

**PENGEMBANGAN *ARABIC ROOT ANALYSIS SYSTEM* (ARAS)
BERBASIS *WEBSITE* DENGAN METODE *INCREMENTAL***

SKRIPSI

Oleh:

MUHAMAD RADIYUDIN

NIM. 220605110120



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

**PENGEMBANGAN *ARABIC ROOT ANALYSIS SYSTEM* (ARAS)
BERBASIS *WEBSITE* DENGAN METODE *INCREMENTAL***

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
MUHAMAD RADIYUDIN
NIM. 220605110120

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

HALAMAN PERSETUJUAN

**PENGEMBANGAN ARABIC ROOT ANALYSIS SYSTEM (ARAS)
BERBASIS WEBSITE DENGAN METODE INCREMENTAL**

SKRIPSI

Oleh :
MUHAMAD RADIYUDIN
NIM. 220605110120

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 18 Juni 2026

Pembimbing I,



Nur Fitriyah Ayu Tunjung Sari, M.Cs
NIP. 19911226 202012 2 001

Pembimbing II,



Roro Inda Melani, M.T., M.Sc
NIP. 19780923 200501 2 008

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Supriyono, M.Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

PENGEMBANGAN ARABIC ROOT ANALYSIS SYSTEM (ARAS) BERBASIS WEBSITE DENGAN METODE INCREMENTAL

SKRIPSI

Oleh :

MUHAMAD RADIYUDIN

NIM. 220605110120

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: Juni 2026

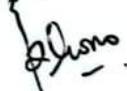
Susunan Dewan Penguji

Ketua Penguji : Fatchurrochman, M.Kom
NIP. 19700731 200501 1 002

Anggota Penguji I : Dr. Agung Teguh Wibowo Almais, M.T
NIP. 19860301 202321 1 016

Anggota Penguji II : Nur Fitriyah Ayu Tunjung Sari, M.Cs
NIP. 19911226 202012 2 001

Anggota Penguji III : Roro Inda Melani, M.T, M.Sc
NIP. 19780925 200501 2 008

()
()
()
()

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Supriyono, M. Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhamad Radiyudin
NIM : 220605110120
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Pengembangan *Arabic Root Analysis System* (Aras)
Berbasis *Website* Dengan Metode *Incremental*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 26 Juni 2026
Yang membuat pernyataan,



Muhamad Radiyudin
NIM.22060511120

MOTTO

... Berikhtiar dengan sungguh-sungguh, berdoa dengan sepenuh hati, dan berhusnuzhan kepada Allah atas setiap ketetapan-Nya." ...

HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim

Puji Syukur kehadiran Allah Swt. atas segala rahmat, nikmat, serta karunia-Nya, sehingga skripsi ini dapat diselesaikan dengan baik dan tepat waktu. Karya sederhana ini penulis persembahkan kepada:

Kedua Orang Tua Tercinta Sebagai wujud rasa hormat, cinta, dan terima kasih yang tak terhingga atas segala doa, kasih sayang, pengorbanan, dukungan moral maupun materi, serta motivasi yang tiada henti diberikan kepada penulis.

Keluarga tercinta yang selalu memberikan semangat, doa, perhatian, dan dukungan sehingga penulis dapat menyelesaikan pendidikan ini.

Dosen Pembimbing yang memberikan bimbingan, arahan, ilmu, kesabaran, serta motivasi yang telah diberikan selama proses penyusunan skripsi ini.

Seluruh Dosen dan Civitas Akademika yang telah ilmu, pengalaman, dan bimbingan yang telah diberikan selama penulis menempuh pendidikan.

Sahabat dan Teman-Teman yang telah memberikan bantuan, semangat, serta dukungan selama masa perkuliahan hingga penyelesaian skripsi ini.

Almamater Tercinta Yang telah menjadi tempat penulis menimba ilmu, mengembangkan kemampuan, serta membentuk karakter selama menempuh pendidikan.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

*Alhamdulillahirabbil'alam**in*, segala puji syukur penulis panjatkan ke hadirat Allah SWT. atas limpahan rahmat, nikmat, dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas skripsi yang berjudul “PENGEMBANGAN *ARABIC ROOT ANALYSIS SISTEM (ARAS) BERBASIS WEBSITE DENGAN METODE INCREMENTAL*” dengan baik dan tepat waktu. Shalawat serta salam tak lupa penulis sampaikan kepada Nabi Muhammad *Shallallahu 'Alaihi Wasallam*, sosok teladan mulia yang telah menuntun kita dari zaman kegelapan menuju zaman kebenaran, Islam, dan ilmu pengetahuan yang kita rasakan manfaatnya hingga hari ini.

Tugas skripsi ini disusun dalam rangka memenuhi salah satu syarat kelulusan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, UIN Maulana Malik Ibrahim Malang. Dalam proses penyusunan tugas ini, penulis mendapatkan bantuan, dukungan, serta doa dari banyak pihak, baik secara langsung maupun tidak langsung. Untuk itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Dr. H. Agus Mulyono, S.Pd., M.Kes, selaku Dekan Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.

3. Supriyono, M.Kom., selaku Ketua Program Studi S1 Teknik Informatika UIN Maulana Malik Ibrahim Malang
4. Juniardi Nur Fadila, M.T., Selaku Dosen Wali
5. Nur Fitriyah Ayu Tunjung Sari, M.Cs., selaku Dosen Pembimbing I yang telah menjadi pintu utama penulis untuk keluar dari permasalahan-permasalahan yang dialami selama proses penulisan skripsi ini. Terima kasih atas segala bimbingan, nasihat, masukan, dan dorongannya selama proses penulisan skripsi ini sehingga penulis mampu menyelesaikan skripsi ini dengan baik dan tepat waktu.
6. Roro Inda Melani, M.T, M.Sc., selaku Dosen Pembimbing II yang telah dengan sabar membimbing penulis untuk menyelesaikan skripsi ini.
7. Fatchurrochman, M.Kom dan Dr. Agung Teguh Wibowo Almais, M.T., selaku Dosen Penguji yang telah memberikan kritik, saran, dan bimbingan untuk menyempurnakan skripsi ini.
8. Nia Faricha, S.Si., admin Program Studi Teknik Informatika, yang membantu dalam urusan administrasi dan selalu mengingatkan tentang kelengkapan berkas.
9. Jauhar Fuadi, S.Pd selaku evaluator yang telah membantu dan memberikan masukan, saran dan dukungan selama proses penulisan skripsi ini
10. Madhasim dan Siti Aminah, Orangtua tercinta yang selalu mendo'akan, memberikan semangat serta mengingatkan penulis agar selalu berada di jalan yang diridhoi oleh Allah SWT.

11. Laeli Padilah, S.H., kakak tercinta yang selalu menjadi tempat yang nyaman bagi penulis untuk menyampaikan keluh kesah selama proses awal perkuliahan hingga saat ini.
12. Seluruh keluarga dan kerabat penulis yang tidak dapat penulis sebutkan satu per satu, yang turut memberikan bantuan, semangat, dukungan, dan doa untuk penulis.
13. Teman-teman sejawat dan sahabat seperjuangan angkatan 2022, rekan bertumbuh yang luar biasa, yang selalu menjadi teman diskusi yang kritis dan saling menguatkan dalam suka maupun duka.

Dan terakhir, tak lupa ucapan terima kasih kepada diri sendiri, yang dengan semangat dan tekad kuat sehingga mampu melewati setiap tantangan, rintangan, dan kesulitan yang dihadapi selama perjalanan ini. Penulis menyadari bahwa penelitian dalam tugas skripsi ini belum sempurna dan masih memiliki beberapa kekurangan. Oleh karena itu, dengan tulus, penulis bersedia menerima masukan dan kritik yang konstruktif dari pembaca sebagai bahan evaluasi dan perbaikan di masa depan. Penelitian ini bisa dilanjutkan dan dikembangkan lebih lanjut dalam penelitian selanjutnya agar dapat memperbaiki kekurangan yang masih ada. Penulis berharap, buku ini tidak hanya membawa manfaat bagi pembaca, tetapi juga bisa berguna dan memberi dampak positif kepada masyarakat secara umum.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 26 Juni 2026

Muhamad Radiyudin

DAFTAR ISI

HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
PERNYATAAN KEASLIAN TULISAN	Error! Bookmark not defined.
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR.....	xv
ABSTRAK	xvi
ABSTRACT	xvii
مستخلص البحث.....	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian	3
BAB II STUDI PUSTAKA	5
2.1 Penelitian Terkait.....	5
2.2 Bahasa Arab dan <i>Morfologi</i>	11
2.2.1 Pengertian Bahasa Arab	11
2.2.2 <i>Morfologi</i> Bahasa Arab	12
2.2.3 <i>Bina</i>	12
2.3 Ekstraksi Akar Kata (<i>ROOT Extraction</i>).....	14
2.4 <i>Finite State Automata (FSA)</i>	14
2.4.1 Pengertian FSA	14
2.4.2 Komponen Dasar FSA	15

2.4.3 Jenis-jenis FSA	16
2.4.4 Aplikasi FSA dalam bidang bahasa	19
2.5 Metode Pengembangan <i>Incremental</i>	19
2.5.1 Pengertian metode <i>incremental</i>	19
2.6 <i>Black-Box Testing</i>	21
2.7 <i>System Usability Scale (SUS)</i>	21
BAB III METODE PENELITIAN	24
3.1 Desain Penelitian	24
3.2 Desain Sistem	25
3.2.1 Alur Kerja Sistem	25
3.2.2 Perancangan Sistem	29
3.4.2 Perancangan Logika FSA	33
3.3 Implementasi	39
3.4 Pengujian Sistem	39
3.4.1 Pengujian Fungsional	39
3.4.2 Pengujian FSA	41
3.4.3 Pengujian <i>Usability</i>	42
BAB IV HASIL DAN PEMBAHASAN	44
4.1 <i>Increment pertama</i>	44
4.1.1 <i>Requirements</i>	44
4.1.2 <i>Analysis</i>	45
4.1.3 Perancangan antarmuka halaman utama	45
4.1.4 Hasil <i>increment pertama</i>	46
4.2 <i>Increment kedua</i>	47
4.2.1 <i>Requirements</i>	47
4.2.2 <i>Analysis</i>	47
4.2.3 Perancangan FSA <i>Ajwaf</i>	48
4.2.4 Hasil Implementasi <i>Ajwaf</i>	49
4.3 <i>Increment ketiga</i>	50
4.3.1 <i>Requirements</i>	50
4.3.2 <i>Analysis</i>	51
4.3.3 Perancangan FSA <i>Mudho'af</i>	51

4.3.4 Hasil Implementasi <i>Mudhoaf</i>	52
4.4 Pengujian dan Evaluasi	53
4.4.1 Pengujian dan Evaluasi Kebutuhan Fungsional dan <i>Non</i> Fungsional..	53
4.4.2 Pengujian dan Evaluasi FSA.....	55
4.4.3 Pengujian dan Evaluasi <i>Usability</i>	60
4.5 Integrasi Islam	64
4.5.1 <i>Muamalah Ma'a Allah</i>	65
4.5.2 <i>Mu'amalah Ma'annas</i>	65
BAB V KESIMPULAN DAN SARAN	67
5.1 Kesimpulan	67
5.2 Saran	68
DAFTAR PUSTAKA	
LAMPIRAN-LAMPIRAN	

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	9
Tabel 3.1 Skenario uji kebutuhan fungsional.....	40
Tabel 3.2 Skenario uji kebutuhan Non fungsional.....	40
Tabel 3.3 Data kata <i>bina Mudho 'af</i> dan <i>ajwaf</i>	41
Tabel 3.4 Pernyataan kuisioner SUS.....	42
Tabel 4.1 Hasil pengujian kebutuhan fungsional.....	54
Tabel 4.2 Hasil pengujian kebutuhan Non fungsional.....	54
Tabel 4.3 Hasil pengujian FSA <i>bina Mudho 'af</i>	55
Tabel 4.4 Hasil pengujian FSA <i>bina ajwaf</i>	57
Tabel 4.5 Daftar nilai kuisioner	61
Tabel 4.6 Perhitungan skor SUS	61
Tabel 4.7 Total skor responden tiap responden (sebelum dikali 2,5)	62
Tabel 4.8 Skor SUS responden (Total dikali 2,5)	62

DAFTAR GAMBAR

Gambar 2.1 <i>Deterministic Finite Automata (DFA)</i>	16
Gambar 2.2 <i>Non-Deterministic Finite Automata (NFA)</i>	18
Gambar 2.3 Alur metode <i>Incremental</i>	20
Gambar 2.4 Kategori <i>Score SUS</i>	22
Gambar 3.1 Desain penelitian.....	24
Gambar 3.2 Diagram Alir sistem	25
Gambar 3.3 <i>Use case</i> diagram	30
Gambar 3.4 <i>Activity</i> Diagram ARAS	31
Gambar 3.5 Diagram ERD.....	32
Gambar 3.6 Model DFA.....	33
Gambar 4.1 Tampilan UI halaman ARAS.....	46
Gambar 4.2 FSA <i>Ajwaf</i>	49
Gambar 4.3 Hasil <i>Testing bina ajwaf</i>	50
Gambar 4.4 FSA <i>mudho'af</i>	52
Gambar 4.5 Hasil <i>Testing bina Mudho'af</i>	53

ABSTRAK

Radiyudin, Muhamad. 2026. **PENGEMBANGAN ARABIC ROOT ANALYSIS SYSTEM (ARAS) BERBASIS WEBSITE DENGAN METODE INCREMENTAL**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Nur Fitriyah Ayu Tunjung Sari, M.Cs (II) Roro Inda Melani, M.T, M.Sc.

Kata kunci: *Finite State Automata, Incremental, Morfologi Bahasa Arab, System Usability Scale*

Bahasa Arab merupakan bahasa yang memiliki kedudukan penting dalam dunia pendidikan Islam karena bahasa arab digunakan sebagai bahasa Al-Qur'an. Oleh sebab itu pembelajaran bahasa arab dilakukan agar dapat memahami isi kandungan dalam Al-Qur'an dengan baik dan benar. Namun, salah satu tantangan yang sering dihadapi dalam proses pembelajaran bahasa Arab adalah memahami perubahan bentuk kata serta menentukan akar kata, karena bahasa arab memiliki aturan bentuk kata yang rumit. Oleh karena itu, penelitian ini bertujuan untuk membangun ARAS (*Arabis ROOT ANALYSIS SYSTEM*), yaitu sistem berbasis *website* yang bisa menganalisis akar kata dalam bahasa Arab dengan menggunakan metode *Finite State Automata* (FSA) dan dikembangkan dengan pendekatan *Incremental*. Sistem yang dikembangkan difokuskan pada analisis *Fi'il Tsulasi Mujarrad Bina Mudhoaf* dan *Fi'il Tsulasi Mujarrad Bina Ajwaf* yang berasal dari kitab *Amtsilatut Tashrifiyyah*. Dari hasil pengujian fungsionalitas sebanyak 9 skenario pengujian, menunjukkan bahwa seluruh skenario pengujian tersebut berhasil dijalankan sesuai dengan kebutuhan sistem. Selain itu, hasil pengujian FSA menunjukkan bahwa ARAS mampu mencapai tingkat akurasi 82% pada kategori *Bina Mudhoaf* dengan 24 kata yang benar dari 29 data uji, serta 78% pada kategori *Bina Ajwaf* dengan 36 kata yang benar dari 46 data uji. Selain pengujian fungsionalitas dan FSA, peneliti juga melakukan pengujian *usability* menggunakan metode *System Usability Scale* (SUS) yang melibatkan 20 responden. Dari hasil perhitungan menggunakan SUS, maka dihasilkan nilai rata-rata dari keseluruhan responden sebesar 64,3. Berdasarkan hasil tersebut, dapat disimpulkan bahwa ARAS berhasil dikembangkan sesuai dengan kebutuhan fungsional sistem.

