BAB II
STUDI LITERATUR

2.1 Studi Literatur

Penerapan teknologi pengenalan suara (speech recognition) terus berkembang di berbagai bidang, termasuk jurnalistik dan otomasi dokumen administratif. Komang Setia Buana (2020) melakukan penelitian terkait penerapan speech-to-text berbasis Python untuk mendukung wartawan dalam mentranskripsikan wawancara secara otomatis. Dengan menggunakan modul speech recognition dan algoritma FastICA, aplikasi ini mampu meningkatkan efisiensi pencatatan hingga mencapai akurasi 94,75% dalam kondisi audio yang bervariasi. Penelitian ini berfokus pada penerapan desktop tanpa pendekatan multibahasa dan belum terintegrasi dengan platform interaktif seperti Unity. Hasilnya menunjukkan bahwa teknologi ini dapat menghemat waktu dan membantu pengguna yang memiliki keterbatasan dalam mengetik secara manual.

Penelitian serupa dilakukan oleh Raihan Rifaldi dan Andhik Budi Cahyono (2024) dengan mengembangkan aplikasi notulensi otomatis bernama SPERCO yang mengintegrasikan model Whisper AI dari OpenAI. Penelitian ini menambahkan fitur Speaker Diarization menggunakan pustaka Pyannote untuk mengidentifikasi pembicara dalam transkrip. Hasil evaluasi menunjukkan tingkat akurasi transkripsi tinggi bahkan dalam lingkungan dengan noise, dan mampu bekerja secara multilingual. Model Whisper dinilai memiliki performa unggul dalam skenario dunia nyata, namun penelitian ini belum mengkaji penerapan Whisper AI dalam konteks edukasi akademik, terutama di lingkungan kampus berbasis Unity.

Tabel 2.1 Research Gap

No.	Peneliti	Insight Penelitian	Hasil	Metode	Keterbatasan	No. Kutipan
1.	Setia Buana (2020)	Speech-to-Text untuk mendukung wartawan mentranskripsi	Akurasi 94,75% dengan Python SpeechRocognition dan FastICA	Eksperimen berbasis rekaman WAV	Tidak mendukung real time interaktif, tidak	[7]

		wawancara			multibahasa, bukan berbasis Unity	
2.	Rifaldi & Cahyono (2024)	Pengembangan SPERCO untuk notulensi otomatis menggunakan Whisper AI + Pyannote	Fitur Speaker Diarization & transkripsi multibahasa dengan tingkat akurasi tinggi	Whisper + Pyannote + VAD	Tidak digunakan untuk sistem akademik atau integrasi Unity	[8]

Dari kedua penelitian tersebut, terlihat bahwa penggunaan model Whisper AI telah menunjukan performa tinggi dalam pengenalan suara dan transkripsi multibahasa. Namun, belum ada penelitian yang mengintegrasikan model ini ke dalam aplikasi interaktif berbasis Unity untuk kebutuhan penyedia informasi akademik mahasiswa. Oleh karena itu, penelitian ini mengusulkan pengembangan asisten virtual akademik berbasis Unity dengan fitur Speech-to-Text menggunakan Whisper AI.

2.2 Kerangka Teori

2.2.1 *Speech Recognition*

Pengenalan suara (Speech Recognition) adalah teknologi yang memungkinkan komputer atau perangkat lain untuk mengidentifikasi dan mengubah ucapan manusia menjadi teks atau perintah yang dapat dimengerti oleh mesin [9]. Proses ini melibatkan beberapa tahapan, mulai dari penangkapan sinyal suara, pra-pemrosesan untuk menghilangkan noise dan normalisasi, ekstraksi fitur untuk mengidentifikasi karakteristik penting dari suara, hingga pemodelan akustik dan bahasa untuk mencocokkan fitur suara dengan unit linguistik seperti fonem dan kata [10]. Teknologi pengenalan suara bertujuan untuk menjembatani komunikasi antara manusia dan komputer secara lebih alami dan intuitif [9]. Sistem pengenalan suara dapat diklasifikasikan berdasarkan berbagai faktor, seperti ketergantungan pada pembicara (speaker-dependent vs. speaker-independent), jenis ucapan (kata terisolasi, kata terhubung, atau ucapan berkelanjutan), dan ukuran kosakata [9]. Tantangan utama dalam pengembangan sistem pengenalan suara meliputi variabilitas dalam ucapan manusia (aksen, intonasi, kecepatan bicara), kebisingan latar belakang, dan ambiguitas bahasa [10].

