

| No | Nama penulis | Tahun | Judul | Metode | Hasil |
|----|--------------|-------|---|--|--|
| | | | | <i>Mixture Model</i> (GMM). | |
| 4. | Nuzul Hikmah | 2018 | Pemanfaatan <i>Text Mining</i> dalam Pencarian Ayat Al Quran menggunakan TF-IDF dan Cosine Similarity | Metode pembobotan TF-IDF dan Cosine Similarity | Pemanfaatan <i>text mining</i> dalam pencarian ayat Al Quran dapat digunakan oleh semua umat muslim untuk mencari topik yang terdapat di dalam AlQuran |

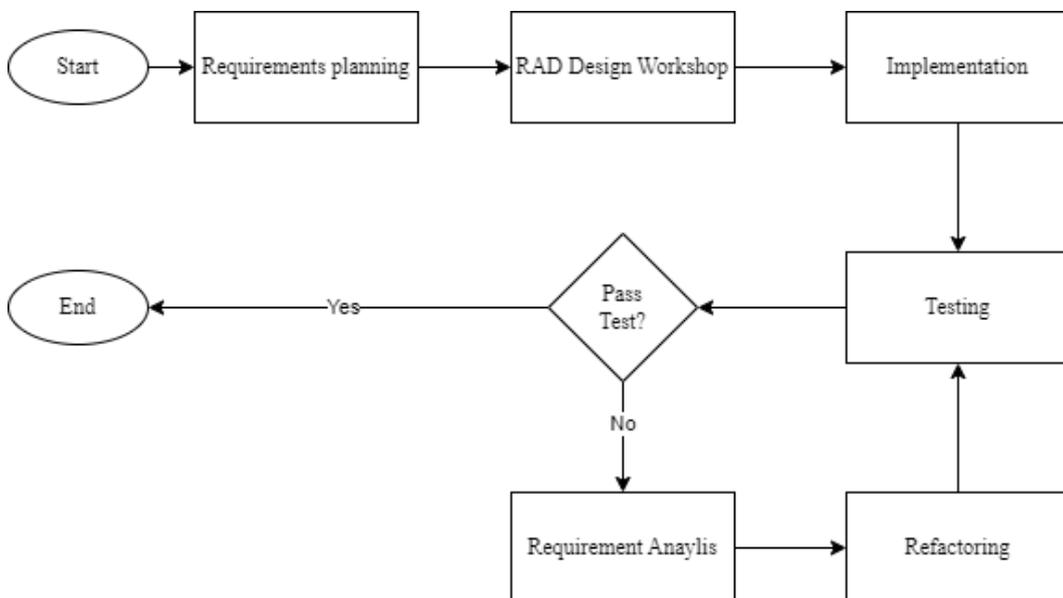
Berdasarkan Tabel 2.5 aplikasi yang akan dibuat sebagai sistem pencari ayat Al-Qur'an yang melibatkan tiga jenis input, yaitu suara dengan menggunakan teknologi *Cloud Speech-To-Text* dari *Google Cloud Platform*, tulisan dengan menggunakan teknologi *Full-Text Search* dengan *Elasticlunr*, serta gambar dengan menggunakan teknologi *Text Recognition* berbasis *Optical Character Recognition* dari *Google Cloud Platform*. Hal ini diharapkan akan memudahkan pengguna dalam mencari dan mempelajari ayat-ayat Al-Qur'an, serta melengkapi keterbatasan dari *platform* maupun penelitian yang sudah ada.

BAB III METODE PENELITIAN

3.1 Tahapan Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah metode *Research and Development* (R&D). Metode penelitian dan pengembangan merupakan metode penelitian yang digunakan untuk menghasilkan produk tertentu, serta menguji keefektifan produk tersebut (Amile & Reesnes, 2015). Sedangkan, Sukmadinata (2009) mendefinisikan penelitian R&D merupakan suatu proses atau langkah-langkah untuk mengembangkan suatu produk baru atau menyempurnakan produk yang telah ada dan dapat dipertanggungjawabkan.