ABSTRACT

Radiyudin, Muhamad. 2026. ***DEVELOPMENT OF A WEB-BASED ARABIC ROOT ANALYSIS SYSTEM (ARAS) USING THE INCREMENTAL METHOD.*** Thesis. Department of Computer Engineering, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University, Malang. Advisors: (I) Nur Fitriyah Ayu Tunjung Sari, M.Cs (II) Roro Inda Melani, M.T, M.Sc.

Key Word: *Finite State Automata, Incremental, Morfologi Arab Language, System Usability Scale*

Arabic is a language that holds an important position in the world of Islamic education because it is used as the language of the Qur'an. Therefore, Arabic is taught to enable students to understand the content of the Qur'an properly and accurately. However, one of the challenges frequently encountered in the process of learning Arabic is understanding word inflection and identifying word roots. Arabic has complex rules governing word forms. Therefore, this research aims to develop ARAS (Arabis Root Analysis System), a web-based System capable of analyzing Arabic word Roots using the Finite State Automata (FSA) method and developed using an incremental approach. The developed System focuses on the Analysis of Fi'il Tsulasi Mujarrad Bina Mudhoaf and Fi'il Tsulasi Mujarrad Bina Ajwaf derived from the book *Amtsilatut Tashrif*. The results of the functionality testing comprising 9 test scenarios indicate that all test scenarios were successfully executed in accordance with system requirements. Additionally, the FSA testing results show that ARAS achieved an accuracy rate of 82% in the Bina Mudhoaf category with 24 correct words out of 29 test data, as well as 78% in the Bina Ajwaf category.

مستخلص البحث

محمد راضي الدّين. 2026. تطوير نظام تحليل الجذور العربية. التراكمي (ARAS). قسم تقنية المعلومات، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: نور فيترياه أيو تونج ساري, M.Cs. المشرف الثاني: رورو إندا ميلاني, M.T, M.Sc.

الكلمات المفتاحية: آلات الحالة المحدودة، التدريجية، مورفولوجي اللغة العربية، مقياس قابلية استخدام النظام

تحتل اللغة العربية مكانة مهمة في عالم التعليم الإسلامي، حيث تُستخدم كلغة القرآن الكريم. ولذلك، يتم تدريس اللغة العربية بهدف فهم محتوى القرآن الكريم فهماً جيداً وصحيحاً. ومع ذلك، فإن أحد التحديات التي غالباً ما تواجه عملية تعلم اللغة العربية هو فهم صيغ الكلمات وتحديد جذورها، لأن اللغة العربية لها قواعد صيغ معقدة. ولذلك، تهدف هذه الدراسة إلى إنشاء (ARABIC ROOT ANALYSIS SYSTEM)، وهو نظام قائم على الويب يمكنه تحليل جذور الكلمات في اللغة العربية باستخدام طريقة آلة الحالات المحدودة (FSA) وتم تطويره بنهج تزايدى. يركز النظام المطور على تحليل الفعل الثلاثي المجرد المبني للمضاعف والفعل الثلاثي المجرد المبني للأجواف المستمدة من كتاب "أمثلة التصريفية". أظهرت نتائج اختبار الوظائف في 9 سيناريوهات اختبار أن جميع سيناريوهات الاختبار تم تنفيذها بنجاح وفقاً لمتطلبات النظام. بالإضافة إلى ذلك، أظهرت نتائج اختبار FSA أن ARAS قادر على تحقيق دقة بنسبة 82% في فئة "بناء المضاعف" مع 24 كلمة صحيحة من أصل 29 بيانات اختبار، و78% في فئة "بناء الأجواف" مع 36 كلمة صحيحة من أصل 46 بيانات اختبار. بالإضافة إلى اختبارات الوظائف واختبار FSA، أجرى الباحثون أيضاً اختبار قابلية الاستخدام باستخدام طريقة مقياس قابلية استخدام النظام (SUS) التي شارك فيها 20 شخصاً.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sebagai negara dengan jumlah penduduk muslim terbanyak di dunia, bahasa Arab diajarkan di banyak tingkat pendidikan, dari sekolah dasar sampai universitas, terutama di sekolah-sekolah Islam (Cahya Edi Setyawan, 2025). Salah satu tantangan dalam belajar bahasa Arab adalah memahami perubahan bentuk kata, hal ini masih banyak dialami oleh pelajar di Indonesia. Penelitian Wiza et al. (2024) menunjukkan bahwa 68,4% mahasiswa mengalami kesulitan dalam menentukan perubahan bentuk kata *Isim* maupun *Fi'il*.

Untuk mengatasi permasalahan tersebut, penelitian oleh Abidin yang berjudul “*Pembangkitan Otomatis Bentuk Kata Arab Berbasis User Interface*” mengembangkan sistem pencarian akar kata berbasis *rule-based*. Namun penelitian tersebut memiliki kekurangan berupa tingkat akurasi yang masih rendah, yaitu kurang dari 60%, khususnya pada bentuk *Bina Mudho 'af* dan *Ajwaf* yang memiliki pola *User Interface* kompleks (Abidin zainal, 2015).

Kompleksitas aturan bahasa Arab memerlukan metode pengembangan yang fleksibel. Oleh karena itu, metode *Incremental* digunakan karena memungkinkan pengembangan sistem dilakukan secara bertahap dan mudah dievaluasi pada setiap iterasi (Pressman R.S., 2010). Kemudian dalam analisis akar kata penggunaan FSA dipilih karena mampu merepresentasikan aturan *User Interface* bahasa Arab dalam bentuk *state* dan transisi secara lebih terstruktur (Fitriana et al., 2024). Bahasa Arab

di dalam agama islam merupakan bahasa yang memiliki kedudukan penting karena merupakan bahasa Al-Qur'an. Sebagaimana Firman Allah SWT dalam QS. Yusuf ayat 2:

إِنَّا أَنْزَلْنَاهُ قُرْآنًا عَرَبِيًّا لَّعَلَّكُمْ تَعْقِلُونَ

"Sesungguhnya Kami menurunkannya berupa Al-Qur'an dengan berbahasa Arab, agar kamu memahaminya".(Q.S Yusuf:2)

Imam Ibnu Katsir seorang ulama besar ahli tafsir, dalam karangannya yaitu *Tafsir Al-Quran Al-Azhim* memberikan penjelasan pada ayat diatas sebagai berikut "Bahasa Arab merupakan bahasa yang paling jelas, paling terang, paling luas, dan memiliki banyak kosakata yang dapat mengekspresikan berbagai makna, sehingga mampu memperbaiki dan mengarahkan jiwa manusia. Karena itulah, Allah menurunkan Kitab-Nya yang terbaik dengan bahasa yang terbaik di antara bahasa-bahasa yang lain, yang dicurahkan-Nya kepada utusan yang terbaik melalui perantaraan malaikat yang terbaik." (*Imam Ibnu Katsir*).

Melihat perkembangan teknologi saat ini, sistem berbasis *website* dinilai memiliki keunggulan berupa fleksibilitas akses, kemudahan distribusi, serta dapat digunakan pada berbagai perangkat (Rahman & Ramdani, 2026). Oleh karena itu, penelitian ini akan membuat sistem yang dinamakan ARAS (*Arabic Root Analysis System*) berbasis *website* menggunakan metode *Finite State Automata* (FSA) dan pendekatan *Incremental* .

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah sebagai berikut:

- a. Bagaimana membangun dan menguji ARAS menggunakan metode FSA dengan pendekatan *incremental*?
- b. Bagaimana hasil evaluasi *usability* ARAS berdasarkan metode *System Usability Scale* (SUS)?

1.3 Batasan Masalah

Agar penelitian ini bisa berjalan dengan baik dan menghasilkan temuan yang jelas serta spesifik. peneliti memberikan batasan masalah yang telah ditetapkan, yaitu sistem yang dikembangkan menangani penentuan kata berupa *Fi'il tsulasi mujarod Bina Mudhoaf* dan *Ajwaf* yang ada di dalam kitab *Amtsilatut Tashrifyyah*.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah sebelumnya, tujuan utama dari penelitian ini adalah sebagai berikut:

- a. Membangun dan menguji sistem ARAS berbasis *website* yang mampu mengekstraksi akar kata Bahasa Arab secara otomatis menggunakan metode FSA dengan pendekatan *incremental* dalam proses pengembangannya, sehingga dapat membantu dalam proses pembelajaran Bahasa Arab.
- b. Mengevaluasi tingkat *usability* ARAS berdasarkan metode SUS.

1.5 Manfaat Penelitian

Penelitian ini memiliki manfaat yang sangat penting. Secara teoritis, penelitian ini memberikan kontribusi besar dalam bidang bahasa arab khususnya memudahkan dalam menentukan akar kata. Sedangkan secara praktis penelitian ini

memberikan manfaat berupa sistem yang menawarkan alat bantu otomatis yang tepat untuk mengenali kata dasar bahasa arab. Ini bisa membantu mengurangi kesulitan dalam proses pencarian akar kata bahasa arab.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Sebelumnya, Fitriana dkk. (2024) telah melakukan penelitian dengan judul “Aplikasi Metode *Finite State Automata* (FSA) dalam Pemangkasan Kata (*Stemming*) Bahasa Indonesia Menggunakan Algoritma Nazief & Adriani” Penelitian ini mendesain penghapusan imbuhan dalam FSA dan menguji akurasi sistem berbasis web. Hasilnya menunjukkan bahwa FSA mampu merepresentasikan proses penghapusan imbuhan secara sistematis sehingga menghasilkan kata dasar dengan baik. (Fitriana et al., 2024).

Penelitian lain yang berjudul “Pembangkitan Otomatis Bentuk Kata Arab Berbasis *User Interface*” telah membuat aplikasi komputer yang bisa menghasilkan bentuk kata dalam bahasa Arab secara otomatis dengan menggunakan ilmu *sharaf*, sehingga bisa digunakan sebagai alat bantu belajar dan membantu pengguna memahami pola-pola pembentukan kata dalam bahasa Arab (Abidin zainal, 2015). Metode yang digunakan adalah dengan membuat aturan pembentukan kata berdasarkan prinsip *tashrif isthilahy* dan *tasrif lughawy* yang diambil dari buku *Al-Amsilah al-Tashrifiyah* karya Muhammad Maksum (1965), kemudian aturan tersebut diterjemahkan ke dalam bentuk algoritma yang bisa dipahami dan digunakan oleh bidang teknologi informasi dengan menggunakan bahasa pemrograman Java. Penelitian menunjukkan bahwa sistem yang dibuat bisa memberikan akurasi rata-rata sebesar 90,12% dalam proses tasrif *isthilahy* dan

66,93% dalam *tasrif lughawy*. Tingkat akurasi tertinggi dicapai pada *Bina shoheh*, yaitu 96,88%. Namun, masih ada beberapa bentuk seperti *Ajwaf* dan *Mudhoaf* yang belum bisa diakui dengan baik, sehingga diperlukan perbaikan aturan untuk meningkatkan akurasi secara keseluruhan.

Tsalitsatun dkk juga telah melakukan penelitian yang berjudul “*Autonomy Stemmer Algorithm for Legal and Illegal Affix Detection Use Finite-State Automata Method*” (Ana; et al., 2019). Penelitian ini mengusulkan algoritma stemming baru tanpa kamus yang mampu mendeteksi *afiks* legal dan ilegal dalam bahasa Indonesia menggunakan metode *Finite-State Automata*. Pengujian dilakukan menggunakan data uji berupa 118 dokumen berita yang memiliki total 15.792 kata. Kemudian penelitian menunjukkan bahwa algoritma yang diajukan berhasil menghasilkan 10.449 kata yang benar dengan tingkat akurasi rata-rata mencapai 66%. Algoritma tersebut juga menunjukkan kemampuan yang cukup baik dalam mengenali akar kata.

Dalam penggunaan metode *Incremental* telah ada penelitian sebelumnya yang berjudul “Penerapan Metode *Iteratif Incremental* dalam Pengembangan *website* Tripinaja untuk Meningkatkan Pelayanan dan Proses Bisnis” penelitian ini mengembangkan aplikasi berbasis web yang bertujuan meningkatkan kualitas layanan dan efektivitas proses bisnis startup Tripinaja di bidang pariwisata. Pengujian sistem dilakukan dengan metode *black box testing* agar memastikan fungsi sistem berjalan dengan baik. Hasil penelitian menunjukkan bahwa semua fitur berjalan dengan baik dan aplikasi memperoleh tingkat penerimaan sebesar 81%, sehingga dinyatakan layak digunakan oleh masyarakat. (Purbaya et al., 2023).

Pengembangan aplikasi *e-commerce* berbasis web mobile dengan pendekatan *iterative Incremental* dikaji oleh Chandra dkk. dalam penelitian berjudul “*Penerapan Model Iterative Incremental dalam Membangun Aplikasi E-Commerce di Toko Kopi Rarukuh Luas Berbasis Web Mobile*”. Penelitian ini diarahkan untuk menyediakan solusi *digital* bagi toko kopi Rarukuh Luas guna meningkatkan pemasaran, mempercepat proses transaksi, serta mempermudah pengelolaan data penjualan. Melalui penerapan pengembangan bertahap dan iteratif, aplikasi yang dihasilkan mampu mendukung berbagai fitur, seperti pencarian produk, pemesanan, penelusuran riwayat transaksi, hingga pencetakan nota pembelian. (Yudi & Kosdiana, 2023).

Model *Incremental* diterapkan oleh Halimah dkk. dalam penelitian yang berjudul “*Penerapan Model Incremental pada Implementasi dan Pengujian Sistem Informasi Pendanaan Usaha Mikro Kecil (PUMK)*.” Penelitian ini berfokus pada pengembangan sistem informasi untuk meningkatkan efisiensi pengelolaan program pendanaan UMK di PT. Angkasa Pura II Kantor Cabang Jambi. Pengembangan sistem dimulai dengan tahap *pra-State* untuk mengidentifikasi kebutuhan, kemudian dilanjutkan dengan dua tahap *State*. Pengujian sistem dilakukan dengan metode *black box Testing* yang menunjukkan bahwa semua fungsi berjalan dengan baik, serta uji penerimaan pengguna (UAT) yang melibatkan 18 responden dengan tingkat penerimaan sebesar 88%, sehingga sistem dianggap efektif dan layak digunakan. (Halimah et al., 2024).

Permasalahan koordinasi antara pedagang keliling dan warga menjadi fokus penelitian Hasyim dkk. dalam penelitian berjudul “*Pengembangan Back-End*”

Aplikasi *Mobile Web* untuk Pedagang Keliling di Lingkungan RT 04 Palem Kota Bandung". Penelitian ini merancang sistem berbasis *website* untuk memudahkan warga dalam mengetahui ketersediaan pedagang secara *real time* dengan menerapkan metode *iterative Incremental* yang memungkinkan pengembangan dilakukan secara bertahap dan adaptif. Uji sistem dilakukan melalui *Blackbox Testing* dan *Stress Testing* untuk memastikan kestabilan dan fungsi aplikasi. Hasil penelitian menunjukkan bahwa pendekatan iteratif *Incremental* mampu meningkatkan efisiensi pengembangan, memudahkan penyesuaian fitur berdasarkan umpan balik pengguna, dan menjaga performa sistem meskipun digunakan oleh 100 hingga 200 pengguna secara bersamaan (Hasyim et al., 2025).

Sementara itu, Saputera dkk. Telah melakukan penelitian yang berjudul "Implementasi Metode *Incremental* Pada Sistem Pengelolaan Arsip Berbasis Web" bertujuan untuk menerapkan pendekatan *Incremental* dalam pengembangan sistem pengelolaan arsip di Kantor Desa Suka Maju. Tujuan ini dilakukan agar pengembangan dapat berlangsung secara bertahap, terorganisir, dan sesuai dengan kebutuhan utama pengguna. Metode *Incremental* dipilih karena memungkinkan penambahan fitur secara bertahap, mengurangi risiko gagalnya proyek, serta memudahkan evaluasi setiap tahap pengembangan. Hasil penelitian menunjukkan bahwa sistem pengelolaan arsip berbasis web yang dikembangkan dengan metode ini berhasil meningkatkan efisiensi dalam pencarian arsip, mengurangi ketergantungan pada arsip fisik, mempercepat proses administrasi, dan meningkatkan tingkat keamanan data. (Saputera et al., 2023).