2.2.2 Whisper AI

Whisper AI adalah model pengenalan ucapan otomatis (Automatic Speech Recognition - ASR) yang dikembangkan oleh OpenAI. Model ini dilatih pada dataset besar dan beragam yang mencakup 680.000 jam data audio multibahasa dan multitasking yang dikumpulkan dari web. Kemampuan utama Whisper AI adalah mentranskripsikan audio dari berbagai bahasa ke dalam teks, serta menerjemahkan ucapan dari bahasa-bahasa tersebut ke dalam bahasa Inggris. Arsitektur Whisper AI didasarkan pada model Transformer end-to-end, yang memproses seluruh sekuens audio dan secara langsung menghasilkan transkripsi teks. Hal ini memungkinkannya untuk menangani berbagai aksen, kebisingan latar belakang, dan bahasa teknis dengan tingkat akurasi yang tinggi. Sifat open-source dari beberapa versi Whisper AI memungkinkan peneliti dan pengembang untuk mengintegrasikannya ke dalam berbagai aplikasi, termasuk asisten virtual, alat transkripsi, dan sistem kontrol suara.

2.2.3 Unity sebagai Platform Pengembangan

Unity adalah mesin permainan (game engine) lintas platform yang dikembangkan oleh Unity Technologies, pertama kali diumumkan dan dirilis pada Juni 2005 di Apple Worldwide Developers Conference sebagai mesin game eksklusif Mac OS X [11]. Sejak saat itu, Unity telah berkembang menjadi salah satu platform pengembangan interaktif 2D dan 3D yang paling populer, tidak hanya untuk game tetapi juga untuk berbagai aplikasi lain seperti simulasi, visualisasi arsitektur, film, rekayasa, dan pengalaman realitas virtual (VR) serta realitas tertambah (AR) [11]. Unity menawarkan lingkungan pengembangan terintegrasi (IDE) yang komprehensif, mencakup editor visual untuk merancang adegan, alat scripting (umumnya menggunakan C#), manajemen aset, dan kemampuan untuk membangun aplikasi ke lebih dari 20 platform berbeda, termasuk Windows, macOS, Linux, Android, iOS, konsol game, dan platform web [11]. Keunggulan Unity terletak pada kemudahan penggunaannya, komunitas pengembang yang besar dan aktif, serta Asset Store yang menyediakan berbagai macam aset siap pakai untuk mempercepat proses pengembangan [11]. Fleksibilitas dan kekayaan fitur yang dimiliki Unity menjadikannya pilihan yang cocok untuk mengembangkan aplikasi yang memerlukan antarmuka pengguna yang interaktif dan dinamis, termasuk integrasi dengan teknologi kecerdasan buatan seperti model pengenalan suara.

2.2.4 Metode Spiral

Metode Spiral adalah model proses pengembangan perangkat lunak yang menggabungkan sifat iteratif dari model prototipe dengan aspek terkontrol dan sistematis dari model air terjun (waterfall) [12]. Model ini menekankan pada analisis risiko pada setiap tahap iterasi. Setiap siklus dalam model Spiral terdiri dari empat aktivitas utama [12]:

1. Perencanaan (Planning): Pada tahap ini, tujuan, alternatif, dan batasan (kendala) untuk iterasi tersebut ditentukan. Ini melibatkan estimasi biaya, jadwal, dan sumber daya untuk iterasi proyek.

2. Analisis Risiko (Risk Analysis): Tahap ini berfokus pada identifikasi dan analisis risiko proyek. Alternatif-alternatif dievaluasi untuk memilih solusi terbaik yang dapat mengurangi risiko yang teridentifikasi. Jika risiko signifikan, strategi untuk mengatasinya dikembangkan.

3. Pengembangan dan Implementasi (Engineering/Development and Implementation): Pada tahap ini, perangkat lunak dikembangkan, bersama dengan pengujian. Produk yang dihasilkan mungkin berupa prototipe, rilis parsial dari perangkat lunak, atau produk akhir.
4. Evaluasi (Evaluation): Hasil dari tahap pengembangan dievaluasi oleh pelanggan atau pengguna. Berdasarkan umpan balik, perencanaan untuk siklus berikutnya dimulai. Jika risiko tidak dapat diterima, proyek dapat dihentikan.

