

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Kemajuan teknologi dalam kehidupan ini tidak dapat kita hentikan karena pasti akan mengikuti kemajuan ilmu pengetahuan. Setiap penemuan dibuat dengan tujuan untuk meningkatkan kehidupan manusia. menawarkan metode baru dalam melakukan tindakan manusia dengan sejumlah kemudahan. Masyarakat telah memperoleh manfaat besar dari berbagai kemajuan yang dicapai dalam sepuluh tahun terakhir, khususnya di bidang teknologi. Kemunculan berbagai model *smartphone* baru merupakan salah satu contoh kemajuan teknis tersebut. Ponsel pintar merupakan perangkat dengan kemampuan komputer bergerak dan fitur perangkat lunak yang menunjang berbagai kebutuhan manusia dan memudahkan pembelajaran penggunanya, termasuk pembelajaran huruf hijaiyah.

Bagi umat Islam, huruf hijaiyah merupakan hal yang krusial. sebagai landasan membaca Al-Quran adalah huruf hijaiyah. Pembelajaran huruf hijaiyah biasanya dimulai ketika kita masih muda. Buku Iqra digunakan untuk mengajarkan huruf hijaiyah kepada anak. Syarat utama membaca Al-Quran adalah huruf Hijaiyah. karena kata dan kalimat dalam Alquran terbentuk dari surat ini. Wajar saja jika Anda sudah mengetahui huruf hijaiyah, Anda bisa membaca Alquran. Dan pada saat ini kita bisa belajar huruf hijaiyah dimanapun dengan adanya *smartphone*.

Berdasarkan Liputan 6, Indonesia merupakan salah satu dari lima negara teratas dalam hal penggunaan *gadget*, khususnya ponsel. Sekitar 47 juta masyarakat

Indonesia, atau 47% dari total populasi negara ini, merupakan pengguna aktif ponsel pintar pada tahun 2014. Di Indonesia, anak-anak dan remaja menggunakan perangkat dengan tingkat yang cukup tinggi sekitar 79,5% dari mereka berselancar. Pada tahun 2014, *Unicef* dan Kementerian Penerangan melaksanakan hal ini. Sesuai uraian permasalahan, maka dibuatlah aplikasi panduan pelafalan huruf hijaiyah dengan fitur *Correcting* dengan memanfaatkan kecintaan anak-anak terhadap *smartphone* guna membantu mereka menjadi lebih mahir dalam menggunakan *smartphone*.

Tujuan aplikasi ini adalah untuk mengenalkan huruf hijaiyah kepada anak sesuai dengan keinginannya. Ini akan mencakup suara dan visual huruf. Diharapkan pengguna dapat belajar sendiri dengan program pembelajaran huruf hijaiyah yang berbasis Android. Aplikasi menyediakan fitur belajar huruf hijaiyah yang akan langsung mengoreksi jika pengucapan huruf salah dengan fitur *audio* untuk membantu dalam belajar huruf hijaiyah. dengan memanfaatkan dan meningkatkan kemampuan pengenalan suara ponsel. Program yang dikembangkan memungkinkan pengguna untuk mengucapkan sebuah huruf, setelah itu akan menafsirkan kata yang diucapkan dan mengubahnya menjadi sebuah *string*. Setelah itu, surat-surat yang diucapkan pengguna akan dibandingkan dengan *database* yang berisi surat-surat yang tepat.

Dengan latar belakang masalah ini, maka penulis mengangkat penelitian berjudul **“Aplikasi Panduan Pengucapan Huruf Hijaiyah dengan Auto Koreksi”**. Diharapkan, implementasi Aplikasi panduan pengucapan ini pengguna

dapat belajar dalam pengucapan huruf hijaiyah yang benar karena memiliki fitur koreksi sehingga dapat mengetahui jika huruf yang diucapkan benar atau salah.

B. Rumusan Masalah

Melihat konteks permasalahan tersebut, maka rumusan masalah penelitiannya adalah bagaimana membuat aplikasi panduan pengucapan huruf hijaiyah sehingga dapat membantu pengguna mengetahui pengucapan huruf benar atau salah?

C. Tujuan Penelitian

Seperti yang telah disampaikan sebelumnya, tujuan penelitian ini adalah untuk membantu pengucapan huruf hijaiyah secara akurat guna memudahkan pembacaan Al-Qur'an dengan benar.

D. Batasan Masalah

Berikut ini adalah keterbatasan penelitian:

1. Aplikasi ini fokus mengajarkan pengguna cara melafalkan huruf hijaiyah agar bisa membaca Al-Qur'an dengan akurat.
2. Kemampuan pengenalan ucapan mempunyai keterbatasan, termasuk potensi kemampuan ucapan-ke-teks yang kurang optimal dan akurasi pengenalan suara, terutama bila terdapat perbedaan intonasi atau pengucapan tertentu.

E. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Bidang Akademik

Program ini dapat digunakan oleh para ulama untuk melakukan kajian pemanfaatan teknologi digital dalam mempelajari huruf hijaiyah.

2. Bidang Masyarakat

Masyarakat dapat menggunakan aplikasi ini untuk membantu mereka belajar pengucapan huruf hijaiyah dengan benar.

3. Bagi Peneliti

Penelitian ini dapat menjadi sarana untuk pembelajaran, dan peneliti dapat mengimplementasikan ilmu yang didapat selama proses perkuliahan.

BAB II

TINJAUAN PUSTAKA

A. Kajian Hasil Penelitian Terdahulu

1. Pada penelitian (Mauliddiyah, 2021) yaitu aplikasi *tahfidz muroja'ah* hafalan Al-Qur'an merupakan sebuah aplikasi menggunakan teknologi *Speech Recognition in Artificial Intelligence (AI)* dengan mengimplementasikan *Google Speech API* dan *Algoritma Rabin Karp* untuk mengubah ucapan pengguna menjadi sebuah *string* yang kemudian dibandingkan dengan *string* ayat yang benar.
2. Penelitian yang dilakukan oleh (Hudayana et al., 2024) tentang pengembangan aplikasi penyimak Al-Qur'an menggunakan teknologi *speech recognition* menunjukkan titik *inovatif* dengan menggabungkan teknologi modern dan praktik keagamaan, dan penelitian. Hasil pengujian performa aplikasi menunjukkan bahwa aplikasi dapat mendeteksi dan mentranskripsikan ayat-ayat pendek Al-Qur'an dengan baik.
3. Dan penelitian yang dilakukan oleh (Wisdhani et al., 2021) pada aplikasi pembelajaran asmaul husna menggunakan *speech recognition*, dapat membantu dalam pengenalan nama-nama asmaul husna dengan mengubah pengucapan kebentuk tulisan sehingga lebih mudah di pahami.

B. Kajian Teori

1. Huruf hijaiyah

Huruf-huruf yang menyusun kata-kata dalam Al-Qur'an dikenal dengan huruf hijaiyah. Jika ada huruf dalam abjad indonesia yang digunakan untuk membentuk kata dan kemudian kalimat, maka huruf hijaiyah mempunyai fungsi yang sama dengan huruf abjad indonesia.

Orang tua biasanya mengajarkan huruf hijaiyah kepada anaknya dengan dua cara. Anak mula-mula akan mendengarkan setelah huruf hijaiyah yang ada di buku iqra dibacakan. Jika orang tua anak paham atau sadar membaca huruf hijiyah, hal ini bisa dilakukan di rumah bersama mereka. Pilihan kedua adalah membayar guru untuk mengajar anak-anak tentang huruf hijaiyah atau menyekolahkan mereka ke TK atau TPA.

Inti Asma Allah Ta'ala terdapat pada huruf hijaiyah. Rahasiannya hanya diketahui oleh Allah SWT. Sekalipun seorang ulama sufi membocorkan rahasia surat-surat itu, itu hanyalah sebagian kecil dari lautan luas misteri yang terkandung di dalamnya. 28 Huruf Arab yang digunakan untuk menulis Al-Qur'an umumnya dikenal dengan huruf Hijaiyah. Kata Arab harf yang berarti huruf, merupakan asal kata huruf. Huruf hijaiyah merupakan nama lain dari huruf arab. Kata kerja hajjaa yang berarti menghitung, membaca, dan mengeja huruf merupakan akar kata hijaiyah (Abim Ridho Pangestu, dkk., 2021).

Dalam bahasa Arab, kata dan kalimat dibentuk dengan huruf hijaiyah. Kata Arab الهجائية yang berarti ejaan atau ejaan, merupakan asal muasal huruf hijaiyah.

Selain digunakan dalam Al-Quran, huruf hijaiyah ini juga sering digunakan dalam tulisan arab.

Batasan terminologi huruf hijaiyah ini berbeda dengan huruf lainnya. Berbeda dengan abjad yang saat ini ditulis dari kiri ke kanan, huruf ini ditulis dari kanan ke kiri. Ke-30 huruf penyusun hijaiyah ini mempunyai gaya penulisan dan bacaan yang beragam berdasarkan tajwid dan harakat.

Rangkaian huruf Hijaiyah seluruhnya berjumlah 30 huruf. Setiap surat dapat dibaca dengan cara yang berbeda. menggunakan *cloud Speech API* untuk mengembangkan aplikasi belajar bahasa Inggris. 30 huruf hijaiyah beserta petunjuk bacaannya disajikan pada tabel di bawah ini.

Tabel 2. 1 *Huruf Hijaiyah*

No	Huruf hijaiyah	Cara Membaca	Transliterasi Latin
1.	ا	Alif	Tidak ada lambang
2.	ب	Ba'	B
3.	ت	Ta'	T
4.	ث	Tsa'	TS
5.	ج	Jim	J
6.	ح	Ha'	H
7.	خ	Kha'	KH
8.	د	Dal	D
9.	ذ	Dzal	Dz
10.	ر	Ra'	R
11.	ز	Za'	Z
12.	س	Sin	S
13.	ش	Syin	SY
14.	ص	Shad	SH
15.	ض	Dhad	DH

Lanjutan Tabel 2. 1Huruf Hijaiyah

16.	ط	Tha'	TH
17.	ظ	Zha'	ZH
18.	ع	'Ain	Huruf vokal dengan tanda petik ('a, 'i, 'u)
19.	غ	Ghain	GH
20.	ف	Fa'	F
21.	ق	Qaf	Q
22.	ك	Kaf	K
23.	ل	Lam	L
24.	م	Mim	M
25.	ن	Nun	N
26.	و	Waw	W
27.	ه	Ha'	H
28.	ء	Hamzah	Huruf vokal yang ditambah dengan tanda petik (a', i', u')
29.	ي	Ya'	Y
30.	لا	lam alif	

Pentingnya bagi umat Islam untuk mempelajari huruf hijaiyah karena merupakan huruf dasar dalam membaca Al-Qur'an. Sangat disayangkan jika umat Islam tidak bisa membaca Al-Quran. Oleh karena itu, suka atau tidak, mempelajari huruf hijaiyah sangatlah penting untuk bisa membaca Al-Quran. Huruf hijaiyah ini biasanya diajarkan kepada kita ketika kita masih kecil.

2. Fitur Koreksi

Fitur koreksi dalam aplikasi ini berfungsi untuk membantu pengguna mengidentifikasi kesalahan dalam pengucapan huruf hijaiyah secara otomatis. Ketika pengguna mengucapkan sebuah huruf, aplikasi menggunakan teknologi pengenalan suara untuk mengubah suara tersebut menjadi teks (*string*) yang

kemudian dibandingkan dengan database huruf hijaiyah yang benar. Jika aplikasi mendeteksi bahwa pengucapan pengguna tidak sesuai dengan bentuk huruf yang diharapkan, fitur koreksi akan memberi tahu pengguna bahwa pengucapannya salah dan memberikan petunjuk atau contoh pelafalan yang benar.

Fitur ini memberikan umpan balik langsung, yang sangat membantu pengguna, terutama anak-anak, dalam belajar dengan lebih mandiri. Dengan adanya koreksi otomatis ini, pengguna dapat memahami apakah pengucapannya sudah benar atau perlu diperbaiki. Fitur koreksi ini didukung oleh algoritma yang dirancang untuk mengenali perbedaan fonetik antara huruf hijaiyah, sehingga dapat membedakan huruf yang mirip, seperti "ba", "ta", dan "tsa".

Tujuan utama fitur koreksi adalah meningkatkan akurasi pembelajaran dan memotivasi pengguna untuk memperbaiki pelafalan mereka hingga mencapai pengucapan yang benar, sehingga proses belajar menjadi lebih efektif dan interaktif. Dalam aplikasi ini fitur koreksi ini akan mengoreksi pengucapan huruf hijaiyah apakah sudah benar atau salah, jika penyebutan huruf hijaiyah benar maka dapat melanjutkan ke huruf selanjutnya dan jika pengucapan huruf hijaiyah salah maka kita akan di suruh mengulang pengucapan.

3. *Google Speech API*

Google Speech API, biasa disebut dengan *Google Voice Search*, dirilis untuk sejumlah model ponsel cerdas di Amerika Serikat pada tahun 2008. Untuk mengenali ucapan, mengubahnya menjadi teks, lalu memasukkannya ke dalam situs penelusuran *Google* sehingga menghasilkan hasil penelusuran dapat ditampilkan melalui input suara, *Google* mengembangkan kerangka *Google Speech API*. *Server*

Google menggunakan pendekatan *Hidden Markov Model* (HMM) untuk pengenalan suara. Dengan kata lain, server *Google* akan mengenali masukan suara yang didapat ponsel pintar Android dan mengubahnya menjadi teks menggunakan algoritma HMM. Setelah konversi suara menjadi teks, hasilnya dimasukkan ke halaman pencarian *Google*, dan *server Google* mengirimkan hasil pencarian ke perangkat Android.

4. *Speech recognition*

Google Speech, juga dikenal sebagai *Speech Recognition*, adalah API yang menggunakan digitalisasi kata dan membandingkan sinyal digital dengan pola yang disimpan dalam *database Google* untuk mendeteksi suara. Kemampuan perangkat lunak, perangkat, atau teknologi untuk mengenali ucapan untuk mengidentifikasi kata-kata yang diucapkan oleh manusia dan mengubahnya menjadi teks yang dapat dibaca. Pada prakteknya teknologi ini juga berevolusi menjadi *voice recognition*, yang dapat membedakan suara satu orang dengan orang lain dan menjadi salah satu ciri *biometrik* yang diandalkan.

Ketika sistem menerima data masukan berupa ucapan atau suara manusia, maka sistem akan mengenali kata atau frasa yang diucapkan dan menghasilkan keluaran teks yang sesuai dengan data masukan tersebut. Dengan mengubah gelombang suara menjadi kumpulan angka, memodifikasinya dengan kode tertentu, dan membandingkannya dengan pola yang disimpan dalam perangkat, masukan berupa kata-kata diubah menjadi sinyal digital. Pola karakteristik yang berbeda akan dihasilkan untuk setiap ujaran yang berbeda.

Dalam praktiknya, teknologi pengenalan suara akan menerjemahkan suara yang ditangkap oleh peralatan yang terhubung, seperti mikrofon, menjadi teks tertulis yang dapat dipahami oleh komputer dan manusia.

Secara umum, proses atau cara kerjanya akan terjadi dalam empat poin besar.

- a) Pertama adalah analisis *audio*, guna memetakan apakah suara yang direkam benar-benar berupa kata-kata yang jelas atau sekedar bunyi *noise* yang tidak dapat diterjemahkan ke dalam kata-kata riil.
- b) Kedua, memecah *audio* menjadi beberapa bagian sehingga lebih mudah ‘dicerna’ oleh sistem yang digunakan.
- c) Ketiga, mengkonversi *audio* ke dalam format yang dapat dibaca komputer sehingga proses penerjemahan file audio dapat berjalan dengan lancar dan tepat.
- d) Keempat, penggunaan algoritma untuk mencocokkan *audio* dengan representasi teks yang dimiliki pada *database*, dan yang paling sesuai sehingga dapat ditampilkan dan dipahami oleh sistem dan manusia.

Praktis teknologi *speech recognition* harus memiliki kemampuan untuk mengenali berbagai jenis pola bicara, gaya bicara, bahasa yang ada, dialek, aksen, serta berbagai kata umum yang ada dalam bahasa tertentu. Semakin besar perbendaharaan *database* sebuah sistem maka dapat dipastikan kemampuan *speech recognition*-nya akan semakin baik pula.

Nah, untuk memperoleh kemampuan tersebut, teknologi ini akan menggunakan dua jenis model utama. Pertama dikenal dengan metode *acoustic model*, yakni model yang mewakili hubungan antara unit linguistik ucapan dan

sinyal audio, dan yang kedua disebut dengan *language model*, yakni pencocokan suara dengan urutan kata untuk membedakan antara kata-kata yang terdengar mirip, sehingga kata-kata yang disajikan sesuai dengan konteks pembicaraan.

5. Android



Gambar 2. 1 Logo Android

Menurut Sutarman,dkk, (2012) menjelaskan bahwa Android adalah sistem operasi yang bersifat *open-source* dan dapat dikembangkan oleh siapa saja. Dengan kata lain, android memberi kebebasan bagi pengembang untuk menciptakan aplikasi yang beragam sesuai kebutuhan pengguna. Android juga mendukung perangkat dengan sistem sentuhan (*touchscreen*) yang memudahkan interaksi pengguna. Sistem operasi seluler android dibangun di atas *kernel Linux* yang dimodifikasi dan alat sumber terbuka lainnya. Perangkat seluler, khususnya yang memiliki layar sentuh seperti tablet dan ponsel cerdas, menjadi fokus penciptaan android. Dengan dukungan finansial *Google*, *Open Handset Alliance* menciptakan android, sebuah sistem operasi yang pertama kali tersedia pada bulan September 2008.

