

BAB 2

TINJAUAN PUSTAKA

2.1 Stranough Firma Indonesia

Stranough Firma Indonesia adalah *workshop* yang berlokasi di Bandung, Jawa Barat, Indonesia, *workshop* ini terkenal sebagai *workshop* pembuatan gitar & bass yang telah disesuaikan dengan preferensi individu, Selama dua dekade, workshop ini telah melayani berbagai kalangan musisi, baik di tingkat lokal maupun internasional.



Gambar 2.1 Logo PT. Stranough Firma Indonesia

2.1.1 Sejarah Stranough Firma Indonesia

PT. Stranough Firma Indonesia, didirikan pada tanggal 29 Mei 2002, telah berkembang selama dua dekade menjadi salah satu perusahaan terkemuka dalam sektor manufaktur gitar. Beberapa tahun pertama dimulai dengan awal yang sederhana di ruang kantor dan *workshop* berukuran 3 x 12 meter di Bandung, Jawa Barat, Indonesia. Seiring dengan berjalannya waktu, perusahaan ini secara bertahap memperluas operasinya, seperti membuat transaksi bisa dilakukan secara online, dan berkolaborasi dengan pengrajin gitar mancanegara.

Pencapaian-pencapaian yang ditandai dengan penghargaan nasional dan eksposur internasional mendorong PT. Stranough Firma Indonesia ke panggung global. Pencapaian yang patut dicatat termasuk partisipasi dalam acara industri terkemuka seperti Winter NAMM Show di Anaheim, California, dan juga

kemitraan strategis dengan pihak terkenal seperti Seymour Duncan dan Dean Guitars.

PT Stranough juga mendapatkan banyak penghargaan seperti Indonesian Young Entrepreneur Award, Juara pertama dari National Entrepreneur Award dan Good Design Indonesia.

2.2 Landasan Teori

Landasan teori bertujuan untuk memberikan kerangka pemahaman bagi penelitian atau kajian yang dilakukan. Dengan memperkenalkan teori-teori yang relevan dan konsep-konsep kunci, landasan teori membantu peneliti untuk mengaitkan hasil penelitian mereka dengan kerangka pemikiran yang telah ada sebelumnya. Berikut landasan teori yang relevan dengan penelitian ini.

2.2.1 Virtual Assistant

Virtual Assistant (VA), adalah aplikasi perangkat lunak yang memanfaatkan kecerdasan buatan untuk memproses bahasa alami dan memberikan bantuan melalui interaksi teks [2]. Agen percakapan cerdas ini, yang juga dikenal sebagai chatbot atau sistem dialog, melakukan tugas-tugas seperti menjawab pertanyaan, membuat rekomendasi, dan menjalankan perintah. Awalnya dirancang untuk meningkatkan produktivitas dan efisiensi di lingkungan desktop, seperti Microsoft Clippy, VA telah berevolusi menjadi bagian integral dari kehidupan sehari-hari, contohnya asisten modern seperti Google Assistant, Apple Siri, Amazon Alexa, dan Microsoft Cortana [3].

2.2.1.1 Fitur Utama

1. Memanfaatkan Pemrosesan Bahasa Alami (NLP)

VA menggunakan teknik NLP untuk memahami dan memproses input bahasa alami dari pengguna, baik dalam bentuk teks maupun lisan. Kemampuan ini memungkinkan mereka untuk berinteraksi dengan pengguna dengan cara percakapan, membuat interaksi menjadi intuitif .

2. Eksekusi Tugas

Di luar perintah sederhana, VA dapat membantu berbagai tugas. Mereka dapat menjawab pertanyaan, memberikan rekomendasi, dan melakukan tindakan

seperti mengatur pengingat, mengirim pesan, dan mengendalikan perangkat rumah pintar. Kemampuan Beradaptasi

VA modern memiliki kemampuan untuk beradaptasi dengan preferensi dan perilaku dari masing-masing pengguna. Personalisasi ini meningkatkan kepuasan dan efisiensi pengguna dalam menyelesaikan tugas.