Tabel 2. 1 Penelitian Terdahulu

Peneliti	Judul	Tujuan	Metode	Hasil	Relevansi
Fitriana et al., 2024	Application of the Finite State Automata (FSA) Method in Indonesian Stemming using the Nazief & Adriani Algorithm	Memodelkan proses stemming bahasa Indonesia menggunakan FSA dan menerapkan algoritma Nazief & Adriani dalam sistem berbasis web.pada perangkat mobile.	Studi literatur algoritma Nazief & Adriani, pemodelan FSA/NFA, penyusunan pseudocode, pengujian sistem web pada kata turunan.	Sistem berhasil melakukan stemming pada 20 kata berimbuhan dengan akurat menggunakan FSA, serta mengungkap kelemahan algoritma terkait <i>under-stemming</i> .	Menunjukkan bagaimana FSA dapat memodelkan analisis <i>User Interface</i> dan berguna sebagai referensi dalam membangun sistem penentu akar kata Arab berbasis aturan dan <i>Incremental</i> .
Abidin zainal, 2015	Pembangkitan Otomatis Bentuk Kata Arab Berbasis <i>User Interface</i>	Membangun aplikasi komputer untuk pembangkit bentuk kata arab secara otomatis.	Rule-Based	Diperoleh akurasi rata-rata 90.12% untuk tasrif isthilahy dan 66.93% untuk tasrif Lughawy	Penelitian ini dapat dijadikan dasar metodologis dalam penelitian penentuan akar kata. Dan jadi pembanding dengan penelitian ini yang menggunakan Metode lain yaitu FSA
Tsalitsatun et al., 2019	Autonomy Stemmer Algorithm for Legal and Illegal Affix Detection Use Finite-State Automata Method	Membuat sistem yang bisa menentukan akar kata secara otomatis	Rule-based	Algoritma yang diajukan berhasil menghasilkan 10.449 kata yang benar dengan tingkat akurasi rata-rata mencapai 66%.	Relevansi penelitian ini sebagai dasar konseptual dan pembanding akurasi bagi penelitian yang akan dilakukan, serta menjadi referensi dalam pengembangan metode yang lebih optimal.
Purbaya et al., 2023	Penerapan Metode Iterative <i>Incremental</i> Dalam Pengembangan	Mengembangkan WEBSITE Tripinaja untuk meningkatkan layanan bisnis.	Iterative <i>Incremental</i> Model, Black Box <i>Testing</i> , UAT.	Sistem berjalan baik dan diterima pengguna dengan nilai UAT 81%.	Relevan sebagai model pengembangan sistem bertahap, cocok untuk membangun

Peneliti	Judul	Tujuan	Metode	Hasil	Relevansi
	WEBSITE Tripinaja				sistem penentu akar kata berbasis WEBSITE.
Yudi & Kosdiana , 2023	Penerapan Model Iterative <i>Incremental</i> dalam Membangun Aplikasi E-Commerce Berbasis Web Mobile	Membangun aplikasi e-commerce untuk meningkatkan transaksi dan pengelolaan data.	Iterative <i>Incremental</i> , PHP–MySQL, Black Box <i>Testing</i> .	Sistem berjalan baik dan fleksibel terhadap perubahan kebutuhan.	Menegaskan bahwa <i>Incremental</i> cocok digunakan untuk pengembangan modul bertahap, sama seperti kebutuhan sistem.
Halimah et al., 2024	Penerapan Model <i>Incremental</i> pada Implementasi dan Pengujian Sistem Informasi Pendanaan Usaha Mikro Kecil (PUMK)	Mengimplementasikan sistem informasi PUMK untuk mempermudah pengelolaan UMK.	<i>Incremental</i> Model, Black Box <i>Testing</i> , UAT.	Semua fitur berjalan baik; tingkat penerimaan pengguna 88%.	Relevan karena menunjukkan keunggulan <i>Incremental</i> dalam pengembangan sistem yang membutuhkan validasi dan penyempurnaan bertahap.
Hasyim et al., 2025	Pengembangan Back-End Aplikasi Mobile Web untuk Pedagang Keliling di Lingkungan RT 04 Palem Kota Bandung	Mengembangkan sistem backend aplikasi pedagang keliling untuk meningkatkan komunikasi dan akses layanan antara pedagang dan warga	Iterative <i>Incremental</i> , Blackbox <i>Testing</i> , Stress <i>Testing</i>	Sistem berhasil dibangun secara bertahap, stabil hingga 100–200 pengguna, dan fleksibel menyesuaikan kebutuhan	Relevan pada aspek penggunaan model <i>Incremental</i> dalam pengembangan sistem berbasis web, yang dapat diterapkan pada pembangunan modul FSA secara bertahap.
Saputera et al., 2023	Implementasi Metode <i>Incremental</i> Pada Sistem Pengelolaan Arsip Berbasis Web	Mengimplementasikan metode <i>Incremental</i> untuk membangun sistem pengelolaan arsip yang efisien dan fleksibel	Metode <i>Incremental</i> (Requirement → Design → Coding → <i>Testing</i>), Blackbox <i>Testing</i>	Sistem arsip berbasis web berjalan efisien, mempermudah pencarian arsip, dan fitur dapat dikembangkan bertahap sesuai kebutuhan	Pendekatan bertahap ini bisa diterapkan dalam pengembangan modul FSA, aturan <i>User Interface</i> , dan fitur web lainnya

Peneliti	Judul	Tujuan	Metode	Hasil	Relevansi
Penelitian ini (ARAS)	Pengembangan ARABIC ROOT ANALYSIS SISTEM (ARAS) Berbasis WEBSITE	Mengembangkan sistem ekstraksi akar kata bahasa Arab yang lebih akurat dan mudah diakses	FSA + <i>Incremental</i>	Diharapkan meningkatkan akurasi dan fleksibilitas sistem	Menggabungkan keunggulan FSA dan <i>Incremental</i> untuk mengatasi keterbatasan penelitian sebelumnya

2.2 Bahasa Arab dan *Morfologi*

Pembahasan mengenai bahasa Arab merupakan dasar yang penting sebelum mempelajari konsep-konsep yang lebih dalam mengenai struktur kata serta proses pembentukannya.

2.2.1 Pengertian Bahasa Arab

Bahasa Arab adalah bahasa yang digunakan oleh banyak orang di berbagai negara, serta digunakan sebagai bahasa resmi dalam agama Islam. Bahasa ini memiliki struktur tata bahasa yang banyak dan rumit, terutama dalam hal bentuk kata. *Morfologi* adalah bagian dari ilmu bahasa yang mempelajari bentuk dan cara kata terbentuk, serta bagaimana kata berubah sesuai dengan peran dan artinya dalam kalimat (Azhar Muhammad, 2025).

Menentukan akar kata dalam bahasa Arab sangat penting karena sebagian besar kosakata dalam bahasa Arab terbentuk berdasarkan sistem akar kata (*jidzr*) dan pola (*wazan*). Dengan memahami asal kata, kita bisa mengetahui arti dasar suatu kata dan dengan lebih mudah memahami hubungan antara kata-kata yang terkait. Selain itu, mengenali akar kata sangat penting dalam belajar bahasa Arab, menganalisis *morfologi* (*shorof*), serta memperkembangkan sistem komputasi

bahasa Arab, seperti pencarian informasi, penerjemahan otomatis, dan pengolahan bahasa alami (Aqil & Hadi, 2019).

2.2.2 Morfologi Bahasa Arab

Morfologi dalam bahasa Arab disebut sebagai *al-sharf* (الصرف), yaitu ilmu yang mempelajari cara kata-kata berubah bentuknya serta aturan-aturan yang mengatur perubahan tersebut. *Morfologi* bahasa Arab mencakup akar kata (جذر), pola (وزن), dan *Afiksasi*. Akar kata yaitu bentuk dasar yang terdiri dari beberapa huruf, biasanya tiga huruf, yang menyimpan makna utama. Selain itu ada pola (وزن), yaitu susunan huruf yang bisa berubah untuk membentuk kata turunan dari akar yang sama. *Afiksasi* adalah penambahan huruf di awal kata (*prefiks*), di tengah kata (*infix*), atau di akhir kata (*sufiks*) agar fungsi kata tersebut berubah (Sekarsari et al., 2025). Contohnya, dari akar ك-ت-ب dapat dibentuk banyak kata seperti كَتَبَ (menulis), مَكْتَب (tempat menulis/meja), dan كِتَاب (*Buku/Kitab*), tergantung pola dan afiks yang digunakan.

2.2.3 Bina

Dalam ilmu sharaf, istilah *Bina* digunakan untuk menunjukkan bentuk perubahan kata berdasarkan susunan huruf asal dan perubahan yang terjadi pada kata tersebut. *Bina* menjadi salah satu bagian penting dalam analisis *morfologi* bahasa Arab karena digunakan untuk mengidentifikasi karakteristik perubahan *Fi'il* maupun *Isim*. Secara umum, *Bina* dalam ilmu sharaf dibagi menjadi beberapa jenis, seperti *shahih*, *mitsal*, *Ajwaf*, *naqis*, dan *mudha'af*. Pemahaman terhadap *Bina* sangat penting karena setiap jenis memiliki pola perubahan yang berbeda pada proses *tashrif* atau pembentukan kata. Penelitian mengenai pembelajaran ilmu

sharaf menunjukkan bahwa pemahaman pola perubahan kata menjadi salah satu aspek utama dalam memahami struktur bahasa Arab secara benar (Amalia, 2024).

2.2.3.1 Bina Ajwaf

Bina Ajwaf merupakan bentuk kata yang huruf tengahnya termasuk huruf illat, yaitu *alif* (ا), *wau* (و), atau *ya* (ي). Keberadaan huruf illat pada bagian tengah menyebabkan terjadinya perubahan bunyi maupun bentuk kata ketika dilakukan proses tashrif. Contohnya adalah kata *قَالَ* yang berasal dari akar kata *ق و ل* dan *بَا ع* yang berasal dari akar kata *ب ي ع*. Pada *Bina Ajwaf*, perubahan huruf tengah sering terjadi karena pengaruh harakat dan pola pembentukan kata sehingga membutuhkan proses analisis *User Interface* yang lebih kompleks dibandingkan kata shahih biasa (Ihwan et al., 2022). Oleh karena itu, dalam sistem analisis *User Interface* bahasa Arab, *Bina Ajwaf* memerlukan aturan khusus agar sistem mampu mengenali perubahan bentuk kata secara tepat. Pemahaman terhadap perubahan bentuk kata dalam ilmu *sharaf* sangatlah penting dalam pembelajaran bahasa Arab karena berkaitan langsung dengan kemampuan memahami struktur kata dan makna dalam teks Arab.

2.2.3.2 Bina Mudho'af

Bina mudha'af adalah bentuk kata yang memiliki pengulangan huruf asli pada bagian akhir atau tengah kata sehingga dua huruf yang sama digabungkan menggunakan tanda *tasydid*. Contohnya seperti *مَدَّ* dan *شَدَّ*. Pada *Bina* ini terjadi proses penggabungan dua huruf identik menjadi satu huruf bertasydid untuk mempermudah pengucapan (Musa & Ghofur, 2024). Dalam analisis *sharaf*, *Bina mudha'af* memiliki karakteristik khusus karena sistem harus mampu mendeteksi

adanya huruf ganda yang secara struktur berasal dari dua huruf asli. Penanganan *Bina mudha'af* menjadi penting dalam penelitian ini karena pola tersebut mempengaruhi proses identifikasi akar kata dan pencocokan pola menggunakan metode Finite State Automata (FSA). Pemahaman terhadap perubahan bentuk kata dalam ilmu sharaf sangat diperlukan dalam pembelajaran bahasa Arab karena membantu proses identifikasi struktur kata dan makna secara tepat.

2.3 Ekstraksi Akar Kata (*ROOT Extraction*)

Dalam bidang komputasi, ekstraksi akar kata adalah proses yang bertujuan untuk menemukan bentuk dasar atau akar dari sebuah kata sebelum mengalami perubahan bentuk karena ditambahkan imbuhan, pola, atau variasi lainnya. Dalam bahasa Arab, ekstraksi akar kata sangat penting karena banyak kata dibentuk dari satu akar dengan cara menambahkan pola dan imbuhan yang rumit. Proses ekstraksi akar kata membantu komputer memahami hubungan antar kata-kata yang memiliki arti dasar sama, meskipun bentuknya berbeda dalam teks (Thalji, 2024).

2.4 Finite State Automata (FSA)

FSA hadir sebagai salah satu model yang mampu menunjukkan hubungan antara *Input*, kondisi, dan hasil melalui mekanisme perpindahan keadaan. Dengan memahami dasar-dasar konsep FSA, penjelasan tentang definisi dan ciri-ciri FSA dapat lebih mudah dipahami.

2.4.1 Pengertian FSA

FSA adalah sebuah model matematika yang digunakan untuk menggambarkan sistem atau mesin yang memiliki jumlah keadaan terbatas, dan

bisa berpindah dari satu keadaan ke keadaan lainnya sesuai dengan *Input* yang diterimanya. FSA menerima masukan berupa rangkaian karakter atau simbol, lalu memakai aturan peralihan untuk menentukan perubahan state hingga akhirnya mencapai state akhir atau tidak (Daffa, 2024).

Dalam bidang komputasi dan pemrograman, FSA berfungsi sebagai mesin yang sederhana, mampu mengenali pola atau struktur tertentu dalam data, seperti mengecek apakah suatu masukan memenuhi syarat atau sesuai dengan aturan yang telah ditetapkan (Daffa, 2024).

2.4.2 Komponen Dasar FSA

Model FSA secara umum memiliki beberapa komponen utama, yaitu Set State yang dilambangkan dengan (Q), yaitu kumpulan semua keadaan yang mungkin ada dalam mesin tersebut. Kemudian Alfabet *Input* yang dilambangkan dengan (Σ), yaitu kumpulan simbol atau karakter yang digunakan sebagai masukan dalam FSA. Selanjutnya Fungsi transisi yang dilambangkan dengan (δ) adalah aturan yang menjelaskan cara FSA beralih dari satu keadaan ke keadaan berikutnya sesuai dengan *Input* yang diterima. Lalu State Awal yang dilambangkan dengan (q_0), yaitu Kondisi awal sebelum mesin mulai membaca masukan. Terakhir State Akhir yang dilambangkan dengan (F), yaitu Keadaan atau beberapa keadaan yang dianggap sebagai hasil dari penerimaan *Input* yang valid (Sandhiyasa et al., 2024).

$$M = (Q, \Sigma, \delta, q_0, F) \quad (2.1)$$

Q = Himpunan State/Kedudukan

Sigma (Σ) = Himpunan Symbol *Input*/masukkan/abjad

Delta (δ) = Fungsi transisi

q_0 = State awal q_0 , dimana q_0 adalah elemen(ϵ) dari Q

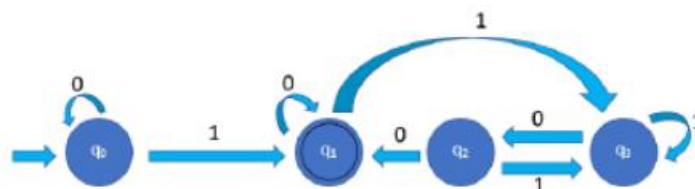
F = Himpunan State Akhir, F bagian dari ($\subseteq$) Q

2.4.3 Jenis-jenis FSA

Ada 2 jenis FSA, yaitu *Deterministic Finite Automata* dan *Non Deterministic Finite Automata*. Di bawah ini penjelasan terkait 2 jenis FSA tersebut.