Proses ini berulang melalui beberapa iterasi, dengan setiap iterasi membangun versi produk yang lebih lengkap atau lebih halus. Model Spiral sangat cocok untuk proyek-proyek besar, kompleks, dan berisiko tinggi di mana persyaratan tidak sepenuhnya dipahami di awal atau mungkin berubah selama pengembangan [12]. Fokus pada manajemen risiko membantu dalam mengendalikan ketidakpastian dan meningkatkan kemungkinan keberhasilan proyek

2.3 Konteks Penelitian

Kebutuhan akan akses informasi akademik yang cepat, tepat, dan mudah merupakan hal krusial bagi mahasiswa di lingkungan perguruan tinggi, termasuk di Program Studi Informatika Universitas Muhammadiyah Malang (UMM). Mahasiswa, sebagai pengguna utama layanan akademik, seringkali dihadapkan pada berbagai tantangan dalam memperoleh informasi terkait jadwal kuliah, status Kartu Rencana Studi (KRS), informasi dosen, lokasi fasilitas, dan berbagai pengumuman penting lainnya. Sistem Informasi Akademik (SIAKAD) yang ada, meskipun bertujuan untuk mempermudah, terkadang masih memiliki kendala seperti kesulitan akses, antarmuka yang kurang intuitif bagi sebagian pengguna, keterlambatan pembaruan informasi, atau bahkan kendala teknis seperti server down pada waktu-waktu krusial [13]. Penelitian oleh Istiqamah dkk. (2024) menunjukkan bahwa efektivitas SIAKAD dalam mendukung layanan mahasiswa belum optimal, terutama dalam hal ketepatan waktu dan pencapaian tujuan layanan informasi [13].

Masalah yang dihadapi mahasiswa tidak hanya terbatas pada aspek teknis sistem informasi yang ada. Seringkali, mahasiswa merasa bingung mengenai prosedur administrasi atau istilah-istilah akademik tertentu [16]. Lebih lanjut, terdapat kecenderungan mahasiswa enggan atau menghadapi kesulitan untuk bertanya secara langsung kepada staf administrasi atau dosen wali, yang mungkin disebabkan oleh kesibukan pihak terkait, antrian panjang,

atau bahkan jarak fisik ke pusat layanan informasi [13][16]. Kondisi ini diperparah dengan terbatasnya jam layanan informasi konvensional, yang tidak selalu dapat mengakomodasi kebutuhan informasi mahasiswa di luar jam kerja.

Dalam konteks inilah, pemanfaatan teknologi informasi, khususnya melalui pengembangan asisten virtual, menjadi sangat relevan untuk meningkatkan kualitas layanan akademik [15]. Asisten virtual, baik dalam bentuk chatbot maupun yang berbasis suara, menawarkan potensi solusi untuk menyediakan layanan informasi yang lebih responsif, personal, dan dapat diakses kapan saja (24/7) [16].

Solusi berbasis suara, seperti yang diusulkan dalam penelitian ini, memiliki relevansi khusus di lingkungan pendidikan. Interaksi menggunakan suara menawarkan cara yang lebih alami dan intuitif bagi pengguna, sejalan dengan kemajuan dalam bidang Kecerdasan Buatan (AI) dan Interaksi Manusia-Komputer (HCI). Teknologi speech recognition memungkinkan pengguna untuk menyampaikan kebutuhannya secara lisan, yang dapat lebih efisien dibandingkan mengetik, terutama untuk pertanyaan-pertanyaan cepat [14]. Bagi mahasiswa Program Studi Informatika UMM yang umumnya telah akrab dengan perkembangan teknologi, adopsi asisten virtual berbasis suara dapat menjadi sebuah inovasi yang menarik dan fungsional. Penggunaan platform Unity untuk pengembangan juga mendukung pembuatan antarmuka pengguna yang interaktif dan dinamis, yang sangat sesuai untuk aplikasi asisten virtual [14]. Dengan demikian, integrasi model Whisper AI pada asisten virtual berbasis Unity diharapkan dapat mengatasi sebagian permasalahan komunikasi dan akses informasi yang dihadapi mahasiswa, menyediakan alternatif yang lebih efisien dan personal dalam mendapatkan informasi akademik.

