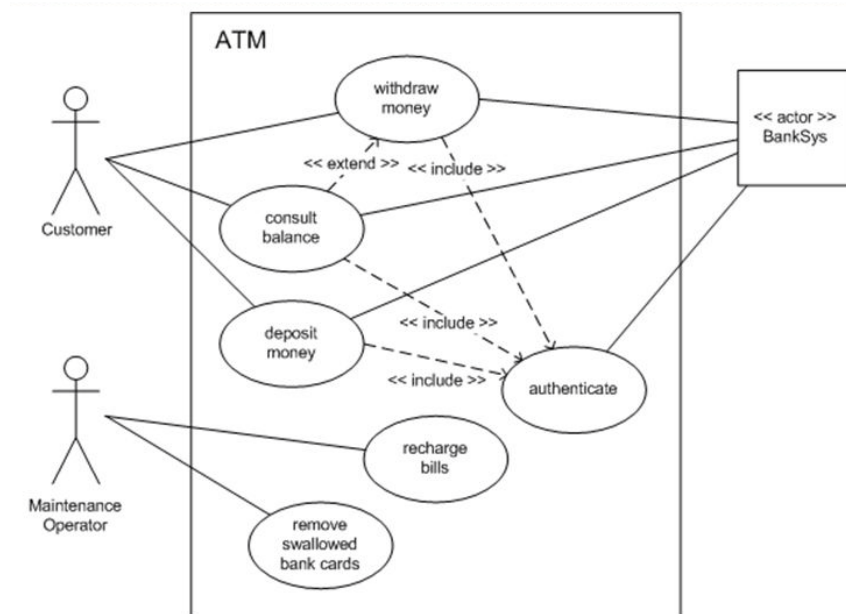
Unified Modeling Language (UML) adalah bahasa pemodelan untuk sistem berorientasi objek [16]. UML memungkinkan pengembang untuk memvisualisasikan, merancang, dan mendokumentasikan berbagai aspek dari sistem perangkat lunak, menggunakan serangkaian diagram dan notasi yang standar. Ini membantu dalam mengomunikasikan struktur dan perilaku sistem kepada berbagai pemangku kepentingan.

UML terdiri dari beberapa jenis diagram, termasuk Use Case Diagram, Class Diagram, Sequence Diagram, dan Activity Diagram. Masing-masing diagram ini memiliki tujuan dan penggunaannya masing-masing, memberikan pandangan yang berbeda pada sistem yang sedang dikembangkan. UML sangat penting dalam tahap perencanaan dan desain perangkat lunak, membantu dalam mengidentifikasi kebutuhan sistem dan potensi masalah.

2.12.1 Use Case Diagram

Use Case diagram digunakan untuk menggambarkan interaksi apa saja yang bisa dilakukan user kepada sistem informasi penjualan yang akan dibangun [17]. Diagram ini membantu dalam mengidentifikasi berbagai fungsi sistem dari perspektif pengguna. Use Case Diagram sangat berguna dalam menentukan fitur-fitur yang diperlukan dan interaksi pengguna dengan sistem.

Diagram ini biasanya terdiri dari aktor, use case, dan asosiasi antara keduanya. Aktor mewakili pengguna atau entitas lain yang berinteraksi dengan sistem, sedangkan use case mewakili serangkaian tindakan yang dilakukan sistem untuk memberikan nilai kepada aktor. Use Case Diagram adalah alat yang efektif untuk mengomunikasikan fungsionalitas sistem secara keseluruhan kepada stakeholder. Adapun contoh use case diagram yang dapat dilihat pada Gambar 2.1 berikut.