Selain itu, Android adalah perangkat lunak sumber terbuka, artinya *Google* mengizinkan pengguna membuat sistem operasinya. *Google Play Store* adalah toko aplikasi lain untuk Android. Bagi Anda yang memiliki *smartphone* Android, tentunya *Google Play Store* menyediakan aplikasi dan game gratis.

Android menawarkan sejumlah lingkungan pengembangan yang berbeda. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dan aplikasi pihak ketiga. Akses ke perangkat keras, termasuk data sistem atau ponsel cerdas, dimungkinkan oleh Antarmuka Pemrograman Aplikasi (API) yang disertakan. Pengguna juga dapat menghapus aplikasi utama dan menukarnya dengan aplikasi pihak ketiga. Adapun kelebihan android yaitu:

- a) Karena merupakan sistem operasi *open source*, pengembang dapat dengan mudah membuatnya.
- b) Mudah beradaptasi dan berubah.
- c) Berbagai macam spesifikasi perangkat keras yang tersedia dapat mengoperasikan sistem operasi Android.
- d) Bantuan dengan berbagai macam aplikasi.
- e) Jika terjadi kerusakan sistem, perbaikan dapat dilakukan karena sistem operasi Android mudah dipahami.
- f) Berbagai macam perangkat listrik dapat menjalankan sistem operasi Android.
- g) Android adalah sistem operasi yang cepat dan responsif.

Adapun kekurangan android yaitu:

- a) Sistem operasi Android menggunakan sejumlah besar RAM karena pengoperasian sistemnya yang ekstensif.
- b) Sistem operasi Android menjadi kurang responsif bila dipasangkan dengan perangkat keras di bawah standar.
- c) Sistem operasi khusus sering kali tidak dapat diandalkan dan di bawah standar.

- d) Membutuhkan koneksi internet aktif dari pengguna. Koneksi internet GPRS setidaknya diperlukan.
- e) Dibandingkan dengan sistem operasi lain, baterai ponsel cerdas akan sangat boros.

6. Android Studio



Gambar 2. 2 Logo *Android Studio*

Android Studio adalah *Integrated Development Environment* (IDE) yang digunakan untuk membuat aplikasi Android. Perangkat lunak ini, yang tersedia secara gratis di bawah lisensi *Apache 2.0*, disediakan oleh *Google* pada 16 Mei 2013. *Eclipse*, alat pemrograman Android sebelumnya, telah digantikan oleh Android Studio. Editor kode, *debugger*, kompiler, dan alat terintegrasi lainnya yang diperlukan untuk menghasilkan perangkat lunak semuanya termasuk dalam Insinyur perangkat lunak dapat menggunakan aplikasi yang disebut IDE (*Integrated Development Environment*). Nadia Firdaus (2017:13) menegaskan bahwa Android Studio “adalah *Integrated Development Environment* (IDE) atau dengan kata lain lingkungan pengembangan terintegrasi resmi yang dirancang khusus untuk pengembangan sistem operasi *Google Android*.”

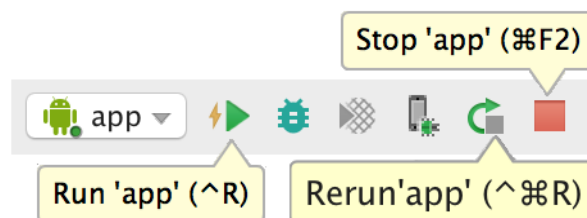
Mirip dengan *Eclipse*, Android Studio dibuat menggunakan *IntelliJ IDEA* dan *plugin ADT* (*Android Development Tools*). Fitur Android Studio meliputi:

- a) Proyek berbasis *Gradle Build*.

- b) Perbaikan *bug* cepat dan pemfaktoran ulang.
- c) Alat baru bernama "*Lint*" menjanjikan pemeriksaan kompatibilitas, kinerja, dan kegunaan aplikasi dengan cepat.
- d) Menyediakan fitur keamanan seperti *Proguard* dan penandatanganan aplikasi
- e) Membuat penggunaan GUI aplikasi Android menjadi lebih sederhana
- f) Setiap aplikasi yang dikembangkan didukung oleh *Google Cloud Platform*.

Karena Android Studio menawarkan sejumlah alat yang memfasilitasi pemrograman, khususnya bagi pemrogram berpengalaman yang ingin mempelajari lebih lanjut tentang Android, maka Android Studio dipilih. Meski menggunakan RAM dalam jumlah besar di PC, Android Studio memiliki sejumlah keunggulan lain, seperti:

1) *Instant RUN*



Gambar 2. 3 *Instant Run*

Saat melakukan perubahan kode, fitur *Instant Run* dapat menjamin bahwa program berjalan dengan cepat tanpa mengharuskan aplikasi dikompilasi ulang atau APK dibuat ulang, sehingga menghasilkan prosedur yang lebih cepat secara keseluruhan.

2) Dioptimalkan untuk semua perangkat Android

Membuat aplikasi untuk berbagai perangkat Android, termasuk *tablet*, *Wear*, TV, dan *Auto*, menjadi mudah dengan Android Studio. Proyek dapat dipecah menjadi beberapa bagian fungsional menggunakan fungsionalitas terstruktur ini, yang kemudian dapat Anda kembangkan, uji, dan debug sesuai kebutuhan.

3) Didesain untuk Tim

Beberapa kontrol versi terkenal, termasuk Git dan *Subversion*, terintegrasi dengan Android Studio. Untuk mempermudah kolaborasi, kita bahkan bisa menggunakan layanan *Github* langsung dari Android Studio. Tim dan pembaca dapat terus bekerja sama secara efektif pada proyek masing-masing yang tersedia dengan cara ini.

7. Java



Gambar 2. 5 Logo Java

Menurut Y. Daniel Liang,dkk, (2017) Program yang besar dan canggih dapat dikembangkan dengan lebih mudah dengan Java, bahasa pemrograman berorientasi objek yang mendukung gagasan modularitas dan enkapsulasi.. Java juga mendukung pemrograman *multithreaded*, yang memungkinkan eksekusi

beberapa proses secara bersamaan, yang sangat efisien untuk aplikasi jaringan dan *server*.

Java adalah salah satu bahasa pemrograman yang paling banyak digunakan untuk mengembangkan aplikasi *web*. Java telah populer di kalangan pengembang selama lebih dari 20 tahun, dan saat ini terdapat jutaan program Java yang digunakan. Java adalah bahasa *multiplatform*, berorientasi objek, berpusat pada jaringan yang dapat digunakan sebagai platform. Aplikasi data besar, perangkat lunak perusahaan, teknologi sisi *server*, dan aplikasi seluler semuanya dapat dibuat dengan Java, bahasa pemrograman yang cepat, aman, dan dapat diandalkan.

Tujuan Java, bahasa pemrograman *generik/non-spesifik* (tujuan umum), adalah untuk meminimalkan ketergantungan implementasi. Slogan "Tulis Sekali, Jalankan Di Mana Saja" juga dikaitkan dengan Java karena kemampuannya membuat aplikasi Java berjalan di berbagai sistem operasi. Java saat ini merupakan bahasa pemrograman yang paling banyak digunakan dan digunakan untuk membuat berbagai macam perangkat lunak.

Bahasa pemrograman Java diciptakan selama 18 bulan Proyek Hijau, yang berlangsung dari awal tahun 1991 hingga musim panas 1992. Versi *Oak* belum digunakan dalam proyek ini. *James Gosling*, *Mike Sheridan*, dan *Patrick Naughton* mengawasi proyek ini selain sembilan programmer *Sun Microsystems* lainnya. Penciptaan *maskot Duke* oleh *Joe Palrang* adalah salah satu hasil dari upaya ini.

Kode sumber untuk Java versi 1.0a2 pertama kali dirilis pada bulan Maret 1995. Kode sumber pertama kali diterbitkan di surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995, mengikuti tonggak sejarahnya. Versi rilis Java tidak

menggunakan nama *Oak* karena program lain telah mendaftarkan merek dagangnya; sebaliknya, "Java" dipilih sebagai nama baru. Biji kopi seduh favorit *Gosling* digunakan untuk menggiling kopi murni yang diberi label ini. Jawa diduga sebagai asal muasal kopi ini. Jadi, kata Java sendiri merupakan asal muasal nama bahasa pemrograman Java (Java adalah istilah bahasa Inggris untuk Jawa).

8. XML (*Extensible Markup Language*)

Rahman (2011) *World Wide Web Consortium* (W3C) telah menetapkan *Standard Generalized Markup Language* (SGML), yang mencakup XML, singkatan dari bahasa markup yang dapat diperluas. Karena memungkinkan pengguna merancang komponen markup mereka sendiri, XML diklasifikasikan sebagai bahasa yang dapat diperluas. Sebaliknya, HTML memerlukan pendefinisian elemen tertentu agar dapat ditampilkan di *browser web*.

Struktur dokumen XML, dokumen serbaguna, merupakan cerminan dari struktur datanya. Struktur dokumen XML, yang terdiri dari elemen seperti *root*, *parent*, dan *child*, umumnya dapat dibandingkan dengan pohon *hierarki*.

Struktur dan konten dokumen ditentukan oleh XML menggunakan sekumpulan elemen, yang masing-masing mencakup bagian teks yang berbeda.

Elemen XML meliputi:

1. *Start Tag* <
2. Isi atau elemen (Dapat berupa teks atau angka)
3. *End Tag* />

Penutupan suatu elemen ditunjukkan oleh Tag Akhir, sedangkan inisialisasinya ditunjukkan oleh Tag Awal. Nama elemen XML harus mengikuti

pedoman peka huruf besar-kecil dan tidak boleh mengandung spasi. Elemen turunan adalah elemen lainnya, dan setiap elemen XML dapat memiliki satu atau lebih elemen turunan.

Elemen XML mempunyai kapasitas untuk memiliki atribut, yaitu deskripsi atau penjelasan. Suatu elemen mungkin memiliki banyak atribut. Penulisan atribut meliputi penulisan nama atribut, diikuti dengan tanda sama dengan (=) dan nilai atribut.

<code><?xml version="1.0"?></code>	--> Deklarasi
<code><belajar pelajaran="Ilmu Pengetahuan Alam"></belajar></code>	--> Elemen
<code><murid nama="Imam" kelas="12 SMA" absen="15"></code>	--> Atribut

Gambar 2. 6 Struktur Dokumen XML

Dua tingkat akurasi yang harus dipenuhi oleh dokumen XML agar dapat dibaca, antara lain:

1. Diformat dengan Benar: Dokumen XML harus mematuhi sintaksis yang benar, yang mencakup elemen pembuat yang menyertakan tag awal (< >) diikuti dengan tag akhir (< />).
2. Valid: konten elemen dalam dokumen XML harus mematuhi tipe data atau standar semantik yang disebutkan; misalnya, elemen tanpa nilai dianggap tidak valid.

Parsing, nama lain untuk membaca dokumen XML, dapat dilakukan dengan berbagai program perangkat lunak atau perpustakaan. Beberapa contohnya melibatkan MSXML yang dikembangkan oleh *Microsoft* untuk pengembangan di

sistem operasi *Windows* dan *Xerces* yang digunakan untuk pengembangan aplikasi Java.

9. *Flowchart*

Diagram alur adalah diagram alur yang memberikan penjelasan rinci tentang keputusan, langkah, dan urutan yang terlibat dalam desain proses. Simbol tertentu digunakan untuk mewakili langkah-langkah ini, dan garis atau panah digunakan untuk menghubungkannya. Berbagai proses dapat dengan mudah dipahami dengan menggunakan diagram alur untuk memvisualisasikannya. Bahkan orang awam pun dapat memahami proses kompleks dengan lebih mudah dan sederhana berkat diagram ini. Inilah alasan mengapa *flowchart* sering kali digunakan di berbagai bidang, baik manajemen proyek, teknologi, hingga bisnis.

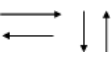
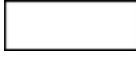
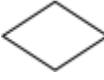
Dalam diagram alir, setiap langkah direpresentasikan oleh simbol dengan makna tertentu untuk menguraikan tugas atau keputusan. Contoh simbol paling umum adalah anak panah yang menunjukkan aliran langkah, atau simbol kotak yang mewakili tugas spesifik yang harus dilakukan dalam proses. Diagram ini biasanya digunakan untuk menggambarkan prosedur proyek, hasil logis, dan pendelegasian wewenang dalam suatu organisasi.

Dengan menggunakan berbagai simbol umum, *flowchart* atau diagram alir ini berupaya memberikan rangkuman yang ringkas, jelas, rapi, dan tentunya komprehensif mengenai langkah-langkah penyelesaian suatu permasalahan. Dengan demikian, diagram alir membuat permasalahan yang sebelumnya sulit menjadi lebih mudah untuk dipahami dan diselesaikan dengan penjelasan yang

lugas. Penggunaan diagram alur adalah cara yang sangat tepat dan tentu saja ringkas untuk mengkomunikasikan alur kerja kepada khalayak umum.

Simbol yang digunakan dalam diagram alur meliputi:

Tabel 2. 2 Simbol-simbol *flowchart*

No	GAMBAR	NAMA	KETERANGAN
1		<i>Arus / Flow</i>	Hubungan antar langkah atau operasi.
2		<i>Connector</i>	Pada lembar atau halaman yang sama terdapat simbol keluar atau masuk untuk suatu proses atau operasi.
3		<i>Off-line Connector</i>	Simbol untuk masuk atau keluar dari suatu proses atau prosedur pada halaman atau lembar yang berbeda.
4		<i>Process</i>	Simbol yang menunjukkan pemrosesan komputer.
5		<i>Decision</i>	Simbol yang menunjukkan suatu kondisi yang dapat mengakibatkan beberapa respon atau tindakan.
6		<i>Predefined Process</i>	Simbol untuk menyiapkan ruang penyimpanan yang akan digunakan untuk pemrosesan.
7		<i>Terminal</i>	Simbol yang menandakan awal atau akhir suatu program.
8		<i>Manual Input</i>	Ini adalah simbol untuk memasukkan kata secara manual ke <i>keyboard online</i> .
9		<i>Document</i>	Tanda bahwa masukan berasal dari dokumen kertas atau keluarannya dicetak di atas kertas.
10		<i>Input-Output</i>	Simbol yang, terlepas dari jenis peralatannya, menggambarkan proses <i>input</i> dan <i>output</i> .
11		<i>Disk and On-Line Storage</i>	Simbol yang menunjukkan apakah input disimpan ke <i>disk</i> atau berasal dari <i>disk</i> .

10. Blackbox testing

Pengujian *BlackBox* adalah teknik pengujian perangkat lunak di mana

penguji mengabaikan organisasi internal dan kode program dan hanya berkonsentrasi pada fungsionalitas aplikasi. Pengujian *black box* digunakan untuk memastikan program berjalan sebagaimana mestinya berdasarkan spesifikasi dan persyaratan yang telah ditetapkan, serta untuk memastikan bahwa sistem berkorelasi dengan masukan pengguna dan menghasilkan hasil yang akurat.

Karakteristik *BlackBox Testing*:

- 1) Fokus pada Fungsi Sistem: Penguji hanya memeriksa apakah aplikasi memberikan hasil yang sesuai dengan *input* yang diberikan, tanpa memperhatikan bagaimana aplikasi tersebut bekerja di dalam.
- 2) Pengujian Berdasarkan Spesifikasi: Penguji menggunakan spesifikasi fungsional atau persyaratan aplikasi untuk menentukan skenario pengujian.
- 3) Tidak Membutuhkan Pengetahuan Kode: Penguji tidak perlu memahami kode sumber aplikasi, karena mereka tidak melihat bagaimana kode tersebut dieksekusi. Mereka hanya berinteraksi dengan antarmuka pengguna dan memperhatikan hasilnya.
- 4) Dapat Dilakukan pada Tahap Pengujian Akhir: *Blackbox* testing sering diterapkan setelah aplikasi selesai dikembangkan atau pada tahap akhir pengembangan untuk memastikan bahwa aplikasi memenuhi kebutuhan pengguna.

Jenis Pengujian *BlackBox*:

- 1) Pengujian Fungsi: Memeriksa apakah fungsionalitas aplikasi bekerja dengan benar sesuai dengan kebutuhan pengguna.
- 2) Pengujian Kinerja: Memeriksa apakah aplikasi dapat menangani beban dan

performa yang dibutuhkan dalam skenario penggunaan.

- 3) Pengujian Keamanan: Mengidentifikasi potensi celah atau kerentanannya yang bisa disalahgunakan oleh pihak luar.
- 4) Pengujian Kompatibilitas: Jadikan aplikasi tersebut berfungsi dengan baik di berbagai perangkat keras, sistem operasi, dan *browser*.

Berikut ini adalah contoh pengujian *blackbox* pada Sistem Jaringan Dokumentasi Dan Informasi (Anisya,dkk,2023), yaitu sebagai berikut:

Tabel 2.3 Contoh Pengujian *BlackBox*

ID	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Detail
S1	Masukkan informasi secara akurat. Nama Jenis Dokumen: Peningat Ijazah Jangka waktu: tiga bulan	Data berhasil disimpan oleh sistem.	Data berhasil disimpan oleh sistem.	Sesuai
S2	Masukkan data tipe nomor pada kolom nama tipe dokumen. Nama Jenis Dokumen: 1	Sistem menolak untuk menyimpan informasi.	Sistem telah berhasil menyimpan data	Tidak Sesuai
S3	Nama tipe dokumen kosong. Nama Jenis Dokumen:	Sistem menolak untuk menyimpan informasi.	Sistem telah berhasil menyimpan data	Tidak Sesuai
S4	Tidak ada durasi pengingat. Durasi Peningat:	Sistem menolak untuk menyimpan informasi.	Sistem telah berhasil menyimpan data	Tidak Sesuai

11. *Whitebox Testing*

Pengujian *Whitebox* adalah semacam pengujian perangkat lunak di mana penguji memiliki pengetahuan luas tentang cara kerja internal atau kode sumber dari program yang diuji. Kadang-kadang juga disebut sebagai Pengujian Kotak Bening, Pengujian Kotak Terbuka, atau Pengujian Kotak Kaca. Pengujian ini

dilakukan dengan fokus pada bagaimana fungsi sistem bekerja secara *internal*, bukan hanya pada *output* yang dihasilkan.