2.4 Model dan Tools yang Digunakan

Pengembangan sistem asisten virtual dalam penelitian ini akan memanfaatkan beberapa model dan perangkat lunak utama. Komponen inti dari sistem ini adalah model Whisper AI dari OpenAI, yang berfungsi sebagai mesin Speech-to-Text (STT). Whisper AI dipilih karena kemampuannya yang telah terbukti dalam mentranskripsikan audio multibahasa dengan akurasi tinggi, bahkan dalam kondisi audio yang beragam dan ber-noise [17][18]. Model ini mampu menangani variasi aksen dan kondisi akustik lebih baik dibandingkan beberapa model lain, yang sangat relevan untuk penggunaan di Indonesia dengan keragaman dialeknya [17]. Penelitian oleh William & Zahra (2025) menunjukkan bahwa Whisper memiliki performa WER (Word Error Rate) yang lebih baik untuk

bahasa Indonesia dibandingkan model XLS-R dan XLSR-53 setelah proses fine-tuning [17]. Meskipun Whisper dirancang untuk penggunaan offline, terdapat upaya dan penelitian untuk mengadaptasinya ke lingkungan real-time seperti yang dilakukan oleh Bevilacqua et al. (2024) dengan sistem Whispy, yang menunjukkan potensi penggunaannya dalam aplikasi interaktif [18].

Bahasa pemrograman utama yang akan digunakan untuk logika pemrosesan dan integrasi adalah Python. Python dipilih karena merupakan bahasa yang populer dalam pengembangan AI dan machine learning, memiliki banyak pustaka pendukung, dan komunitas yang besar. Python akan digunakan untuk mengelola input audio, memanggil model Whisper AI untuk transkripsi, memproses teks hasil transkripsi, dan berinteraksi dengan game engine Unity [19][20]. Integrasi antara Python dan Unity dapat dilakukan, misalnya, melalui protokol komunikasi seperti UDP (User Datagram Protocol) untuk pertukaran data antara agen AI yang dikembangkan dengan Python dan lingkungan simulasi atau antarmuka yang dibuat di Unity [20].

Platform pengembangan antarmuka pengguna (UI) dan pengalaman pengguna (UX) interaktif akan menggunakan Unity. Unity adalah game engine yang kuat dan fleksibel, tidak hanya untuk pengembangan game tetapi juga untuk aplikasi interaktif lainnya, termasuk simulasi dan visualisasi [11][19]. Unity mendukung scripting dengan C# dan menyediakan berbagai alat untuk membuat antarmuka 2D/3D yang menarik dan responsif. Dalam konteks penelitian ini, Unity akan digunakan untuk menciptakan avatar virtual, menampilkan informasi akademik dalam bentuk teks atau visual, dan mengelola interaksi pengguna dengan asisten virtual [19]. ML-Agents dari Unity juga menunjukkan bagaimana Unity dapat diintegrasikan dengan algoritma machine learning yang dikembangkan menggunakan Python [19].

Untuk mempercepat proses inferensi model Whisper AI, terutama jika menggunakan model yang lebih besar atau untuk pemrosesan real-time yang lebih responsif, akan dipertimbangkan penggunaan PyTorch dengan dukungan CUDA (Compute Unified Device Architecture). PyTorch adalah sebuah framework machine learning open-source yang banyak digunakan untuk aplikasi seperti computer vision dan natural language processing, dan merupakan dasar dari banyak implementasi model Transformer seperti Whisper [19]. CUDA adalah platform komputasi paralel dan model pemrograman yang dikembangkan oleh NVIDIA untuk komputasi umum pada unit

pemrosesan grafis (GPU) mereka [21]. Dengan memanfaatkan GPU melalui CUDA, operasi matriks dan tensor yang intensif secara komputasi dalam model deep learning seperti Whisper dapat dipercepat secara signifikan, mengurangi latensi dan memungkinkan respons yang lebih cepat dari asisten virtual [18][21]. Penelitian Galvez et al. (2024) menunjukkan bagaimana fitur CUDA baru dapat digunakan untuk mengoptimalkan decoding pada model speech recognition berbasis RNN-T, yang menggarisbawahi pentingnya akselerasi GPU dalam tugas-tugas semacam ini [21].