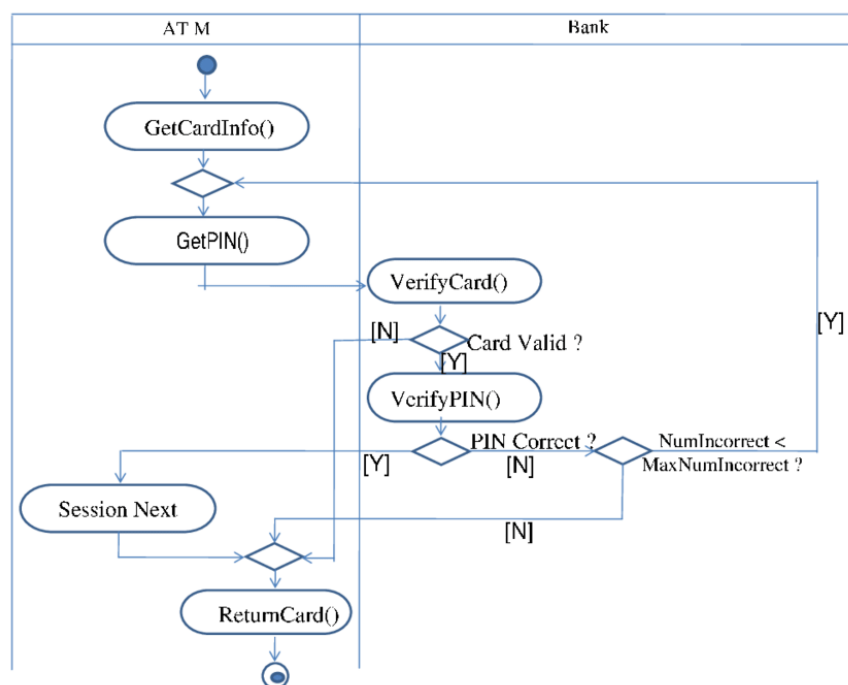
Gambar 2.1 Contoh Use Case Diagram

Diagram Use Case ini menggambarkan interaksi antara pengguna (Customer dan Maintenance Operator) dengan sistem ATM, dan juga melibatkan sistem perbankan eksternal (BankSys). Pengguna 'Customer' dapat melakukan operasi seperti 'withdraw money', 'consult balance', dan 'deposit money', yang semuanya memerlukan 'authentication'. Operasi 'withdraw money' memiliki ekstensi opsional 'consult balance' yang bisa dilakukan sebelumnya, sedangkan 'deposit money' termasuk dalam kasus penggunaan 'authenticate'. Pengguna 'Maintenance Operator' memiliki kasus penggunaan 'remove swallowed bank cards' dan 'recharge bills'. Setiap aksi yang dilakukan oleh pengguna ATM berinteraksi dengan sistem perbankan (BankSys), yang memainkan peran sebagai aktor dalam transaksi yang berhubungan dengan ATM. Relasi 'include' menunjukkan bahwa beberapa fungsi termasuk sebagai bagian dari proses yang lain, sementara 'extend' menunjukkan bahwa fungsi tersebut adalah tambahan opsional yang dapat terjadi dalam konteks tertentu.

2.12.2 Activity Diagram

Activity diagram atau disebut juga dengan diagram aktivitas, merupakan diagram yang menggambarkan berbagai aliran aktivitas dalam sebuah sistem yang sedang dirancang dan bagaimana masing-masing aliran berawal, keputusan yang mungkin terjadi, dan bagaimana aktivitas tersebut berakhir [17]. Diagram ini menggambarkan alur kontrol dari satu aktivitas ke aktivitas lainnya, dan sering digunakan untuk mendokumentasikan aspek operasional dari proses bisnis. Activity Diagram membantu dalam memvisualisasikan alur kerja, memungkinkan pemahaman yang lebih baik tentang bagaimana komponen sistem berinteraksi.

Diagram ini menggunakan berbagai simbol seperti lingkaran, panah, dan persegi panjang untuk mewakili langkah-langkah, alur kontrol, dan pengambilan keputusan dalam proses. Activity Diagram sangat berguna dalam merencanakan dan merancang alur kerja sistem, serta membantu dalam mengidentifikasi kemungkinan perbaikan atau pengoptimalan proses. Adapun contoh Activity Diagram yang dapat dilihat pada Gambar 2.2 berikut.



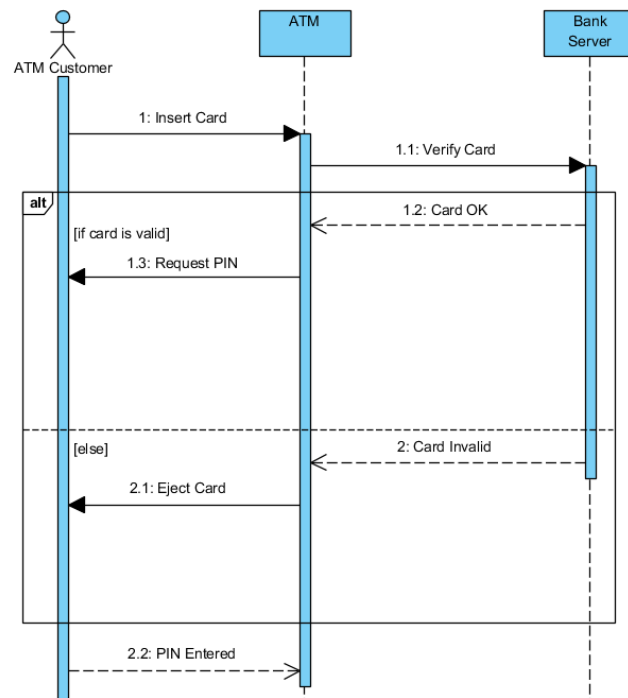
Gambar 2.2 Contoh Activity Diagram

Diagram aktivitas ini menggambarkan proses verifikasi kartu dan PIN di ATM yang melibatkan interaksi antara ATM dan bank. Proses dimulai dengan ATM yang meminta informasi kartu dan PIN dari pengguna. Setelah itu, informasi tersebut dikirim ke bank untuk verifikasi. Jika kartu tidak valid, proses berakhir dan kartu dikembalikan; jika valid, bank kemudian memverifikasi PIN. Jika PIN salah dan jumlah kesalahan belum melebihi batas maksimal, pengguna diminta memasukkan PIN lagi; jika benar, sesi transaksi berikutnya dimulai, jika tidak, kartu dikembalikan.