Karakteristik *Whitebox Testing*

- 1) Akses ke Kode Sumber: Penguji harus memiliki akses ke kode program dan memahami bagaimana logika implementasi berjalan.
- 2) Berbasis Logika dan Struktur: Pengujian difokuskan pada alur logika, kondisi, dan struktur kode program.
- 3) Menguji Jalur *Internal*: Termasuk menguji jalur alur kerja, kondisi, *loop*, dan penanganan *error*.

Teknik *Whitebox Testing*

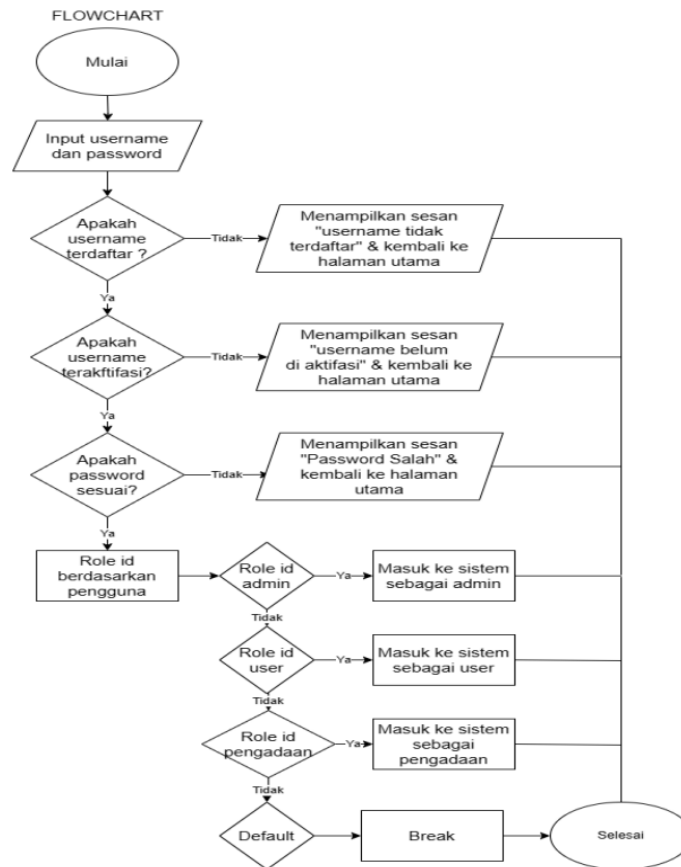
Beberapa teknik umum yang digunakan dalam pengujian *Whitebox* meliputi:

- 1) *Testing Jalur (Path Testing)*: Menguji setiap jalur logis dalam kode untuk memastikan semua cabang (*if, else*) telah dilalui.
- 2) *Coverage Testing*:
 - a) *Statement Coverage*: Memastikan setiap pernyataan dalam kode dijalankan minimal sekali.
 - b) *Branch Coverage*: Memastikan setiap cabang logika (*true/false*) telah diuji.
- 3) *Loop Testing*: Memvalidasi struktur *loop* seperti *for, while*, atau *do-while* untuk berbagai skenario (tanpa iterasi, satu iterasi, dan banyak iterasi).
- 4) *Data Flow Testing*: Mengevaluasi bagaimana data bergerak melalui program, memastikan variabel diinisialisasi sebelum digunakan, dan memeriksa pola pemakaian data.

Berikut ini adalah contoh pengujian *blackbox* pada Sistem Jaringan

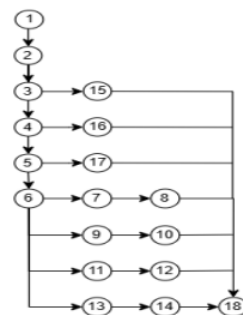
Dokumentasi Dan Informasi (Anisya,dkk,2023), yaitu sebagai berikut:

a. *Flowchart Login*



Gambar 2. 7 Contoh *Flowchart Login*

b. *Flowgraph Login*



Gambar 2. 8 Contoh *Flowgraph Login*

Flowgraph Gambar 3 menunjukkan kepada kita bahwa ada 18 *node* dan 23 *edge*. Setelah perolehan nilai *node* dan *edge*, dilakukan perhitungan selanjutnya

$$V(G) = E - N + 2$$

$$V(G) = 23 - 18 + 2$$

$$V(G) = 7$$

Perhitungan ini menghasilkan tujuh jalur *independen*, yaitu sebagai berikut:

- 1) 1-3-5-6-7-8-18 (Skenario *login* peran admin yang berhasil)
- 2) (Skenario *login* berhasil dengan peran pengguna) 1-2-3-4-5-6-9-10-18
- 3) 11-12-18 1-2-3-4-5-6 (Berhasil masuk dengan peran pengadaan)
- 4) 1-2-3-15-18 (kasus nama pengguna yang tidak terdaftar)
- 5) Skenario nama pengguna belum dipicu: 1-2-3-4-16-18
- 6) Skenario kata sandi salah: 1-2-3-4-5-17-18
- 7) 1-2-3-4-5-6-13-14-18 (Apa yang akan terjadi jika ID peran tidak ditentukan)

Setelah perolehan jalur *independen*, perhitungan kompleksitas *siklomatik* menggunakan tabel 1 akan digunakan untuk memastikan tingkat risiko jalur *independen*. Karena memiliki beberapa jalur berbeda, Fungsi *login file otentikasi* adalah pendekatan yang stabil dan terorganisir dengan risiko minimal.

12. UML (*Unified Modelling Language*)

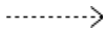
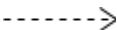
Situs ilmukomputer.com mengklaim industri perangkat lunak telah menjadikan UML (*Unified Modeling Language*) sebagai bahasa standar untuk mengembangkan, mendokumentasikan, dan memvisualisasikan sistem perangkat lunak. UML dapat digunakan untuk membangun model aplikasi perangkat lunak

apa pun yang ditulis dalam bahasa pemrograman apa pun dan dapat dijalankan pada perangkat keras, sistem operasi, atau jaringan apa pun.

Pada masa kini, kebanyakan perancang sistem informasi menggunakan diagram UML untuk menggambarkan informasi. Hal ini bertujuan untuk memvalidasi desain arsitektur perangkat lunak atau program yang dikembangkan, menyelidiki kemungkinan desain, dan meningkatkan komunikasi dalam tim proyek. Saat ini, para ahli menganggapnya lebih berguna untuk analisis, desain, atau pemodelan sistem karena UML menawarkan serangkaian pedoman dan representasi visual yang lebih eksplisit. Diagram struktur, diagram perilaku, dan diagram interaksi adalah tiga kategori utama UML. dimana diagram yang menjelaskan arsitektur sistem dan bagaimana terintegrasi satu sama lain disertakan untuk setiap kategori.

Berikut ini adalah daftar simbol UML:

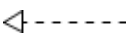
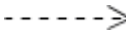
Tabel 2. 4 *Symbol Use Case Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menjelaskan berbagai peran yang dapat dimainkan pengguna dalam interaksi dengan kasus Penggunaan.
2		<i>Dependency</i>	Elemen yang mengandalkan komponen <i>non-standalone</i> akan terpengaruh ketika elemen <i>stand-alone</i> diubah.
3		<i>Generalization</i>	Koneksi di mana nenek moyang (objek induk) lebih unggul dari keturunan (objek anak), yang memiliki struktur data dan perilaku berbeda.
4		<i>Include</i>	Tunjukkan bahwa sumber <i>Use Case</i> eksplisit.

Lanjut Tabel 2. 4 *Symbol Use Case Diagram*

5		<i>Extend</i>	Menunjukkan bahwa perilaku <i>use case</i> sumber pada saat tertentu diperluas oleh <i>use case</i> target.
6		<i>Association</i>	Ini membuat hubungan antara dua objek.
7		<i>System</i>	Mengidentifikasi paket yang memberikan pandangan terbatas pada sistem.
8		<i>Use Case</i>	Penjelasan tentang urutan tindakan yang diambil sistem untuk memberikan hasil yang dapat diukur kepada para pelaku.
9		<i>Collaboration</i>	Aturan dan elemen lainnya bekerja sama untuk menciptakan sinergi, yaitu perilaku yang lebih baik daripada gabungan bagian-bagiannya.
10		<i>Note</i>	Elemen fisik yang ada saat aplikasi beroperasi dan berfungsi sebagai representasi sumber daya komputer.

Tabel 2. 5 *Symbol Class Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	suatu hubungan di mana objek anak berbagi struktur data dan perilaku objek induk (<i>aksesor</i>).
2		<i>Nary Association</i>	Mencoba menjauhi asosiasi dengan lebih dari dua item.
3		<i>Class</i>	Sekelompok barang yang mempunyai sifat dan fungsi serupa.
4		<i>Collaboration</i>	Penjelasan tentang urutan tindakan yang dilakukan sistem untuk memberikan hasil yang dapat diukur kepada agen.
5		<i>Realization</i>	Tindakan yang sebenarnya dilakukan suatu objek.
6		<i>Dependency</i>	Perubahan pada suatu unsur yang berdiri sendiri akan berdampak pada unsur yang bergantung pada komponen yang tidak berdiri sendiri.
7		<i>Association</i>	Yang menghubungkan antara dua objek.

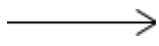
Tabel 2. 6 *Symbol Sequence Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek entitas adalah antarmuka yang dapat berkomunikasi satu sama lain.
2		<i>Message</i>	Persyaratan untuk komunikasi antar objek yang menyimpan informasi tentang kejadian.
3		<i>Message</i>	Informasi mengenai tindakan yang terjadi terdapat dalam spesifikasi komunikasi antar objek.

Tabel 2. 7 *Symbol State Chart Diagram*

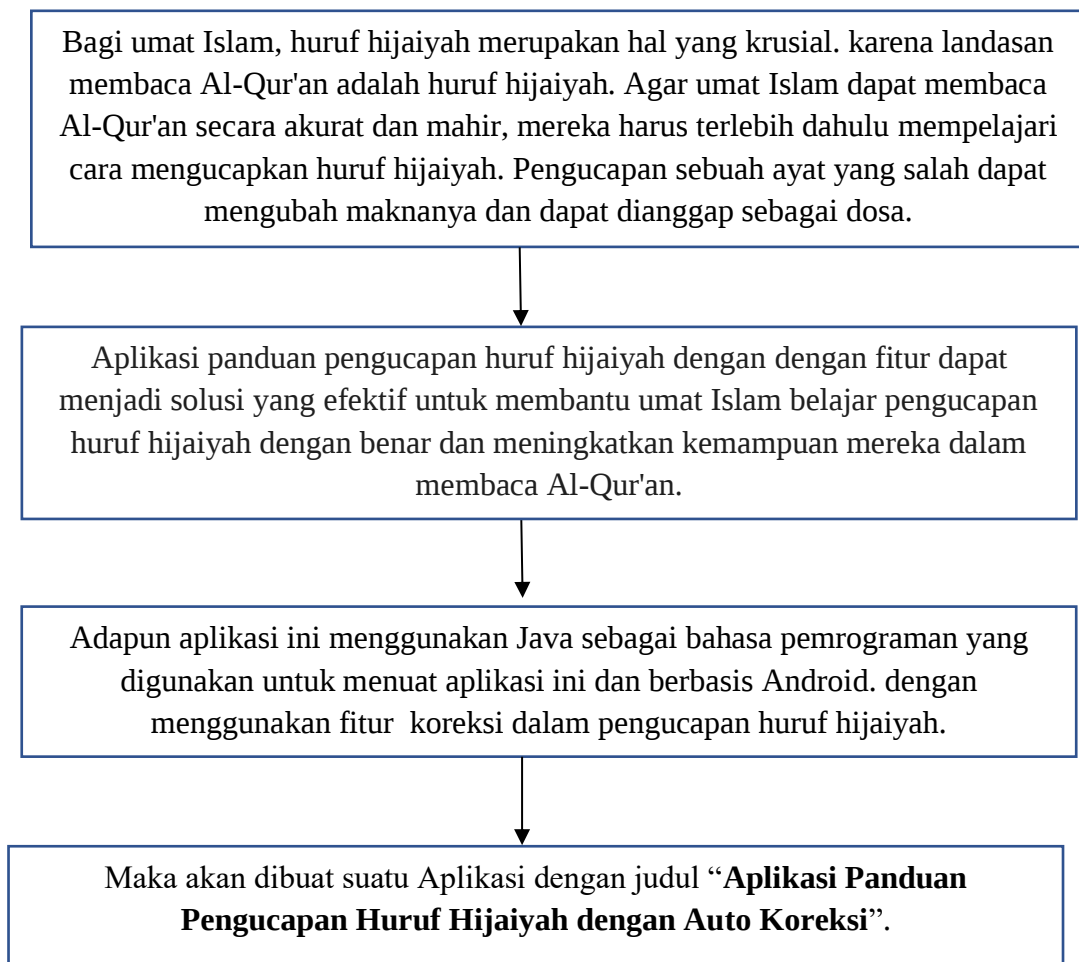
No	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Karakteristik dan nilai tautan yang dimiliki suatu objek pada waktu tertentu.
2		<i>Initial Pseudo State</i>	Karakteristik yang menjadi ciri objek yang dibuat atau diprakarsai.
3		<i>Final State</i>	Atribut objek diproduksi dan dihapus.

Tabel 2. 8 *Symbol Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
4		<i>Transition</i>	Suatu kejadian yang mengubah satu atau lebih nilai atribut suatu objek untuk memulai keadaan objek
5		<i>Association</i>	Atribut untuk menghubungkan antara dua objek
6		<i>Node</i>	Komponen fisik yang mewakili sumber daya komputasi dan hadir saat program dijalankan.
1		<i>Activity</i>	Mendemonstrasikan interaksi antara setiap <i>class</i> antarmuka.
2		<i>Action</i>	Kondisi sistem yang menunjukkan bagaimana suatu aktivitas dilakukan
3		<i>Initial Node</i>	Atribut untuk menggambarkan awal objek dibentuk atau diawali.
4		<i>Activity FinalNode</i>	Atribut untuk menggambarkan objek dibentuk dan menghapus.
5		<i>Fork Node</i>	Satu aliran yang pada akhirnya terbagi menjadi beberapa aliran.

C. Kerangka Pikir

Untuk memberikan pemahaman yang lebih baik mengenai alur penelitian yang dijelaskan, diagram berikut ini menjadi kerangkanya:



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Penelitian *kualitatif* adalah jenis penelitian yang digunakan, dan pengguna memperoleh manfaat dari pendekatan penelitian ini. Pengguna sudah terbiasa menggunakan aplikasi panduan pengucapan huruf Hijaiyah dengan fitur koreksi dengan berbasis Android.

B. Waktu dan Tempat Penelitian

Durasi rencana yang digunakan untuk penelitian adalah $\pm$ 2 bulan. dan bertempat di Masjid Babutaqwa, Desa Sipatuo, Kec.Patampanua, Kota Pinrang.

Tabel 3. 1 Uraian Kegiatan

No	Uraian Kegiatan	Bulan 2024		
		September	Oktober	November
1	Analisis Kebutuhan			
2	Perancangan/Desain			
3	Implementasi			
4	Pengujian			
5	Pemeliharaan			

B. Alat dan Bahan Penelitian

1. *Hardware*

Persyaratan perangkat keras berikut digunakan untuk mempersiapkan skripsi ini:

Tabel 3. 2 Perangkat Keras

Jenis	Spesifikasi
Laptop	Laptop ASUS DESKTOP-C2SGMRL
<i>Processor</i>	Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz 1.70 GHz
<i>Memory</i>	4,00 GB
<i>Hardisk</i>	465 GB
<i>SSD</i>	238

2. *Software*

Program ini dibuat dengan menggunakan *software* berikut:

Tabel 3. 3 Perangkat Lunak

Jenis	Spesifikasi
Sistem Operasi	<i>Windows 10 Pro</i>
Bahasa Pemrograman	<i>Java</i>
<i>Tools</i>	<i>Android Studio</i>
	<i>Google Speech API sebagai framework pengenalan kata</i>

D. Metode Pengumpulan Data

1. Secara tidak langsung

Metode tidak langsung ini melibatkan pengumpulan pengetahuan dan data dari buku, jurnal, makalah, dan internet.

2. Secara Langsung

Metode langsung melibatkan pengumpulan data atau informasi dengan mengevaluasi keakuratan aplikasi, khususnya yang berkaitan dengan cara

pengucapan huruf hijaiyaiyah.

E. Tahapan Penelitian

Berikut tahapan penelitian yang disebutkan dalam penelitian ini:

1. Analisis Kebutuhan

Pada titik ini dimulai dengan pemahaman terhadap kebutuhan dan tujuan perangkat lunak yang sedang dikembangkan. Peneliti akan mempelajari kebutuhan dan persyaratan pengguna, serta menetapkan fitur-fitur dan fungsi yang diperlukan.

2. Perancangan

Setelah itu peneliti menggunakan alternatif pemecahan masalah untuk merancang aplikasi yang ingin dikembangkan.

3. Implementasi

Peneliti mengeksekusi hasil desain berupa kode program setelah selesai merancang.

4. Pengujian

Setelah mengimplementasikan hasil rancangan kedalam bentuk kode program, kemudian peneliti melakukan pengujian terhadap aplikasi. Tahap ini untuk memastikan apakah ada kesalahan pada kode program atau *error* yang tampil saat di akses dan memperbaiki jika ada *error* tersebut.