2.4.3.1 *Deterministic Finite Automata (DFA)*

Dalam DFA, setiap state hanya memiliki satu jalur transisi yang terdefinisi dengan jelas untuk setiap simbol *Input*. Artinya, dari satu kondisi dan satu simbol masukan, mesin hanya bisa pindah ke satu kondisi berikutnya saja (Sandhiyasa et al., 2024). Diagram DFA dapat dilihat pada gambar 2.1.



Gambar 2.1 *Deterministic Finite Automata (DFA)*

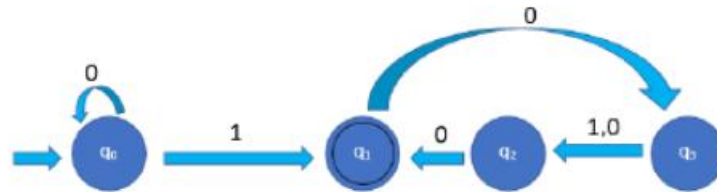
(Sumber gambar: (Dandy et al., 2020))

Gambar 2.1 menunjukkan sebuah Mesin FSA yang memiliki empat state, yaitu q_0 , q_1 , q_2 , dan q_3 . State q_0 berfungsi sebagai state awal, sedangkan state q_1 adalah state akhir yang ditandai dengan dua lingkaran. Perpindahan dari satu state ke state lainnya ditentukan oleh masukan berupa simbol 0 dan 1. Di state q_0 ada

loop sendiri untuk simbol 0, sehingga automata tetap berada di state q_0 ketika menerima input 0. Jika mendapat masukan 1, otomata akan pindah dari keadaan q_0 ke keadaan q_1 . Di state q_1 juga ada *self-loop* untuk simbol 0, sehingga automata tetap berada di state tersebut ketika menerima input 0. Selain itu, ada peralihan dari keadaan q_1 ke q_3 dengan input 1. State q_2 dan q_3 saling terhubung dengan cara dua arah, yaitu dari q_2 ke q_3 jika ada input 1, serta dari q_3 ke q_2 jika ada input 0. Selanjutnya, terdapat peralihan dari keadaan q_2 ke q_1 dengan input 0, sedangkan pada keadaan q_3 terdapat *self-loop* untuk simbol 1. Berdasarkan struktur transisi tersebut, automata akan menerima sebuah string jika setelah semua simbol dalam string tersebut diproses, posisi akhir automata berada pada state q_1 yang merupakan state penerima. Diagram ini menunjukkan cara FSA menggunakan perpindahan dari satu state ke state lainnya untuk mengenali pola tertentu pada kombinasi simbol 0 dan 1.

2.4.3.2 *Non-Deterministic Finite Automata (NFA)*

Berbeda dengan DFA, pada NFA sebuah state bisa memiliki beberapa kemungkinan perpindahan untuk simbol *Input* yang sama, atau bahkan perpindahan tanpa membaca *Input* apa pun. Keuntungan NFA adalah Desainnya lebih fleksibel, tetapi cara kerjanya tetap sama dengan DFA (Sandhiyasa et al., 2024). Diagram NFA dapat dilihat pada gambar 2.2.



Gambar 2.2 *Non-Deterministic Finite Automata* (NFA)

(Sumber gambar: (Dandy et al., 2020))

Berbeda dengan automata pada gambar 2.1 sebelumnya yang memiliki banyak jalur transisi dan perpindahan antar state yang lebih rumit, Gambar 2.2 menunjukkan FSA dengan struktur yang lebih mudah dan alur transisi yang lebih lurus. Mesin ini memiliki empat keadaan, yaitu q_0 , q_1 , q_2 , dan q_3 , di mana q_0 adalah keadaan awal dan q_1 adalah keadaan penerima yang ditandai dengan dua lingkaran. Di state q_0 ada loop sendiri untuk simbol 0, sehingga mesin otomata akan tetap berada di state tersebut selama menerima input 0. Ketika mendapat tanda 1, mesin otomata berpindah dari keadaan q_0 ke keadaan q_1 . Selanjutnya, terjadi peralihan dari state q_1 ke q_3 dengan menggunakan simbol 0, lalu dari q_3 ke q_2 dengan simbol 1 atau 0, serta dari q_2 kembali ke q_1 dengan menggunakan simbol 0. Berbeda dengan gambar sebelumnya yang menunjukkan beberapa kemungkinan perpindahan antar state dan self-loop pada lebih dari satu state, automata pada gambar ini memiliki pola transisi yang lebih terbatas, sehingga lebih mudah untuk dianalisis. Struktur tersebut memungkinkan FSA mampu mengenali pola tertentu dari kombinasi simbol 0 dan 1, dengan sebuah string dianggap diterima jika proses membacanya berakhir di state q_1 yang merupakan state akhir penerima.

2.4.4 Aplikasi FSA dalam bidang bahasa

FSA memiliki peran penting dalam mengolah bahasa dan teks digital, misalnya dalam proses tokenisasi teks, yaitu membagi teks menjadi bagian-bagian kecil yang bisa dianalisis oleh komputer. Kemudian dalam Pengenalan pola, yaitu proses mengecek apakah sebuah kata atau bentuk tertentu sesuai dengan aturan yang diterapkan oleh sistem (Postiglione, 2024) .

2.5 Metode Pengembangan *Incremental*

Incremental metode yang akan digunakan oleh peneliti dalam proses pengembangan ARAS.

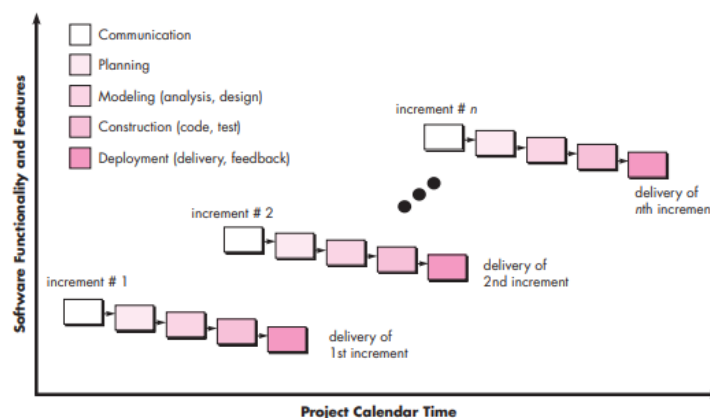
2.5.1 Pengertian metode *incremental*

Incremental merupakan model pengembangan perangkat lunak yang menghasilkan perangkat lunak dalam beberapa increment atau bertahap. Setiap increment menyediakan fungsionalitas yang semakin lengkap hingga seluruh kebutuhan sistem terpenuhi. Model ini memungkinkan pengguna memperoleh produk yang dapat digunakan lebih awal serta memberikan umpan balik untuk pengembangan increment berikutnya (Pressman R.S., 2010)

Metode pengembangan *Incremental* adalah suatu pendekatan dalam pembuatan perangkat lunak di mana suatu sistem dibangun dan dikembangkan secara perlahan dalam bentuk bagian demi bagian kecil yang disebut *State* atau iterasi (Evi Dwi Wahyuni et al., 2025).

Dalam model *Incremental* , pengembangan sistem dilakukan secara bertahap, mulai dari tahap perencanaan, pembuatan, pengujian, sampai pemasangan atau penggabungan pada setiap bagian sistem. Setelah satu bagian (*State*) selesai,

hasilnya bisa langsung digunakan, lalu bagian berikutnya dikembangkan berdasarkan masukan dari pengguna atau klien. Pendekatan ini memberi manfaat karena hasil perangkat lunak sudah bisa digunakan lebih dahulu, dan tim bisa mendapatkan masukan untuk memperbaiki atau menambahkan fitur berikutnya. Alur pendekatan *Incremental* dapat dilihat pada gambar 2.3.



Gambar 2.3 Alur metode *Incremental*
(Sumber gambar: (Pressman R.S., 2010))

Gambar 2.3 menunjukkan proses kerja dari metode pengembangan *Incremental*, yaitu cara membuat perangkat lunak dengan langkah perlahan secara bertahap melalui beberapa tahap hingga akhirnya terbentuk produk yang siap digunakan. Proses dimulai dengan mengumpulkan kebutuhan, lalu dilanjutkan dengan menganalisis, merancang, dan menulis kode dalam tahap pertama. Hasilnya adalah produk perangkat lunak awal yang sudah bisa digunakan meskipun fiturnya belum lengkap. Selanjutnya, pada iterasi kedua dan iterasi berikutnya, proses yang sama dilakukan kembali dengan menambahkan kebutuhan baru atau melakukan perbaikan berdasarkan hasil evaluasi dan masukan dari pengguna, sehingga setiap iterasi menghasilkan versi sistem yang lebih lengkap. Proses ini terus dilakukan

sampai semua kebutuhan terpenuhi dan tercipta produk perangkat lunak yang lengkap dan siap digunakan, sehingga metode *Incremental* memungkinkan pengembangan sistem yang lebih fleksibel, terkontrol, serta mampu menyesuaikan perubahan kebutuhan sepanjang proses pengembangan.

2.6 Black-Box Testing

Blackbox merupakan salah satu cara untuk menguji perangkat lunak dengan memeriksa bagian-bagian fungsional dan hasil yang dikeluarkan oleh sistem tanpa melihat atau memahami kode yang ada di dalamnya. Dalam pendekatan ini, seorang penguji menyebutkan apakah aplikasi berjalan sesuai dengan kebutuhan dengan memberikan berbagai masukan dan melihat hasil yang muncul. Dengan begitu, kesalahan pada fitur, proses, atau tampilan bisa terlihat dari perspektif pengguna.

Penelitian terbaru oleh Ridho dkk yang berjudul “*Implementation of Black-Box Testing on the Information system for the Smart Indonesia Card College Recommendation*” menunjukkan bahwa pengujian black-box adalah salah satu metode yang sering digunakan untuk mengurangi kesalahan dan memastikan kualitas perangkat lunak sebelum digunakan oleh pengguna, termasuk dalam pengujian sistem informasi dan aplikasi berbasis web (Sholehurrohman et al., 2025).

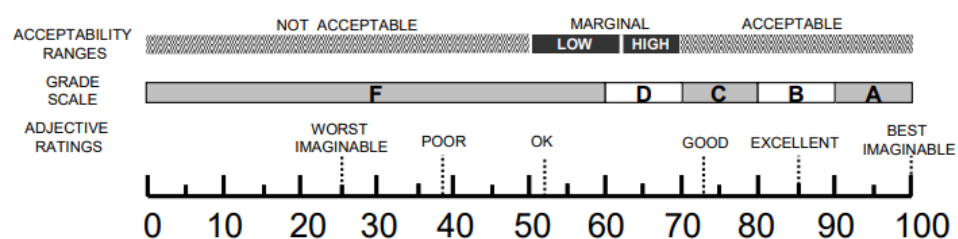
2.7 System Usability Scale (SUS)

Menurut Brooke, *Usability* tidak bisa didefinisikan dengan pasti karena sangat tergantung pada situasi penggunaan, sifat pengguna, pekerjaan yang dilakukan, serta lingkungan di mana sistem itu digunakan. Oleh karena itu, dibutuhkan cara yang bisa memberi penjelasan umum tentang sejauh mana suatu

sistem mudah digunakan, tanpa harus melakukan pengecekan yang rumit. SUS dikembangkan untuk memenuhi kebutuhan tersebut dengan memberikan cara penilaian yang mudah digunakan namun tetap bisa menghasilkan hasil yang konsisten dan dapat dipercaya (Brooke, 2020).

SUS berupa 10 pernyataan yang akan dinilai menggunakan skala *Likert* lima poin, mulai dari Sangat Tidak Setuju (1) hingga Sangat Setuju (5). Pernyataan-pernyataan tersebut dibuat untuk mengukur berapa aspek terkait *Usability*, seperti kemudahan dalam penggunaan, kompleksitas sistem, konsistensi fitur, kemudahan pembelajaran, serta tingkat kepercayaan pengguna saat menggunakan sistem.

Skor SUS berada pada rentang 0–100, di mana semakin tinggi skor yang diperoleh maka semakin baik tingkat *usability* sistem tersebut. Selain itu, skor SUS dapat dikategorikan ke dalam beberapa tingkat penerimaan, mulai dari *Not Acceptable*, *Marginal*, hingga *Acceptable* (Bangor et al., 2009). Gambar 4.4 merupakan hasil kategori nilai SUS yang dikemukakan oleh Bangor et al., 2009.



Gambar 2.4 Kategori *Score* SUS
(Sumber gambar: (Bangor et al., 2009))

Gambar 2.4 menjelaskan cara menilai tingkat kegunaan sistem menggunakan skor dari SUS. Skor SUS dibagi menjadi beberapa kategori, yaitu kategori *acceptability* yang terdiri dari *not acceptable*, *marginal*, dan *acceptable*,

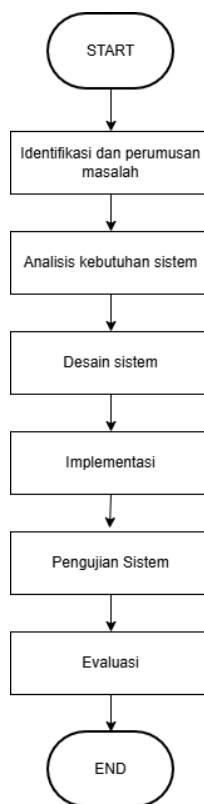
skala nilai berupa huruf F hingga A, serta penilaian berupa kata-kata seperti *worst imaginable* hingga *best imaginable*. Klasifikasi ini membantu mengubah angka SUS menjadi penilaian yang lebih sederhana dan mudah dipahami mengenai kualitas serta tingkat kemudahan penggunaan suatu sistem.

BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Proses penelitian dimulai dari tahap identifikasi dan perumusan masalah sebagai dasar penelitian, kemudian dilanjutkan dengan analisis kebutuhan sistem untuk menentukan spesifikasi yang diperlukan. Setelah itu dilakukan tahap Desain sistem sebagai perancangan solusi, yang kemudian direalisasikan pada tahap implementasi. Tahap akhir adalah pengujian sistem yang dilakukan untuk memastikan seluruh fungsi mampu berjalan sesuai dengan tujuan yang telah ditetapkan. Desain penelitian dapat dilihat pada gambar 3.1.



Gambar 3.1 Desain Penelitian

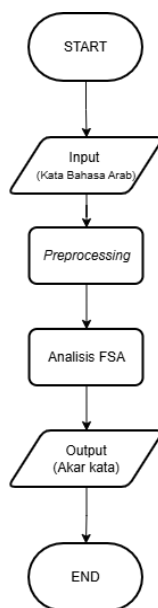
Gambar 3.1 menunjukkan bahwa penelitian dilaksanakan secara terstruktur mulai dari perencanaan hingga evaluasi hasil.

3.2 Desain Sistem

Untuk mendukung proses analisis *morfologi* bahasa Arab, penelitian ini merancang sebuah Desain sistem yang menggambarkan alur kerja aplikasi secara keseluruhan dan perancangan aplikasi.

3.2.1 Alur Kerja Sistem

Desain sistem tersebut menunjukkan tahapan proses mulai dari *Input* kata, *Preprocessing*, proses analisis menggunakan pendekatan FSA, hingga menghasilkan *Output* berupa informasi *hasil ekstraksi* kata. Perancangan ini bertujuan untuk memberikan gambaran terstruktur mengenai mekanisme kerja sistem dalam melakukan identifikasi pola dan akar kata bahasa Arab. Adapun Desain sistem yang digunakan dalam penelitian ini terdapat pada gambar 3.2.



Gambar 3.2 Diagram Alir sistem

Gambar 3.2 menunjukkan *flowchart* sistem yang digunakan dalam penelitian untuk menggambarkan alur proses analisis kata bahasa Arab menggunakan FSA. Alur pada sistem terdiri dari 4 tahapan utama, yaitu *Input*, *Preprocessing* (FSA), Analisis FSA dan *Output*, yang saling terhubung secara berurutan. Alur ini menunjukkan bagaimana data diproses mulai dari kata dimasukkan oleh pengguna hingga menghasilkan informasi analisis akhir berupa akar kata.