Berdasarkan penjelasan terkait metode R&D adalah metode penelitian yang digunakan untuk menghasilkan dan menyempurnakan produk tertentu yang sesuai dengan acuan dan kriteria, sehingga menghasilkan suatu produk dengan melalui tahapan proses dan validasi. Oleh karena itu, dalam penelitian ini penulis menggunakan jenis metode R&D berupa metode pengembangan aplikasi cepat (*Rapid Application Development/RAD*). Gambar 3.1 memperlihatkan tahapan penelitian yang dilakukan.



Gambar 3.1 Diagram alir penelitian

Berdasarkan Gambar 3.1 merupakan diagram alir penelitian dilakukan dengan beberapa tahapan proses untuk mengumpulkan sejumlah data yang dibutuhkan selanjutnya dilakukan pengembangan sistem, pengujian dan evaluasi

terhadap sistem yang dibuat. Dalam penelitian ini, tahapan pengembangan sistem yang dilakukan adalah sebagai berikut.

a. Analisis Kebutuhan (*Requirements planning*)

Pada tahapan ini penulis bertujuan membangun aplikasi pencari ayat Al-Qur'an untuk memenuhi kebutuhan masyarakat khususnya penyandang disabilitas tuna netra, kemudian aplikasi dapat digunakan untuk mencari potongan ayat Al-Qur'an dan maknanya. Pada aplikasi ini terdapat fitur pencari ayat Al-Qur'an menggunakan suara menggunakan *Speech recognition Cloud Speech to Text*.

b. Pengembangan Aplikasi Cepat (*RAD Design Workshop*)

Setelah mengetahui tahapan analisis kebutuhan, penulis melakukan desain perancangan dengan membuat pemodelan terhadap aplikasi. Desain yang dimaksud meliputi perancangan proses aplikasi dan desain antarmuka. Pada perancangan sistem, penulis menggunakan *flowchart*, diagram UML dan *storyboard*.

c. Implementasi

Pada tahapan ini akan dilakukan implementasi dari aplikasi yang terdiri dari sebagai berikut.

1. Membangun aplikasi, dalam tahapan ini dirancang menggunakan bahasa pemrograman *typescript* dan teknologi AI.
2. Menguji aplikasi, pada tahapan ini penulis melakukan *testing* terhadap keseluruhan aplikasi yang dikembangkan untuk mengetahui keberhasilan dari aplikasi yang dibangun.

d. Pengujian

Tahapan ini dilakukan setelah aplikasi selesai dibuat dengan melakukan 2 kategori pengujian, yaitu:

1. Pengujian kelayakan; dan
2. Performa dan akurasi.

Apabila aplikasi tidak memenuhi dua kriteria pengujian, maka dilakukan tahap *refactoring*.

3.2 Analisis Kebutuhan

Pada tahapan analisis kebutuhan ini terdiri dari analisa kebutuhan sistem, analisis kebutuhan perangkat lunak dan perangkat keras. Analisis kebutuhan sistem

merupakan tahapan yang bertujuan untuk mengetahui kebutuhan-kebutuhan yang diperlukan oleh sistem. Sistem yang dianalisis merupakan sistem yang berisi informasi tentang segala sesuatu yang berkaitan dengan pembuatan sistem tersebut. Hal ini menyebabkan tahapan yang paling penting dalam proses pembuatan sistem ini. Adapun kebutuhan sistem yang akan digunakan pada penelitian ini adalah sebagai berikut.

1. Sistem dapat melakukan sistesis suara dalam bahasa Arab menjadi transkripsi huruf hijaiyah;
2. Sistem dapat melakukan pencarian ayat Al-Quran; dan

Selain itu, terdapat kebutuhan input dari sistem informasi yang digunakan dalam pengembangan aplikasi ini adalah sebagai berikut.

1. Data informasi ayat dan surah Al-Qur'an, dan data makna dan tafsir terjemahan dari ayat Al-Qur'an.
2. Aplikasi dapat mendeteksi suara pengguna.
3. Perangkat bisa digunakan mengetik teks untuk melakukan pencarian ayat Al-Qur'an.