5. Pemeliharaan

Tahap terakhir yaitu melakukan pemeliharaan aplikasi, pada tahap pemeliharaan dapat mereplikasi seluruh proses pengembangan, mulai dari

analisis spesifikasi hingga modifikasi perangkat lunak, namun tidak dapat mereplikasi pembuatan perangkat lunak baru.

F. Metode Pengujian

Dalam penelitian ini digunakan dua (dua) metode pengujian data yaitu pengujian *black box* dan *white box*.

1. Blackbox testing

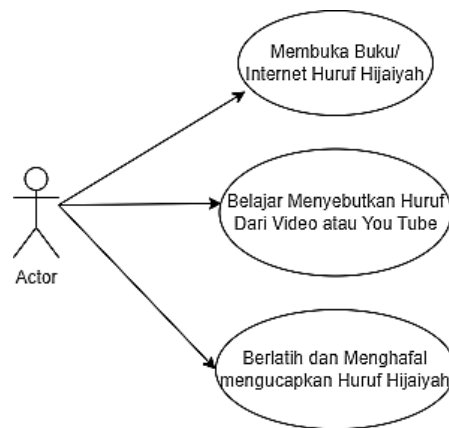
Fungsionalitas program yang sudah ada adalah penekanan utama dari pengujian *black box*. Pada *Blackbox testing* diuji dengan cara menjalankan program kemudian diamati apakah program tersebut apakah berhasil atau tidak. Teknik partisi *ekivalen*, yaitu pengujian berdasarkan masukan dari setiap menu pada perangkat lunak, digunakan dalam pengujian *black box*. Berdasarkan perannya, setiap menu masukan dikategorikan dan dikelompokkan untuk pengujian.

2. Whitebox testing

Pengujian *whitebox* berupaya menentukan apakah struktur aplikasi sesuai dengan aturan. *Whitebox testing* menitikberatkan pada pengujian dengan mengecek detail perancangan perangkat lunak. *Whitebox testing* dinilai dengan mendefinisikan semua alur dari perangkat lunak, kemudian membangun kasus yang akan digunakan dalam proses pengujian. Setelah itu, uji kasus tersebut untuk mendapatkan hasilnya.

G. Desain Sistem

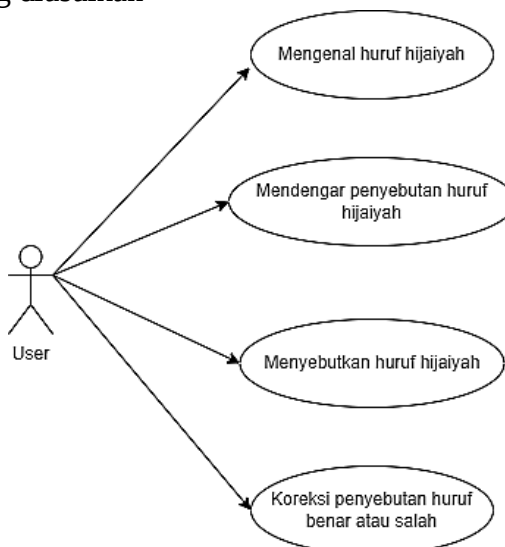
1) Desain sistem yang berjalan



Gambar 3. 1 Desain Sistem Yang Berjalan

Pada gambar 3.1 menjelaskan tentang desain sistem yang berjalan yaitu actor membuka buku iqra atau internet tentang huruf hijaiyah. Selanjutnya belajar menyebutkan huruf hijaiyah dari video atau you tube. Kemudian actor menghafal dan berlatih mengucapkan huruf hijaiyah.

2) Desain sistem yang diusulkan



Gambar 3. 2 Desain Sistem Yang Diusulkan

Pada gambar 3.2 menjelaskan tentang desain sistem yang diusulkan yaitu user dapat mengakses menu mengenal huruf dan mendengarkan prnyrbutan huruf. Selanjutnya user juga dapat mengakses menu menyebutkan huruf, Dimana user dapat berlatih menyebutkan huruf dengan menyebutkan huruf hijaiyah yang dipilih kemudian sistem akan mengoreksi penyebutan huruf sudah benar atau salah.

BAB IV

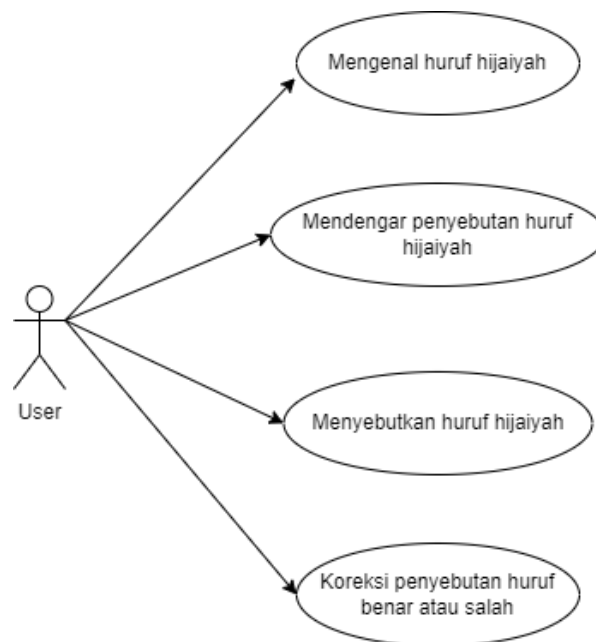
HASIL PENELITIAN DAN PEMBAHASAN

A. Analisis Aliran Data Diagram UML

Melalui analisis aliran data, objek sistem diorientasikan menggunakan diagram *use case*, diagram aktivitas, dan diagram urutan untuk memberikan rincian tentang fase dan alur aplikasi.

1. *Use Case Diagram*

Contoh use case menjelaskan bagaimana aktor atau pengguna terlibat dengan sistem. Gambar ini menunjukkan operasi utama sistem serta perspektif aktor luar.



Gambar 4. 1 *Use Case Diagram*

Ketika pengguna membuka aplikasi pada gambar di atas tanpa *login* terlebih dahulu, maka akan muncul layar yang mengidentifikasi dan merujuk huruf hijaiyah.

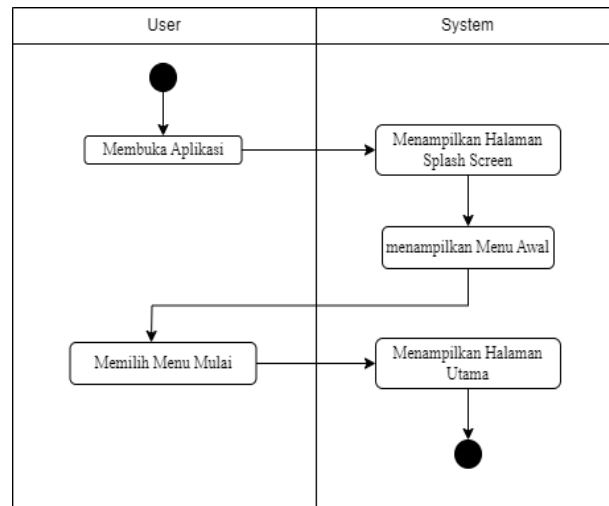
Tabel 4.1 Keterangan dari Desain Sistem Yang Diusulkan

Nama use case	Deskripsi
Mengenal huruf hijaiyah	<i>Use case</i> ini akan menampilkan huruf hijaiyah dari huruf alif (ا) hingga ya (ي)
Mendengarkan penyebutan huruf hijaiyah	<i>Use case</i> ini berada dalam use case mengenal huruf, memunculkan suara cara menyebutkan masing-masing huruf hijaiyah
Menyebutkan huruf hijaiyah	<i>Use case</i> ini berfungsi agar kita dapat memasukkan suara kita untuk menyebutkan huruf hijaiyah
Koreksi penyebutah huruf benar/salah	<i>Use case</i> ini berfungsi agar kita mengetahui apakah penyebutan huruf hijaiyah yang disebutkan benar atau salah

1. *Activity Diagram*

Alur kerja atau alur aktivitas dalam suatu sistem atau proses dapat dimodelkan dengan menggunakan diagram aktivitas, suatu bentuk diagram dalam UML (*Unified Modeling Language*). Diagram ini membantu menjelaskan operasi sistem dan urutan pelaksanaannya.

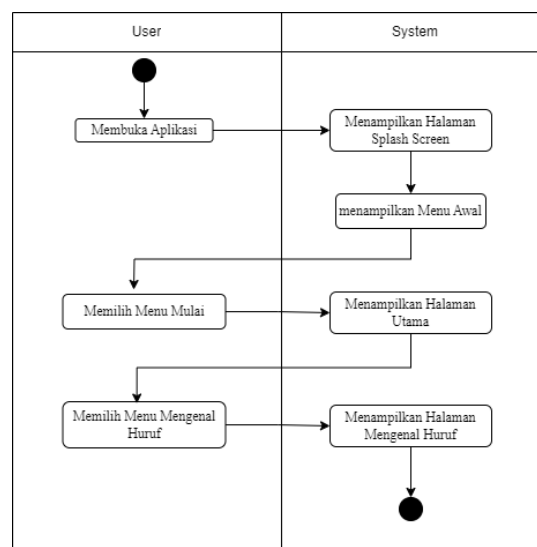
a. *Activity diagram* tampilan menu utama



Gambar 4. 2 Activity Diagram Tampilan Menu Utama

Gambar 4.2 menjelaskan tampilan start menu. Sistem menampilkan halaman menu awal ketika pengguna membuka aplikasi. Sistem akan menampilkan menu utama jika memutuskan untuk memilih menu *Start*.

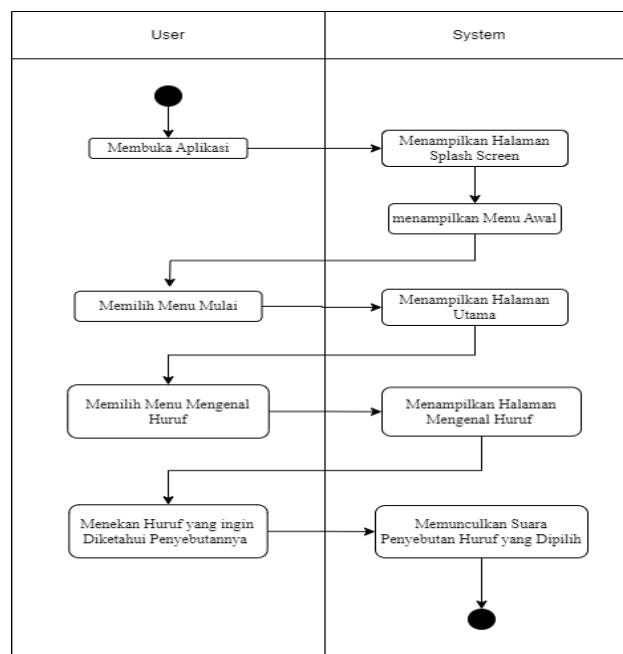
b. *Activity diagram* Tampilan Mengenal Huruf Hijaiyah



Gambar 4. 3 Activity Diagram Tampilan Mengenal Huruf Hijaiyah

Tampilan untuk mengidentifikasi huruf hijaiyah dijelaskan pada Gambar 4.3. Saat pengguna meluncurkan aplikasi, sistem akan menampilkan halaman menu awal. Sistem akan menampilkan menu utama jika memutuskan untuk memilih menu *Start*. Ketika pengguna memilih opsi pengenalan huruf hijaiyah pada menu utama, sistem akan menampilkan halaman informasi pengenalan huruf hijaiyah.

c. *Activity diagram* Tampilan Mendengarkan Huruf Hijaiyah

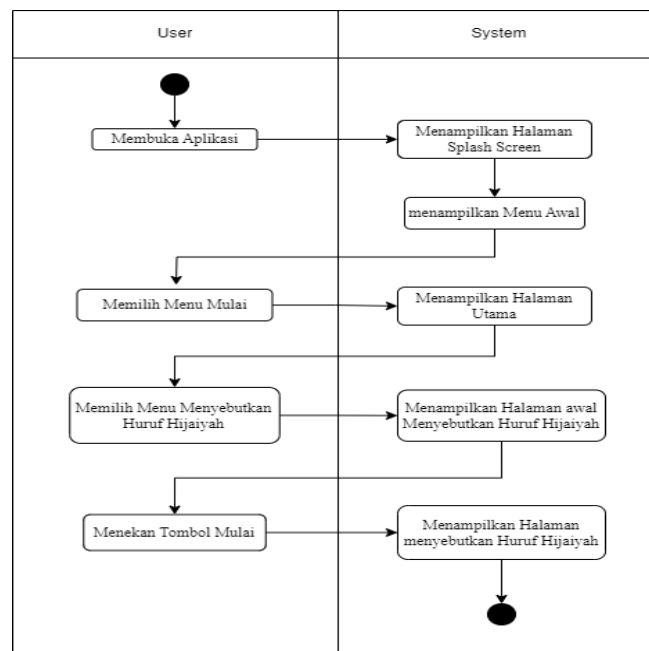


Gambar 4. 4 Activity Diagram Tampilan Mendengarkan Huruf

Tampilan mendengarkan atau mengucapkan huruf hijaiyah dijelaskan pada Gambar 4.4 diatas. Saat pengguna meluncurkan aplikasi, sistem akan menampilkan halaman menu awal. Pilih menu *start*, dan sistem kemudian akan menampilkan menu utama. Pengguna memilih opsi untuk mengenal huruf hijaiyah dari menu utama. Setelah itu akan ditampilkan halaman identifikasi huruf hijaiyah oleh sistem. Pengguna memilih huruf yang ingin dipelajari cara mengucapkannya pada halaman

pengenalan huruf, dan sistem kemudian akan menghasilkan keluaran suara yang menyatakan huruf yang dipilih.

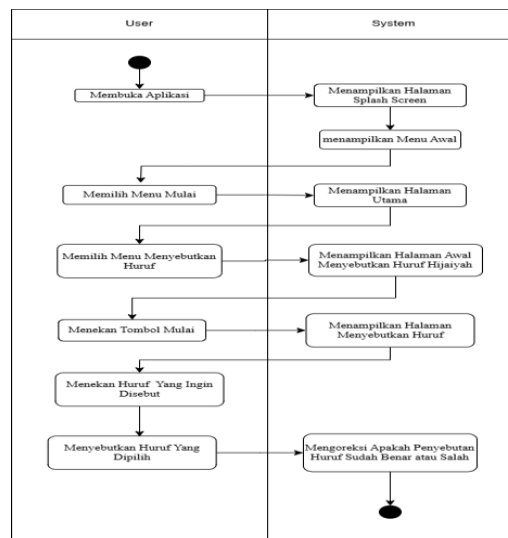
d. *Activity* diagram Tampilan Menyebutkan Huruf Hijaiyah



Gambar 4. 5 *Activity* Diagram Tampilan Menyebutkan Huruf Hijaiyah

Gambar 4.5 diatas memberikan penjelasan tampilan yang berkaitan dengan huruf hijaiyah. Sistem akan menampilkan halaman menu pertama saat pengguna meluncurkan program. Halaman menu utama akan ditampilkan oleh sistem setelah Anda memilih menu *Start*. Ketika pengguna memilih menu diskusi dari menu utama, sistem akan menampilkan halaman pertama yang merujuk pada huruf hijaiyah. Sistem kemudian akan menampilkan halaman yang membahas tentang huruf hijaiyah jika pengguna memilih menu *Start*.

e. Activity diagram Tampilan Mengoreksi Penyebutan Huruf Hijaiyah



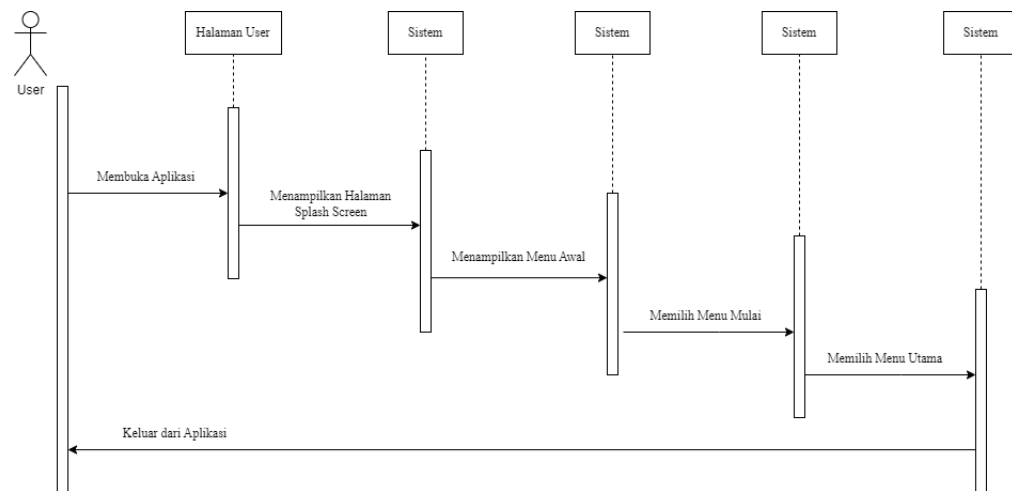
Gambar 4. 6 Activity Diagram Tampilan Mengoreksi Penyebutan Huruf Hijaiyah

Pada gambar 4.6 menjelaskan tentang tampilan mengoreksi penyebutan huruf hijaiyah, dimana Saat pengguna meluncurkan aplikasi, halaman awal ditampilkan oleh sistem, setelah itu pilih menu mulai. Kemudian halaman menu utama akan ditampilkan oleh sistem. Ketika pengguna memilih menu penyebutan huruf pada halaman utama, maka sistem akan menampilkan terlebih dahulu tampilan awal penyebutan huruf. Ketika Anda memilih menu *start*, sistem kemudian akan menampilkan halaman yang membahas tentang huruf hijaiyah. Setelah itu pilih huruf yang ingin disebutkan, kemudian *user* menyebutkan huruf yang telah dipilih, kemudian sistem akan merekan penyebutan huruf yang dipilih kemudian menerjemahkan suara kedalam bentuk teks menggunakan *speech recognizer* untuk menerjemahkan suara kedalam bentuk tulisan. Selanjutnya sistem akan mencocokkan hasil suara ke dalam bentuk tulisan tersebut dengan *string*, kemudian mengeluarkan *output* suara ketika penyebutan sesuai atau tidak.