2.12.3 Sequence Diagram

Sequence Diagram menggambarkan bagaimana user melakukan interaksi dengan sistem informasi untuk mendapatkan informasi yang dibutuhkan [17]. Diagram ini menunjukkan bagaimana objek-objek berkomunikasi satu sama lain dalam urutan tertentu untuk melaksanakan fungsionalitas dalam sistem. Sequence Diagram sangat berguna untuk menganalisis alur kontrol dan data dalam skenario tertentu.

Diagram ini menggunakan garis vertikal untuk mewakili waktu hidup objek dan panah untuk menunjukkan pesan antara objek. Sequence Diagram membantu dalam memahami alur pesan dan interaksi antar objek, yang sangat penting dalam desain sistem berorientasi objek. Adapun contoh Sequence Diagram yang dapat dilihat pada Gambar 2.3 berikut.

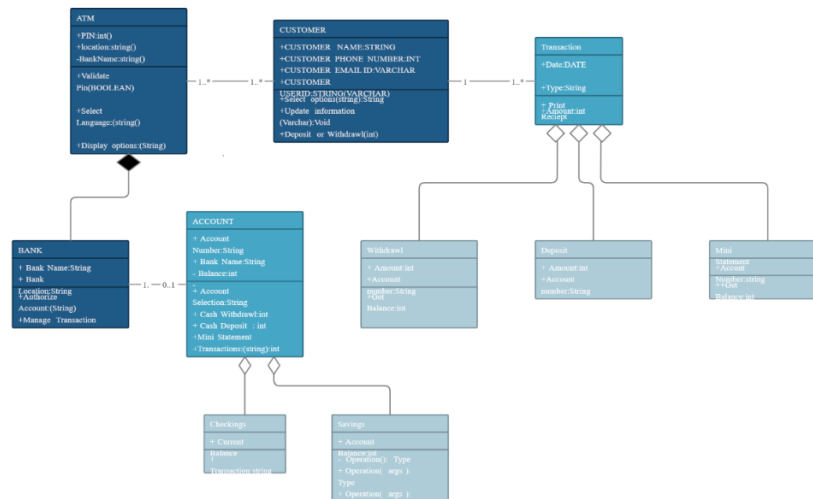


Gambar 2.3 Contoh Sequence Diagram

Diagram ini menggambarkan interaksi antara pelanggan, ATM, dan server bank selama proses verifikasi kartu dan permintaan PIN. Prosesnya dimulai ketika pelanggan memasukkan kartu ke dalam ATM. ATM kemudian mengirim permintaan untuk memverifikasi kartu ke server bank. Jika kartu dikonfirmasi valid oleh server bank ("Card OK"), ATM selanjutnya meminta pelanggan untuk memasukkan PIN mereka. Jika kartu dinyatakan tidak valid ("Card Invalid"), ATM akan segera mengeluarkan kartu tersebut kembali kepada pelanggan. Ini menunjukkan dua jalur proses yang mungkin tergantung pada validitas kartu yang dimasukkan.

2.12.4 Class Diagram

Class Diagram adalah jenis diagram UML yang digunakan untuk menggambarkan struktur kelas dalam sistem. Diagram ini menunjukkan kelas, atribut, metode, dan hubungan antar kelas, seperti warisan dan asosiasi. Sebuah Class Diagram akan menunjukkan bagaimana skema dari arsitektur sebuah sistem yang sedang dirancang [18]. Dalam Class Diagram, kelas digambarkan dengan kotak yang terbagi menjadi tiga bagian, mewakili nama kelas, atribut, dan metode. Hubungan seperti asosiasi, ketergantungan, dan pewarisan digambarkan menggunakan berbagai jenis panah. Adapun contoh Class Diagram yang dapat dilihat pada Gambar 2.4 berikut.



Gambar 2.4 Contoh Class Diagram

Gambar 2.4 menggambarkan arsitektur sistem perbankan yang mencakup kelas untuk ATM, Customer, Bank, Account, dan berbagai transaksi. Kelas ATM menyediakan fungsionalitas untuk inisiasi, validasi PIN, dan tampilan pilihan, sedangkan kelas Customer menyimpan informasi dasar pelanggan dan memfasilitasi pembaruan informasi serta transaksi keuangan. Bank bertanggung jawab untuk manajemen dan otorisasi transaksi, sedangkan Account bertindak sebagai kelas dasar untuk jenis akun berbeda seperti Checking dan Savings, masing-masing dengan operasi khusus seperti penarikan dan deposit. Kelas Transaction secara khusus menangani pencatatan dan pencetakan detail transaksi, memastikan bahwa semua kegiatan keuangan tercatat dan terdokumentasi dengan baik.