Adapun *output* dari sistem informasi dapat berupa tampilan teks ayat, makna dan tafsir dari ayat Al-Qur'an adalah sebagai berikut.

1. Aplikasi dapat menerjemahkan masukan suara bahasa Arab ke dalam transkrip huruf hijaiyah.
2. Aplikasi dapat mencari ayat Al-Quran berdasarkan kalimat dalam huruf hijaiyah, transliterasi latin, juga terjemahan.

3.3 Alat dan Bahan

Penelitian ini memerlukan peralatan dan bahan yang terbagi menjadi tiga kategori yaitu perangkat keras, lunak dan *services*. Perangkat keras yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.1. Sedangkan perangkat lunak yang digunakan dapat dilihat pada Tabel 3.2. Kemudian, jenis *services* yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.3.

Tabel 3.1 Spesifikasi perangkat keras

| Aspek | Spesifikasi |
|--------------|-------------------------|
| Processor | AMD Ryzen 7 5700X |
| Graphic Card | NVIDIA GeForce RTX 3070 |

| | |
|------------------------------|--|
| Random Access Memory | 32GB DDR4 3200MHz |
| Storage | 1TB SSD M.2 NVMe PCIe 4.0 |
| Motherboard | Asrock B550M Pro4 |
| Handphone Realme (GT Master) | Qualcomm Snapdragon 778G, Random Access Memory 8GB |

Tabel 3.2 Spesifikasi perangkat lunak

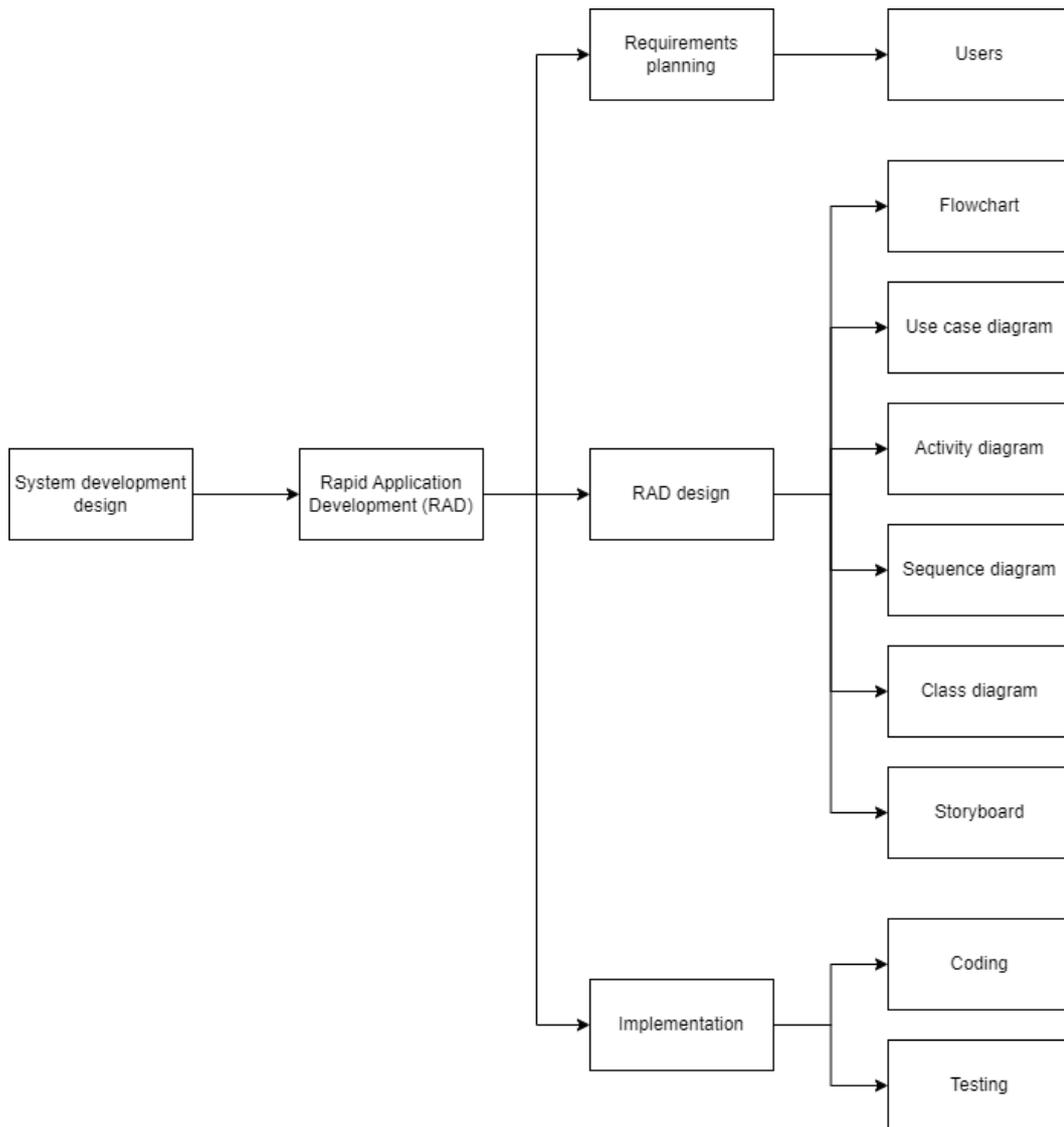
| Nama | Kegunaan |
|--------------------|---|
| Windows 11 Pro | <i>Operating System (OS)</i> |
| Visual Studio Code | <i>Code editor</i> |
| Nodejs | <i>Javascript runtime</i> |
| Microsoft Edge | <i>Web browser</i> |
| Git | <i>Version control</i> |
| Javascript | Bahasa pemrograman web |
| React Native | Kompilasi javascript untuk aplikasi <i>mobile</i> |