3. Sequence Diagram

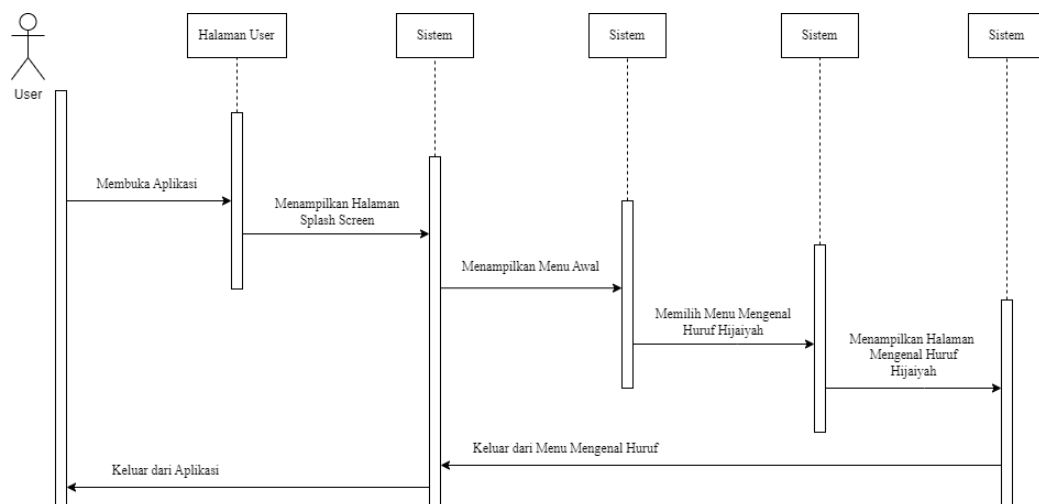
Sequence diagram merupakan diagram interaksi yang menggambarkan bagaimana suatu tindakan dilakukan, pesan apa yang disampaikan, dan kapan tindakan tersebut dilakukan.

a. Sequence Diagram Tampilan untuk Menu Utama



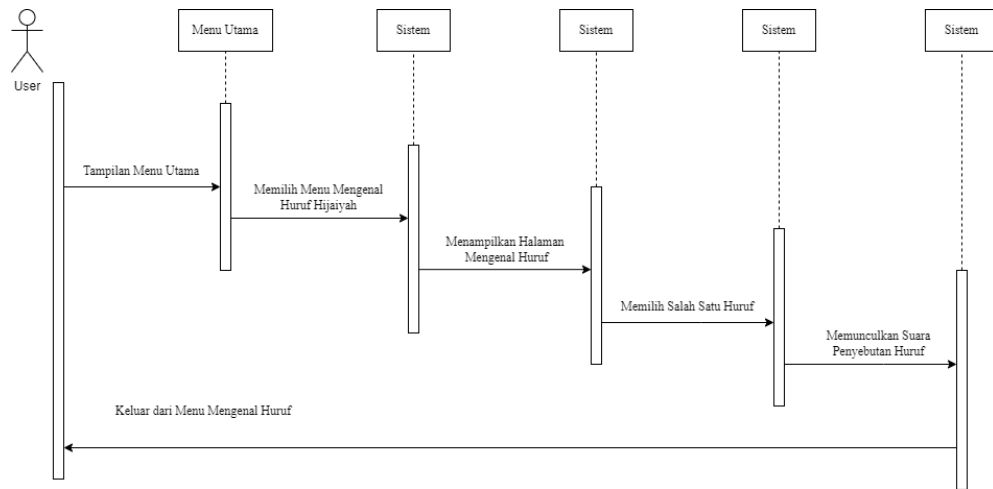
Gambar 4. 7 Sequence Diagram untuk Tampilan Menu Utama

b. Sequence Diagram untuk Tampilan Mengenal Huruf Hijaiyah



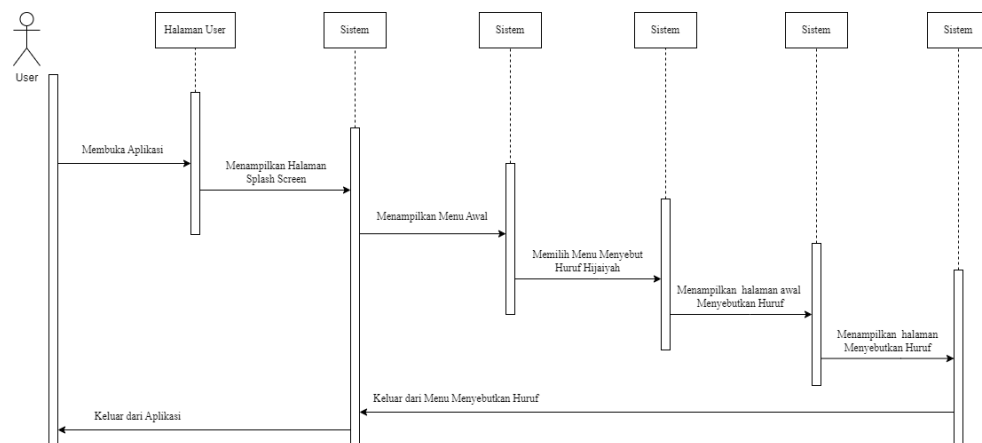
Gambar 4. 8 Sequence Diagram Tampilan untuk Mengenal Huruf Hijaiyah

c. *Sequence Diagram* untuk Tampilan mendengarkan Huruf Hijaiyah



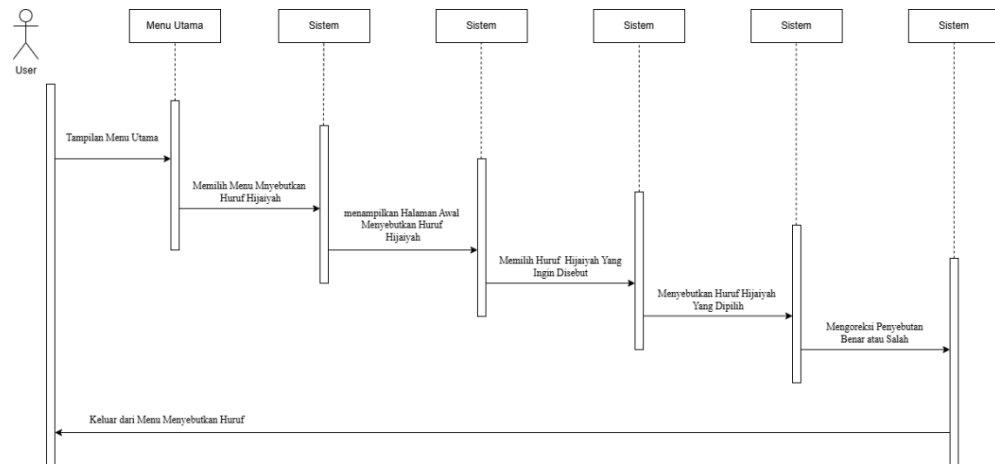
Gambar 4. 9 *Sequence Diagram* Tampilan Mendengarkan Huruf Hijaiyah

d. *Sequence Diagram* Tampilan Menyebutkan Huruf Hijaiyah



Gambar 4. 10 *Sequence Diagram* Tampilan Menyebutkan Huruf Hijaiyah

e. *Sequence Diagram* Tampilan Mengoreksi Penyebutan Huruf Hijaiyah



Gambar 4. 11 *Sequence Diagram* Tampilan Koreksi Penyebutan Huruf Hijaiyah

B. Detail Sistem

1. Tampilan Menu Awal

Gambar di bawah menunjukkan tampilan pertama menu program yang berisi tombol *start* dan *exit*.



Gambar 4.12 Tampilan Awal Aplikasi

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background_1"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/logo"
        android:layout_width="431dp"
        android:layout_height="267dp"
        android:layout_marginTop="4dp"
        android:contentDescription="logo"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/logo"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="-1dp" />


```

Gambar 4.13 Script untuk layout untuk tampilan awal aplikasi

```
<ImageButton
    android:id="@+id/mulai"
    android:layout_width="261dp"
    android:layout_height="125dp"
    android:layout_marginTop="476dp"
    android:background="@drawable/mulai"
    android:contentDescription="Tombol untuk mulai aplikasi"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/logo"
    tools:ignore="MissingConstraints" />

<ImageButton
    android:id="@+id/keluar"
    android:layout_width="92dp"
    android:layout_height="101dp"
    android:layout_marginBottom="4dp"
    android:background="@drawable/keluar"
    android:contentDescription="Tombol untuk info aplikasi"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.949"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Gambar 4.14 Script layout tombol mulai dan exit (keluar)

```
public class MainActivity extends AppCompatActivity {

    // Deklarasi variabel Button
    ImageButton mulai; 2 usages
    ImageButton keluar; 2 usages

    @SuppressWarnings("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Inisialisasi variabel Button
        mulai = findViewById(R.id.mulai);
        keluar = findViewById(R.id.keluar);

        // Set OnClickListener untuk Button Mulai
        mulai.setOnClickListener(v -> {
            Intent intent = new Intent( packageName, MainActivity.this, ActivityMulai.class);
            startActivity(intent);
        });

        // Set OnClickListener untuk Button Keluar
        keluar.setOnClickListener(v -> {
            finishAffinity(); // Menutup semua activity dan keluar dari aplikasi
        });
    }
}
```

Gambar 4.15 Script proses tombol mulai dan exit (keluar)

2. Tampilan Menu Utama

Gambar di bawah menunjukkan tampilan menu utama aplikasi yang memiliki dua fitur yaitu kemampuan mengenal dan mengucapkan huruf hijaiyah.



Gambar 4.16 Tampilan Menu pada Utama Aplikasi

a. Scrip background dan kembali

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background_1"
    android:descendantFocusability="afterDescendants"
    tools:context=".ActivityMulai">

    <!-- Perbaikan di sini -->

    <ImageButton
        android:id="@+id/kembali"
        android:layout_width="79dp"
        android:layout_height="72dp"
        android:layout_marginEnd="324dp"
        android:background="@drawable/kembali_"
        android:contentDescription="Tombol untuk memulai fitur kembali"
        app:layout_constraintEnd_toEndOf="parent"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteY="10dp" />
```

Gambar 4.17 Script untuk layout tampilan menu utama

```
public class ActivityMulai extends AppCompatActivity {
    ImageButton kembali; 2 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mulai); // Pastikan nama file XML sesuai

        kembali = findViewById(R.id.kembali);

        // Tombol "Kembali"
        kembali.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Menutup activity ini dan kembali ke activity sebelumnya
                finish();
            }
        });
    }
}
```

Gambar 4.18 Script proses tombol kembali

```

<ImageButton
    android:id="@+id/mengenal"
    android:layout_width="202dp"
    android:layout_height="219dp"
    android:layout_marginTop="120dp"
    android:background="@drawable/mengenal"
    android:contentDescription="Tombol untuk memulai fitur mengenal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.401"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<ImageButton
    android:id="@+id/menyebut"
    android:layout_width="203dp"
    android:layout_height="222dp"
    android:background="@drawable/menyebut"
    android:contentDescription="Tombol untuk memulai fitur menyebut"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.74" />

```

Gambar 4.19 Script untuk layout tampilan tombol menu

```

public class ActivityMulai extends AppCompatActivity {
    ImageButton mengenal; 2 usages
    ImageButton menyebut; 2 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mulai); // Pastikan nama file XML sesuai

        // Inisialisasi tombol untuk mengenal huruf hijaiyah
        mengenal = findViewById(R.id.mengenal);
        menyebut = findViewById(R.id.menyebut);

        // Tombol "Mengenal"
        mengenal.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Mulai activity MengenalHuruf
                Intent intent = new Intent(getApplicationContext(), MengenalHuruf.class);
                startActivity(intent);
            }
        });

        // Tombol "Menyebut"
        menyebut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Mulai activity MenyebutkanHuruf
                Intent intent = new Intent(getApplicationContext(), MenyebutkanHuruf.class);
                startActivity(intent);
            }
        });
    }
}

```

Gambar 4.20 Script proses tombol menu

3. Tampilan Menu Mengenal Huruf Hijaiyah

Halaman untuk mengidentifikasi huruf hijaiyah ditunjukkan pada gambar di bawah ini. Ketika kita menekan salah satu huruf hijaiyah maka akan terdengar bunyi yang menandakan bahwa huruf tersebut telah ditekan.



Gambar 4.21 Tampilan Halaman Mengenal huruf

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background_2"
    tools:context=".mengenai_huruf">

    <!-- Tombol Kembali -->
    <ImageButton
        android:id="@+id/kembali"
        android:layout_width="48dp"
        android:layout_height="48dp"
        android:layout_margin="16dp"
        android:background="@drawable/kembali_"
        android:contentDescription="Tombol kembali"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <!-- Tombol Lanjut -->
    <ImageButton
        android:id="@+id/lanjut"
        android:layout_width="48dp"
        android:layout_height="48dp"
        android:layout_margin="16dp"
        android:background="@drawable/lanjut_"
        android:contentDescription="Tombol lanjut"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

```

Gambar 4.22 Script untuk tampilan menu mengenal huruf dengan tombol kembali dan lanjut

```
public class mengenai_huruf extends AppCompatActivity {

    ImageButton kembali; 2 usages
    ImageButton lanjut; 2 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mengenai_huruf1); // Pastikan nama file XML sesuai

        // Inisialisasi tombol
        kembali = findViewById(R.id.kembali);
        lanjut = findViewById(R.id.lanjut);

        // Tombol "Kembali"
        kembali.setOnClickListener(v -> {
            finish(); // Menutup activity ini dan kembali ke activity sebelumnya
        });

        // Tombol "Lanjut"
        lanjut.setOnClickListener(v -> {
            Intent intent = new Intent(packageContext, mengenai_huruf_2.class);
            startActivity(intent); // Beralih ke activity berikutnya
        });
    }
}

```

Gambar 4.23 Script proses tombol lanjut dan kembali

```
<!-- Susunan huruf menggunakan LinearLayout -->
<LinearLayout
    android:id="@+id/container_huruf"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/kembali"
    android:layout_marginTop="16dp">

    <!-- Baris 1 -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginBottom="16dp">

        <!-- Huruf ta -->
        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:orientation="vertical"
            android:gravity="center">

            <ImageButton
                android:id="@+id/ta"
                android:layout_width="96dp"
                android:layout_height="100dp"
                android:background="@drawable/ta_"
                android:contentDescription="Huruf Ta" />

            <ImageButton
                android:id="@+id/huruf_ta"
                android:layout_width="96dp"
                android:layout_height="21dp"
                android:background="@drawable/huruf_ta_"
                android:contentDescription="Nama huruf Ta"
                tools:ignore="TouchTargetSizeCheck"/>

        </LinearLayout>
    </LinearLayout>

```

Gambar 4.24 Script untuk layout tampilan huruf

```

private void setupLetterButtons() { 1 usage

    // Contoh tombol Ta
    ImageButton ta = findViewById(R.id.ta);
    ta.setOnClickListener(v -> playSound(R.raw.ta)); // Pastikan file ba.mp3 ada di res/raw

private void playSound(int soundResource) { 10 usages
    // Pastikan untuk menghentikan suara yang sedang diputar sebelum memutar yang baru
    if (mediaPlayer != null) {
        mediaPlayer.release(); // Hentikan suara sebelumnya
        mediaPlayer = null; // Setel ke null untuk mencegah kebocoran memori
    }

    // Buat media player baru dengan sumber suara yang dipilih
    mediaPlayer = MediaPlayer.create(context: this, soundResource);
    mediaPlayer.start(); // Mulai memutar suara

    // Menghentikan dan melepaskan MediaPlayer setelah selesai memutar
    mediaPlayer.setOnCompletionListener(mp -> {
        mediaPlayer.release();
        mediaPlayer = null;
    });
}

@Override
protected void onDestroy() {
    // Hentikan dan lepaskan media player jika masih aktif
    if (mediaPlayer != null) {
        mediaPlayer.release();
        mediaPlayer = null;
    }
    super.onDestroy();
}
}

```

Gambar 4.25 Script proses suara

4. Tampilan Menu Awal Menyebutkan Huruf Hijaiyah

Yampilan awal menyebutkan huruf hijaiyah terlihat pada halaman pertama pada gambar dibawah ini. terdapat tombol mulai untuk masuk ke halaman menyebutkan huruf.