3. Bantuan Proaktif

Virtual Assistant dirancang untuk menawarkan bantuan proaktif. Mereka dapat mengantisipasi kebutuhan pengguna berdasarkan konteks dan interaksi sebelumnya.

4. Panduan Interaktif

VA menawarkan panduan interaktif yang memudahkan pengguna dalam menjalankan berbagai fungsi atau tugas. Dengan fitur ini, pengguna dapat menerima instruksi langkah demi langkah dalam menjalankan suatu proses, seperti memasak resep, memecahkan masalah teknis, atau menjalankan rutinitas kebugaran.

2.2.2 Natural Language Processing (NLP)

Pemrosesan Bahasa Alami (Natural Language Processing / NLP) adalah domain pada artificial intelligence yang menyelidiki cara-cara di mana komputer dapat memahami dan berinteraksi dengan bahasa alami, baik dalam bentuk tekstual maupun lisan, untuk melakukan tugas-tugas yang bermanfaat [4].

Upaya untuk membuat NLP bisa memahami bahasa manusia telah mengalami perkembangan yang signifikan selama beberapa decade terakhir. Awalnya Pada tahun 1950-an, NLP mengandalkan metode berbasis aturan (*rule based*), di mana para ahli secara manual membuat aturan yang mengatur analisis kata, struktur kalimat, jawaban pertanyaan, dan terjemahan mesin. Namun, sistem ini membutuhkan banyak tenaga manusia dan menjadi semakin tidak praktis untuk dikelola dengan semakin banyaknya aturan.

Pada tahun 1990-an, dengan meningkatnya jumlah data di internet membuka jalan bagi metode pembelajaran statistik dalam NLP. Model-model ini,

yang dilatih pada data berlabel atau data yang dikumpulkan oleh peneliti menghasilkan kemajuan penting di berbagai bidang seperti penerjemahan mesin dan teknologi mesin pencari.

Kemudian pada tahun 2012 Munculnya teknik *deep learning* dalam NLP menandai perubahan yang signifikan. Berdasarkan keberhasilannya dalam pengenalan objek dan ucapan, *deep learning* dengan cepat melampaui metode statistik dalam menangani permasalahan NLP. Dampaknya, pendekatan berbasis *deep learning* ini menjadi paradigma utama untuk tugas-tugas seperti penerjemahan mesin, mesin pemahaman bacaan, dan *chatbot*.

2.2.2.1 Langkah NLP

1. Analisis Leksikal

Analisis Leksikal bertanggung jawab dalam memecah teks masukan menjadi token yang merupakan unit terkecil yang masih mempunyai makna [5]. Langkah ini mencakup:

a. Tokenisasi

Memisahkan teks menjadi kata, frasa, simbol, atau elemen bermakna lainnya yang disebut token.

b. Normalisasi

Mengonversi teks ke bentuk standar, seperti mengubah semua huruf menjadi huruf kecil, menghapus tanda baca, dan menangani kontraksi.

c. Part-of-Speech Tagging

Menandai token dengan kelas kata (kata benda, kata kerja, kata sifat, dsb.).

Contoh: Masukan: "Pemrosesan Bahasa Alami sangat menarik."

Token: ["Pemrosesan", "Bahasa", "Alami", "sangat", "menarik"]

2. Analisis Sintaktik

Analisis Sintaktik, atau parsing, melibatkan analisis struktur gramatikal dari teks. Langkah ini memastikan bahwa kalimat mengikuti aturan sintaksis dan membangun pohon sintaktik untuk merepresentasikan struktur kalimat [5].

a. Parsing

Menentukan struktur gramatikal sebuah kalimat berdasarkan tata bahasa formal yang diberikan.

b. Dependency Parsing

Mengidentifikasi ketergantungan antar kata dalam kalimat, seperti kata mana yang menjadi subjek, objek, dll.