3.2.1.1 *Input*

Tahap input adalah proses pertama dalam sistem yang menerima masukkan kata bahasa arab yang dilakukan oleh pengguna. Kata yang telah dimasukkan inilah yang akan dianalisis untuk mengetahui akar katanya. Dalam penelitian ini, input yang digunakan adalah kata yang termasuk dalam kelompok *bina* bentuk ajwaf dan *bina* bentuk *mudha'af*.

3.2.1.2 *Preprocessing*

Tahap *preprocessing* dilakukan untuk mempersiapkan data agar dapat digunakan dalam analisis dengan memanfaatkan FSA. Pada tahap ini dilakukan beberapa proses, yakni menghilangkan tanda *harakat*, mengatur ulang bentuk karakter, dan membagi karakter menjadi bagian-bagian terpisah. Proses *preprocessing* diperlukan agar sistem bisa bekerja dengan lebih konsisten dan tepat.

1. Penghapusan Harakat

Menghilangkan harakat adalah proses menghapus semua tanda baca yang ada pada kata dalam bahasa Arab. Harakat yang dihapus mencakup *fathah*, *kasrah*, *dhammah*, *sukun*, serta berbagai jenis tanwin. Tujuan dari proses ini adalah

menyederhanakan bentuk kata sehingga sistem bisa fokus pada huruf-huruf penyusunnya, tanpa terganggu oleh cara pelafalannya yang berbeda. Misalnya, kata "يَمْدُ" akan diubah menjadi "يمد", sedangkan kata "قَالَ" akan menjadi "قال".

2. Normalisasi Karakter

Normalisasi karakter adalah proses mengatur kembali bentuk huruf Arab yang bisa ditulis dalam beberapa variasi berbeda. Pada tahap ini, karakter-karakter yang memiliki bentuk yang sama akan diubah menjadi satu bentuk yang standar. Misalnya, huruf ا, أ, dan آ diubah menjadi ا, sedangkan huruf ى diubah menjadi ي. Proses normalisasi bertujuan untuk mengurangi perbedaan cara menulis yang bisa menyebabkan kesalahan ketika sistem menganalisis pola kata.

3. Segmentasi Karakter

Segmentasi karakter adalah proses memecah kata yang sudah dinormalisasi menjadi huruf-huruf penyusunnya. Setiap huruf akan dipisahkan dan diatur berdasarkan urutan munculnya, sehingga bisa dibaca satu per satu oleh FSA. Misalnya, kata يمد akan dibagi menjadi tiga huruf, yaitu م, ي, dan د. Tahap ini diperlukan karena FSA bekerja dengan cara mengalihkan state setiap kali membaca karakter satu per satu secara berurutan.

3.2.1.3 Analisis FSA

Tahap analisis dilakukan menggunakan FSA. Tahapan ini adalah bagian utama dari sistem yang dikembangkan. Pada tahap ini, sistem akan membaca setiap karakter yang telah diproses sebelumnya melalui serangkaian keadaan dan perpindahan yang sudah ditentukan sebelumnya. Tujuan utama dari proses ini

adalah untuk mengenali bentuk kata, mengetahui jenis kata kerja, serta mengambil akar kata yang ada di dalamnya.

1. Pencocokan Pola

Pencocokan pola kata adalah proses mengenali struktur morfologi yang ada pada kata yang diberikan. Sistem akan memeriksa susunan huruf dengan pola-pola yang sudah ditentukan sebelumnya. Dengan cara ini, sistem bisa mengenali bentuk kata yang sesuai dengan aturan-aturan yang berlaku, sehingga memudahkan proses dalam mencari akar kata.

2. Identifikasi jenis bina

Setelah pola kata terdeteksi, sistem akan mengetahui jenis kata berdasarkan ciri bentuknya. Dalam penelitian ini, kategori yang digunakan adalah *Bina ajwaf* *Bina mudhaaf*. Proses mengenali dilakukan dengan menggunakan aturan-aturan transisi yang sudah dibuat dalam FSA sehingga sistem bisa membedakan kedua jenis kata tersebut secara otomatis.

3. Ekstraksi Akar Kata

Ekstraksi akar kata adalah proses mengambil huruf-huruf asli yang membentuk kata dasar dalam bahasa Arab. Setelah pola dan jenis kata kerja berhasil dikenali, sistem akan menentukan posisi huruf yang termasuk dalam akar kata. Misalnya, kata "يمد" akan menghasilkan akar kata "م د د", sedangkan kata "يقول" akan menghasilkan akar kata "ق و ل". Hasil ekstraksi ini adalah keluaran utama yang dihasilkan oleh sistem tersebut.

3.2.1.4 Output

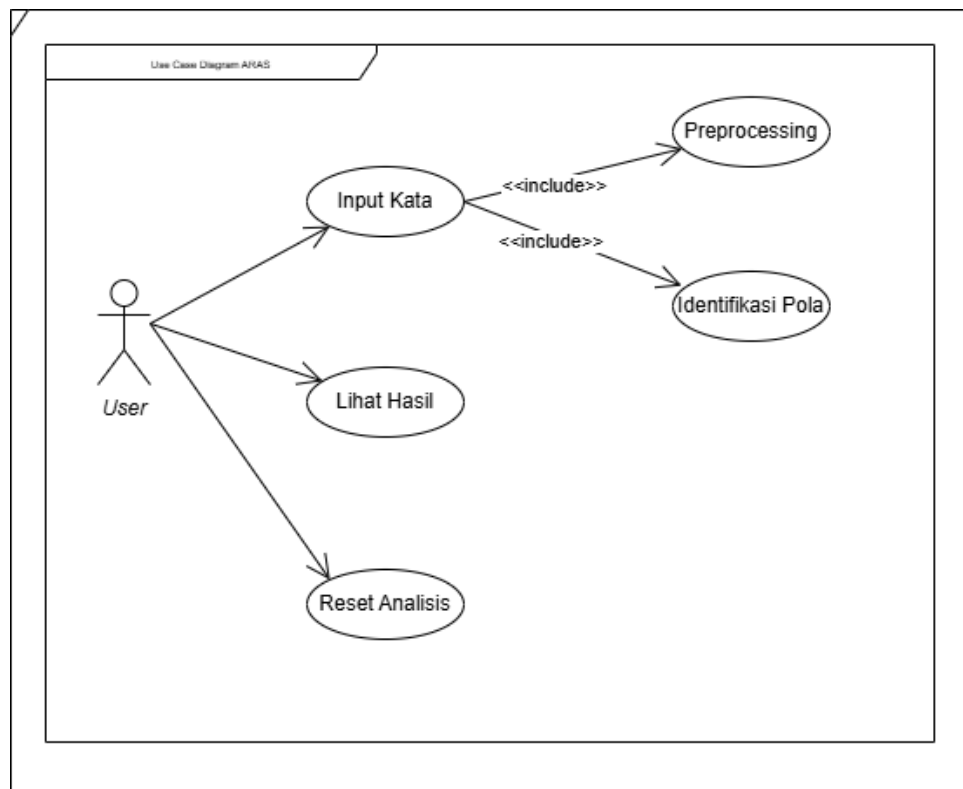
Output adalah tahap terakhir yang digunakan untuk menampilkan hasil analisis tersebut kepada pengguna. Sistem akan menampilkan informasi yang didapat selama proses pengolahan data, seperti kata yang dimasukkan, jenis bina, shigot serta akar kata yang berhasil ditarik. Informasi tersebut ditampilkan di layar sistem supaya pengguna bisa memahami hasil analisis yang sudah dilakukan dengan cara yang mudah dan terstruktur.

3.2.2 Perancangan Sistem

Pada tahap perancangan sistem, peneliti melakukan rancangan sisten menggunakan *Unified Modeling Language* (UML). UML adalah suatu metode atau cara untuk membuat model secara visual yang digunakan sebagai alat dalam merancang sistem berbasis objek. Berikut adalah rancangan sistem menggunakan UML.

3.2.2.1 Use Case Diagram

Use Case Diagram merupakan bagian dari diagram UML yang menunjukkan interaksi antara sistem dan pengguna. Pada penelitian ini, *Use case* dapat mendeskripsikan tipe interaksi antara pengguuna ARAS dengan ARAS. *Use Case* diagram ARAS dapat dilihat pada gambar 3.3 yang memberikan gambaran lengkap hubungan interaksi antara pengguna ARAS dengan ARAS itu sendiri.



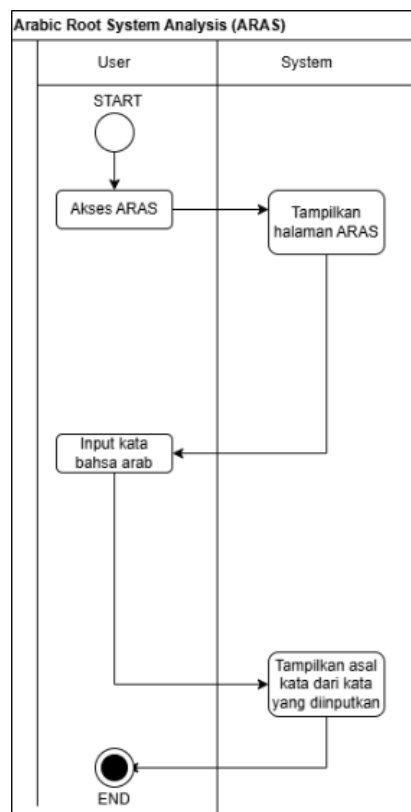
Gambar 3.3 Use case Diagram

Gambar 3.3 menggambarkan interaksi antara pengguna dengan sistem analisis bahasa Arab berbasis FSA. Pada diagram tersebut, *user* berperan sebagai aktor utama yang melakukan *input* kata bahasa Arab ke dalam sistem. Selanjutnya sistem melakukan *preprocessing*, analisis pola menggunakan FSA, identifikasi akar kata dan *wazn*, hingga menampilkan hasil analisis kepada pengguna. Diagram ini membantu menggambarkan fungsi utama sistem serta alur interaksi yang terjadi selama proses analisis berlangsung.

3.2.2.2 Activity Diagram

Activity diagram diartikan sebagai diagram aktivitas, yaitu sebuah jenis diagram yang digunakan untuk menggambarkan alur kerja atau rangkaian aktivitas

yang terjadi di dalam suatu sistem dengan cara yang terstruktur dan sistematis. Diagram aktivitas tidak hanya menunjukkan urutan langkah-langkah dalam proses, tetapi juga menjelaskan bagaimana alur dari satu aktivitas ke aktivitas berikutnya, sehingga membantu pengembang memahami cara kerja sistem secara lebih dalam. Gambar 3.4 menunjukkan diagram aktivitas pada ARAS yang memperlihatkan secara lengkap alur proses yang terjadi dalam sistem. Gambar 3.4 akan dijadikan acuan oleh peneliti dalam membangun sistem agar mampu menjadi sistem yang baik digunakan untuk semua kalangan.

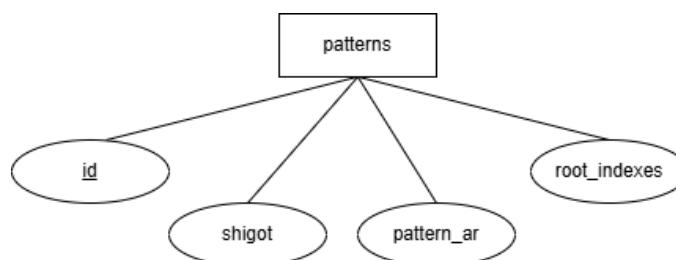


Gambar 3.4 Activity Diagram ARA

Gambar 3.4 menunjukkan proses kerja sistem ARAS mulai dari awal sampai akhir. Proses dimulai saat pengguna mengunjungi *website*, lalu sistem akan menampilkan halaman utama ARAS. Selanjutnya, pengguna memasukkan sebuah kata dalam bahasa Arab, kemudian sistem akan memproses kata tersebut dan menampilkan akar kata dari kata yang telah dimasukkan.

3.2.2.3. Entity Relationship Diagram (ERD)

ERD digunakan untuk menampilkan struktur basis data yang digunakan dalam sistem. ERD memberikan penjelasan tentang *entitas*, sifat-sifatnya, serta cara hubungan antar *entitas* dalam sebuah basis data. Dalam penelitian ini, *database* yang digunakan memiliki bentuk yang sederhana dan hanya terdiri dari satu *entitas*, yaitu tabel *patterns*. Tabel ini berfungsi sebagai tempat menyimpan pola-pola (*wazn*) bahasa Arab yang digunakan dalam proses mengenali dan mengekstrak akar kata. Diagram ERD dapat dilihat pada gambar 3.5.



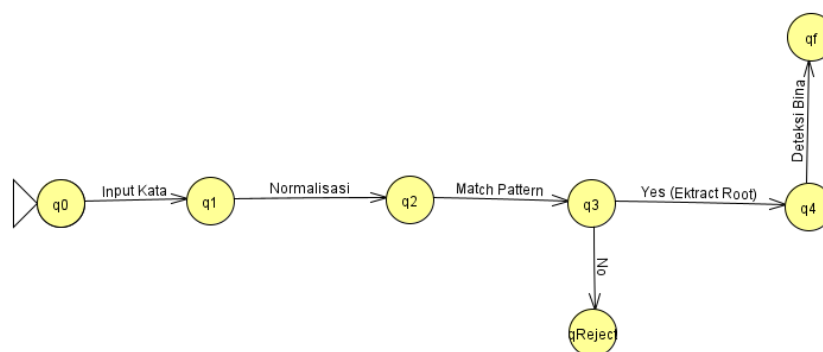
Gambar 3.5 Diagram ERD

Berdasarkan gambar 3.5 tabel *patterns* memiliki empat atribut, yaitu *id* sebagai *primary key*, *shigot*, *pattern_ar*, dan *root_indexes*. Atribut *id* ditetapkan sebagai *primary key* yang digunakan untuk membedakan setiap data pola secara unik. Atribut *shigot* digunakan untuk menyimpan jenis bentuk kata, seperti *Fi'il madhi*, *Fi'il mudhari*,

Fi'il amr, *Fi'il nahyi*, *masdar*, *isim fa'il*, *isim maf'ul*, dan bentuk-bentuk lainnya. Atribut *pattern_ar* menyimpan pola atau bentuk dalam huruf Arab yang nantinya akan dijadikan patokan oleh mesin otomatis untuk melakukan pencocokan. Sementara itu, atribut *root_indexes* menyimpan lokasi huruf-huruf dasar kata dalam pola yang digunakan dalam proses mengekstrak akar kata setelah pola tersebut berhasil dikenali.

3.4.2 Perancangan Logika FSA

Model FSA yang digunakan dalam penelitian ini adalah *Deterministic Finite Automata* (DFA), yaitu jenis model automata yang setiap *state* hanya memiliki satu transisi untuk setiap simbol input dan automata hanya berada pada satu *state* aktif pada satu waktu. DFA digunakan untuk mencocokkan kata Arab terhadap pola *sharaf* yang tersimpan pada basis data, kemudian hasil pencocokan digunakan untuk mengekstraksi akar kata dan menentukan jenis bina. Model DFA sistem yang telah dibuat untuk sistem ARAS dapat dilihat pada gambar 3.6.



Gambar 3.6 Model DFA

Pada gambar 3.6 menunjukkan rancangan FSA model DFA, penjelasan Berdasarkan gambar 3.6. Secara formal, setiap state dapat didefinisikan sebagai berikut.

