PENGEMBANGAN ALAT TRANSKRIPSI *REAL-TIME* BERBASIS *ON-DEVICE SPEECH RECOGNITION* BAGI PENGGUNA DENGAN GANGGUAN PENDENGARAN

Marlon Brando Layanto[1], Hadian Satria Utama[2], Wahidin Wahab[3]

[123,] Program Studi Teknik Elektro, Fakultas Teknik, Universitas Tarumanagara

Jl. Letjen S. Parman No.1, Grogol Petamburan, Jakarta Barat
[1] marlon.525210011@stu.untar.ac.id, [2] hadianu@ft.untar.ac.id, [3] wahidinwahab@ft.untar.ac.id

Abstract

Hearing loss can severely diminish social communication and overall quality of life. To broaden access beyond hearing aids and cochlear implants, this study presents a real-time transcription device using on-device automatic speech recognition (ASR). Its hardware consists of microphone, microcontroller with bluetooth, heads-up display and power supply, that will be placed inside the device and mounted on eyewear. The microphone captures speech and transmits it to a smartphone, where the speech will be transcribed using. Transducer-architecture on-device ASR model with a Zipformer encoder before returning the text to the heads-up display directly to the user's eye. Performance was evaluated in terms of word error rate (WER) and real-time factor (RTF). The ASR achieved a lowest average WER of 9.81% in four datasets. In live trials at 20–100 cm in controlled environments (40–60 dB), it yielded 12.93% WER and 0.015 RTF; in noisier settings (60–70 dB), yielded 27.53% WER and 0.015 RTF. Notably, at a 20 cm distance, WER difference between environments was only 0.19%. These findings demonstrate that the device can deliver rapid and sufficiently accurate speech transcription at typical conversational distances.

Keywords : *Automatic Speech Recognition, Hearing Loss, Real-Time Transcription, On-Device ASR, Assistive Technology*

Abstrak

Gangguan pendengaran berdampak signifikan terhadap kualitas hidup penderitanya, terutama dalam aspek komunikasi sosial. Untuk mengatasi keterbatasan akses terhadap teknologi seperti *hearing aid* dan implan koklea, penelitian ini merancang alat bantu berupa perangkat transkripsi *real-time* berbasis teknologi *automatic speech recognition* (*ASR*). Perangkat terdiri dari mikrofon, mikrokontroler dengan modul *bluetooth*, *heads-up display* dan *power supply* yang ditanam dalam alat yang dapat dipasang pada kacamata. Suara yang ditangkap mikrofon dikirim ke smartphone, ditranskripsikan menggunakan model *ASR on-device* berbasis arsitektur *Transducer* dengan encoder *Zipformer*, kemudian hasil teks dikirim kembali ke alat untuk ditampilkan langsung ke mata pengguna melalui *heads-up display.* Pengujian dilakukan terhadap akurasi transkripsi menggunakan metrik *word error rate* (*WER*) dan kecepatan transkripsi *menggunakan real-time factor* (*RTF*). Model ASR menunjukkan performa baik dengan rata-rata *WER* terendah sebesar 9,81% pada empat dataset uji. Uji coba alat secara langsung di lingkungan terkontrol (40-60 dB) pada jarak 20-100 cm menunjukkan rata-rata *WER* sebesar 12,93% dan RTF sebesar 0,015, sedangkan di lingkungan tidak terkendali (60-70 dB), alat memiliki *WER* sebesar 27,53% dan *RTF* sebesar 0,015. Pada jarak 20 cm, *WER* alat pada kedua lingkungan memiliki perbedaan 0,19%. Hasil ini menunjukkan alat mampu melakukan transkripsi dengan cepat dan cukup akurat pada jarak alat yang dekat dengan pembicara.

Kata kunci : *Automatic Speech Recognition, Gangguan Pendengaran, Transkripsi Real-Time, ASR On-Device, Teknologi Pembantu*

1. PENDAHULUAN

Gangguan pendengaran merupakan kondisi yang dimiliki oleh banyak orang. Menurut World Health Organization (WHO), sebanyak 430 juta penduduk di seluruh dunia membutuhkan bantuan untuk mengatasi gangguan pendengarannya [1]. Orang yang memiliki gangguan pendengaran akan mengalami penurunan kondisi kesehatan lebih besar daripada orang yang mempunyai pendengaran normal. Kondisi yang dialami seperti depresi, isolasi sosial, dan penurunan kualitas hidup [2]. Hal ini dikarenakan oleh kesulitan besar yang dialami orang tersebut untuk berkomunikasi dibandingkan dengan orang normal pada umumnya [3]. Bahkan, gangguan pendengaran merupakan salah satu faktor yang dapat menyebabkan demensia. Di atas 8% kasus global demensia yang diperkirakan disebabkan oleh gangguan pendengaran [4]. Oleh karena itu, aksesibilitas merupakan salah satu faktor yang penting untuk mengatasi masalah kualitas hidup orang tersebut.

Berbagai teknologi telah dikembangkan untuk membantu penderita gangguan pendengaran, seperti *hearing aid* dan implan koklea. Namun, akses teknologi tersebut di atas sangat terbatas. Untuk 400 juta lebih individu, hanya kurang dari 20% mempunyai akses teknologi tersebut [5]. Hal ini disebabkan oleh akses dan kualitas pelayanan kesehatan yang kurang, kurangnya kesadaran akan manfaat teknologi tersebut dalam masyarakat, dan biaya yang tinggi untuk teknologi tersebut [6]. Oleh karena itu, diperlukan teknologi alternatif yang murah dan mudah diakses individu yang memiliki akses terbatas terhadap *hearing aid* atau implan koklea. Akses untuk teknologi alternatif ini akan membantu individu tersebut untuk meningkatkan kualitas hidupnya sehingga kesehatannya semakin meningkat.

Salah satu teknologi alternatif yang dapat digunakan adalah kacamata transkripsi dengan *automatic speech recognition* (*ASR*). *Automatic speech recognition* adalah teknologi yang menerima suara ucapan sebagai *input*, dan mengubah suara yang diterima menjadi teks [7]. Teknologi tersebut telah dibuat dengan desain yang beragam. Kacamata yang dibuat oleh Sinha dan Cavelry [8] menggunakan kacamata Google Glass sebagai layar visual dan memiliki fitur transkripsi *real-time* menggunakan teknologi *automatic speech recognition* berbasis *cloud* dari IBM. Pada penelitan Ridha dan Shehieb [9], prototipe kacamata yang dibuat menggunakan kacamata MadGaze X5 dan memiliki fitur transkripsi *real-time* menggunakan teknologi *automatic speech recognition* berbasis cloud dari Google serta indikasi suara lingkungan, seperti suara tembakan atau gelas yang pecah. Kedua penelitian tersebut menggunakan teknologi *ASR* berbasis cloud, yaitu komputasi yang dilakukan untuk mengubah suara menjadi teks dikerahkan ke *server* luar, sehingga menimbulkan masalah latensi dan keandalan, terutama di daerah dengan akses internet terbatas. Kedua penelitian tersebut juga menggunakan kacamata yang telah dijual atau dirancang dengan harga yang mahal, sehingga kurangnya aksesibillitas yang diberikan. Oleh karena itu, alat yang dikembangkan perlu memberikan aksesibilitas yang luas,

Alat yang dikembangkan adalah kacamata dengan kebaruan implementasi menggunakan *on-device ASR*, yaitu jenis *ASR* yang dapat dijalankan dari alat yang dipakai oleh pengguna itu sendiri, tanpa harus terhubung ke koneksi internet [10]. Alat yang akan dipakai untuk implementasi tersebut adalah *smartphone*. Implementasi ini akan mengurangi latensi, meningkatkan keandalan dan kesediaan dari model *ASR* yang dipakai, serta keuntungan privasi tanpa terhubung ke internet [11]. Kacamata ini juga akan dibuat menggunakan *3D printer*, sehingga masyarakat tidak perlu membeli kacamata pintar jadi yang mahal.