3. Analisis Semantik

Analisis Semantik bertujuan untuk memahami makna dari teks. bagian ini melibatkan interpretasi konteks dan memberikan makna pada struktur yang benar secara sintaksis [5].

a. Word Sense Disambiguation

Menentukan makna mana dari sebuah kata yang digunakan dalam konteks yang diberikan.

b. Named Entity Recognition (NER)

Mengidentifikasi dan mengklasifikasikan entitas (nama, tanggal, lokasi, dll.) dalam teks.

Contoh: Masukan: "Apple sedang mempertimbangkan untuk membeli startup di U.K. seharga \$1 miliar."

Entitas: [("Apple", Perusahaan), ("U.K.", Lokasi), ("\$1 miliar", Uang)]

4. Discourse Integration

Bagian ini berkaitan dengan bagaimana kalimat sebelumnya dapat mempengaruhi interpretasi kalimat berikutnya [5].

a. Resolusi Koreferensi

Mengidentifikasi ketika ekspresi yang berbeda merujuk pada entitas yang sama dalam teks.

b. Discourse Relation

Memahami hubungan logis dan semantik antara kalimat-kalimat dan bagian-bagian teks (misalnya, sebab-akibat, kontras).

Contoh: "Ani pergi ke toko. Dia membeli beras."

Koreferensi: "Dia" merujuk pada "Ani".

Discourse Relation: Kedua kalimat berhubungan melalui tindakan ani pergi ke toko dan kemudian membeli beras.

5. Analisis Pragmatik

Analisis pragmatik berhubungan dengan pengetahuan yang berada di luar dokumen dan/atau teks. Analisis pragmatik berfokus pada apa yang dideskripsikan ditafsirkan kembali dengan apa yang sebenarnya dimaksudkan, memperoleh berbagai aspek bahasa yang membutuhkan pengetahuan dunia nyata [5].

Contoh :

"bagus! Kemacetan lalu lintas lagi" (Sarkasme tentang ketidaknyamanan)

“Ani adalah bunga desa” (Metafora untuk kata cantik)

2.2.3 Large Language Model (LLM)

Model Bahasa Besar (Large Language Model / LLM) adalah model pembelajaran mesin yang mampu melaksanakan berbagai tugas NLP, seperti pembuatan dan klasifikasi teks, menjawab pertanyaan percakapan, dan penerjemahan bahasa. Model-model ini menjalani pelatihan yang intensif menggunakan kumpulan data yang besar, sehingga mendapatkan julukan “(Large) Language Model”. [6]

LLM dapat membuat teks dengan mengambil sebuah teks input dan kemudian secara berulang kali memprediksi token atau kata berikutnya, untuk membuat LLM bekerja dalam domain yang spesifik (contohnya, sebagai chatbot), teknik yang dapat dilakukan adalah *fine-tuning* dan embedding data.

2.2.4 OpenAI API

OpenAI API adalah platform berbasis cloud yang menyediakan akses ke model pembelajaran mesin OpenAI, model ini dapat diterapkan pada tugas yang membutuhkan NLP [7]. API ini menawarkan berbagai model dengan berbagai

kemampuan dan titik harga, dan memungkinkan fine-tuning model khusus agar sesuai dengan kebutuhan spesifik.

2.2.4.1 Konsep Utama

1. Model Pembuatan Teks

Model-model pembangkitan teks OpenAI, yang dikenal sebagai generative pre-trained transformers (GPT), termasuk GPT-4 dan GPT-3.5. Model-model ini dilatih untuk memahami dan menghasilkan bahasa alami berdasarkan input yang diberikan, yang disebut sebagai "prompt." Merancang prompt yang efektif melibatkan pemberian instruksi atau contoh untuk mencapai hasil yang diinginkan. Model GPT dapat digunakan untuk berbagai macam aplikasi, seperti pembuatan konten, pembuatan kode, rangkuman, percakapan, dan penulisan kreatif [8].

2. Asisten

Asisten, yang didukung oleh model bahasa GPT-4, melakukan tugas untuk pengguna berdasarkan instruksi yang disematkan di jendela konteks model. Asisten ini juga dapat mengakses alat bantu untuk menangani tugas yang lebih kompleks, seperti menjalankan kode atau mengambil informasi dari file [8].