1. Himpunan *State* (Q)

- a. *q0 State awal (Start State)*: *State* awal ketika sistem menerima *input* berupa kata bahasa Arab dari pengguna untuk dilakukan proses analisis morfologi.
- b. *q1 State normalisasi (Normalization State)*: *State* yang bertugas melakukan pembersihan dan normalisasi kata, seperti penghapusan *harakat*, normalisasi *hamzah*, serta penyesuaian bentuk kata agar siap diproses oleh *automata*.
- c. *q2 State transformasi (Transformation State)*: *State* yang melakukan proses transformasi kata, seperti *ekspansi tasydid* dan rekonstruksi bentuk khusus (misalnya *Ajwaf Amr* dan *Ajwaf Nahyi*) sehingga kata sesuai dengan format yang digunakan pada proses pencocokan pola.
- d. *q3 State pencocokan pola (Pattern Matching State)*: *State* yang melakukan pencocokan kata terhadap pola (*wazn*) yang tersimpan pada basis data menggunakan mekanisme DFA. Jika pola tidak sesuai maka sistem berpindah ke *state* penolakan (*Reject State*).
- e. *q4 State ekstraksi dan analisis akar kata (Root Extraction State)*: *State* yang melakukan ekstraksi akar kata berdasarkan *indeks* akar (*Root indexes*) dari pola yang berhasil dikenali. Pada tahap ini juga dilakukan analisis morfologi untuk menentukan jenis bina *ajwaf* dan *mudhoaf*.

- f. *qf State* akhir (*Final/Accept State*): *State* akhir yang menunjukkan bahwa kata berhasil dikenali.
- g. *qReject State* penolakan (*Reject State*): *State* yang dicapai apabila tidak ditemukan pola yang sesuai sehingga kata tidak dapat dikenali oleh sistem.

2. Alfabet *Input* (Σ)

Alfabet *Input* berupa:

- a. Huruf *hijaiyah* (ﻱـﻻ)
- b. Harakat Arab (*fathah*(◌َ), *kashroh*(◌ِ), *dhommah*(◌ُ), *sukun*(◌ْ), *tasydid*(◌ّ))

3. Fungsi Transisi (δ)

Fungsi transisi ditentukan oleh:

- a. Karakter *Input* (huruf/harakat)
- b. Kondisi *karakter* (hidup/mati, *tasydid*, huruf illat)

4. State Awal (q_0)

State awal adalah q_0 , yaitu saat sistem menerima *Input* kata dari pengguna.

5. State Akhir (q_f)

State akhir adalah q_f , yaitu kondisi ketika sistem berhasil melakukan ekstarksi akar kata.

Berikut adalah contoh penerapan model FSA pada gambar 3.6 untuk dua jenis kata: *Mudho 'af* (مضعف) dan *Ajwaf* (أجوف).

1. Contoh kata *Mudhoaf* (مضعف)

q0 → q1 (Prepare) = Kata مَمْدُودٌ diterima sebagai input dari pengguna.

Sistem kemudian melakukan proses normalisasi dengan menghapus harakat dan menyesuaikan bentuk karakter Arab yang diperlukan.

Input: مَمْدُودٌ

Hasil normalisasi: ممدود

q1 → q2 (Transform) = Pada tahap transformasi, sistem mempersiapkan kata agar dapat diproses oleh *automata*. Sistem melakukan pengecekan terhadap bentuk kata dan menjalankan proses transformasi yang diperlukan, seperti normalisasi *Ajwaf* atau ekspansi *tasydid* apabila ditemukan pada input.

Untuk kata ممدود tidak diperlukan transformasi tambahan sehingga kata tetap ممدود

q2 → q3 (Match Pattern) = FSA melakukan pencocokan kata terhadap pola (*wazan*) yang tersimpan pada basis data.

Kata: ممدود

berhasil dicocokkan dengan pola: مفعول

q3 → q4 (Extraction) = Setelah pola dikenali, sistem melakukan ekstraksi akar kata menggunakan nilai *root_indexes* yang tersimpan pada pola.

Dari pola yang dikenali yaitu مفعول maka diperoleh akar kata: مدد

Pada tahap ini sistem juga melakukan analisis morfologi terhadap akar kata yang telah diekstraksi.

Karena huruf kedua dan ketiga sama: د = د

maka akar kata dikategorikan sebagai Bina Mudhoaf.

Selanjutnya sistem membentuk *Fi'il Madhi* dari *root* yang ditemukan, agar sama dengan yang tertulis dalam kitab yaitu مَدَّ

q4 → qf (Final State) = Setelah seluruh proses berhasil diselesaikan, automata mencapai *Final State* (qf).

Dengan demikian sistem berhasil mengenali kata مَمْلُوءٌ sebagai kata yang berasal dari kata مَلَأَ, termasuk kategori *Bina Mudhoaf*, dan menghasilkan bentuk *Fi'il Madhi* (akar kata) مَدَّ.

2. Contoh kata *Ajwaf* (أَجْوَف)

q0 → q1 (Prepare) = Kata صُنْ diterima sebagai input dari pengguna. Sistem kemudian melakukan proses normalisasi dengan menghapus *harakat* dan menyesuaikan bentuk karakter Arab yang diperlukan.

Input: صُنْ

Hasil normalisasi: صن

q1 → q2 (Transform) = Pada tahap transformasi, sistem mempersiapkan kata agar dapat diproses oleh *automata*. Sistem melakukan pengecekan terhadap bentuk kata dan menjalankan proses transformasi yang diperlukan, Kata صن merupakan bentuk *Fi'il Amr Ajwaf* yang mengalami penghilangan huruf *illat* pada posisi tengah. Oleh karena itu sistem menjalankan fungsi *normalisasiAjwafAmr()* untuk merekonstruksi huruf *illat* yang hilang.

Proses transformasi: صون → صن

Dengan demikian kata yang akan diproses oleh FSA menjadi: صون

q2 → q3 (Match Pattern) = FSA melakukan pencocokan kata terhadap pola *wazan* yang tersimpan pada basis data.

Kata: صون

berhasil dicocokkan dengan pola: فعل

Pola tersebut dikenali melalui proses transisi *state* pada automata hingga mencapai *state* penerimaan pola.

q3 → q4 (Extraction) = Setelah pola dikenali, sistem melakukan ekstraksi akar kata menggunakan nilai *root_indexes* yang tersimpan pada tabel *patterns*.

Dari kata: صون

diperoleh huruf akar: ص و ن

sehingga menghasilkan root: صون

Pada tahap ini sistem juga melakukan analisis morfologi terhadap akar kata yang telah diekstraksi.

Karena huruf kedua (عين الفعل) berupa huruf illat و, maka akar kata tersebut dikategorikan sebagai *Bina Ajwaf*.

Selanjutnya sistem membentuk *Fi'il Madhi* dari root yang ditemukan sehingga diperoleh: صان

q4 → qf (Final State) = Karena seluruh proses berhasil diselesaikan, automata mencapai *Final State* (qf).

Dengan demikian sistem berhasil mengenali kata صُنْ sebagai bentuk *Fi'il madhi* yang berasal dari kata صون, termasuk kategori *Bina Ajwaf*, dan menghasilkan bentuk *Fi'il Madhi* صان

3.3 Implementasi

Desain *Sistem*, alur proses, dan logika FSA yang sudah dibuat sebelumnya akan diubah menjadi sistem berbasis *website* yang bisa digunakan oleh pengguna. Implementasi sistem dilakukan secara bertahap sesuai dengan metode pengembangan *Incremental*. Setiap tahapan dibuat dan dicek terlebih dahulu sebelum digabungkan dengan modul lainnya. Tujuan ini adalah agar kesalahan bisa terlihat lebih dini dan sistem bisa dibangun dengan cara yang rapi.

Sistem yang akan dibangun menggunakan bahasa pemrograman PHP sebagai bahasa pemrograman utama. PHP digunakan untuk mengimplementasikan seluruh logika sistem, termasuk proses normalisasi kata Arab, pengolahan pola *morfologi*, penerapan FSA, ekstraksi akar kata, dan koneksi dengan basis data MySQL yang menyimpan pola-pola *wazan* bahasa Arab.

3.4 Pengujian Sistem

Proses uji coba diperlukan untuk memastikan bahwa semua bagian sudah berjalan sesuai dengan rencana dan kebutuhan pengguna. Selain itu, uji coba juga bertujuan untuk menemukan kesalahan, mengecek bagaimana sistem bekerja, serta memastikan bahwa fitur yang dibuat bisa memberikan hasil yang sesuai dengan yang diharapkan.

3.4.1 Pengujian Fungsional

Pengujian terhadap sistem ARAS dilakukan dengan dua pendekatan utama untuk memastikan sistem berjalan sesuai dengan tujuan yang ditetapkan. Pertama, pengujian dilakukan terhadap kebutuhan fungsional dan *Non fungsional* menggunakan metode *blackbox testing*, di mana setiap fitur diuji berdasarkan *input*

dan *output* tanpa melihat proses internal sistem. Pengujian ini memastikan bahwa sistem mampu menerima *input* berupa kata Arab, mengidentifikasi pola, menentukan akar kata, menampilkan hasil analisis, serta menjalankan fitur. Skenario uji kebutuhan fungsional sistem dapat dilihat pada tabel 3.1.

Tabel 3.1 Skenario uji kebutuhan fungsional

NO	KEBUTUHAN FUNGSIONAL	SKENARIO UJI (BLACK BOX)	INPUT	HASIL YANG DIHARAPKAN
KF1	Sistem mampu menerima masukan berupa kata dalam bahasa Arab	Uji <i>Input</i> kata Arab tanpa harakat	ممدود	Sistem menerima <i>Input</i> dan memproses kata
		Uji <i>Input</i> teks arab berharakat	مَمْدُودٌ	Sistem menerima <i>Input</i> dan melakukan normalisasi
KF2	Sistem mampu melakukan analisis <i>User Interface</i>	<i>Input</i> kata <i>Bina Mudhoaf</i>	يَمْدُ	Sistem menganalisis struktur kata (<i>Bina Mudhoaf</i>)
		<i>Input</i> kata <i>Bina Ajwaf</i>	خَوْفًا	Sistem menganalisis struktur kata (<i>Bina mahmuz</i>)
KF3	Sistem mampu menerapkan logika FSA	<i>Input</i> kata sesuai pola yang tersedia	مَمْدُودٌ	Sistem menelusuri state FSA dan mencapai final state
KF4	Sistem mampu menghasilkan dan menampilkan akar kata	<i>Input</i> kata <i>Bina Mudhoaf</i>	مَمْدُودٌ	Sistem menampilkan <i>ROOT</i> : مَدَّ
		<i>Input</i> kata <i>Bina Ajwaf</i>	خَوْفًا	Sistem menampilkan <i>root</i> : خَافَ

Adapun untuk skenario uji kebutuhan *Non* fungsional sistem dapat dilihat pada tabel 3.2.

Tabel 3.2 Skenario uji kebutuhan *Non* fungsional

NO	KEBUTUHAN NON FUNGSIONAL	SKENARIO UJI (BLACK BOX)	HASIL YANG DIHARAPKAN
KNF1	Sistem berbasis WEBSITE dan dapat diakses melalui	Akses sistem melalui beberapa	Sistem dapat dibuka dan berjalan normal di berbagai browser tanpa instalasi tambahan

	browser tanpa instalasi tambahan	browser (Chrome, Firefox, Edge)	
KNF2	Sistem memiliki waktu respons yang cepat dalam memproses <i>Input</i> kata	<i>Input</i> beberapa kata Arab secara berulang	Sistem menampilkan hasil ekstraksi dalam waktu ≤ 3 detik

3.4.2 Pengujian FSA

Pengujian juga dilakukan terhadap ketepatan FSA dalam menentukan akar kata pada *Fi'il mujarrod* dengan melibatkan Seorang pengajar bahasa arab di MA Muallimat Kota Malang sebagai evaluator untuk menguji kata yang terdapat dalam kitab *amtsilati tashrifiyyah* yaitu sebanyak 180 kata, dengan rincian 29 bentuk *Bina Mudho'af* dan 46 bentuk *Bina ajwaf*. Ahli bahasa tersebut menilai kesesuaian hasil ekstraksi akar kata oleh sistem dengan aturan *morfologi* bahasa Arab yang benar. pengujian metode ini memastikan bahwa sistem ARAS tidak hanya berfungsi dengan baik secara teknis, tetapi menghasilkan analisis yang valid dan dapat dipertanggungjawabkan. Adapun data yang akan di ujikan untuk pengujian FSA dapat dilihat pada table 3.3.

Tabel 3.3 Data kata *bina Mudho 'af* dan *ajwaf*

No	<i>Fi'il</i> Madī	<i>Fi'il</i> Mudari'	<i>Masdar</i>	<i>Masdar</i> Mim	<i>Isim</i> <i>Fa'il</i>	<i>Isim</i> <i>Ma'ul</i>	<i>Fi'il</i> <i>Amr</i>	<i>Fi'il</i> Nahy	<i>Dhorof</i> (Makan/ Zaman)	<i>Isim</i> Alat	<i>Bina</i>
1	مَدَّ	يُمَدُّ	مَدًّا	مَمَدًّا	مَادٌّ	مَمْدُودٌ	مُدَّ	لَا تَمُدَّ	مَمَدٌّ	مِمْدٌ	<i>Mudho'af</i>
2	قَرَّ	يَقَرُّ	قَرًّا	مَقَرًّا	قَارٌّ	مَقْرُورٌ	قَرَّ	لَا تَقَرَّ	مَقَرٌّ	مَقَرٌّ	<i>Mudho'af</i>
3	عَضَّ	يَعَضُّ	عَضًّا	مَعَضًّا	عَاضٌ	مَعْضُوضٌ	عَضَّ	لَا تَعَضَّ	مَعَضٌّ		<i>Mudho'af</i>
4	صَنَّ	يَصْنُ	صَنْئًا	مَصْنَأًا	صَائِنٌ	مَصْنُونٌ	صَنَّ	لَا تَصْنَنَّ	مَصْنَانٌ	مَصْنُونٌ	<i>Ajwaf</i>
5	سَارَ	يَسِيرُ	سَيْرًا	مَسِيرَةً	سَائِرٌ	مَسِيرٌ	سِرَّ	لَا تَسِيرَنَّ	مَسِيرٌ	مَسِيرٌ	<i>Ajwaf</i>
6	خَافَ	يَخَافُ	خَوْفًا	مَخَافًا	خَائِفٌ	مَخُوفٌ	خَفَّ	لَا تَخَفَنَّ	مَخَافٌ		<i>Ajwaf</i>
7	هَابَ	يَهَابُ	هَيْبَةً	مَهَابَةً	هَائِبٌ	مَهِيْبٌ	هَبَّ	لَا تَهَبَنَّ	مَهَابٌ		<i>Ajwaf</i>
8	طَالَ	يَطُولُ	طَوْلًا	مَطَالًا	طَوِيلٌ		طَلَّ	لَا تَطْلَنَّ	مَطَالٌ		<i>Ajwaf</i>

3.4.3 Pengujian *Usability*

Evaluasi sistem dilakukan agar mengetahui sejauh mana sistem yang sudah dibuat itu mudah digunakan dan sesuai dengan pengalaman pengguna saat memakai sistem tersebut. Metode yang akan digunakan adalah SUS, yang dikemukakan oleh *John Brooke* pada tahun 1986.

Pada evaluasi sistem menggunakan SUS responden akan diminta untuk menggunakan sistem terlebih dahulu, kemudian mengisi kuesioner SUS berdasarkan pengalaman yang diperoleh selama menggunakan sistem. Pengisian kuesioner dilakukan melalui *Google Form (gform)* setelah proses penggunaan sistem selesai agar tanggapan yang diberikan mencerminkan pengalaman pengguna secara langsung. Pernyataan yang akan peneliti gunakan adalah hasil terjemahan dari pernyataan *usability* SUS oleh *Brooke*. Pernyataan hasil terjemahan dapat dilihat pada tabel 3.4.