Pada penelitian ini, akan dirancang alat yang mengubah suara ucapan menjadi teks dan ditampilkan pada alat. Alat ini akan menangkap suara dari mikrofon untuk dikirimkan ke *smartphone* melalui *bluetooth*. *Smartphone* yang dilengkapi dengan aplikasi *ASR* akan memproses data suaranya dan mengubah suara menjadi teks, lalu hasil teksnya dikirim kembali pada alat untuk ditampilkan. Alat ini memiliki berbagai keuntungan, seperti kenyamanan bagi pengguna, karena dengan alat ini, pengguna tidak perlu memegang *smartphone* dan melihat layarnya. Hal ini dikarenakan hasil teks dari aplikasi *ASR* akan ditampilkan langsung ke mata pengguna pada alat tersebut, sehingga interaksi sosial dengan lawan pembicara lancar dan tidak canggung. Selain itu, aplikasi pada *smartphone* menggunakan implementasi *on-device ASR*, sehingga pengguna tidak memerlukan koneksi internet untuk menjalankan aplikasi. Hal ini sangat berguna apabila pengguna berada di wilayah dengan jangkuan internet yang lemah atau tidak ada.

2. METODOLOGI PENELITIAN

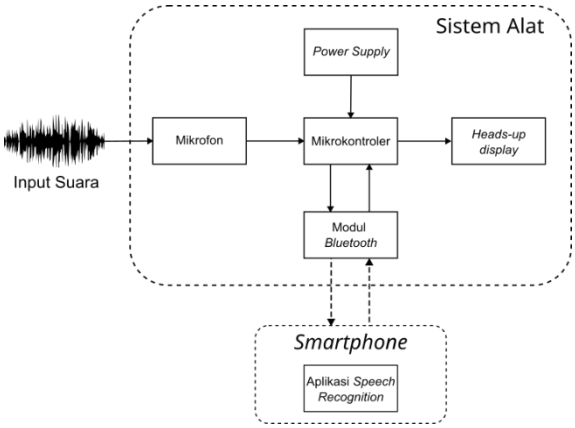
2.1. Alur Penelitian

Penelitian akan dilakukan dalam 5 tahap, yaitu identifikasi masalah, studi literatur, perancangan *hardware* dan *software*, pengujian

sistem, dan kesimpulan. Penelitian diawali dengan mengidentifikasi masalah yang menjadi persoalan dalam penelitian ini, yaitu diperlukannya teknologi alternatif untuk individu dengan gangguan pendengaran. Selanjutnya akan dilakukan studi literatur, yaitu proses penelusuran dari berbagai referensi/sumber informasi yang relevan dengan topik penelitian. Tahap selanjutnya adalah perancangan *hardware* dan *software* alat. Perancangan *hardware* meliputi tiga bagian, yaitu bagian badan, bagian elektronik, dan bagian *heads-up display*, sedangkan perancangan *software* meliputi dua bagian, yaitu aplikasi *ASR* dan program mikrokontroler. Setelah perancangan *hardware* dan *software*, sistem akan diuji keakuratan transkripsinya serta kecepatan tranksripsi dapat dihasilkan. Kesimpulan dari penelitian akan dibuat berdasarkan data-data yang dihasilkan dari pengujian sistem.

2.2. Perancangan Alat

Alat yang dirancang akan membantu individu dengan gangguan pendengaran dengan mengubah suara menjadi teks. Diagram blok alat perancangan dapat dilihat pada Gambar 2.

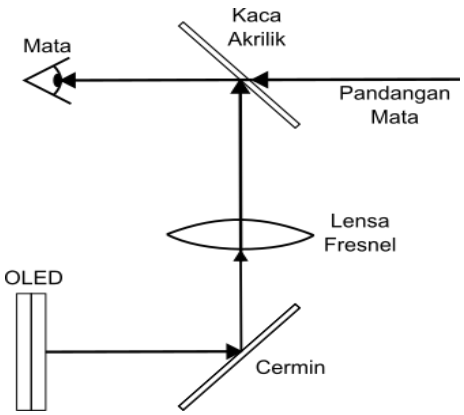


Gambar 1. Diagram Blok Alat Rancangan

Suara yang datang dari pembicara akan ditangkap mikrofon. Mikrofon akan mengubah gelombang suara yang ditangkap menjadi sinyal listrik. Sinyal tersebut akan diproses oleh mikrokontroler untuk dikirim ke smartphone melalui modul *bluetooth*. Aplikasi dari *smartphone* akan menerima sinyal dari modul *bluetooth* untuk mengubah suara menjadi teks menggunakan *ASR*. Hasil teks dari pengubahan suara akan dikirim ke mikrokontroler melalui modul bluetooth dan akan ditampilkan ke *heads-up display*. Alat ini perlu dipasang ke kacamata pengguna untuk melihat tampilan *heads-up display*.

2.3. Perancangan *Heads-Up Display*

Heads-up display (*HUD*) adalah suatu perangkat tampilan yang berfungsi untuk menampilkan informasi yang penting seperti gambar atau tulisan tanpa mengalihkan pandangan utama dari pengguna. Skema perancangan *HUD* dapat dilihat pada Gambar 3.



Gambar 2. Skema Rancangan *Heads-Up Display*

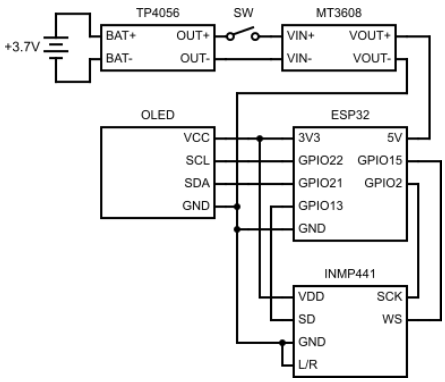
Layar *OLED* 0,49 inci akan digunakan untuk menampilkan hasil transkripsi. Hasil teks dari layar *OLED* akan dipantulkan oleh cermin. Hasil pantulan dari cermin akan diarahkan ke lensa *fresnel*, lensa *fresnel* akan meneruskan ke kaca akrilik yang memantulkannya ke mata. Lensa *fresnel* adalah lensa cembung yang terdiri dari serangkaian alur konsentris [12]. Lensa ini digunakan karena ketebalannya yang tipis serta lebih ringan dibandingkan lensa cembung konvensional [13]. Lensa tersebut digunakan untuk menghasilkan gambar maya yang dapat dilihat oleh pengguna. Jarak minimum objek yang dapat dilihat dengan jelas oleh mata manusia adalah 25 cm [14]. Jarak fokus pada lensa yang digunakan adalah 10 cm. Jarak layar *OLED* agar hasil teks dapat terlihat dengan jelas dapat dihitung dengan persamaan 1:

$$\frac{1}{f} = \frac{1}{i} + \frac{1}{o} \quad (1)$$

yang di mana f adalah jarak fokus lensa, i adalah jarak gambar maya dari lensa, dan o adalah jarak objek dari lensa. Apabila jarak fokus lensa adalah 10 cm dan jarak gambar maya yang diperlukan adalah 25 cm, maka dapat dihitung jarak objek pada layar *OLED* dari lensa yang diperlukan adalah 7,2 cm. Hasil gambar maya dari teks akan dipantulkan ke kaca akrilik yang dilapisi dengan film semi reflektif, sehingga diarahkan menuju mata pengguna. Hal ini memungkinkan pengguna untuk dapat melihat hasil teks serta tanpa mengganggu pandangan utama pengguna.