3. Embedding

Embedding adalah representasi vektor data, seperti teks, yang mempertahankan konten dan maknanya. Potongan data yang serupa memiliki embedding yang lebih dekat satu sama lain. OpenAI menyediakan model penyematan teks yang mengubah string teks menjadi vektor penyematan, yang berguna untuk pencarian, pengelompokan, rekomendasi, deteksi anomali, dan klasifikasi [8].

4. Token

Model pembangkitan teks dan embedding memproses teks dalam unit yang disebut token. Token dalam konteks ini memiliki arti urutan karakter yang umum ditemukan dalam satu set teks. Misalnya, "tokenization" dipecah menjadi

"token" dan "ization", contoh lain adalah teks “saya ingin membuat gitar” dipecah menjadi “s”, “aya”, “ingin”, “membuat”, “g” dan “itar”. Biasanya, token pertama dari setiap kata dimulai dengan karakter spasi. Umumnya, 1 token adalah sekitar 4 karakter atau 0,75 kata dalam teks bahasa Inggris [8].

OpenAI menyediakan Alat tokenizer yang dapat digunakan untuk melihat bagaimana string dibagi menjadi token.

5. Context Windows

Untuk model pembuatan teks, prompt dan output yang dihasilkan tidak boleh melebihi panjang konteks maksimum model. Untuk model embedding, input harus lebih pendek dari panjang konteks maksimum model [8].

2.2.4.2 Model Yang Tersedia

1. GPT-4o

GPT-4o ("o" untuk "omni") adalah model chatgpt yang paling baru. Model ini multimodal (menerima input teks atau gambar dan mengeluarkan teks), dan memiliki kecerdasan tinggi yang sama dengan GPT-4 Turbo tetapi jauh lebih efisien-menghasilkan teks 2x lebih cepat dan 50% lebih murah. Selain itu, GPT-4o memiliki kinerja terbaik di seluruh bahasa non-Inggris dari semua model openAI. Model ini memiliki jendela konteks sebesar 128,000 token [8].

2. GPT-4 Turbo dan GPT-4

GPT-4 adalah model multimodal yang besar (menerima input teks atau gambar dan mengeluarkan teks) yang dapat memecahkan masalah sulit dengan akurasi yang lebih tinggi daripada model-model kami sebelumnya, berkat pengetahuan umum yang lebih luas dan kemampuan penalaran yang canggih. Seperti gpt-3.5-turbo, GPT-4 dioptimalkan untuk mode obrolan. GPT-4 memiliki jendela konteks sebesar 8192 tokens, sedangkan GPT-4 Turbo memiliki jendela konteks sebesar 128,000 token [8].

3. GPT-3.5 Turbo

Model GPT-3.5 Turbo dapat memahami dan menghasilkan bahasa atau kode alami dan telah dioptimalkan untuk obrolan Model ini memiliki jendela konteks sebesar 16,385 token [8].

4. Model untuk embedding

Semua model embedding memiliki jendela konteks sebesar 8191, yang berbeda hanyalah panjang vektor yang dibangkitkan yaitu 1536 untuk text-embedding-3-small dan 3072 untuk text-embedding-3-large [8].

2.2.5 Multimedia

Multimedia adalah suatu bentuk komunikasi yang mengintegrasikan berbagai bentuk konten seperti teks, audio, gambar, animasi, video, dan elemen interaktif [9]. Penggabungan berbagai jenis media ini digunakan untuk menciptakan penyajian informasi yang menarik dan dinamis, meningkatkan pengalaman pengguna melalui stimulasi sensorik dan interaktivitas. Aplikasi umum multimedia mencakup program pendidikan, periklanan digital, video game, realitas virtual, dan konten online. Dengan menggabungkan unsur-unsur ini, multimedia dapat secara efektif menyampaikan pesan yang kompleks, menghibur, mendidik, dan melibatkan pengguna dengan cara yang tidak dapat dilakukan oleh media tradisional.