Gambar 4.26 Tampilan Halaman Awal Menyebutkan Huruf

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background_1"
    tools:context=".menyebutkan_huruf">
    <ImageButton
        android:id="@+id/kembali"
        android:layout_width="76dp"
        android:layout_height="75dp"
        android:layout_marginEnd="336dp"
        android:background="@drawable/kembali_"
        android:contentDescription="Tombol untuk memulai fitur kembali"
        app:layout_constraintEnd_toEndOf="parent"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteY="30dp" />
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="273dp"
        android:layout_height="451dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.326"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.359"
        app:srcCompat="@drawable/oke" />
    <ImageButton
        android:id="@+id/mulai"
        android:layout_width="261dp"
        android:layout_height="125dp"
        android:layout_marginBottom="260dp"
        android:background="@drawable/mulai_"
        android:contentDescription="Tombol untuk mulai aplikasi"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        tools:ignore="MissingConstraints" />

```

Gambar 4.27 Script untuk layout tampilan awal menyebutkan huruf

```

public class menyebutkan_huruf extends AppCompatActivity {

    // Deklarasi variabel Button
    ImageButton mulai; 2 usages
    ImageButton kembali; 2 usages

    @SuppressWarnings("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menyebutkan_huruf);

        // Inisialisasi variabel Button
        mulai = findViewById(R.id.mulai);
        kembali = findViewById(R.id.kembali);

        // Set OnClickListener untuk Button Mulai
        mulai.setOnClickListener(v -> {
            Intent intent = new Intent(getApplicationContext(), menyebutkan_huruf_1.class);
            startActivity(intent);
        });

        // Tombol "Kembali"
        kembali.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Menutup activity ini dan kembali ke activity sebelumnya
                finish();
            }
        });
    }
}

```

Gambar 4.28 Script proses tombol mulai dan kembali

5. Tampilan Menu Menyebutkan Huruf Hijaiyah

Terdapat huruf hijaiyah pada menu yang terlihat pada gambar dibawah ini..



Gambar 4.29 Tampilan Halaman Menyebutkan Huruf Hijaiyah

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background_2"
    tools:context=".menyebutkan_huruf_1">

    <TextView
        android:id="@+id/tv_result"
        android:layout_width="366dp"
        android:layout_height="75dp"
        android:gravity="center"
        android:text="Silakan sebutkan huruf sebanyak 2 kali"
        android:textColor="#000000"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.488"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.125" />

    <TextView
        android:id="@+id/hasil_ucapan"
        android:layout_width="400dp"
        android:layout_height="50dp"
        android:layout_marginTop="720dp"
        android:fontFamily="sans-serif-black"
        android:gravity="center"
        android:text="Hasil ucapan akan muncul di sini"
        android:textColor="#000000"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

Gambar 4.30 Script layout tampilan menu menyebutkan huruf untuk teks

```

<!-- Tombol Kembali -->
<ImageButton
    android:id="@+id/kembali"
    android:layout_width="48dp"
    android:layout_height="48dp"
    android:layout_margin="16dp"
    android:background="@drawable/kembali_"
    android:contentDescription="Tombol kembali"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<!-- Tombol Lanjut -->
<ImageButton
    android:id="@+id/lanjut"
    android:layout_width="48dp"
    android:layout_height="48dp"
    android:layout_margin="16dp"
    android:background="@drawable/lanjut_"
    android:contentDescription="Tombol lanjut"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

Gambar 4.31 Script layout tombol kembali dan lanjut

```

@SuppressLint("MissingInflatedId")
@override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menyebutkan_huruf);

    // Inisialisasi komponen UI
    kembali = findViewById(R.id.kembali);
    lanjut = findViewById(R.id.lanjut);

    // Tombol "Kembali"
    kembali.setOnClickListener(v -> {
        Intent intent = new Intent( packageName, menyebutkan_huruf_1.this, menyebutkan_huruf.class);
        startActivity(intent);
        finish();
    });

    // Tombol "Lanjut"
    lanjut.setOnClickListener(v -> {
        Intent intent = new Intent( packageName, menyebutkan_huruf_1.this, menyebutkan_huruf_2.class);
        startActivity(intent);
    });
}

```

Gambar 4.32 Script proses tombol kembali dan lanjut

```

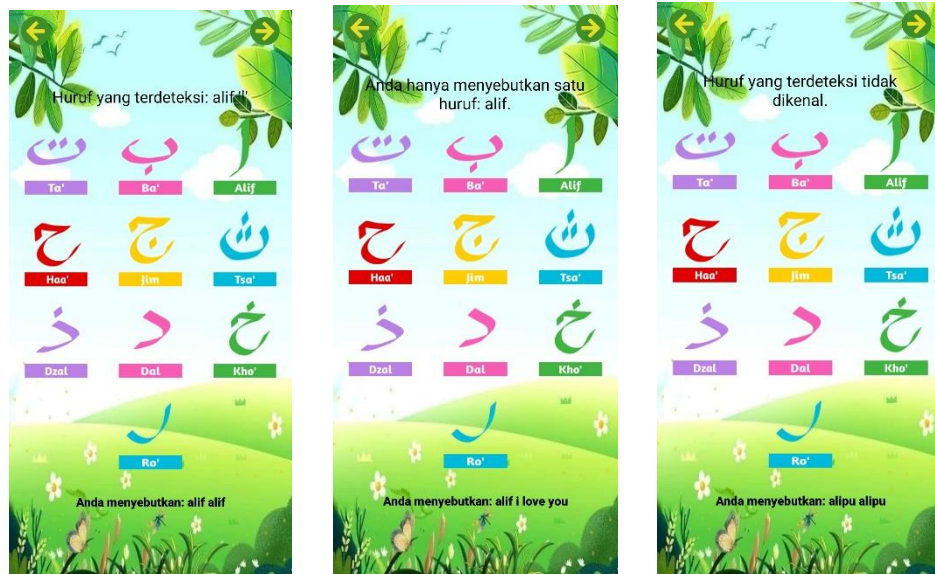
<!-- Susunan huruf menggunakan LinearLayout -->
<LinearLayout
    android:id="@+id/container_huruf"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="100dp"
    android:orientation="vertical"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/kembali">
    <!-- Baris 1 -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:orientation="horizontal">
        <!-- Huruf ta -->
        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:gravity="center"
            android:orientation="vertical">
            <ImageButton
                android:id="@+id/ta"
                android:layout_width="96dp"
                android:layout_height="100dp"
                android:background="@drawable/ta_"
                android:contentDescription="Huruf Ta" />
            <ImageButton
                android:id="@+id/huruf_ta"
                android:layout_width="94dp"
                android:layout_height="21dp"
                android:background="@drawable/huruf_ta_"
                android:contentDescription="Nama huruf Ta"
                tools:ignore="TouchTargetSizeCheck" />
        </LinearLayout>
    </LinearLayout>

```

Gambar 4.33 Script layout tampilan huruf

6. Tampilan Tampilan Koreksi Ketika Menyebutkan Huruf Hijaiyah

Tampilan halaman saat menyatakan huruf hijaiyah terlihat pada gambar di bawah ini, ada 3 bagian yaitu tampilan ketika pengucapan benar, salah satu huruf salah, dan pengucapan salah.



Gambar 4.34 Tampilan Halaman Koreksi ketika Menyebutkan Huruf Hijaiyah

```
// Inisialisasi SpeechRecognizer
speechRecognizer = SpeechRecognizer.createSpeechRecognizer(context, this);
Intent speechIntent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
speechIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
speechIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());

// Tombol huruf Ta
hurufTa.setOnClickListener(v -> {
    selectedLetter = "ta";
    tvResult.setText("Mendengarkan...");
    speechRecognizer.startListening(speechIntent);
});
```

Gambar 4.35 Script proses merekam suara ketika menekan salah satu huruf

```
// Menangani hasil pengucapan
speechRecognizer.setRecognitionListener(new RecognitionListener() {
    @Override
    public void onResults(Bundle results) {
        ArrayList<String> resultList = results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
        if (resultList != null && !resultList.isEmpty()) {
            String spokenText = resultList.get(0).toLowerCase().trim();
            HasilUcapan.setText("Anda menyebutkan: " + spokenText);

            // Tunda evaluasi untuk memberi waktu menampilkan pengucapan
            new Handler().postDelayed(() -> {
                switch (selectedLetter) {
                    case "ta":
                        handleTa(spokenText);
                        break;
                }
            }, 1000);
        }
    }
});
```

Gambar 4.36 Script proses mencocokkan huruf yang dipilih dengan hasil koreksi

```

private void handleTa(String spokenText) { // usage
    if (spokenText.equalsIgnoreCase( anotherString: "ta ta" ) || spokenText.equalsIgnoreCase( anotherString: "taa taa" ) || spokenText.equalsIgnoreCase( anotherString: "tata" )) {
        tvResult.setText("Huruf yang terdeteksi: ta 'ا'");
        playSound(R.raw.ta_benar);
    } else if (spokenText.contains("ta" ) || spokenText.equalsIgnoreCase( anotherString: "taa" )) {
        tvResult.setText("Anda hanya menyebutkan satu huruf: ta.");
        playSound(R.raw.satu_ta_salah);
    } else {
        tvResult.setText("Huruf yang terdeteksi tidak dikenal.");
        playSound(R.raw.ta_salah);
    }
}
}

```

Gambar 4.37 Script proses mengoreksi penyebutan huruf yang dipilih

C. Penjelasan fitur koreksi

1. Pengenalan Suara

Proses ini berfokus pada mendeteksi suara yang dihasilkan oleh pengguna dan menyimpannya dalam bentuk *string* (S) untuk dibandingkan dengan huruf yang benar (T).

Definisi:

- a. **S**: Suara yang terdeteksi (*output* dari sistem pengenalan suara).
- b. **T**: Teks yang diharapkan, yaitu huruf yang ingin dikenali (misalnya "alif", "ba", dll).

$S = \text{SpeechRecognizer.detectedSpeech}()$ (Suara yang terdeteksi dari pengguna)

Setelah suara terdeteksi, kita akan membandingkannya dengan huruf yang diinginkan, yang ditandai dengan variabel **T**.

2. Validasi Pengucapan

Pada tahap ini, kita memeriksa kesesuaian pengucapan dengan huruf yang diinginkan. Ada tiga kemungkinan hasil:

- a. Pengucapan **benar**: Suara yang terdeteksi sama persis dengan huruf yang diinginkan.
- b. Pengucapan **sebagian benar**: Suara yang terdeteksi mengandung sebagian dari huruf yang diinginkan, tetapi tidak sepenuhnya cocok.
- c. Pengucapan **salah**: Suara yang terdeteksi sama sekali tidak cocok dengan huruf yang diinginkan.

Rumus Validasi:

Untuk memvalidasi pengucapan, kita membandingkan **S** dengan **T** dengan tiga kondisi:

- 1) Pengucapan Benar: Jika **S** sama persis dengan **T**

$$\text{Validasi } (S,T) = \begin{cases} 1 & \text{jika } S = T \\ 0 & \text{jika } S \neq T \end{cases}$$

Artinya: Suara yang terdeteksi **S** persis sama dengan huruf yang diinginkan **T**.

- 2) Pengucapan Satu Benar: Jika **S** mengandung sebagian dari **T**, misalnya kata "ba" untuk "ba", atau pengucapan singkat yang serupa.

$$\text{Validasi Sebagian } (S,T) = \begin{cases} 0,5 & \text{jika } S \in T \text{ tetapi } S \neq T \\ 0 & \text{jika } S \notin T \end{cases}$$

Artinya: Suara yang terdeteksi **S** mengandung sebagian huruf **T** (misalnya "ba" untuk "baa"), tapi tidak persis.

- 3) Pengucapan Salah: Jika **S** sama sekali tidak cocok dengan **T**.

$$\text{Validasi Salah } (S,T) = \text{jika } S \notin T$$

Artinya: Suara yang terdeteksi **S** tidak mengandung **T** sama sekali.

3. Fungsi Koreksi dan Umpan Balik

Setelah kita memvalidasi suara, sistem akan memberikan umpan balik kepada pengguna berdasarkan hasil perbandingan antara **S** dan **T**. Fungsi koreksi ini memiliki tiga hasil yang berbeda, yaitu pengucapan benar, sebagian benar, atau salah.

Rumus Fungsi Koreksi:

Fungsi koreksi menghasilkan umpan balik yang diberikan kepada pengguna berdasarkan hasil validasi:

$$f(S) = \begin{cases} \text{"Huruf yang terdeteksi:"} + T & \text{jika validasi } (S,T) = 1 \\ \text{"Anda hanya menyebutkan satu huruf:'} + S & \text{jika validasi sebagian } (S,T) = 0,5 \\ \text{"Huruf yang terdeteksi tidak dikenal."} & \text{jika validasi salah } (S,T) = 0 \end{cases}$$

Umpan Balik Berdasarkan Kode:

- Pengucapan Benar: Jika suara yang terdeteksi sesuai dengan huruf yang diinginkan, maka sistem memberikan umpan balik yang positif dan memainkan suara yang benar.

Output = "Huruf yang terdeteksi:" +T

Dan memainkan suara yang benar.

- Pengucapan Satu Benar: Jika suara yang terdeteksi mengandung sebagian huruf yang benar, sistem memberikan umpan balik bahwa hanya sebagian huruf yang disebutkan dan memainkan suara yang salah.

Output = "Anda hanya menyebutkan satu huruf:" +S

Dan memainkan suara yang salah (misalnya, suara untuk pengucapan yang kurang tepat).

- c. Pengucapan Salah: Jika suara yang terdeteksi tidak cocok dengan huruf yang diinginkan, maka sistem memberikan umpan balik negatif dan memainkan suara yang salah.

Output = "Huruf yang terdeteksi tidak dikenal: "

Dan mainkan suara salah.

3. Rumus Koreksi untuk Semua Huruf

Untuk setiap huruf yang ingin dideteksi (misalnya alif, ba, ta, dll), kita bisa mendefinisikan rumus koreksi ini. Setiap huruf akan diuji menggunakan rumus Validasi (**S**, **T**) dan kemudian memberikan umpan balik yang sesuai.

Misalnya, untuk huruf Alif:

$$\text{Validasi Alif (S)} = \begin{cases} 1 & \text{jika } S = \text{"alif"} \\ 0,5 & \text{jika } S \in \text{"alif"} \\ 0 & \text{jika } S \notin \text{"alif"} \end{cases} \quad \text{tetapi } S \neq \text{"alif"}$$

Fungsi koreksi:

$$f(S) = \begin{cases} \text{"Huruf yang terdeteksi: alif"} & \text{jika validasi Alif (S) = 1} \\ \text{"Anda hanya menyebutkan satu huruf: ' + S} & \text{jika validasi Alif (S) = 0,5} \\ \text{"Huruf yang terdeteksi tidak dikenal.} & \text{jika validasi Alif (S) 0} \end{cases}$$

Secara keseluruhan, rumus matematis untuk mengenali suara dan mengoreksi pengucapan berdasarkan kode yang diberikan dapat disusun sebagai berikut:

$$\text{Validasi Alif } (S) = \begin{cases} 1 & \text{jika } S = T \\ 0,5 & \text{jika } S \in T \\ 0 & \text{jika } S \notin T \end{cases} \quad \text{tetapi } S \neq T$$

$$f(S) = \begin{cases} \text{"Huruf yang terdeteksi:"} + T & \text{jika validasi } (S,T) = 1 \\ \text{"Anda hanya menyebutkan satu huruf:"} + S & \text{jika validasi sebagian } (S,T) = 0,5 \\ \text{"Huruf yang terdeteksi tidak dikenal."} & \text{jika validasi salah } (S,T) = 0 \end{cases}$$

Di mana S adalah suara yang terdeteksi dan T adalah huruf yang diinginkan. Fungsi validasi memeriksa kesesuaian pengucapan, dan fungsi koreksi memberikan umpan balik berdasarkan hasil tersebut.

D. Pengujian Sistem

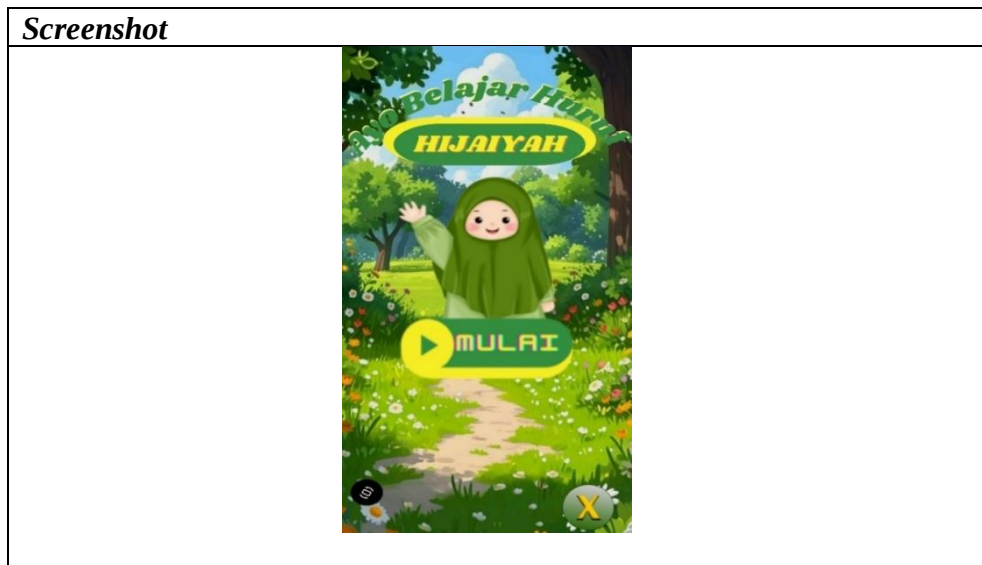
Dalam penelitian ini, dua pendekatan pengujian sistem digunakan: pengujian *blackbox* dan pengujian *whit box*. Ini adalah hasil percobaan yang dilakukan dengan metodologi ini.