2.4. Perancangan Bagian Elektronik

Bagian elektronik berfungsi untuk menerima suara dari mikrofon dan mengirim data suara ke *smartphone* serta menerima hasil teks dari *smartphone* untuk ditampilkan ke *heads-up display*. Komponen elektronik yang akan digunakan pada alat ini adalah mikrokontroler ESP32 WROVER-E, modul *charger* TP4056, baterai *Li-Po* 3,7 V, modul *boost converter* MT3608, mikrofon INMP441, sakelar geser dan layar *OLED* SSD1306 0,49 inci. Mikrokontroler ESP32 WROVER-E adalah mikrokontroler dengan mikroprosesor Xtensa *dual-core 32-bit* LX6 dengan kecepatan *clock* sebesar 240 MHz lengkap dengan fitur WiFi/Bluetooth. Mikrokontroler ini juga memiliki memori *flash* untuk menyimpan data berkelanjutan sebesar 4 MB [15]. Mikrokontroler ini digunakan karena memori yang cukup untuk mengendalikan aplikasi *audio* dan memiliki modul *Bluetooth* yang terintegrasi. Mikrofon INMP441 adalah mikrofon dengan ukuran yang kecil serta memiliki protokol I2S, yaitu protokol komunikasi untuk mengirimkan data *audio* antara perangkat digital lainnya secara langsung, tanpa alat eksternal. Mikrofon ini dipilih karena ukurannya yang kecil dan memiliki protokol I2S. Layar *OLED* yang digunakan memiliki *driver* SSD1306 untuk mengendalikan layar *OLED* tersebut. *Driver* tersebut juga memiliki fitur protokol I2C (*Inter-Integrated Circuit*), yaitu protokol komunikasi serial yang digunakan untuk menghubungkan mikrokontroler dengan perangkat lainnya [16]. TP4056 adalah modul *charger* yang memiliki *output* arus di atas 1 A serta memiliki fitur *constant-voltage/constant-current* agar arus dan tegangan yang diberikan lebih stabil [17]. MT3608 adalah modul untuk menaikkan tegangan, yang disebut dengan *boost converter*. Modul ini dapat menaikkan tegangan dari tegangan masukkan 2-24 V menjadi tegangan keluaran 2-28 V [18]. Skematik rangkaian elektronik alat dapat dilihat pada Gambar 4.



Gambar 3. Skematik Elektronik Alat

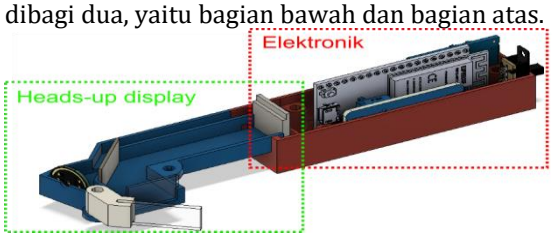
Baterai dihubungkan ke modul *charger* TP4056 agar baterai dapat diisi ulang. Sakelar geser SW digunakan menghidupkan atau mematikan alat rancangan. Untuk memenuhi tegangan ESP32, tegangan 3,7 V dari baterai akan diubah menjadi 5 V dengan *boost converter* MT3608. ESP32 akan mengirimkan data serial hasil teks ke layar *OLED* dengan protokol I2C melalui pin SDA dan SCL.. Mikrofon INMP441 akan mengirimkan data *audio* dengan protokol I2S melalui pin WS, SD, dan SCK.

2.5. Perancangan Badan Alat

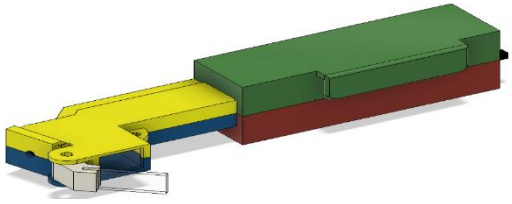
Hasil rancangan elektronik dan *heads-up display* alat akan ditempatkan pada badan alat. Badan alat didesain dengan software Fusion 360 dan dibuat menggunakan *3D printer*. Badan alat dapat dilihat pada Gambar 5.

Gambar 4. Badan-Badan Alat Rancangan

Badan alat dibagi menjadi dua, yaitu badan untuk *heads-up display* dan badan untuk *casing* elektronik. Badan *heads-up display* akan menempatkan komponen cermin, lensa *fresnel* dan kaca akrilik, sedangkan badan *casing* elektronik akan menempatkan komponen mikrokontroler dan modul *bluetooth ESP32 WROVER-E*, modul layar *OLED*, baterai Li-Po 550 mAh, modul *boost converter* MT3608, modul *charger* TP4056, dan sakelar geser. Mikrofon INMP441 ditempatkan di badan *heads-up display* agar mikrofon dapat menangkap suara pembicara. Kedua bagian badan tersebut masing-masing juga dibagi dua, yaitu bagian bawah dan bagian atas.



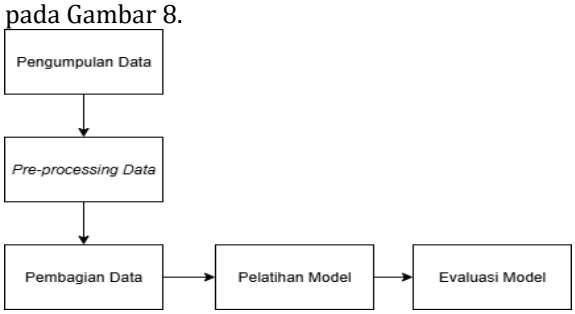
Gambar 5. Dua Bagian Badan Alat



Gambar 6. Badan Alat Komplit

2.6. Perancangan Model *ASR*

Untuk melatih model yang digunakan dalam aplikasi *ASR*, diperlukan data suara dari berbagai sumber. Perancangan model *ASR* akan dilakukan dengan bahasa pemrograman Python. Tahapan perancangan model *ASR* tersebut diilustrasikan pada Gambar 8.



Gambar 7. Tahap-Tahap Perancangan Model *ASR*

2.6.1. Pengumpulan Data

Untuk melatih model *ASR*, data-data yang digunakan adalah kumpulan rekaman suara dari berbagai sumber beserta transkripsinya. Data akan diperoleh dari HuggingFace, yaitu *platform* komunitas untuk mengembangkan, membagi dan menerapkan model *AI* [19]. *Dataset* yang digunakan adalah YODAS, Common-Voice, FLEURS, dan LibriVox.

YODAS [20] memiliki data-data dari video Youtube yang mempunyai transkripsi manual. Satu video dari Youtube akan dibagi menjadi beberapa rekaman suara sesuai dengan waktu transkripsinya. YODAS memiliki versi alternatif bernama YODAS2, yaitu *dataset* yang sama dengan YODAS tetapi rekaman suaranya tidak dibagi. Oleh karena itu, YODAS2 memiliki format satu rekaman suara per video dengan berbagai transkripsi. Untuk penelitian ini, YODAS2 akan digunakan karena *dataset* YODAS memiliki banyak transkripsi yang tidak akurat karena rekaman suara yang terpotong. FLEURS [21] dan Common Voice [22] terdiri dari beberapa rekaman ucapan yang telah divalidasi oleh pembuat *dataset* untuk digunakan. Librivox terdiri dari rekaman suara dari beberapa buku audio berbahasa Indonesia. Total durasi *dataset* terlihat pada Tabel I.