Tabel 3.4 Pernyataan kuisoner SUS

NO	Pernyataan
P1	Saya ingin menggunakan sistem ini secara sering.
P2	Saya merasa sistem ini terlalu rumit.
P3	Sistem ini mudah digunakan
P4	Saya memerlukan bantuan orang teknis untuk menggunakan sistem ini.
P5	Fitur-fitur dalam sistem ini terintegrasi dengan baik.
P6	Saya menemukan banyak ketidakkonsistenan dalam sistem ini.
P7	Saya merasa kebanyakan orang akan belajar menggunakan sistem ini dengan sangat cepat.
P8	Saya merasa sistem ini sangat sulit digunakan.
P9	Saya merasa percaya diri saat menggunakan sistem ini.
P10	Saya perlu mempelajari banyak hal sebelum dapat menggunakan sistem ini.

Skor SUS dihitung dengan cara mengurangi angka 1 dari nilai jawaban pada pernyataan positif yaitu nomor 1, 3, 5, 7, dan 9. Sementara itu, untuk pernyataan negatif yaitu nomor 2, 4, 6, 8, dan 10, nilai jawaban responden dikurangi dengan

angka 5. Seluruh nilai kontribusi kemudian dijumlahkan dan hasilnya dikalikan dengan 2,5 sehingga diperoleh skor SUS dengan rentang nilai 0–100. Semakin tinggi skor yang diperoleh, maka semakin baik tingkat *Usability* sistem yang diuji (Brooke, 2020).

BAB IV

HASIL DAN PEMBAHASAN

Pada bagian ini, peneliti akan menjelaskan bagaimana cara melaksanakan dan hasil dari pembangunan ARAS yang menggunakan FSA dalam menganalisis *User Interface* bahasa Arab. Di bagian akhir, akan dibahas hasil pengujian sistem untuk melihat seberapa berhasil sistem yang telah dibangun. Berikut adalah tahapan pengembangan sistem secara *Incremental* .

4.1 Increment pertama

Tahapan ini dimulai dengan fokus pada pengembangan antarmuka pengguna (*User Interface/UI*) untuk sistem yang akan dibangun. Tahap ini bertujuan untuk membuat cara berinteraksi antara pengguna dan sistem, sehingga proses memasukkan kata, mengirim data, dan menampilkan hasil analisis dapat dilakukan dengan mudah. Pada *increment* pertama, penerapan FSA belum sepenuhnya dilaksanakan. Fokus utama pengembangan adalah merancang halaman utama sistem, membuat form untuk memasukkan kata-kata dalam bahasa Arab.

4.1.1 Requirements

Pada *increment* pertama dilakukan identifikasi kebutuhan dasar sistem yang akan digunakan sebagai fondasi pembangunan ARAS. Berdasarkan kebutuhan pengguna, sistem harus mampu menampilkan antarmuka yang sederhana untuk menerima masukan kata bahasa Arab dan menampilkan hasil analisis yang diproses oleh sistem. *Requirement* yang ditetapkan pada *increment* pertama meliputi:

1. Sistem dapat menerima *input* kata bahasa Arab dari pengguna.
2. Sistem menyediakan tombol untuk menjalankan proses analisis.
3. Sistem dapat menampilkan hasil proses analisis.
4. Sistem memiliki tampilan yang mudah digunakan oleh pengguna.

4.1.2 Analysis

Berdasarkan kebutuhan yang telah ditentukan, dilakukan analisis terhadap alur kerja sistem. Hasil analisis menunjukkan bahwa proses interaksi pengguna terdiri dari beberapa tahapan utama, yaitu memasukkan kata, melakukan pemrosesan, dan menampilkan hasil ke halaman web.

Analisis juga menunjukkan bahwa antarmuka harus mampu mendukung pengembangan *increment* berikutnya karena seluruh proses analisis *morfologi* nantinya akan dimulai dari data yang dimasukkan melalui halaman utama. Oleh karena itu, diperlukan komponen input yang mendukung karakter Arab, tombol eksekusi analisis, serta area untuk menampilkan hasil proses.

4.1.3 Perancangan antarmuka halaman utama

Perancangan antarmuka dilakukan untuk memudahkan pengguna dalam melakukan analisis *root* kata bahasa Arab. Desain UI dibuat sederhana agar pengguna dapat langsung memasukkan kata tanpa melalui proses konfigurasi tambahan. Komponen utama pada halaman terdiri dari:

1. Form *Input* kata Arab
2. Tombol proses analisis
3. Informasi hasil analisis

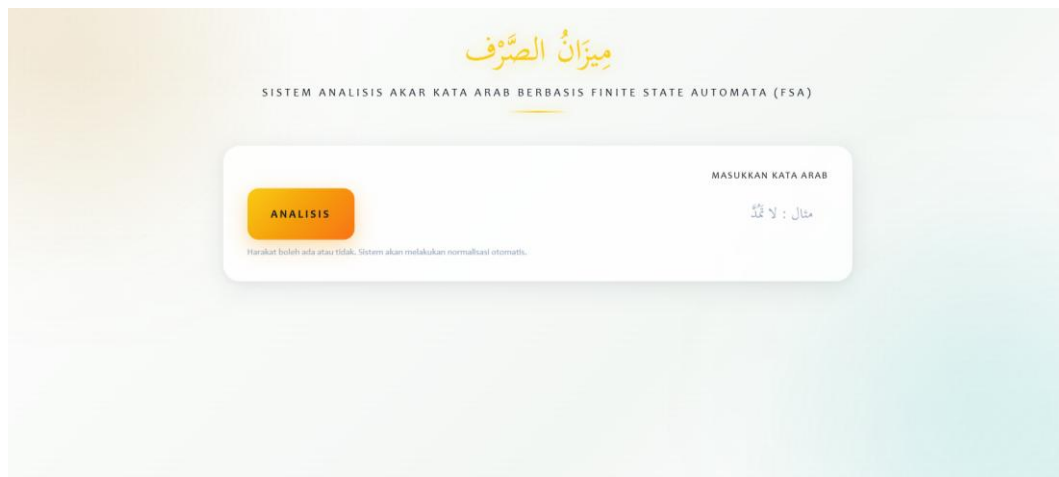
Secara konseptual, alur antarmuka sistem adalah sebagai berikut:

1. Pengguna memasukkan kata bahasa Arab
2. Sistem menerima *Input*
3. Sistem memproses analisis
4. Hasil ditampilkan pada halaman

Antarmuka ini menjadi fondasi implementasi FSA karena seluruh proses *state transition* nantinya dimulai dari *Input* pengguna pada halaman utama.

4.1.4 Hasil *increment* pertama

Gambar 4.1 adalah hasil *increment* pertama yaitu berupa tampilan UI untuk halaman utama ARAS.



Gambar 4.1 Tampilan UI halaman ARAS

Hasil *increment* pertama menunjukkan bahwa sistem telah mampu menampilkan halaman utama analisis dan menerima *Input* kata bahasa Arab.

4.2 *Increment* kedua

Increment kedua difokuskan pada implementasi FSA untuk menangani *Fi'il Ajwaf*. *Fi'il Ajwaf* adalah kata yang dimana pada huruf tengahnya atau huruf keduanya (*root*) berupa huruf *illat*. Implementasi *increment* kedua bertujuan agar sistem mampu merekonstruksi huruf *illat* yang hilang akibat perubahan kata bahasa Arab.

4.2.1 *Requirements*

Setelah antarmuka berhasil dibangun pada *State* pertama, kebutuhan berikutnya adalah kemampuan sistem untuk mengenali dan merekonstruksi kata yang termasuk dalam kategori *Fi'il Ajwaf*.

Kebutuhan fungsional pada *increment* kedua meliputi:

1. Sistem mampu mengenali pola *Fi'il Ajwaf*.
2. Sistem mampu merekonstruksi huruf *illat* yang hilang.
3. Sistem mampu membedakan bentuk *Ajwaf Amr* dan *Ajwaf Nahyi*.
4. Sistem dapat menghasilkan bentuk *root* yang sesuai sebelum proses ekstraksi akar kata dilakukan.
5. Sistem dapat mengintegrasikan proses rekonstruksi dengan mekanisme FSA.

4.2.2 *Analysis*

Fi'il Ajwaf adalah kata kerja yang memiliki huruf *illat* (و atau ي) yang berada di posisi huruf kedua dalam akar kata tersebut. Dalam pembentukan kata, huruf *illat* sering mengalami perubahan bentuk atau bahkan dihilangkan, sehingga

bentuk permukaan kata berbeda dengan bentuk aslinya. Kondisi ini membuat proses pencocokan pola di FSA tidak selalu bisa mengenali akar kata secara langsung. Selain itu, diketahui bahwa bentuk *Fi'il Ajwaf Amr* dan *Fi'il Ajwaf Nahyi* memiliki perbedaan dalam karakteristik morfologinya. Dalam beberapa situasi, huruf *illat* tidak muncul dalam bentuk kata yang dimasukkan, sehingga diperlukan langkah rekonstruksi terlebih dahulu sebelum melakukan ekstraksi akar kata. Tanpa melalui proses rekonstruksi, sistem bisa menghasilkan akar kata yang tidak sesuai dengan aturan sharaf.

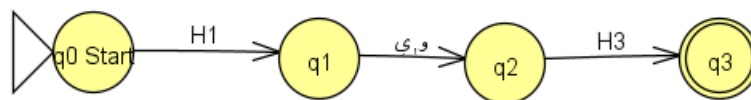
Untuk menyelesaikan masalah tersebut, diperlukan sebuah mekanisme tambahan yang mampu mengenali pola *Fi'il Ajwaf*, membedakan jenis perubahan yang terjadi, serta merekonstruksi huruf *illat* yang telah hilang. Dengan demikian, sistem bisa menghasilkan bentuk akar kata yang lebih tepat meskipun kata yang dimasukkan sudah mengalami perubahan bentuk karena sifat *Fi'il Ajwaf*.

4.2.3 Perancangan FSA *Ajwaf*

Berdasarkan hasil analisis terhadap pola *fi'il ajwaf*, diketahui bahwa jenis kata kerja ini memiliki ciri khusus berupa adanya huruf *illat* (ا atau ي) pada posisi *'ain fi'il*. Dalam beberapa bentuk *tashrif*, huruf tersebut bisa mengalami perubahan atau bahkan tidak terlihat dalam bentuk kata yang digunakan. Situasi ini membuat bentuk kata yang diterima oleh sistem berbeda dari bentuk dasarnya, sehingga diperlukan langkah identifikasi dan penyesuaian terlebih dahulu sebelum mencari akar kata.

Untuk menggambarkan proses pengenalan akar kata dari *fi'il ajwaf*, digunakan model Finite State Automata (FSA) yang dibuat berdasarkan urutan

huruf yang membentuk akar kata tersebut. Dalam model ini, automata beralih dari state awal ke state akhir dengan membaca huruf pertama dari akar kata, lalu huruf illat (ا atau ي) yang merupakan ciri khas dari fi'il ajwaf, serta huruf ketiga dari akar kata. Rencana FSA untuk fi'il ajwaf dapat dilihat pada Gambar 4.2.



Gambar 4.2 FSA *Ajwaf*

Gambar 4.2 menampilkan Finite State Automata (FSA) yang digunakan untuk mengenali akar kata fi'il ajwaf. State q0 adalah state awal yang menerima huruf pertama dari akar kata (fa' fi'il). Setelah itu automata berpindah ke state q1. Dari state q1 ke q2 hanya bisa dilakukan jika huruf yang dibaca adalah huruf illat, yaitu ا atau ي, yang berada pada posisi 'ain fi'il dalam fi'il ajwaf'. Selanjutnya, automata membaca huruf ketiga dari akar kata (lam fi'il) melalui transisi H3 dan berpindah ke state q3 sebagai state akhir. Jika semua transisi berhasil dilalui hingga mencapai Q3, maka kata tersebut diterima sebagai akar kata yang memiliki ciri khas fi'il ajwaf.

4.2.4 Hasil Implementasi *Ajwaf*

Implementasi FSA pada *incremental* kedua berhasil meningkatkan kemampuan sistem dalam menangani perubahan huruf *illat*, karena sistem menunjukkan berhasil mengenali pola *Fi'il Ajwaf*. Hasil dari increment ke-2 dapat dilihat pada gambar 4.3.

			HASIL ANALISIS FSA
AKAR KATA	SHIGOT	BINA	KATA ASLI
صان	Masdar	Ajwaf	صَوْنًا

Gambar 4.3 Hasil *Testing bina ajwaf*

Gambar 4.3 menampilkan hasil dari kata yang diinputkan. Hasil tersebut menampilkan kata asli, bina shigot, dan akar kata. Dengan adanya output yang jelas seperti pada gambar 4.3 user akan dengan mudah memahami bentuk kata yang diinputkan tersebut.

4.3 *Increment* ketiga

Increment ketiga difokuskan pada implementasi FSA untuk *Fi'il Mudhoaf*. *Fi'il Mudhoaf* merupakan kata kerja yang memiliki huruf kedua dan ketiga sama sehingga mengalami *tasydid*. Tujuan *increment* ketiga adalah agar sistem mampu melakukan ekspansi *tasydid* dan merekonstruksi *ROOT* secara benar.

4.3.1 *Requirements*

Increment ketiga berfokus pada penanganan *Fi'il Mudho'af*. Pada jenis kata ini terdapat dua huruf yang sama dan digabungkan menggunakan *tasydid* sehingga menyebabkan jumlah huruf yang terlihat berbeda dengan bentuk *root* aslinya. Kebutuhan fungsional yang ditetapkan meliputi:

1. Sistem mampu mendeteksi keberadaan *tasydid*.
2. Sistem mampu melakukan ekspansi huruf yang mengalami *tasydid*.
3. Sistem mampu mengembalikan bentuk kata menjadi susunan huruf lengkap.

4. Sistem mampu mengekstraksi *root Fi'il Mudho'af* secara benar.
5. Sistem mampu mengintegrasikan proses ekspansi *tasydid* dengan mekanisme FSA.

4.3.2 Analysis

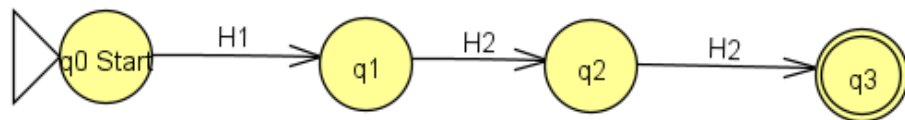
Hasil analisis menunjukkan bahwa tantangan utama pada *Fi'il Mudho'af* terletak pada penggunaan *tasydid* yang merepresentasikan pengulangan huruf. Jika *tasydid* tidak diekspansi terlebih dahulu, maka sistem akan membaca jumlah huruf lebih sedikit dibandingkan bentuk *root* sebenarnya sehingga proses identifikasi pola menjadi tidak akurat.

Berdasarkan hasil analisis tersebut, diperlukan mekanisme yang mampu mendeteksi simbol *tasydid* dan menggandakan huruf sebelumnya sebelum proses pencocokan pola dilakukan. Mekanisme ini kemudian diimplementasikan untuk menangani transformasi *Fi'il Mudho'af*.

4.3.3 Perancangan FSA *Mudho'af*

Berdasarkan hasil analisis terhadap pola *fi'il mudha'af*, terlihat bahwa jenis kata kerja ini memiliki ciri khusus yaitu kesamaan antara huruf 'ain *fi'il* dan lam *fi'il*. Dalam penulisannya, kedua huruf tersebut biasanya digabungkan dengan *tasydid*, sehingga salah satu huruf tidak terlihat secara jelas. Kondisi ini membuat bentuk kata yang diterima oleh sistem berbeda dari bentuk dasar katanya. Oleh karena itu, dibutuhkan proses mengenali dan memperluas huruf yang memiliki *tasydid* sebelum mencari akar kata. Untuk menggambarkan proses tersebut, dibuat Finite State Automata (FSA) yang mengatur pembacaan huruf pertama dari akar

kata, lalu huruf kedua, dan diakhiri dengan huruf ketiga yang nilainya sama dengan huruf kedua. Diagram FSA untuk fi'il mudha'af dapat dilihat pada Gambar 4.4.