1. Black Box Testing

a. Halaman Menu Beranda Pengujian Black Box

Tabel 4. 2 *Black Box Testing* untuk Halaman Menu Awal

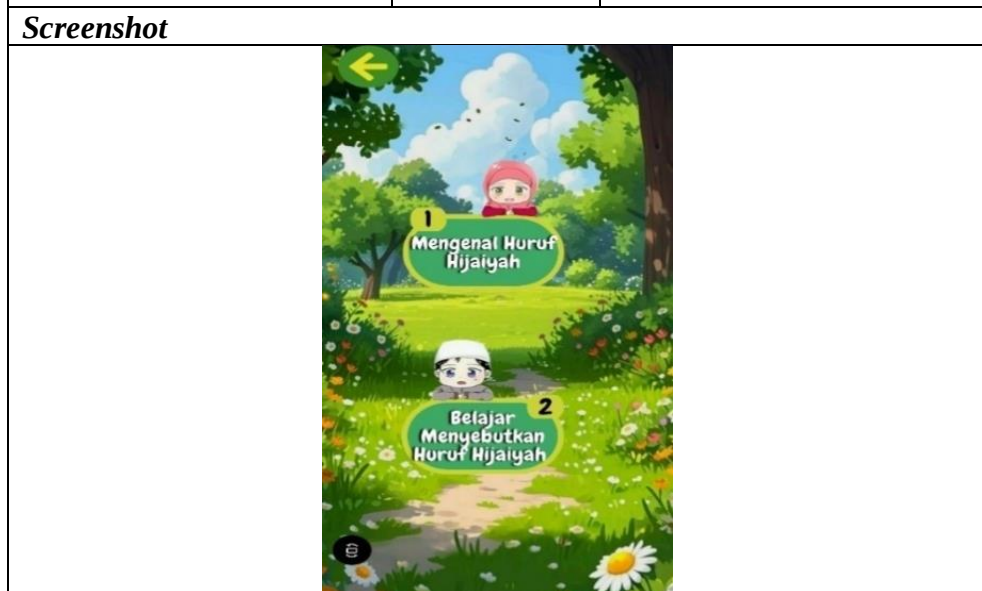
Tes Faktor	Hasil	Keterangan
Pengguna meluncurkan program.	✓	menampilkan halaman menu awal dengan sukses.



b. Menguji Halaman Menu Utama dengan *Black Box*

Tabel 4. 3 *Black Box Testing* Halaman Menu Utama

Tes Faktor	Hasil	Keterangan
Pengguna memilih menu Mulai.	✓	menampilkan halaman menu utama dengan sukses.



c. Halaman Menu Pengujian *Black Box*: Mengenal Huruf Hijaiyah

Tabel 4. 4 *Black Box Testing* Halaman Menu Tentang Aplikasi

Tes Faktor	Hasil	Keterangan
Menekan menu pertama yaitu mengenal huruf hijaiyah.	✓	menampilkan halaman yang berhasil mengenali huruf hijaiyah.
Screenshot		
		

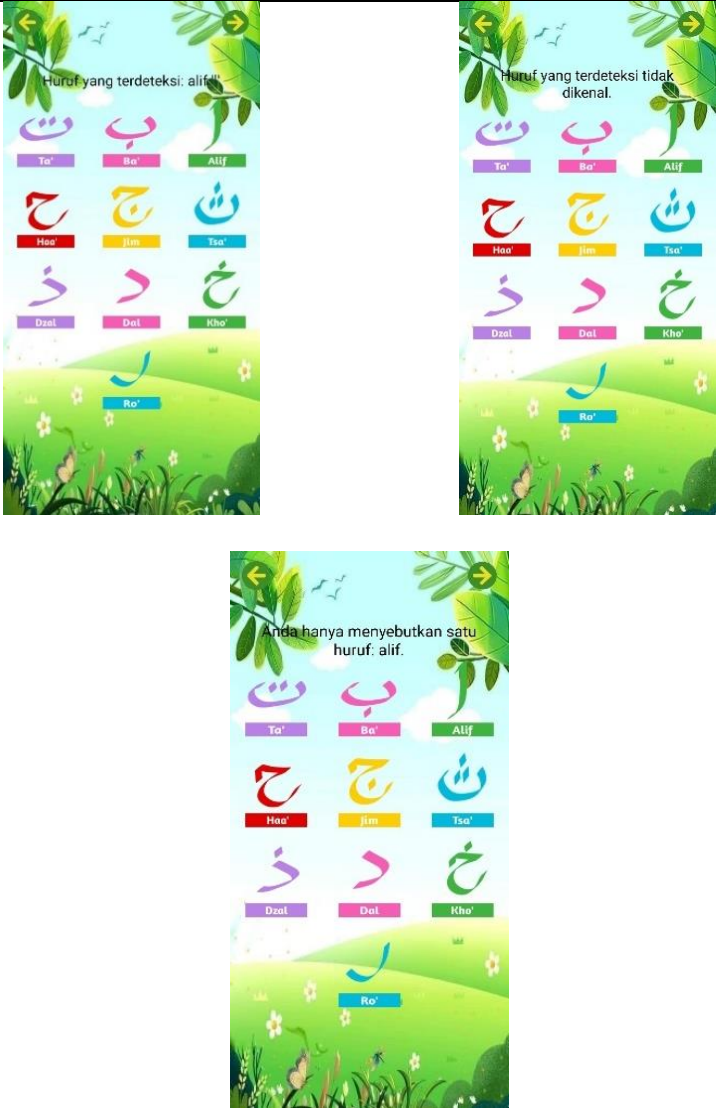
d. Untuk Halaman Menu Penyebutan Huruf Hijaiyah, *Black Box Testing*

Tabel 4. 5 *Black Box Testing* untuk Halaman Halaman Menyebutkan Huruf Hijaiyah

Tes Faktor	Hasil	Keterangan
Pengguna menyebutkan huruf hijaiyah dengan menekan menu.	✓	menampilkan halaman referensi huruf hijaiyah berhasil.
Screenshot		
		

e. *Black Box Testing* Tampilan koreksi Menyebutkan Huruf Hijaiyah

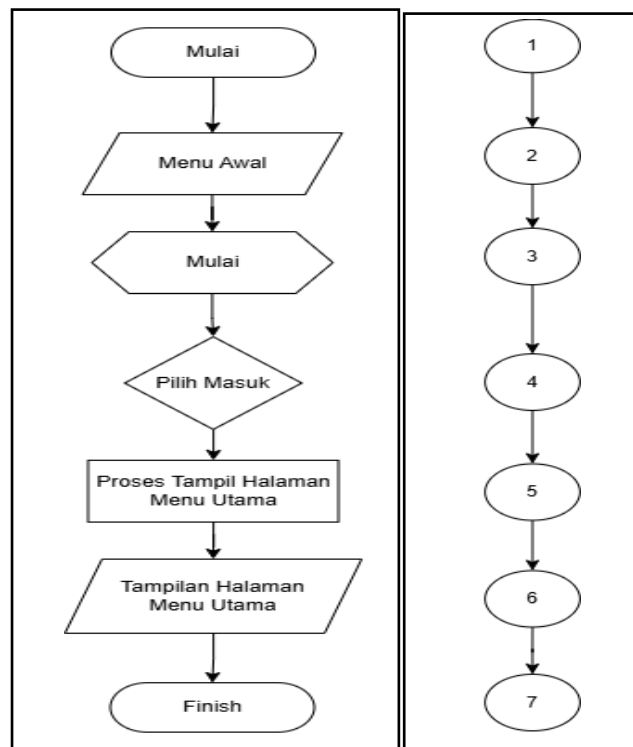
Tabel 4. 6 *Black Box Testing* Ketika Menyebutkan Huruf Hijaiyah

Tes Faktor	Hasil	Keterangan
User menekan salah satu huruf hijaiyah kemudian menyebutkan huruf yang ditekan	✓	Berhasil tampil tampilan koreksi Ketika penyebutan huruf tersebut benar atau salah
Screenshot		
		

5) White Box Testing

Flowchart dan *flowgraph* sistem yang dihasilkan akan ditampilkan pada pengujian ini. Hasil keseluruhan pengujian dengan *white box* pengujian adalah sebagai berikut.

a) *Flowchart* dan *Flowgraph* Menu Awal



Gambar 4.38 *Flowchart* dan *Flowgraph* Menu Awal

Perhitungan dapat dilakukan sebagai berikut setelah meninjau diagram alir dan diagram alir pada menu pertama di atas:

1. Kompleksitas siklomatik $V(G)$ pada tepi dan simpul dihitung.

$$\text{Rumus untuk : } V(G) = E - N + 2$$

Diketahui :

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 7$$

$$P(\text{Predikat node}) = 0$$

Penyelesaian :

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 6 - 7 + 2 \\ &= 1 \end{aligned}$$

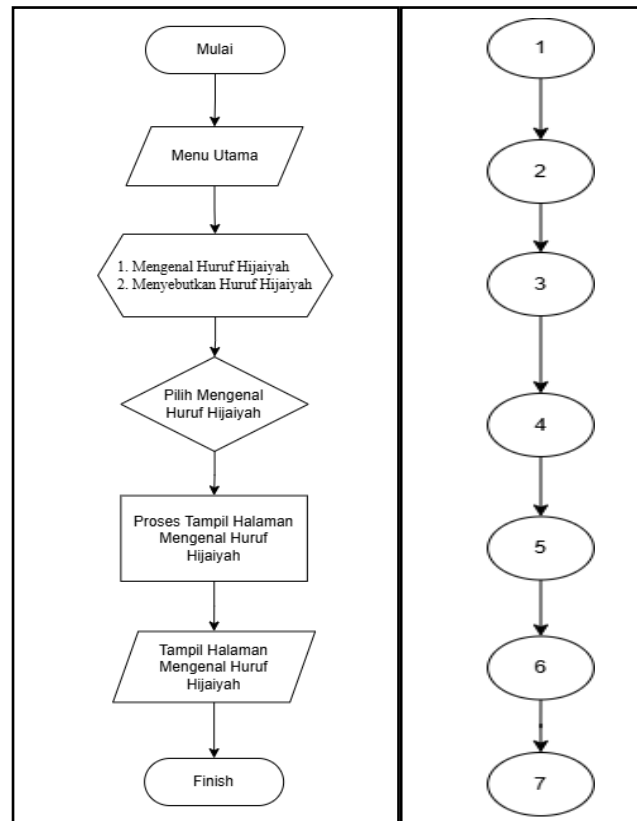
$$\begin{aligned}\text{Predikat (P)} &= P + 1 \\ &= 0 + 1 \\ &= 1\end{aligned}$$

2. Berdasarkan perhitungan diatas siklomatik *flowgraph* menu tampilan awal mempunyai = 1.
3. Jalur independen *flowgraph* ini adalah:
Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7
4. Gambar *matriks* menu menu awal

Tabel 4.7 *Grafik matriks menu menu awal*

[illegible]

b). *Flowchart* dan *Flowgraph* untuk halaman Menu Mengenal Huruf



Gambar 4.39 *Flowchart* dan *Flowgraph* untuk menu Mengenal Huruf

Perhitungannya dapat diselesaikan sebagai berikut setelah meninjau diagram alir dari menu mengenal huruf dan diagram alir di atas:

1. Kompleksitas siklomatik $V(G)$ pada tepi dan simpul dihitung.

Rumus untuk : $V(G) = E - N + 2$

Diketahui :

E (edge) = 6

N (node) = 7

P (Predikat node) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 6 - 7 + 2$$

$$= 1$$

Predikat (P) = P + 1

$$= 0 + 1$$

$$= 1$$

2. Berdasarkan perhitungan kompleksitas siklomatik *flowgraph* dari menu mengenal huruf mempunyai = 1.

3. Jalan independen pada *flowgraph* ini adalah:

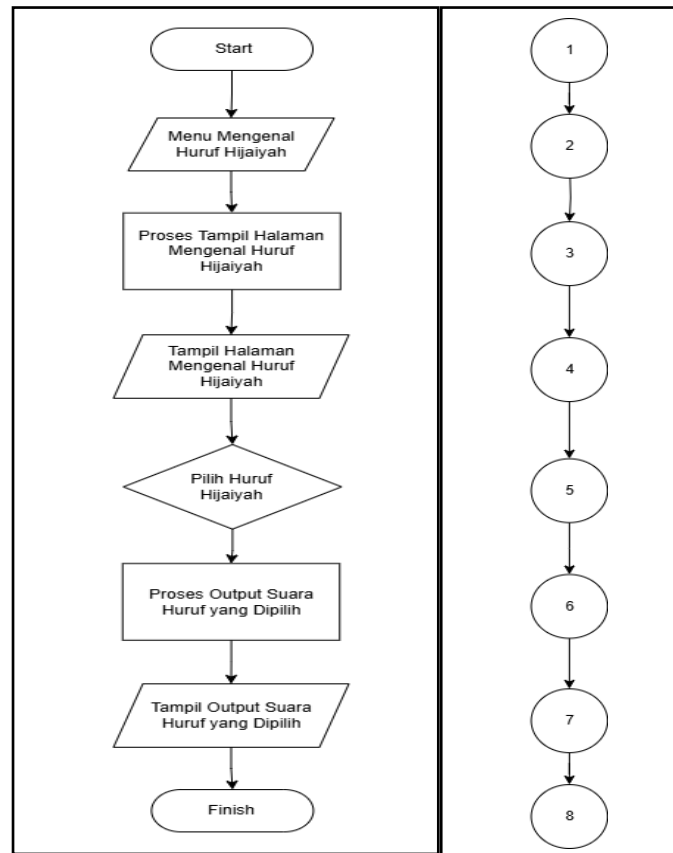
Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7

4. *Grafik matriks* menu mengenal huruf

Tabel 4. 8 *Grafik matriks* menu mengenal huruf

	1	2	3	4	5	6	7	E - 1
1		1						$1 - 1 = 0$
2			1					$1 - 1 = 0$
3				1				$1 - 1 = 0$
4					1			$1 - 1 = 0$
5						1		$1 - 1 = 0$
6							1	$1 - 1 = 0$
7								0
	SUM (E + 1)							$0 + 1 = 1$

c). *Flowchart* dan *Flowgraph* Mendengarkan Penyebutan Huruf Hijaiyah



Gambar 4.40 *Flowchart* dan *Flowgraph* Mendengarkan Penyebutan Huruf

Perhitungannya dapat diselesaikan sebagai berikut setelah meninjau diagram alir pada menu mendengar penyebutan huruf dan diagram alir di atas:

1. Kompleksitas siklomatik $V(G)$ pada tepi dan simpul dihitung.

$$\text{Rumus untuk : } V(G) = E - N + 2$$

Diketahui :

$$E \text{ (edge)} = 7$$

$$N \text{ (node)} = 8$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 7 - 8 + 2 \\ &= 1 \end{aligned}$$

$$\begin{aligned}\text{Predikat (P)} &= P + 1 \\ &= 0 + 1 \\ &= 1\end{aligned}$$

2. Berdasarkan perhitungan kompleksitas siklomatik dari *flowgraph* tampilan mendengarkan huruf mempunyai = 1.

3. Jalan independen pada *flowgraph* ini adalah:

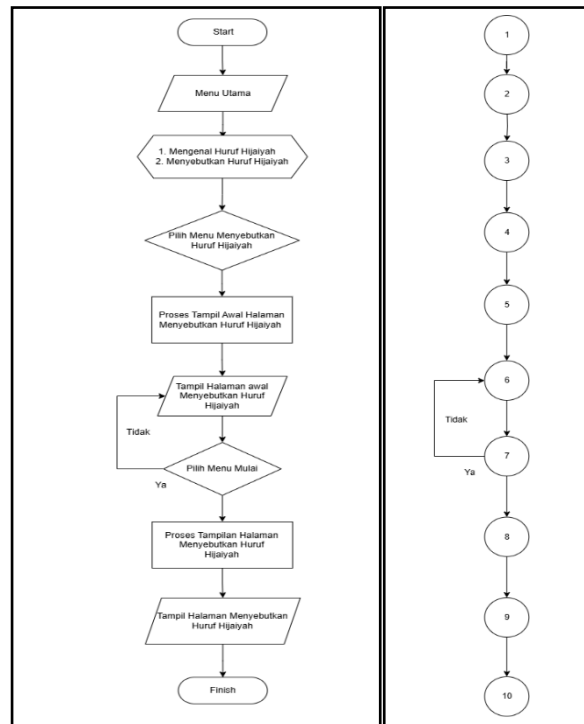
Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

4. *Grafik matriks* alur mendengarkan penyebutan huruf hijaiyah

Tabel 4.9 Grafik matriks tampilan mendengarkan penyebutan huruf hijaiyah

	1	2	3	4	5	6	7	8	E - 1
1		1							$1 - 1 = 0$
2			1						$1 - 1 = 0$
3				1					$1 - 1 = 0$
4					1				$1 - 1 = 1$
5						1			$1 - 1 = 0$
6							1		$1 - 1 = 0$
7								1	$1 - 1 = 0$
8									0
	SUM (E + 1)								$1 + 1 = 2$

d). *Flowchart* dan *Flowgraph* Menu Menyebutkan Huruf Hijaiyah



Gambar 4.41 *Flowchart* dan *Flowgraph* Menu Menyebutkan Huruf Hijaiyah

Perhitungan berikut dapat dilakukan dengan mempelajari diagram alir dan diagram alir menu yang mengacu pada huruf hijaiyah di atas:

1. Kompleksitas siklomatik $V(G)$ pada tepi dan simpul dihitung.

$$\text{Rumus untuk : } V(G) = E - N + 2$$

Diketahui :

$$E (\text{edge}) = 10$$

$$N (\text{node}) = 10$$

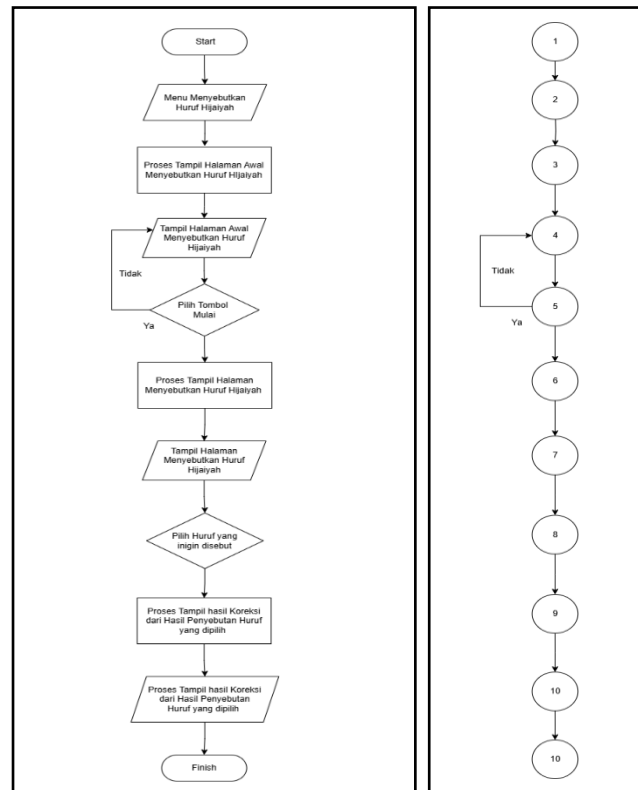
$$P (\text{Predikat node}) = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 10 - 10 + 2$$

e). *Flowchart* dan *Flowgraph* Koreksi Penyebutan Huruf Hijaiyah



Gambar 4.42 *Flowchart* dan *Flowgraph* Koreksi Penyebutan Huruf Hijaiyah

Perhitungan dapat dilakukan sebagai berikut setelah meninjau diagram alir dan diagram alir pada tampilan koreksi huruf hijaiyah tersebut di bawah:

1. Kompleksitas siklomatik $V(G)$ pada tepi dan simpul dihitung.

$$\text{Rumus untuk : } V(G) = E - N + 2$$

Diketahui :

$$E \text{ (edge)} = 11$$

$$N \text{ (node)} = 11$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 11 - 11 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

2. Berdasarkan perhitungan kompleksitas siklomatik *flowgraph* pada tampilan koreksi penyebutan huruf hijaiyah mempunyai = 2.