TABEL I. TOTAL DURASI *DATASET*

Nama	Total Durasi (jam)
YODAS2	1420,1
Common-Voice	12
FLEURS	15
LibriVox	7
Total	1454,1

2.6.2. *Pre-processing* Data

Pre-processing data adalah proses modifikasi data menjadi data yang siap digunakan. Modifikasi data akan diimplementasi menggunakan Lhotse. Lhotse adalah perpustakaan *software* untuk merepresentasikan data suara sehingga data tersebut dapat dimodifikasi sesuai kebutuhan [23]. Modifikasi pertama yang akan dilakukan pada setiap *dataset* adalah mengubah laju sampel *dataset*. Laju sampel pada setiap rekaman suara akan diubah menjadi 16.000 Hz. Selanjutnya, setiap *dataset* akan dimodifikasi dengan cara yang berbeda berdasarkan tingkat *noise*nya. Dataset pertama yang akan dimodifikasi adalah YODAS. *Dataset* tersebut memiliki *noise* yang bervariasi, seperti transkripsi yang salah, rekaman suara yang tidak berbahasa Indonesia, dll. Oleh karena itu, data-data tersebut perlu dibersihkan dengan tiga tahap, yaitu identifikasi bahasa, saring teks, dan *forced alignment*.

Walaupun *dataset* YODAS terbilang memiliki data rekaman suara berbahasa Indonesia, lebih dari setengah data tersebut adalah rekaman suara *noise* dan tidak berbahasa Indonesia. Oleh karena itu, data-data tersebut harus disaring sehingga rekaman suara tersebut dapat digunakan. Model Whisper Base dari OpenAI [24] akan digunakan untuk mengidentifikasi bahasa dari data-data tersebut. Untuk setiap rekaman *audio*, Whisper akan mengidentifikasi bahasa di segmen 30 detik dipilih dari tengah rekaman. Whisper kemudian akan memberikan skor dari 0 sampai 1 untuk mengetahui keakuratan identifikasi bahasa. Data *audio* tidak akan digunakan apabila skor kurang dari 0,5.

Apabila *dataset* sudah diidentifikasi bahasanya, tahap selanjutnya adalah penyaringan teks. Untuk setiap teks, terdapat beberapa perubahan yang diperlukan secara bertahap agar teks dapat digunakan, yaitu:

a. Semua huruf diubah menjadi huruf besar.
b. Mengubah operator matematika menjadi bentuk verbal, contohnya "+" diubah menjadi "TAMBAH".
c. Emotikon dan karakter ekspresif akan dihapus
d. Angka berurutan seperti 123456 dipisahkan menjadi angka individu, seperti "1 2 3 4 5 6"
e. Ubah format waktu menjadi bentuk verbal. Contoh, "pukul 10.00" menjadi "pukul sepuluh".
f. Hapus tanda titik pada angka ribuan. Contoh, angka "12.000" diganti menjadi "12000".
g. Ubah tanda titik dalam alamat IP menjadi bentuk verbal. Contoh, angka "192.168.32" menjadi "192 titik 168 titik 32"
h. Ubah satuan menjadi bentuk verbal. Contoh, satuan km/jam menjadi "kilometer per jam"

i. Tambahkan spasi pada setiap nomor telepon. Pola nomor telepon adalah empat digit yang dipisah dengan spasi menjadi dua atau empat grup.. Contoh, angka "4356 2613" diubah menjadi "4 3 5 6 2 6 1 3".
j. Angka desimal akan dipisah menjadi beberapa digit individu dan tanda koma diubah mnjadi bentuk verbal. Contoh, angka 1,6342 diubah menjadi "1 KOMA 6 3 4 2".
k. Tambahkan spasi antara angka dan huruf.
l. Ubah angka pecahan menjadi bentuk verbal. Contoh, angka "5/2" diubah menjadi "5 PER 2"
m. Hapus semua teks yang berhubungan dengan Hypertext Markup Language (HTML).
n. Hapus label dialog pada transkripsi. Contoh, kata "Dimas:" akan dihapus.
o. Hapus karakter yang berulang-ulang dan non-alfanumerik
p. Normalisasi dan romanisasi teks.

Setelah teks disaring, langkah berikutnya adalah forced alignment menggunakan model CTC dari TorchAudio. Rekaman dibagi berdasarkan cap waktu transkripsi, ditambah *padding* 2 detik di depan dan belakang, dan transkripsi diberi tanda bintang di awal/akhir untuk menandai batas penyelarasan. Fitur diekstrak dengan model akustik Wav2Vec 2.0 yang menghasilkan distribusi probabilitas token setiap 20 ms, lalu transkripsi di-tokenisasi ke token-model yang sama, lalu CTC mencari jalur penyelarasan optimal antara urutan token dan frame audio. Hasilnya adalah pemetaan waktu untuk tiap token yang kemudian digabung menjadi kata, dengan setiap kata diberi skor keyakinan 0-100. Rata-rata skor per transkripsi digunakan untuk menyaring. Transkripsi dengan skor kurang dari 70 akan dihapus, dari yang tersisa kata-kata dengan skor kurang dari 10 dihapus, dan kata umum seperti "hai" atau "halo" dihapus jika memiliki skor kurang dari 70.

Untuk dataset Common-Voice, FLEURS, dan LibriVox, hanya dilakukan penyaringan teks seperti penghapusan tanda baca, normalisasi, dan romanisasi. Setelah semua dataset termodifikasi, dibuatlah token untuk model ASR melalui subword tokenization, yaitu memecah kata jarang muncul menjadi subkata, Fitur audio diekstrak menggunakan log-mel filter bank. Fitur kemudian diaugmentasi dengan menambahkan noise dari MUSAN dan teknik SpecAugment, yaitu menutup rentang frekuensi tertentu dan pembengkokkan waktu fitur untuk meningkatkan performa. Untuk kestabilan pelatihan, audio berdurasi lebih dari 20 detik atau kurang dari 1 detik dihapus. Akhirnya data disusun menjadi file tar dan dibagi menjadi beberapa shard untuk mempercepat akses saat pelatihan.

2.6.3. Pembagian Data

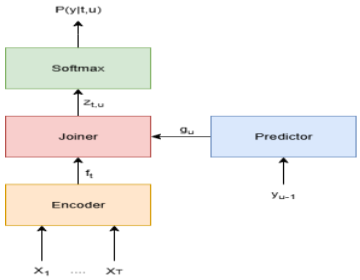
Setelah *pre-processing* data, setiap *dataset* dibagi menjadi tiga bagian, yaitu bagian *training, validiation*, dan *test*. Data training adalah data yang digunakan untuk melatih model dan data *validation* digunakan untuk mengevaluasi model selama pelatihan. Hasil akhir dari model yang sudah dilatih akan diuji oleh data *test* [32]. Pembagian data ini dilakukan agar model *ASR* dapat melakukan prediksi untuk data-data yang tidak ada dalam pelatihan [33]. Total durasi pada setiap bagian *dataset* dapat diluhat pada Tabel II.

TABEL II. PEMBAGIAN *DATASET*

Nama	Total Durasi (jam)		
	Training	*Validation*	*Test*
YODAS2	533	12	12
Common-Voice	9	1	2
FLEURS	7	4	4
LibriVox	6	-	1
Total	555	17	19

2.6.4. Pelatihan Model

Pada tahap ini, model *ASR* akan dilatih dengan data yang sudah disaring. Arsitektur *ASR* yang digunakan pada penelitian ini adalah *transducer*, yang arsitekturnya dapat dilihat pada Gambar 9.