2.2.6 Objek Multimedia

Menurut Sutopo, objek-objek multimedia adalah sebagai berikut [10]:

1. Teks

Teks adalah data atau informasi dalam bentuk susunan-susunan karakter. Sebagian besar multimedia menggunakan tesks karena sangat efektif untuk menyampaikan ide kepada pengguna.

2. Gambar

Gambar sangat bermanfaat untuk mengilustrasikan data atau informasi yang akan disampaikan, terutama suatu informasi yang sulit dijelaskan melalui kata-kata .

3. Animasi

Animasi merupakan simulasi gerakan yang dihasilkan dengan menggerakkan sejumlah gambar dari frame ke frame yang lain. Frame adalah gambar tunggal pada sejumlah rentetan gambar yang membentuk animasi.

4. Video

Video menyediakan informasi yang kaya dan hidup untuk multimedia. Video dapat digunakan untuk menjelaskan sesuatu hal yang sangat sulit dijelaskan sama halnya dengan gambar.

5. Audio

Multimedia akan cepat bosan jika tidak tersedianya audio (suara). Suara disini bisa berbentuk percakapan, musik, atau efek yang digunakan untuk menunjang multimedia yang lain.

2.2.7 Pemrograman Berbasis Objek

Pemrograman Berorientasi Objek (PBO) adalah paradigma pemrograman yang menggunakan objek sebagai unit dasar penyusun perangkat lunak. Objek dalam PBO merepresentasikan entitas yang dapat memiliki data dan perilaku, serta dapat berinteraksi satu sama lain. Paradigma ini menekankan konsep pewarisan, enkapsulasi, dan polimorfisme [11].

Beberapa konsep kunci dalam pemrograman berbasis objek meliputi [11] :

1. Kelas, merupakan sebuah *blueprint* untuk menciptakan objek. Kelas mendefinisikan atribut (data) dan metode (fungsi) yang akan dimiliki oleh objek.
2. Objek, entitas konkret dari suatu kelas. Objek memiliki atribut dan metode yang telah ditentukan oleh kelasnya.
3. Metode / Behaviour, fungsi atau prosedur yang terkait dengan suatu objek atau kelas, merepresentasikan bagaimana perilaku objek. Metode pada dasarnya adalah cara untuk berinteraksi dengan objek atau untuk melakukan operasi tertentu pada objek tersebut.

4. Enkapsulasi, menyembunyikan detail implementasi dan menyediakan antarmuka yang jelas untuk berinteraksi dengan objek. Hanya metode-metode yang didefinisikan secara publik yang dapat diakses dari luar kelas.
5. Pewarisan, Mekanisme yang memungkinkan kelas turunan untuk mengambil atribut dan metode dari kelas yang sudah ada. Hal ini memungkinkan penggunaan kembali kode dan mempermudah pengorganisasian dan pemeliharaan kode.
6. Polimorfisme, Konsep yang memungkinkan objek dari kelas yang berbeda untuk merespon dengan cara yang sama terhadap panggilan metode. Polimorfisme memungkinkan penggunaan yang lebih fleksibel dari metode dan meningkatkan reusable code.

2.2.8 Javascript

JavaScript (JS) adalah bahasa pemrograman tingkat tinggi yang secara umum digunakan untuk pengembangan aplikasi web [12]. Dikembangkan pertama kali oleh Netscape pada tahun 1995, JavaScript telah menjadi salah satu elemen kunci dalam pengembangan web modern. Bahasa ini bersifat dinamis, dapat dijalankan di berbagai platform, dan memungkinkan interaksi pengguna secara langsung di dalam browser.

JavaScript umumnya dieksekusi di sisi klien (client-side), artinya kode tersebut dijalankan oleh peramban (browser) pada perangkat pengguna. Hal ini memungkinkan pembuatan antarmuka pengguna yang dinamis dan responsif.

Javascript paling sering digunakan untuk memodifikasi *document object model* (DOM) yaitu representasi hierarkis dokumen HTML/XML yang dihasilkan oleh browser. JavaScript menggunakan DOM untuk berinteraksi dengan elemen-elemen HTML, mengubah struktur dan tampilan halaman secara dinamis.