3. Jalan independen pada *flowgraph* ini adalah:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 4$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11$$

4. *Grafik matriks* alir tampilan koreksi penyebutan huruf hijaiyah

Tabel 4. 11 *Grafik matriks* koreksi penyebutan huruf hijaiyah

	1	2	3	4	5	6	7	8	9	10	11	E - 1
1		1										1 - 1 = 0
2			1									1 - 1 = 0
3				1								1 - 1 = 0
4					1							1 - 1 = 0
5				1		1						2 - 1 = 1
6							1					1 - 1 = 0
7								1				1 - 1 = 0
8									1			1 - 1 = 0
9										1		1 - 1 = 0
10											1	1 - 1 = 0
11												
	SUM (E + 1)											1 + 1 = 2

- b. Pengujian Akurasi Penyebutan Huruf Hijaiyah

Tujuan dari pengujian ini adalah untuk mengukur dan menilai seberapa baik sistem pendeteksi aplikasi dapat memperbaiki kejadian huruf hijaiyah yang salah disebutkan.

Tabel 4. 12 Akurasi Penyebutan Huruf Hijaiyah

Huruf	terdeteksi	Tidak terdeteksi
Alif (‘ا’)	10	0
Ba (‘ب’)	4	6
Ta (‘ت’)	3	7
Tsa (‘ث’)	6	4
Jim (‘ج’)	10	0
Haa (‘ح’)	6	4
Kho (‘خ’)	2	8
Dal (‘د’)	10	0
Dzal (‘ذ’)	10	0
Ro (‘ر’)	5	5
Zay (‘ز’)	6	4
Sin (‘س’)	7	3
Syin (‘ش’)	7	3
Shod (‘ص’)	3	7
Dhod (‘ض’)	0	10
Tho (‘ط’)	4	6
Dzho (‘ظ’)	0	10
Ain (‘ع’)	7	3
Ghoin (‘غ’)	8	2
Fa (‘ف’)	5	5
Qof (‘ق’)	3	7
Kaf (‘ك’)	5	5
Lam (‘ل’)	9	1
Mim (‘م’)	6	4
Nun (‘ن’)	8	2
Waw (‘و’)	10	0
Ha (‘ه’)	8	2
Lam Alif (‘لا’)	6	4
Hamzah (‘ء’)	9	1
Ya (‘ي’)	7	3
Jumlah	184	116

a. Rata-rata terdeteksi

$$\text{Rata-rata akurasi} = \frac{184}{300} \times 100\% = 61\%$$

b. Rata-rata tidak terdeteksi

$$\text{Rata-rata akurasi} = \frac{120}{300} \times 100\% = 39\%$$

Berdasarkan hasil pengujian, sistem pengenalan pengucapan huruf hijaiyah menunjukkan tingkat akurasi rata-rata sebesar 61% untuk huruf yang terdeteksi dengan benar, sedangkan 39% dari data uji tidak berhasil terdeteksi. Hal ini menunjukkan bahwa sistem memiliki performa yang cukup baik, tetapi masih memerlukan penyempurnaan untuk meningkatkan akurasi, terutama dalam mendeteksi pengucapan yang belum sesuai.

Huruf dengan akurasi tinggi seperti Alif (‘ا’), Jim (‘ج’), Dal (‘د’), Dzal (‘ذ’), dan Waw (‘و’) menunjukkan bahwa model pengenalan suara bekerja sangat baik pada huruf-huruf ini. Beberapa alasan yang mungkin adalah:

1) Sifat Fonem yang Jelas dan Khas:

- a) Huruf seperti Alif (‘ا’) dan Waw (‘و’) memiliki fonem yang jelas dan khas sehingga lebih mudah dikenali oleh sistem.
- b) Tidak banyak variasi dalam cara pengucapan huruf-huruf ini di berbagai dialek atau aksen.

2) Data Latihan yang Cukup:

Mungkin huruf-huruf ini lebih sering muncul dalam dataset pelatihan, sehingga model lebih terlatih untuk mengenalinya.

3) Tidak Banyak Kemiripan dengan Huruf Lain:

Huruf seperti Dal (‘د’) dan Jim (‘ج’) memiliki suara yang unik dan tidak banyak tumpang tindih dengan huruf lain dalam bahasa Arab.

4) Kesederhanaan Fonetik:

Huruf dengan fonetik sederhana atau tanpa unsur kompleks seperti tekanan (stress) atau aspirasi lebih mudah dikenali oleh algoritma pengenalan suara.

Huruf dengan akurasi rendah seperti Dhod (‘ض’) dan Dzho (‘ظ’) mengindikasikan kesulitan model dalam mengenali suara huruf-huruf ini. Penyebabnya bisa meliputi:

1) Fonem yang Mirip dengan Huruf Lain:

Dhod (‘ض’) dan Dzho (‘ظ’) memiliki bunyi yang sering kali mirip dengan huruf lain seperti Shod (‘ص’) atau Zay (‘ز’). Hal ini menyebabkan kebingungan dalam pengenalan.

2) Fonem Kompleks atau Tidak Umum:

Huruf seperti Dzho (‘ظ’) memiliki fonem yang lebih kompleks atau jarang digunakan dalam percakapan sehari-hari, sehingga model mungkin tidak cukup terlatih untuk mengenalinya.

3) Variasi dalam Pengucapan:

Beberapa huruf seperti Dhod (‘ض’) sering kali diucapkan dengan sedikit variasi dalam dialek yang berbeda, misalnya antara dialek Arab klasik dan dialek modern.

4) Kurangnya Data Latihan:

Jika dataset pelatihan memiliki jumlah sampel yang tidak merata, huruf-huruf seperti Dhod (‘ض’) dan Dzho (‘ظ’) mungkin memiliki lebih sedikit sampel dibandingkan huruf-huruf lain, yang menyebabkan performa pengenalan yang buruk.

5) Kelemahan dalam Algoritma Model:

Algoritma yang digunakan mungkin tidak cukup baik dalam menangani suara dengan nada rendah atau fitur akustik kompleks yang khas dari huruf tertentu.

BAB V

PENUTUP

A. Kesimpulan

Penulis dapat membuat kesimpulan berikut berdasarkan hasil dari penelitian ini :

1. Bahasa pemrograman Java dan Android Studio digunakan hingga berhasil membuat aplikasi Koreksi Huruf Hijaiyah. Aplikasi ini dilengkapi dengan fitur koreksi untuk membantu pengguna dalam belajar dan mengoreksi pengucapan huruf hijaiyah secara akurat. Fitur-fitur yang ada dalam aplikasi ini antara lain pengenalan huruf hijaiyah, cara pelafalan yang benar, koreksi dalam pengucapan, dan analisis suara untuk memberikan umpan balik yang tepat.
2. Fitur Pendeteksi Kesalahan Pengucapan Huruf Hijaiyah berhasil dibangun menggunakan teknologi pengenalan suara berbasis pembelajaran mesin. Model ini dilatih dengan menggunakan *speech recognizer* dan kemudian suara diubah menjadi bentuk teks, yang diintegrasikan dalam aplikasi Android. Teknologi ini memungkinkan aplikasi untuk mengidentifikasi dan memberi umpan balik mengenai kesalahan dalam pengucapan huruf hijaiyah.
3. Untuk memastikan bahwa organisasi internal kode dan logika aplikasi berfungsi sebagaimana mestinya, pengujian *blackbox* dan *whitebox* dilakukan. Seluruh fungsi dan tombol dalam aplikasi dapat beroperasi dengan baik tanpa adanya kesalahan (*error*).

4. Pengujian Akurasi Aplikasi menunjukkan tingkat akurasi rata-rata sebesar 60% untuk huruf yang terdeteksi dengan benar, sedangkan 40% dari data uji tidak berhasil terdeteksi. Dimana Model pengenalan suara menunjukkan performa yang sangat baik dalam mengenali huruf-huruf dengan akurasi tinggi seperti Alif ('ا'), Jim ('ج'), Dal ('د'), Dzal ('ذ'), dan Waw ('و'). Hal ini disebabkan oleh fonem yang jelas dan khas, tidak adanya variasi pengucapan yang signifikan, data latihan yang cukup, suara yang unik tanpa kemiripan dengan huruf lain, serta kesederhanaan fonetik huruf-huruf tersebut. Sebaliknya, huruf-huruf dengan akurasi rendah seperti Dhod ('ض') dan Dzho ('ظ') menghadapi tantangan dalam pengenalan karena fonem yang mirip dengan huruf lain, kompleksitas fonetik, variasi pengucapan, kurangnya data latihan, dan keterbatasan algoritma dalam menangani suara yang kompleks atau kurang umum. Faktor-faktor ini menunjukkan bahwa pengembangan dataset dan algoritma yang lebih baik diperlukan untuk meningkatkan akurasi pengenalan huruf-huruf ini. Hal ini menunjukkan bahwa sistem memiliki performa yang cukup baik, tetapi masih memerlukan penyempurnaan untuk meningkatkan akurasi, terutama dalam mendeteksi pengucapan yang belum sesuai.

Saran

Penulis penelitian ini mengakui bahwa masih banyak permasalahan yang memerlukan pengembangan dan koreksi pada penelitian selanjutnya. Oleh karena itu, penulis menawarkan rekomendasi sebagai berikut untuk mengembangkan program perbaikan penyebutan huruf hijaiyah dalam menyikapi:

1. Penggunaan *Database*: Aplikasi ini disarankan untuk menggunakan database yang lebih terstruktur guna memperkaya informasi mengenai huruf hijaiyah dan pelafalannya. Dengan menggunakan *database*, aplikasi dapat mengurangi volume kode program serta mengoptimalkan penggunaan penyimpanan aplikasi, sehingga mempermudah pembaruan dan pengelolaan data.
2. Peningkatan Akurasi Deteksi Suara: Dalam pengujian pengucapan huruf hijaiyah, beberapa huruf menunjukkan tingkat akurasi yang lebih rendah. Oleh karena itu, disarankan untuk melakukan peningkatan pada model pembelajaran mesin yang digunakan, khususnya dalam mengatasi variasi suara pengguna dari berbagai aksen atau kecepatan berbicara.

DAFTAR PUSTAKA

- Akbar, A., Husodo, A. Y., Zubaidi, A., Studi, P., Informatika, T., Teknik, F., & Mataram, U. (2019). Implementasi Google Speech API pada Aplikasi Koreksi Hafalan Al-Qur ' an Berbasis Android (The Implementation of the Google Speech on Qur ' an Recitation Correction. *Jtika*, 1(1), 1–8.
- Assisi, M., Septiarini, A., Kridalaksana, A. H., & Wati, M. (2022). Rancang Bangun Aplikasi Hafalan Al-Quran dengan Google Speech API Berbasis Android. *Jurnal Rekayasa Teknologi Informasi (JURTI)*, 6(1), 26. <https://doi.org/10.30872/jurti.v6i1.8006>
- Fadhilla, A., SUheri, & Khaliq, A. (2021). Aplikasi Mobile Learning Berbasis Android Untuk Belajar Huruf Hijaiyah. *Jurnal Nasional Teknologi Komputer*, 1(1), 28–36. <https://doi.org/10.61306/jnastek.v1i1.5>
- Hastuty, A., Muh, B., & Amir, A. (2021). Sistem Pengenalan Ucapan Bahasa Daerah Menggunakan Metode Mel Frequency Cepstral Coefficient (MFCC) dan Adaptive Neuro-Fuzzy Inference System (ANFIS). *Jurnal Sintaks Logika*, 1(2), 76–81. <https://doi.org/10.31850/jsilog.v1i2.812>
- Hudayana, N. A., Haris, A., & Zakaria, A. R. (2024). *Pengembangan Aplikasi Penyimak Al- Qur ' an Menggunakan Teknologi AI dengan Metode Speech Recognition pada Platform Android*. 2(November), 83–93.
- Kurniawati, K., & Pawelloi, A. I. (2023). Aplikasi kalkulator menggunakan suara Berbasis android. *Jurnal Sintaks Logika*, 3(3), 24–28. <https://doi.org/10.31850/jsilog.v3i3.2584>
- Malia, R. (2022). Aplikasi Education Pembelajaran Al-Islam Kemuhammadiyaan Berbasis Android. *Jurnal Sintaks Logika*, 2(2), 61–70. <https://doi.org/10.31850/jsilog.v2i2.1062>
- Manurung, M. K. A. A. (2021). Implementasi Google Speech API Pada Aplikasi Tahfidz Muroja'ah Hafalan Al-Qur'an Menggunakan Algoritma Rabin Karp Berbasis Android (Doctoral dissertation, Universitas Islam Negeri Sumatera Utara).
- Mauliddiyah, N. L. (2021). Penerapan Speech Recognition pada Aplikasi Android Penentuan Level Pembelajaran Yanbu'a Jilid 1 Untuk Anak Usia Dini Menggunakan Google Speech API. 6. [https://sipora.polije.ac.id/6053/3/Daftar Pustaka.pdf](https://sipora.polije.ac.id/6053/3/Daftar%20Pustaka.pdf)

- Muqsith, F. I., Supriyati, E., & Listyorini, T. (2025). Klasifikasi Pengucapan Huruf Hijaiyah Berbasis Android Menggunakan CNN dengan Fitur Mel-Spectrogram. *10*(1), 67–78. <https://doi.org/10.30591/jpit.v10i1.8145>
- Pangestu, A. R., & Purwanto, A. (2021). Aplikasi Pembelajaran Huruf Hijaiyah dan Iqro untuk Anak Berbasis Android . *01*(01), 16–21.
- Prtama, R. W., & Ulum, M. B. (2022). Aplikasi Pembelajaran Huruf Hijaiyah Dan Game Edukasi Berbasis Android. Prosiding Seminar Nasional ..., 919–938. <https://conference.upnvj.ac.id/index.php/senamika/article/view/2283%0Ah>
<https://conference.upnvj.ac.id/index.php/senamika/article/download/2283/1723>
- Raya, R., Supianto, A. A., & Furqon, M. T. (2021). Pengembangan Aplikasi Pendeteksi Kesalahan Dalam Membaca Al-Qur'an Berbasis Android Menggunakan Speech Recognition Dengan Menerapkan Metode Jaro Winkler Distance. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Kompute*, 5(8), 3488–3497. <http://j-ptiik.ub.ac.id>
- Ridho Pangestu., & Agung Purwanto. (2021). Aplikasi Pembelajaran Huruf Hijaiyah dan Iqro untuk Anak Berbasis Android. *Sistem Informasi*, Universitas Darwan Ali.
- Septiarini, A., Kridalaksana, A. H., & Wati, M. (2022). Rancang Bangun Aplikasi Hafalan Al-Quran dengan Google Speech API Berbasis Androi. *Jurnal Rekayasa Teknologi Informasi (JURTI)*, 6(1), 26-35.
- Wahyuddin, W., & Hasnawati, H. (2023). Aplikasi Media Pembelajaran Pengenalan Pancasila Menggunakan Augmented Reality. *Jurnal Sintaks Logika*, 3(3), 8–15. <https://doi.org/10.31850/jsilog.v3i3.2579>
- Wahyu Pratama., & Bahrul Ulum. (2022). *Aplikasi Pembelajaran Huruf Hijaiyah Dan Game Edukasi Berbasis Android*. Teknik Informatika, Fakultas Ilmu Komputer-Universitas Esa Unggul.
- Widya Khafa Nofaa, Dewi Anggraini Puspa Hapsari, & Dinda Salsabila Putri. (2023). *Aplikasi Pembelajaran Huruf Hijaiyah Berbasis Android*. *Jurnal Ilmiah Teknik*, 2(1), 11–19. <https://doi.org/10.56127/juit.v2i1.473>
- Wisdhani, Z., Ontowirjo, A. H. J., & Kambey, F. D. (2021). Asmaul Husna Learning Application Using The Speech recognition Aplikasi Pembelajaran Asmaul Husna Menggunakan Speech recognition. <https://ejournal.unsrat.ac.id/index.php/informatika>

Zubaidi, A., Akbar, A., & Husodo, A. Y. (2019). Implementasi Google Speech API Pada Aplikasi Koreksi Hapalan Al-Qur'an Berbasis Android. Jurnal Teknologi Informasi, Komputer, Dan Aplikasinya (JTIKA), 1(1), 1-8.