Gambar 8. Arsitektur *Transducer*

Encoder berfungsi untuk memodelkan fitur akustik menjadi vektor f_t dari input fitur suara x_1, x_2, dan seterusnya sampai x_T. *Predictor* berfungsi sebagai model bahasa dengan inputnya adalah hasil output dari prediksi sebelumnya, yaitu y_{u-1} dan *output*nya adalah vektor g_u. *Joiner* menggabungkan hasil *encoder* dan *decoder* untuk mengeluarkan output prediksi teks $z_{t,u}$ [34]. Hasil *output* vektor $z_{t,u}$ akan diubah menjadi probabilitas $P(y|t,u)$ dari setiap *token* menggunakan fungsi *softmax*. Arsitektur *transducer* ini dipilih karena dibandingkan dengan arsitektur alternatif seperti *Connectionist*

Temporal Classification (CTC) atau *encoder-decoder* dengan fitur *attention*, akurasi *transducer* lebih akurat daripada CTC [35] dan walaupun akurasi lebih tinggi pada model *encoder-decoder*, memori yang digunakan lebih besar dibandingkan model *transducer*.

Encoder yang akan digunakan pada arsitektur tersebut adalah *zipformer*, yaitu jenis *encoder* dalam *automatic speech recognition* (*ASR*) yang dirancang lebih efisien dan cepat dibanding arsitektur *Transformer* standar, dengan teknik khusus seperti *downsampling* dan *upsampling* untuk "mengompres" dan "mendekompresi" representasi fitur suara secara bertahap sehingga proses komputasi menjadi lebih cepat tanpa kehilangan akurasi [36]. *Predictor* yang digunakan adalah *embedding layer* dan satu *1-D convolutional layer*. *Joiner* yang digunakan adalah satu *linear layer*.

Untuk mengubah suara menjadi teks secara *real-time*, jenis *ASR* yang digunakan adalah jenis *streaming*. *Streaming ASR* akan menghasilkan teks dari setiap *audio chunk* atau potongan *audio* secara berurutan [37]. Besarnya *chunk* atau *chunk size* akan memengaruhi latensi dan akurasi dari model *ASR*.

Model akan dilatih menggunakan Google Colab Free Tier dan dilatih dengan dua tahap. Model akan dilatih menggunakan *dataset* YODAS2 terlebih dahulu. Apabila sudah dilatih, model akan dilatih lebih lanjut menggunakan *dataset* Common-Voice, FLEURS, LibriVox. Tahap pertama pelatihan akan dilakukan dengan *parameter* model sebesar 66 juta, *learning rate* sebesar 0,045, serta durasi maksimum *audio* yang dapat diakses saat pelatihan model sebesar 400 detik. Model tersebut akan dilatih sebanyak 202000 langkah. Untuk pelatihan tahap kedua, model akan dilatih sebanyak 100000 langkah dan *learning rate* yang digunakan sebesar 0,0035 dengan durasi maksimum *audio* yang dapat diakses sebesar 500 detik. Pemilihan *hyperparameter* model *ASR* yang lain mengikuti model Zipformer-M [36]. Model dilatih menggunakan *library* Icefall.

2.6.5. Evaluasi Model

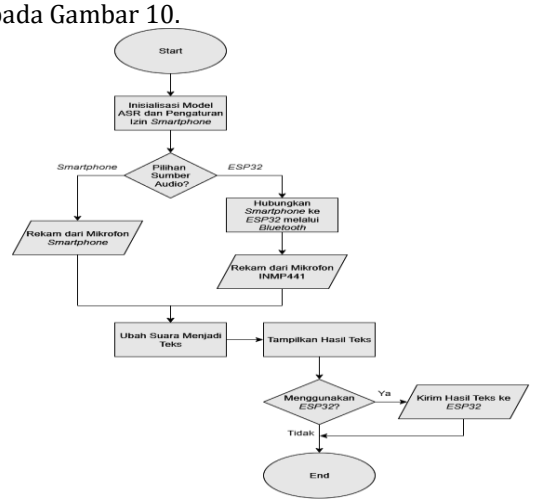
Model *ASR* yang sudah dilatih akan dievaluasi dengan menghitung *word error rate* (*WER*) pada bagian *test* setiap *dataset*. *Word error rate* adalah perhitungan standar untuk mengevaluasi keakuratan transkripsi dari *automatic speech recognition* [39]. *WER* dapat dihitung dengan persamaan 2.

$$\mathrm{WER} = \frac{\mathrm{I} + \mathrm{D} + \mathrm{S}}{\mathrm{N}} \times 100\% \qquad (2)$$

Di mana I adalah jumlah kata pada *output* yang tidak ada di referensi teks (*Insertion*), S adalah jumlah kata pada *output* yang terganti dari referensi teks (*Substitution*), D adalah jumlah kata pada *output* yang terhapus dari referensi teks (*Deletion*), dan N adalah jumlah kata dari referensi teks.

2.7. Perancangan Aplikasi *ASR*

Model *ASR* yang sudah dilatih akan dikonversi menjadi format ONNX untuk digunakan pada aplikasi Android. Pembuatan aplikasi akan menggunakan Android Studio. Implementasi model *ASR* akan dijalankan dengan kerangka kerja Sherpa-onnx. Sherpa-onnx dipilih karena kerangka kerja tersebut mempunyai bantuan untuk menjalankan *ASR* secara waktu nyata.. Diagram alir aplikasi tersebut dapat dilihat pada Gambar 10.



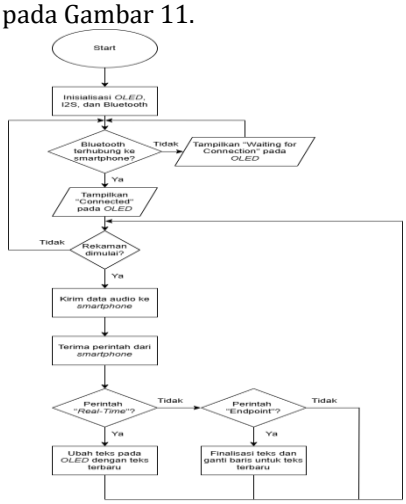
Gambar 9. Diagram Alir Aplikasi *ASR*

Pertama, program akan menginisalisasi model *ASR* dan mengatur izin pada *smartphone*, seperti izin untuk menyalakan *bluetooth*. Pengguna lalu memilih sumber *audio* yang diinginkan, mikrofon dari *smartphone* atau mikrofon INMP441. Jika pengguna memilih mikrofon INMP441, aplikasi akan menghubungkan *smartphone* ke *ESP32* melalui *bluetooth*, lalu rekam suara dari mikrofon INMP441. Hasil rekaman suara akan diubah menjadi teks dengan model *ASR* yang sudah dilatih lalu hasil teks ditampilkan pada *smartphone*. Apabila *smartphone* terhubung ke *ESP32*, kirim hasil teks ke *ESP32*.

2.8. Perancangan Program Mikrokontroler

Untuk menampilkan hasil transkripsi pada layar *OLED*, mikrokontroler harus mengirim dan menerima data *audio* secara *real-time*. Program mikrokontroler akan dibuat dengan Arduino IDE.

Diagram alir program mikrokontroler dapat dilihat pada Gambar 11.



Gambar 10. Diagram Alir Program Mikrokontroler

Program diawali dengan inisialisasi layar *OLED*, I2S, dan *bluetooth*. Apabila mikrokontroler terhubung ke *smartphone* melalui *bluetooth*, layar *OLED* akan menampilkan teks "Connected". Teks "Waiting for Connection" akan ditampilkan pada layar *OLED* jika *bluetooth* tidak terhubung ke *smartphone*. Apabila rekaman dimulai, mikrofon INMP441 akan tangkap suara yang berbicara dan kirim data suara tersebut ke *smartphone*, kemudian *smartphone* akan mengirimkan perintah "*Real-Time*" atau "*Endpoint*". Untuk perintah "*Real-Time*", mikrokontroler akan mengubah teks pada layar *OLED* dengan hasil transkripsi yang terbaru. Namun, jika perintah yang didapatkan adalah "*Endpoint*", mikro kontroler akan mengganti baris untuk hasil transkripsi yang terbaru. Proses tersebut akan diulang sampai pengguna ingin berhenti merekam suara.