Tabel 3.3 *Services* dan kegunaannya

| Nama | Kegunaan |
|-------------------|---------------------------------------|
| NeuralSpace Voice | <i>Speech recognition</i> bahasa Arab |
| Typesense | <i>Full-text search</i> |
| Expo | Deployment aplikasi Android |
| Github | <i>Repository git</i> |

3.4 Tahapan Perancangan Desain

Tahapan perancangan desain merupakan serangkaian langkah interaktif yang memungkinkan desain yang menggambarkan semua aspek perangkat lunak yang dibangun. Pada tahapan ini terdiri dari pembuatan *flowchart*, pembuatan model, *use case*, dan *activity diagram*. Adapun *flowchart* pada penelitian ini dapat dilihat dari Gambar 3.2.



Gambar 3.2 Perancangan desain

Sumber : Rolly & Hakiem (2015)

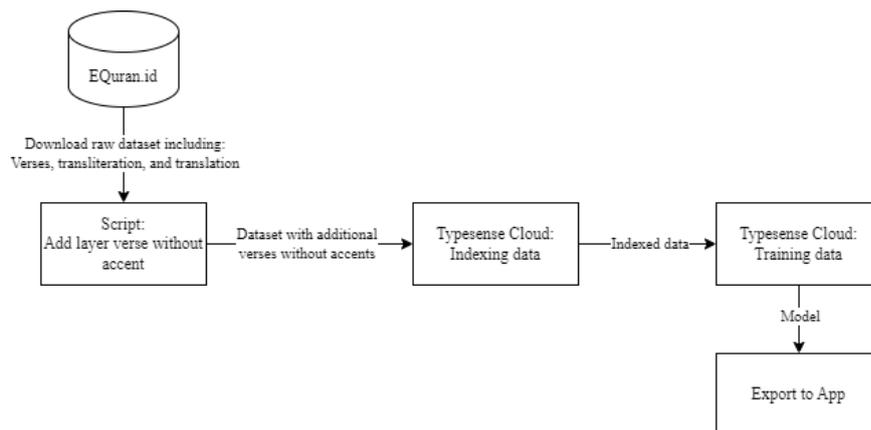
Tahapan pertama adalah perencanaan, di mana proyek direncanakan dan dijadwalkan. Ini melibatkan kegiatan-kegiatan seperti menentukan ruang lingkup proyek, mengidentifikasi sumber daya yang dibutuhkan, dan membuat timeline. Tahapan selanjutnya adalah desain, di mana sistem dirancang. Tahap ini menggunakan berbagai alat dan teknik desain seperti:

1. *Use case diagram*: Diagram ini menunjukkan interaksi antara pengguna dan sistem.
2. *Flowchart*: Diagram ini menunjukkan aliran data melalui sistem.

3. *Activity diagram*: Diagram ini menunjukkan aktivitas yang terlibat dalam suatu proses.
4. *Class diagram*: Diagram ini menunjukkan kelas dan objek dalam sistem.
5. *Storyboard*: Ini adalah representasi visual dari alur pengguna melalui sistem.

Tahapan ketiga adalah Implementasi, di mana sistem dikodekan menggunakan bahasa pemrograman *Typescript*. Tahapan keempat adalah pengujian, di mana sistem diuji untuk mengidentifikasi dan memperbaiki kesalahan atau bug.

Selain itu, pada tahapan perancangan desain dilakukan pembuatan model sistem, yang bertujuan untuk mengetahui proses kerja sistem yang akan dibuat. Model sistem dapat dilihat pada Gambar 3.3.



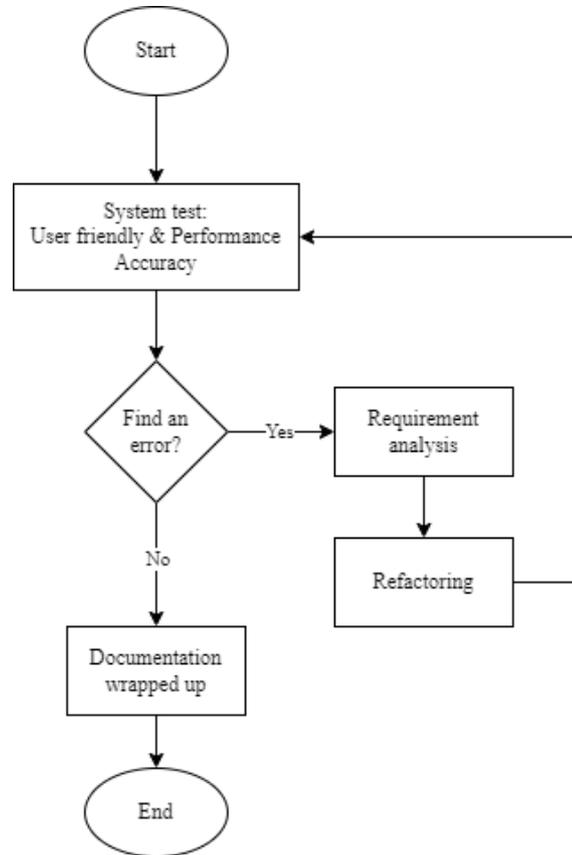
Gambar 3.3 Diagram model

Pada tahapan pembuatan model, dataset akan diunduh dari sumber EQuran.id. Lalu dataset diolah lebih lanjut dengan menambahkan layer ayat tanpa harakat. Kemudian dataset diupload ke *Typesense Cloud* untuk dilakukan *Indexing* dan *Training* yang akan menghasilkan model yang siap digunakan oleh aplikasi.