2.2.9 Basis Data

Basis data adalah arsip virtual yang terorganisir untuk menyimpan sejumlah besar informasi terstruktur dengan cara yang memungkinkan pengambilan, pengelolaan, dan manipulasi yang efisien[13]. Database bertindak sebagai tempat penyimpanan terpusat, di mana data diorganisasikan ke dalam tabel, baris, dan kolom, mirip dengan spreadsheet.

Pada intinya, database terdiri dari dua komponen utama: data itu sendiri dan sistem manajemen basis data (DBMS). DBMS adalah perangkat lunak yang memungkinkan pengguna untuk berinteraksi dengan database, melakukan tugas-tugas seperti menambah, mengakses, memperbarui, dan menghapus data, serta menerapkan langkah-langkah keamanan dan memastikan integritas data.

Basis data terdiri dari berbagai jenis, termasuk relasional, NoSQL, dan berorientasi objek, yang masing-masing cocok untuk kebutuhan penyimpanan dan pengambilan data yang berbeda. Basis data relasional, seperti MySQL, PostgreSQL, dan Oracle, mengatur data ke dalam tabel dengan hubungan yang sudah ditentukan sebelumnya, sehingga memudahkan kueri dan transaksi yang kompleks. Basis data NoSQL, seperti MongoDB dan Cassandra, menawarkan fleksibilitas lebih dalam menangani data yang tidak terstruktur atau semi-terstruktur. Database berorientasi objek, seperti Couchbase dan db4o, menyimpan data dalam bentuk objek, yang mencerminkan struktur bahasa pemrograman berorientasi objek.

2.2.10 Pixi JS

PixiJS adalah library rendering yang menggunakan WebGL atau Canvas untuk menampilkan gambar dan konten visual 2D. PixiJS menyediakan grafik adegan lengkap (hierarki objek untuk dirender), dan menyediakan dukungan interaksi untuk memungkinkan penanganan peristiwa klik dan sentuhan [14]. PixiJS adalah pengganti untuk Flash di dunia HTML5 modern, tetapi memberikan kinerja yang lebih baik dan efek tingkat piksel yang melampaui apa yang dapat dicapai oleh Flash [15]. Library ini cocok untuk game online, konten pendidikan, iklan interaktif, dan visualisasi data.

2.2.10.1 Komponen PixiJS

1. Renderer

Inti dari sistem PixiJS adalah renderer, yang menampilkan grafik adegan dan menggambarinya ke layar. PixiJS akan secara otomatis menentukan apakah akan memberikan Anda renderer WebGPU atau WebGL [14].

2. Container

Objek scene utama yang membuat scene graph: pohon objek yang dapat dirender untuk ditampilkan, seperti sprite, grafik, dan teks [14].

3. Assets

Sistem Aset menyediakan alat untuk memuat sumber daya secara asinkron, seperti gambar dan file audio [14].

4. Graphics

Kelas yang digunakan untuk membuat sebuah geometri yang akan ditampilkan pada layar [14].

berikut ini daftar primitif dasar yang digunakan sebagai blok dasar pada pembuatan sebuah geometri [14] :

1. Line
2. Rect
3. RoundRect
4. Circle
5. Ellipse
6. Arc
7. Bezier and Quadratic Curve
8. Torus
9. Chamfer Rect
10. Fillet Rect
11. Regular Polygon
12. Star
13. Rounded Polygon

5. Ticker

Ticker menyediakan *callback* secara berkala berdasarkan waktu. Logika pembaruan game umumnya akan dijalankan sebagai respons terhadap ticker sekali per frame.

Sebuah aplikasi dapat memiliki beberapa ticker yang digunakan pada satu waktu [14]

6. Application

Kelas ini adalah pembantu sederhana yang membungkus Loader, Ticker, dan Renderer menjadi satu objek tunggal yang mudah digunakan. Dapat

digunakan untuk memulai proyek dengan cepat, membuat prototipe, dan membangun proyek sederhana [14].

7. Events

PixiJS mendukung interaksi berbasis pointer - membuat objek dapat diklik, menembakkan hover event, dll [14].