3. HASIL DAN PEMBAHASAN

3.1 Pengujian Akurasi *ASR*

Akurasi *ASR* akan diuji menggunakan metode *WER*. *Dataset* yang diuji adalah bagian *test* dari YODAS2, Common-Voice, FLEURS, dan LibriVox. Terdapat dua nilai *chunk size* dan tiga nilai konteks kiri yang diuji, masing-masing 320ms dan 640ms, serta 64, 128, dan 256. Model *ASR* akan dibandingkan dengan model *ASR* Whisper Small Hasil *WER* dari *ASR* dapat dilihat pada Tabel III.

TABEL III. HASIL *WER* (%) MODEL *ASR*

Dataset	*Chunk Size*	*Frame* Konteks Kiri		
		64	128	256
YODAS2	320ms	12,82	12,67	12,64
	640ms	11,98	11,90	11,90
Common-Voice	320ms	12,90	12,82	12,82
	640ms	11,66	11,57	11,*58*
FLEURS	320ms	10,63	10,01	9,62
	640ms	9,71	9,11	8,96
LibriVox	320ms	7,88	7,74	7,82
	640ms	6,91	6,86	6,79
Rata-rata	320ms	11,06	10,81	10,73
	640ms	10,07	9,86	9,81

Berdasarkan hasil *WER* pada Tabel III, semakin besar nilai *chunk size* dan *frame* konteks kiri, semakin akurat transkripsi dari model *ASR*. *Chunk size* 640ms memiliki nilai *WER* rata-rata yang lebih kecil (10,07%, 9,86%, dan 9,81%) dibandingkan *chunk size* 320ms (11,06%, 10,81%, dan 10,73%) pada setiap *frame* konteks kiri. Nilai rata-rata *frame* konteks kiri 256 (10,73% dan 9,81%) memiliki nilai *WER* yang paling kecil dibandingkan nilai *frame* konteks kiri lainnya. *Dataset* yang memiliki nilai *WER* paling kecil pada setiap *chunk size* dan *frame* konteks kiri adalah *dataset* LibriVox, sedangkan *dataset* yang memiliki nilai yang paling besar adalah YODAS2. Nilai *WER* LibriVox paling kecil karena data *test* yang mirip dengan data *training* dan *validation*, sedangkan nilai *WER* YODAS2 paling besar karena walaupun *dataset*nya sudah dibersihkan, terdapat beberapa transkripsi yang masih memiliki *noise* serta kata-kata yang terhapus saat *pre-processing*. Untuk perbandingan data

Untuk hasil yang akurat, model *ASR* yang digunakan pada aplikasi *ASR* adalah *chunk size* 640ms dengan *frame* konteks kiri 256. Untuk menunjukkan keunggulan model *ASR* ini, model *ASR* dengan akurasi paling tinggi akan dibandingkan dengan model Whisper-medium dari OpenAI. Model tersebut dipilih karena skala model yang sama dengan model *ASR* penelitian, yaitu *medium* dan karena Whisper merupakan salah satu model *ASR* yang *open-source*.

TABEL IV. PERBANDINGAN *WHISPER* DENGAN MODEL *ASR* PENELITIAN

Dataset	*WER* (%)	
	Whisper-medium	Model *ASR Penelitian*
YODAS2	25,90	11,90
Common-Voice	12,04	11,58
FLEURS	12,91	8,96
LibriVox	28,65	6,79
Rata-rata	19,88	9,81

Pada Tabel IV, terlihat bahwa model *ASR* penelitian memiliki nilai *WER* lebih rendah dibandingkan Whisper-medium dalam semua *dataset*. Nilai *WER* pada Librivox merupakan nilai

tertinggi dari Whisper-medium. Hal ini dikarenakan terdapat nama-nama yang tidak dikenal pada *dataset* oleh Whisper. Nilai *WER* pada *dataset* YODAS2 juga tinggi dikarenakan banyak *noise* pada *dataset*.

3.2 Pengujian Aplikasi *ASR*

Alat yang dirancang akan diuji hasil *WER*nya dan seberapa cepat transkripsi yang dapat dihasilkan. Pengujian kecepatan transkripsi tersebut dilakukan dengan metode *real-time factor* (*RTF*), yaitu berapa lama model *ASR* dapat memproses suara menjadi teks dibagi dengan berapa lama suara yang diucap. Apabila nilai *RTF*nya kurang dari atau sama dengan 1, maka model *ASR* tersebut disebut *real-time* [41]. Alat akan diuji dengan 10 sampel yang berisi suara ucapan yang ditangkap oleh mikrofon. Sampel tersebut akan dihitung keakuratannya dengan *WER*nya dan *real-time factor*. Alat akan diuji di dua tempat, yaitu lingkungan yang terkontrol dengan nilai intensitas suara berada pada *range* 40-60 dB dan lingkungan yang tidak terkontrol dengan nilai intensitas suara berada pada range 60-70 dB. Intensitas suara diukur menggunakan *sound level* meter. Pada setiap tempat tersebut, pengaruh jarak antara alat dan pembicara dengan *WER* dan *RTF* alat juga akan diuji. Tiga jarak yang diuji adalah 20 cm, 50 cm, dan 100 cm. Sampel 1-5 akan menggunakan transkripsi dari berita Kompas [42] dan sampel 6-10 akan menggunakan transkripsi dari Wikipedia [43]. Hasil pengujian alat rancangan dalam lingkungan yang terkontrol terdapat pada Tabel V dan hasil pengujian alat rancangan dalam lingkungan tidak terkontrol terdapat pada Tabel VI.

TABEL IV. HASIL PENGUJIAN ALAT DARI LINGKUNGAN TERKONTROL

Sampel	Jarak Alat dan Pembicara (cm)	*WER* (%)	RTF
1	20	3,33	0,016
	50	10,00	0,019
	100	13,33	0,018
2	20	6,67	0,014
	50	0,00	0,018
	100	10,00	0,017
3	20	7,41	0,014
	50	11,11	0,018
	100	14,81	0,014
4	20	23,08	0,014
	50	7,69	0,016
	100	30,77	0,014
5	20	0,00	0,013
	50	12,50	0,014
	100	25,00	0,012
6	20	8,33	0,013
	50	0,00	0,014
	100	8,33	0,012
7	20	12,50	0,013
	50	25,00	0,014
	100	25,00	0,018
8	20	13,33	0,013
	50	13,33	0,014
	100	13,33	0,016
9	20	4,76	0,012
	50	4,76	0,013
	100	9,52	0,017
10	20	26,09	0.011
	50	13,04	0,016
	100	34,78	0,016
Rata-rata	20	10,55	0,013
	50	9,74	0,016
	100	18,49	0,015
	Semua	12,93	0,015

TABEL VI. HASIL PENGUJIAN ALAT DARI LINGKUNGAN TIDAK TERKONTROL

Sampel	Jarak Alat dan Pembicara (cm)	*WER* (%)	RTF
1	20	13,33	0,017
	50	13,33	0,017
	100	66,67	0,018
2	20	6,67	0,015
	50	10,00	0,018
	100	40,00	0,019
3	20	14,81	0,015
	50	18,52	0,017
	100	55,56	0,017
4	20	23,08	0,013
	50	23,08	0,018
	100	46,15	0,015
5	20	0,00	0,013
	50	12,50	0,011
	100	37,50	0,013
6	20	0,00	0,013
	50	25,00	0,011
	100	33,33	0,016
7	20	12,50	0,013
	50	25,00	0,012
	100	75,00	0,016
8	20	6,67	0,013
	50	20,00	0,016
	100	46,67	0,017
9	20	4,76	0,014
	50	19,05	0,017
	100	71,40	0,015
10	20	21,74	0,013
	50	39,13	0,015

	100	50,00	0,012
Rata-rata	20	10,36	0,014
	50	20,00	0,015
	100	52,23	0,016

Berdasarkan pengamatan pada Tabel IV dan Tabel V, nilai *RTF* alat yang paling besar adalah 0,019, sehingga alat memenuhi syarat *real-time.* Terdapat *WER* bernilai 0 pada kedua tabel tersebut dan nilai *WER* rata-rata pada alat berjarak 20 cm dari pembicara memiliki perbedaan yang kecil, yaitu 0,19%. Hasil ini menunjukkan bahwa pada jarak dekat, alat ini mampu melakukan transkripsi yang akurat dalam kondisi lingkungan yang terkontrol dan lingkungan yang bising.