3.5 Pengujian

Pada tahapan pengujian ini mengadopsi salah satu fase utama dari *metode-D* yaitu *system Test & Fix*. Secara umum fase *system Test & Fix* menjelaskan terkait proses dan hasil pengujian sistem yang telah diimplementasikan sebelumnya. Tujuan dari *system Test & Fix* untuk mengidentifikasi adanya *bug* atau *error* pada sistem serta untuk melakukan peninjauan kembali terkait kesesuaian fungsi yang

ada pada sistem dengan diharapkan oleh pengguna sebelum penulis melakukan perilisan pada aplikasi ini. Proses pengujian dengan menggunakan *system Test & Fix* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Proses pengujian

System Test (pengujian sistem) merupakan tahapan untuk mengetahui hasil dari implementasi sistem uji. *Error* yang ditemukan pada saat pengujian kemudian dituliskan pada daftar kebutuhan untuk tujuan iterasi perbaikan. Pengujian ini mengadopsi tahap *system test*, *documentation wrap-up*, dan *release day*. Pengujian sistem aplikasi akan menggunakan metode *black-box testing*, dikarenakan yang akan diuji pada penelitian ini adalah kelayakan dari aplikasi yang dikembangkan (Simarmata, 2010; Nurandini, 2023). Pada aplikasi pencari ayat Al-Qur'an akan dilakukan juga pengujian *performance accuracy*. Pengujian ini bertujuan untuk mengetahui bagaimana *performance* aplikasi dan sampai sistem operasi Android berupa aplikasi pencari ayat Al-Qur'an berbasis Android.

3.5.1 Kelayakan (*user friendly*)

Pengujian dengan *black-box testing* dapat dilakukan dengan membuat kasus uji berdasarkan kelayakan perangkat lunak dengan menggunakan aplikasi secara langsung dan mencoba semua fungsi apakah telah meliputi *test case*, skenario pengujian, hasil yang diharapkan dan keterangan format pengujian dapat dilihat pada Lampiran 1. Adapun Tabel hasil pengujian *black-box testing* dapat dilihat pada Tabel 3.4.

Tabel 3.4 *Black-box testing*

| No | Test Case | Sesuai | Tidak Sesuai |
|----|-----------------|--|--|
| 1 | Case yang diuji | Jumlah responden yang menjawab sesuai antara <i>test case</i> dengan hasil yang diharapkan | Jumlah responden yang menjawab tidak sesuai antara <i>test case</i> dengan hasil yang diharapkan |

Format pengujian diatas merupakan skenario yang akan dijalankan dengan menguji kelayakan sistem. Pengujian aspek kelayakan dilakukan dengan melakukan tes pada setiap fungsi perangkat lunak oleh responden. Instrumen pengujian menggunakan *test case* dengan format skala Guttman. Skala pengukuran tipe ini akan didapat jawaban yang tegas yaitu ‘sesuai’ yang ditandai dengan angka 1 (satu) dan ‘tidak sesuai’ yang ditandai dengan angka 0 (nol), untuk mengetahui kelayakan perangkat lunak dari sisi *functionality*, pada penelitian ini menggunakan interpretasi standar yang ditetapkan oleh ISO 9126. Rumus analisis data yang digunakan adalah sebagai berikut.

$$X = 1 - \frac{A}{B} \dots \dots \dots (3.1)$$

Keterangan:

X = Tingkat fungsionalitas

A = Jumlah fungsi yang gagal

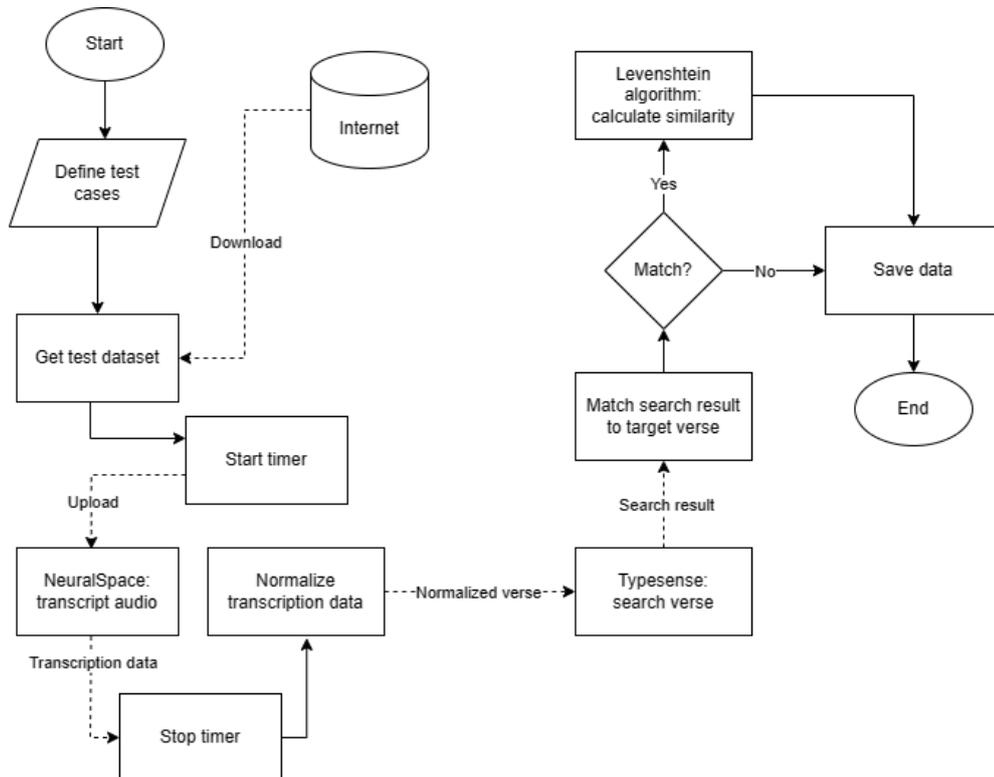
B = Jumlah keseluruhan fungsi yang diuji

Berdasarkan rumus 3.1 tingkat fungsionalitas aplikasi merujuk pada sejauh mana suatu aplikasi dapat melakukan tugas dan fungsi yang telah ditentukan. Fungsionalitas ini mencakup fitur-fitur dan kemampuan yang dimiliki oleh aplikasi untuk memenuhi kebutuhan pengguna atau tujuan tertentu. Nilai ini bervariasi antara 0-1, di mana nilai 1 menunjukkan bahwa aplikasi berjalan sesuai dengan

skenario pengujian secara sempurna, sedangkan nilai 0 menunjukkan bahwa aplikasi sama sekali tidak berjalan sesuai skenario pengujian (Zahara, 2017).

3.5.2 Performa & Akurasi

Pengujian aspek performa & akurasi dilakukan untuk menguji kinerja aplikasi pada sumber daya yang digunakan dalam kondisi tertentu (Nurandini, 2023). Pengujian performa & akurasi diantaranya dengan menguji transkripsi dan *search* dengan menggunakan Node.js. Menurut McCall dalam Roger S. Pressman (2010), *performance* merupakan jumlah sumber daya komputasi dan kode yang diperlukan program untuk mampu melaksanakan fungsinya secara baik dan benar. Gambar 3.5 menunjukkan proses pengujian performa & akurasi, langkah pertama yang dilakukan adalah melakukan pengkategorian pengujian berdasarkan surah terpendek yang dilakukan dengan surah Al-Fatihah, Al-Ikhlash, Al-Falaq, dan An-Nas. Selanjutnya, dilakukan pengujian pada surah dengan kategori *medium* (Al-Kahfi), kemudian pengujian performa & akurasi pada surah terpanjang yaitu Al-Baqarah.



Gambar 3.5 Alur pengujian performa & akurasi

3.5.3 Hubungan Antara Panjang Ayat dengan Performa dan Akurasi

Adapun metode yang digunakan untuk menganalisis hubungan antara panjang ayat dengan performa & akurasi adalah salah satu library Python, yaitu numpy dengan rumus sebagai berikut:

$$Y = a + bX \dots\dots\dots (3.2)$$

Keterangan :

Y = Variabel terikat (performa & akurasi)

X = Variabel bebas (panjang ayat)

a = Intersep (waktu yang dibutuhkan sistem untuk memulai)

b = Koefisien regresi (besar pengaruh panjang ayat terhadap performa & akurasi)