Pada Tabel IV, alat yang berjarak 50 cm dari pembicara memiliki nilai *WER* rata-rata 9,74%, lebih kecil dibandingkan dengan nilai *WER* rata-rata alat yang berjarak 20 cm dari pembicara, yaitu 10,55%. Hal tersebut disebabkan oleh artkikulasi dari pengucapan beberapa sampel yang lebih jelas pada saat pengujian alat dengan jarak 50 cm dengan pembicara, sehingga *ASR* dapat memproses suaranya lebih mudah. Nilai *RTF* pada pengujian alat berjarak 50 cm dari pembicara (0,016) lebih besar daripada pengujian alat berjarak 100 cm dari pembicara (0,015). Hal ini dikarenakan suara *noise* menutupi suara dari pembicara akibat jarak alat yang jauh, sehingga waktu pemrosesan suara oleh *ASR* berkurang dengan melewati bagian *noise* dari suara. Pada Tabel V, nilai rata-rata *WER* pada setiap jarak alat dan pembicara adalah 10,36%, 20%, dan 52,23% untuk masing-masing jarak 20 cm, 50 cm, dan 100 cm. Nilai rata-rata *RTF* pada setiap jarak alat dan pembicara adalah 0,014, 0,015, 0,016 untuk masing-masing jarak 20 cm, 50 cm, dan 100 cm. Semakin jauh jarak alat dengan pembicara dan lingkungan yang bising, semakin tinggi nilai WER dan *RTF* alat. Hal ini disebabkan oleh *ASR* yang memiiki kesulitan mengekstraksi fitur dari suara, memengaruhi waktu pemrosesan suara.

4. Kesimpulan dan Saran

Pada penelitian ini, dapat disimpulkan bahwa alat yang dirancang dapat melakukan transkripsi dengan cepat dan akurat pada jarak alat yang dekat dengan pembicara walaupun dalam kondisi lingkungan yang tidak terkontrol, yang ditunjukkan oleh perbedaan nilai rata-rata *WER* pada jarak alat dan pembicara 20 cm dengan lingkungan terkontrol (10,55%) dan lingkungan bising (10,36%), yaitu 0,19%. Pada model *ASR,* nilai *chunk size* dan *frame* konteks kiri berpengaruh pada akurasi transkripsi. Semakin besar nilai *chunk size* dan *frame* konteks kiri, maka semakin akurat transkripsi yang dihasilkan. Jarak alat dengan pembicara serta lingkungan yang bising memengaruhi akurasi dan kecepatan transkripsi. Pada lingkungan yang tidak bising, artikulasi dari pembicara juga memengaruhi akurasi transkripsi, dengan artikulasi yang jelas akan menghasilkan transkripsi yang lebih akurat. Hal ini ditunjukkan pada Tabel IV oleh nilai rata-rata *WER* pada jarak 50 cm (9,74%) yang lebih kecil dibanding jarak 20 cm (10,55%)

Untuk penelitian selanjutnya, model *ASR* yang dilatih dapat ditingkatkan akurasinya dengan melatih model bahasa eksternal. Model bahasa tersebut dapat dilatih dengan data-data teks yang lebih banyak sehingga performa *ASR* akan lebih akurat. Data-data untuk *ASR* juga perlu dibuat lebih *noisy* serta *noise* yang beragam agar model *ASR* dapat menghasilkan teks akurat dari suara yang bising. Alat perancangan dapat dibuat lebih kecil dengan merancang elektronik alat menggunakan *Printed Circuit Board* (*PCB*) khusus. Modul-modul dalam elektronik alat ini serta kabel-kabel yang dipakai membutuhkan ruang dalam alat lebih besar.

Daftar Pustaka:

[1] World Health Organization, "Deafness and hearing loss." Diakses: 9 Februari 2025. [Daring]. Tersedia pada: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[2] J. L. Punch, R. Hitt, dan S. W. Smith, "Hearing Loss and Quality of Life," *Journal of communication disorders*, vol. 78, hlm. 33–45, 2019.

[3] J. Gao, H. Hu, dan L. Yao, "The Role of Social Engagement in the Association of Self-Reported Hearing Loss and Health-Related Quality of Life," *BMC Geriatr*, vol. 20, no. 1, hlm. 182, Des 2020, doi: 10.1186/s12877-020-01581-0.

[4] D. S. Powell, E. S. Oh, N. S. Reed, F. R. Lin, dan J. A. Deal, "Hearing Loss and Cognition: What We Know and Where We Need to Go," *Front. Aging Neurosci.*, vol. 13, hlm. 769405, Feb 2022, doi: 10.3389/fnagi.2021.769405.

[5] "Who Issues Guidance to Improve Access to Hearing Care in Low- and Middle-Income Settings." Diakses: 16 Agustus 2024. [Daring]. Tersedia pada: https://www.who.int/news/item/01-03-2024-who-issues-guidance-to-improve-access-to-hearing-care-in-low--and-middle-income-settings

[6] A. Orji, K. Kamenov, M. Dirac, A. Davis, S. Chadha, dan T. Vos, "Global and Regional Needs, Unmet Needs and Access to Hearing Aids," *International Journal of Audiology*, vol. 59, no. 3, hlm. 166–172, Mar 2020, doi: 10.1080/14992027.2020.1721577.

[7] Yana Karisma, N. D. S. Ismail, Shinta Esabella, Erwin Mardinata, dan Rodianto, "PENERAPAN SPEECH TO TEXT PADA APLIKASI KAMUS BAHASA SUMBAWA INDONESIA INGGRIS BERBASIS ANDROID," *JIRE*, vol. 5, no. 2, hlm. 230–241, Des 2022, doi: 10.36595/jire.v5i2.751.

[8] I. Sinha dan O. Caverly, "EyeHear: Smart Glasses for the Hearing Impaired," dalam *HCI International 2020 – Late Breaking Papers: Universal Access and Inclusive Design*, vol. 12426, C. Stephanidis, M. Antona, Q. Gao, dan J. Zhou, Ed., dalam Lecture Notes in Computer Science, vol. 12426. , Cham: Springer International Publishing, 2020, hlm. 358–370. doi: 10.1007/978-3-030-60149-2_28.

[9] A. M. Ridha dan W. Shehieb, "Assistive Technology for Hearing-Impaired and Deaf Students Utilizing Augmented Reality," dalam *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, ON, Canada: IEEE, Sep 2021, hlm. 1–5. doi: 10.1109/CCECE53047.2021.9569193.

[10] B. Li *dkk.*, "A Language Agnostic Multilingual Streaming On-Device ASR System," 2022, *arXiv*. doi: 10.48550/ARXIV.2208.13916.

[11] J. Macoskey, G. P. Strimel, dan A. Rastrow, "Bifocal Neural ASR: Exploiting Keyword Spotting for Inference Optimization," dalam *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada: IEEE, Jun 2021, hlm. 5999–6003. doi: 10.1109/ICASSP39728.2021.9414652.

[12] M. Ali, F. Alam, I. Ahmed, B. AlQattan, A. K. Yetisen, dan H. Butt, "3D printing of Fresnel lenses with wavelength selective tinted materials," *Additive Manufacturing*, vol. 47, hlm. 102281, 2021.

[13] S. Deng *dkk.*, "Carbon Nanotube Array Based Binary Gabor Zone Plate Lenses," *Sci Rep*, vol. 7, no. 1, hlm. 15256, Nov 2017, doi: 10.1038/s41598-017-15472-9.

[14] "The human eye." Diakses: 6 Juli 2025. [Daring]. Tersedia pada: http://labman.phys.utk.edu/phys222core/modules/m8/human_eye.html

[15] Espressif Systems, "32-bit MCU & 2.4 GHz Wi-Fi & Bluetooth/Bluetooth LE." ESP32-WROVER-E datasheet, 2025. [Daring]. Tersedia pada: https://documentation.espressif.com/esp32-wrover-e_esp32-wrover-ie_datasheet_en.pdf

[16] R. K. Kodali dan K. S. Mahesh, "Low cost ambient monitoring using ESP8266," dipresentasikan pada 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), IEEE, 2016, hlm. 779–782.

[17] NanJing Top Power ASIC Corp., "1A Standalone Linear Li-Ion Battery Charger." TP4056 datasheet, n.d. [Daring]. Tersedia pada: https://datasheet.lcsc.com/lcsc/1809261820_TOPPOWER-Nanjing-Extension-Microelectronics-TP4056-42-ESOP8_C16581.pdf

[18] Aerosemi Technology, "High Efficiency 1.2 MHz 2A Step Up Converter." MT3608 datasheet, n.d. [Daring]. Tersedia pada: https://www.olimex.com/Products/Breadboarding/BB-PWR-3608/resources/MT3608.pdf

[19] J. Castaño, S. Martínez-Fernández, X. Franch, dan J. Bogner, "Analyzing the evolution and maintenance of ml models on hugging face," dipresentasikan pada Proceedings of the 21st International Conference on Mining Software Repositories, 2024, hlm. 607–618.

[20] X. Li, S. Takamichi, T. Saeki, W. Chen, S. Shiota, dan S. Watanabe, "Yodas: Youtube-Oriented Dataset for Audio and Speech," dalam *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Taipei, Taiwan: IEEE, Des 2023, hlm. 1–8. doi: 10.1109/ASRU57964.2023.10389689.

[21] A. Conneau *dkk.*, "FLEURS: FEW-Shot Learning Evaluation of Universal Representations of Speech," dalam *2022 IEEE Spoken Language Technology Workshop (SLT)*, Doha, Qatar: IEEE, Jan 2023, hlm. 798–805. doi: 10.1109/SLT54892.2023.10023141.

[22] R. Ardila *dkk.*, "Common Voice: A Massively-Multilingual Speech Corpus," 2019, *arXiv*. doi: 10.48550/ARXIV.1912.06670.

[23] P. Żelasko, D. Povey, J. Trmal, dan S. Khudanpur, "Lhotse: a speech data representation library for the modern deep learning ecosystem," *arXiv preprint arXiv:2110.12561*, 2021.

[24] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, dan I. Sutskever, "Robust speech recognition via large-scale weak supervision," dipresentasikan pada International conference on machine learning, PMLR, 2023, hlm. 28492–28518.

[25] J. Hwang *dkk.*, "TorchAudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for PyTorch," dipresentasikan pada 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE, 2023, hlm. 1–9.

[26] A. Baevski, H. Zhou, A. Mohamed, dan M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," 2020, *arXiv*. doi: 10.48550/ARXIV.2006.11477.

[27] L. A. Mullen, K. Benoit, O. Keyes, D. Selivanov, dan J. Arnold, "Fast, consistent tokenization of natural language text," *Journal of Open Source Software*, vol. 3, no. 23, hlm. 655, 2018.

[28] K. Batsuren *dkk.*, "Evaluating subword tokenization: Alien subword composition and oov generalization challenge," *arXiv preprint arXiv:2404.13292*, 2024.

[29] R. Mushi dan Y.-P. Huang, "Assessment of mel-filter bank features on sound classifications using deep convolutional neural network," dipresentasikan pada 2021 International Conference on System Science and Engineering (ICSSE), IEEE, 2021, hlm. 334–339.

[30] D. Snyder, G. Chen, dan D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 28 Oktober 2015, *arXiv*: arXiv:1510.08484. doi: 10.48550/arXiv.1510.08484.

[31] D. S. Park *dkk.*, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[32] Y. Xu dan R. Goodacre, "On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning," *J. Anal. Test.*, vol. 2, no. 3, hlm. 249–262, Jul 2018, doi: 10.1007/s41664-018-0068-2.

[33] Didi Kurniawan dan Dhani Ariatmanto, "IDENTIFIKASI VARIETAS BIBIT DURIAN MENGGUNAKAN MOBILENETV2 BERDASARKAN GAMBAR DAUN," *JIRE*, vol. 7, no. 2, hlm. 231–240, Nov 2024, doi: 10.36595/jire.v7i2.1236.

[34] F. Boyer, Y. Shinohara, T. Ishii, H. Inaguma, dan S. Watanabe, "A Study of Transducer Based End-to-End ASR with ESPnet: Architecture, Auxiliary Loss and Decoding Strategies," dalam *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Cartagena, Colombia: IEEE, Des 2021, hlm. 16–23. doi: 10.1109/ASRU51503.2021.9688251.

[35] E. Battenberg *dkk.*, "Exploring neural transducers for end-to-end speech recognition," dipresentasikan pada 2017 IEEE automatic speech recognition and understanding workshop (ASRU), IEEE, 2017, hlm. 206–213.

[36] Z. Yao *dkk.*, "Zipformer: A faster and better encoder for automatic speech recognition," *arXiv preprint arXiv:2310.11230*, 2023.

[37] S. Kumar *dkk.*, "XLSR-Transducer: Streaming ASR for Self-Supervised Pretrained Models," 2024, *arXiv*. doi: 10.48550/ARXIV.2407.04439.

[38] P. Swietojanski *dkk.*, "Variable Attention Masking for Configurable Transformer Transducer Speech Recognition," 2022, doi: 10.48550/ARXIV.2211.01438.

[39] A. Ali dan S. Renals, "Word Error Rate Estimation for Speech Recognition: e-WER," dalam *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia: Association for Computational Linguistics, 2018, hlm. 20–24. doi: 10.18653/v1/P18-2004.

[40] Cahya, "whisper-medium-id · Hugging Face." Diakses: 26 Oktober 2025. [Daring]. Tersedia pada: https://huggingface.co/cahya/whisper-medium-id

[41] A. V. Ivanov *dkk.*, "Speed vs. accuracy: Designing an optimal asr system for spontaneous non-native speech in a real-time application," *Proc. of the IWSDS, Saariselk, Finland*, 2016.

[42] "Korea Utara Dukung Palestina, Sebut Pendudukan Israel Ilegal dan Kritik Kebijakan Luar Negeri AS," Kompas.tv. Diakses: 26 Juni 2025. [Daring]. Tersedia pada: https://www.kompas.tv/internasional/506895/korea-utara-dukung-palestina-sebut-pendudukan-israel-ilegal-dan-kritik-kebijakan-luar-negeri-as

[43] "Proklamasi Kemerdekaan Indonesia," *Wikipedia bahasa Indonesia, ensiklopedia bebas*. 4 Mei 2025. Diakses: 26 Juni 2025. [Daring]. Tersedia pada: https://id.wikipedia.org/w/index.php?title=Proklamasi_Kemerdekaan_Indonesia&oldid=27228882