Gambar 4.4 FSA *mudho'af*

Gambar 4.4 menunjukkan FSA yang digunakan untuk menggambarkan pola fi'il mudha'af. Proses dimulai dari state awal yaitu q0 dengan membaca huruf pertama dari akar kata (H1), sehingga automata berpindah ke state q1. Selanjutnya, huruf kedua dari akar kata (H2) dibaca dan menyebabkan perpindahan ke state q2. Ciri khas dari fi'il mudha'af terdapat pada kesamaan huruf kedua (huruf 'ain pada fi'il) dan huruf ketiga (huruf lam pada fi'il). Oleh karena itu, peralihan dari q2 ke q3 juga menggunakan simbol H2 yang sama. Jika semua langkah transisi berhasil dilalui hingga mencapai state akhir q3, maka kata dasar tersebut diterima sebagai bentuk yang sesuai dengan ciri-ciri fi'il mudha'af. Sebagai contoh, akar kata *مَدَد* diproses dengan huruf *م* sebagai H1, kemudian diikuti oleh *د* sebagai H2, dan diakhiri dengan huruf *د* yang sama sebagai huruf ketiga, sehingga automata mencapai keadaan akhir.

4.3.4 Hasil Implementasi *Mudhoaf*

Hasil pengujian *increment* ketiga menunjukkan bahwa sistem berhasil melakukan rekonstruksi *root Fi'il Mudhoaf*. Hasil dari *increment* ketiga ini adalah

FSA mampu meningkatkan kemampuan sistem dalam menangani pola *tasydid* dan *Fi'il Mudhoaf*. Hasil implementasi mudho'af dapat dilihat pada gambar 4.5.

			HASIL ANALISIS FSA
AKAR KATA	SHIGOT	BINA	KATA ASLI
مَدَّ	Isim Ma'ful	Mudhoaf	مَشْدُوذٌ

Gambar 4.5 Hasil *Testing bina Mudho'af*

Sama halnya dengan Gambar 4.3. Gambar 4.5 menampilkan hasil dari kata yang diinputkan berupa mudhoaf. Hasil tersebut menampilkan kata asli, bina shigot, dan akar kata.

4.4 Pengujian dan Evaluasi

Pada penelitian ini ada 3 aspek pengujian, yang pertama yaitu pengujian kebutuhan fungsional yang dilakukan untuk memastikan bahwa seluruh fungsi utama pada sistem analisis bahasa Arab berbasis *Finite State Automata* (FSA) dapat berjalan sesuai dengan kebutuhan yang telah dirancang. Metode pengujian yang digunakan adalah *Black Box Testing*, yaitu metode pengujian yang berfokus pada fungsi sistem tanpa melihat struktur internal kode program. Kemudian pengujian FSA yang dilakukan untuk mengetahui seberapa besar akurasi FSA dalam menganalisis akar kata. Terakhir adalah pengujian *usability*, pengujian ini dilakukan untuk mengetahui kelayakan sistem untuk digunakan oleh pengguna.

4.4.1 Pengujian dan Evaluasi Kebutuhan Fungsional dan *Non Fungsional*

Pada pengujian ini, sistem diuji berdasarkan *Input* dan *Output* yang dihasilkan. Peneliti melakukan pengujian menggunakan *blackbox testing* apakah

fitur yang tersedia dapat menghasilkan keluaran sesuai dengan spesifikasi kebutuhan fungsional. Berikut adalah hasil pengujian menggunakan blackbox *Testing*.

Tabel 4.1 Hasil pengujian kebutuhan fungsional

NO	KEBUTUHAN FUNGSIONAL	SKENARIO UJI (BLACK BOX)	INPUT	HASIL YANG DIHARAPKAN	HASIL PENGUJIAN
KF1	Sistem mampu menerima masukan berupa kata dalam bahasa Arab	Uji <i>Input</i> kata Arab tanpa harakat	ممدود	Sistem menerima <i>Input</i> dan memproses kata	Berhasil
		Uji <i>Input</i> teks arab berharakat	مَمْدُودٌ	Sistem menerima <i>Input</i> dan melakukan normalisasi	Berhasil
KF2	Sistem mampu melakukan analisis <i>User Interface</i>	<i>Input</i> kata <i>Bina Mudhoaf</i>	يَمْدُ	Sistem menganalisis struktur kata (<i>Bina Mudhoaf</i>)	Berhasil
		<i>Input</i> kata <i>Bina Ajwaf</i>	حَوْفًا	Sistem menganalisis struktur kata (<i>Bina mahmuz</i>)	Berhasil
KF3	Sistem mampu menerapkan logika FSA	<i>Input</i> kata sesuai pola yang tersedia	مَمْدُودٌ	Sistem menelusuri state FSA dan mencapai final state	Berhasil
KF4	Sistem mampu menghasilkan dan menampilkan akar kata	<i>Input</i> kata <i>Bina Mudhoaf</i>	مَمْدُودٌ	Sistem menampilkan <i>ROOT</i> : مَدَّ	Berhasil
		<i>Input</i> kata <i>Bina Ajwaf</i>	حَوْفًا	Sistem menampilkan <i>ROOT</i> : خَافَتْ	Berhasil

Tabel 4.2 Hasil pengujian kebutuhan *Non* fungsional

NO	KEBUTUHAN NON FUNGSIONAL	SKENARIO UJI (BLACK BOX)	HASIL YANG DIHARAPKAN	HASIL PENGUJIAN
KNF1	Sistem berbasis WEBSITE dan dapat diakses melalui browser tanpa instalasi tambahan	Akses sistem melalui beberapa browser (Chrome, Firefox, Edge)	Sistem dapat dibuka dan berjalan normal di berbagai browser tanpa instalasi tambahan	Berhasil
KNF2	Sistem memiliki waktu respons yang cepat dalam memproses <i>Input</i> kata	<i>Input</i> beberapa kata Arab secara berulang	Sistem menampilkan hasil ekstraksi dalam waktu ≤ 3 detik	Berhasil

Dari hasil pengujian kebutuhan fungsional dan *Non* fungsional dapat diketahui dengan menghitung akurasi dengan melihat pada formula 4.1:

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah pengujian berhasil}}{\text{Total pengujian}} \times 100 \quad (4.1)$$

Jumlah pengujian berhasil = 9

Total pengujian = 9

Maka:

Persentase keberhasilan = $9/9 * 100\%$

Hasil perhitungan:

Persentase keberhasilan = 100%

Hasil pengujian di atas menunjukkan persentase akurasi sebesar 100% pada pengujian kebutuhan fungsional dan *Non* fungsional. Hasil ini menunjukkan bahwa sistem yang dibangun sesuai dengan apa yang dibutuhkan secara fungsional maupun *Non* fungsional.

4.4.2 Pengujian dan Evaluasi FSA

Pengujian dilakukan terhadap ketepatan FSA dalam menentukan akar kata pada *fil mujarrod* dengan melibatkan Seorang pengajar bahasa arab di MA Muallimat Kota Malang sebagai evaluator untuk menguji kata pada kitab *amtsilati tashrifiiyyah* yang termasuk kedalam *Bina Mudhoaf* dan *Ajwaf*, 29 kata *Bina Mudhoaf* dan 46 kata *Bina Ajwaf*. Tabel 4.3 menunjukkan hasil pengujian dari setiap kata.

Tabel 4.3 Hasil pengujian FSA *bina Mudho'af*

No	Kata	<i>Bina</i> Kitab	<i>Shigot</i> Kitab	<i>ROOT</i> Kitab	<i>Bina</i> Sistem	<i>Shigot</i> Sistem	<i>ROOT</i> Sistem	Penilaian
1	مَدَّ	<i>Mudhoaf</i>	<i>Fi'il</i> <i>Madhi</i>	مَدَّ	<i>Mudhoaf</i>	<i>Fi'il</i> <i>Madhi</i>	مَدَّ	SESUAI
2	يَمُدُّ	<i>Mudhoaf</i>	<i>Fi'il</i> <i>Mudhari</i>	مَدَّ	<i>Mudhoaf</i>	<i>Fi'il</i> <i>Mudhari</i>	مَدَّ	SESUAI
3	مَدَّا	<i>Mudhoaf</i>	<i>Masdar</i>	مَدَّ	<i>Mudhoaf</i>	<i>Masdar</i>	مَدَّ	SESUAI
4	مَمَدَّا	<i>Mudhoaf</i>	<i>Masdar</i>	مَدَّ	<i>Mudhoaf</i>	<i>Masdar</i>	مَدَّ	SESUAI
5	مَادَّ	<i>Mudhoaf</i>	<i>Isim</i> <i>Fa'il</i>	مَدَّ	<i>Mudhoaf</i>	<i>Isim</i> <i>Fa'il</i>	مَدَّ	SESUAI

No	Kata	Bina Kitab	Shigot Kitab	ROOT Kitab	Bina Sistem	Shigot Sistem	ROOT Sistem	Penilaian
6	مَمْدُودٌ	Mudhoaf	Isim Maf'ul	مَدَّ	Mudhoaf	Isim Maf'ul	مَدَّ	SESUAI
7	مُدَّ	Mudhoaf	Fi'il Amr	مَدَّ	Mudhoaf	Fi'il Madhi	مَدَّ	TIDAK
8	لَا تَمُدَّ	Mudhoaf	Fi'il Nahyi	مَدَّ	Mudhoaf	Fi'il Nahyi	مَدَّ	SESUAI
9	مَمَدَّ	Mudhoaf	Dhorof	مَدَّ	Mudhoaf	Dhorof	مَدَّ	SESUAI
10	مِمَدَّ	Mudhoaf	Isim Alat	مَدَّ	Mudhoaf	Dhorof	مَدَّ	TIDAK
11	فَرَّ	Mudhoaf	Fi'il Madhi	فَرَّ	Mudhoaf	Fi'il Madhi	فَرَّ	SESUAI
12	يَفِرُّ	Mudhoaf	Fi'il Mudhari	فَرَّ	Mudhoaf	Fi'il Mudhari	فَرَّ	SESUAI
13	فَرَّأَ	Mudhoaf	Masdar	فَرَّ	Mudhoaf	Masdar	فَرَّ	SESUAI
14	مَفَرَّأَ	Mudhoaf	Masdar	فَرَّ	Mudhoaf	Masdar	فَرَّ	SESUAI
15	فَارَّ	Mudhoaf	Isim Fa'il	فَرَّ	Mudhoaf	Isim Fa'il	فَرَّ	SESUAI
16	مَفَرَّوْرٌ	Mudhoaf	Isim Maf'ul	فَرَّ	Mudhoaf	Isim Maf'ul	فَرَّ	SESUAI
17	فَرَّ	Mudhoaf	Fi'il Amr	فَرَّ	Mudhoaf	Fi'il Madhi	فَرَّ	TIDAK
18	لَا تَفِرَّ	Mudhoaf	Fi'il Nahyi	فَرَّ	Mudhoaf	Fi'il Nahyi	فَرَّ	SESUAI
19	مَفَرَّ	Mudhoaf	Dhorof	فَرَّ	Mudhoaf	Dhorof	فَرَّ	SESUAI
20	مِيفَرَّ	Mudhoaf	Isim Alat	فَرَّ	Mudhoaf	Dhorof	فَرَّ	TIDAK
21	عَضَّ	Mudhoaf	Fi'il Madhi	فَرَّ	Mudhoaf	Fi'il Madhi	فَرَّ	SESUAI
22	يَعَضُّ	Mudhoaf	Fi'il Mudhari	فَرَّ	Mudhoaf	Fi'il Mudhari	فَرَّ	SESUAI
23	عَضَّا	Mudhoaf	Masdar	فَرَّ	Mudhoaf	Masdar	فَرَّ	SESUAI
24	مَعَضَّا	Mudhoaf	Masdar	فَرَّ	Mudhoaf	Masdar	فَرَّ	SESUAI
25	عَاضُّ	Mudhoaf	Isim Fa'il	فَرَّ	Mudhoaf	Isim Fa'il	فَرَّ	SESUAI
26	مَعَضُّو ضٌ	Mudhoaf	Isim Maf'ul	فَرَّ	Mudhoaf	Isim Maf'ul	فَرَّ	SESUAI
27	عَضَّ	Mudhoaf	Fi'il Amr	فَرَّ	Mudhoaf	Fi'il Madhi	فَرَّ	TIDAK
28	لَا تَعْضُّ	Mudhoaf	Fi'il Nahyi	فَرَّ	Mudhoaf	Fi'il Nahyi	فَرَّ	SESUAI
29	مَعَضَّ	Mudhoaf	Dhorof	فَرَّ	Mudhoaf	Dhorof	فَرَّ	SESUAI

Dilihat dari hasil pengujian kata *Bina Mudhoaf* di atas, dapat diketahui bahwa:

$$\text{Akurasi FSA} = \frac{\text{Jumlah kata sesuai}}{\text{Total pengujian}} \times 100 \quad (4.2)$$

Jumlah kata sesuai = 24 kata

Total pengujian = 29 kata

Maka:

Akurasi FSA = $24/29 \times 100\%$

Hasil perhitungan:

Persentase Akurasi = 82%

Berdasarkan hasil perhitungan, diperoleh tingkat akurasi sebesar 82% pada pengujian kata Bina Mudho'af menggunakan metode Finite State Automata (FSA). Hasil tersebut menunjukkan bahwa sebagian besar data uji berhasil diproses dan diidentifikasi dengan benar oleh sistem, meskipun masih terdapat beberapa kasus yang belum dapat dikenali secara akurat.

Tabel 4.4 Hasil pengujian FSA *bina ajwaf*

No	Kata	<i>Bina Kitab</i>	<i>Shigot</i>	<i>ROOT Kitab</i>	<i>Bina Sistem</i>	<i>Shigot Sistem</i>	<i>ROOT Sistem</i>	Penilaian
1	صَانَ	<i>Ajwaf</i>	<i>Fi'il Madhi</i>	صَانَ	<i>Ajwaf</i>	<i>Fi'il Madhi</i>	صَانَ	SESUAI
2	يَصُونُ	<i>Ajwaf</i>	<i>Fi'il Mudhari</i>	صَانَ	<i>Ajwaf</i>	<i>Fi'il Mudhari</i>	صَانَ	SESUAI
3	صَوْنًا	<i>Ajwaf</i>	<i>Masdar</i>	صَانَ	<i>Ajwaf</i>	<i>Masdar</i>	صَانَ	SESUAI
4	مَصَانًا	<i>Ajwaf</i>	<i>Masdar</i>	صَانَ	<i>Ajwaf</i>	<i>Masdar</i>	صَانَ	SESUAI
5	صَائِنُ	<i>Ajwaf</i>	<i>Isim Fa'il</i>	صَانَ	<i>Ajwaf</i>	<i>Isim Fa'il</i>	صَانَ	SESUAI
6	مَصُونُ	<i>Ajwaf</i>	<i>Isim Maf'ul</i>	صَانَ	<i>Ajwaf</i>	<i>Dhorof</i>	صَانَ	TIDAK
7	صُنْ	<i>Ajwaf</i>	<i>Fi'il Amr</i>	صَانَ	<i>Ajwaf</i>	<i>Fi'il Madhi</i>	صَانَ	TIDAK

No	Kata	Bina Kitab	Shigot	ROOT Kitab	Bina Sistem	Shigot Sistem	ROOT Sistem	Penilaian
8	لَا تَنْصُنْ	Ajwaf	Fi'il Nahyi	صَنَّ	Ajwaf	Fi'il Nahyi	صَنَّ	SESUAI
9	مَصَانٌ	Ajwaf	Dhorof	صَنَّ	Ajwaf	Dhorof	صَنَّ	SESUAI
10	مَصُونٌ	Ajwaf	Isim Alat	صَنَّ	Ajwaf	Dhorof	صَنَّ	TIDAK
11	سَارَ	Ajwaf	Fi'il Madhi	سَارَ	Ajwaf	Fi'il Madhi	سَارَ	SESUAI
12	يَسِيرُ	Ajwaf	Fi'il Mudhari	سَارَ	Ajwaf	Fi'il Mudhari	سَارَ	SESUAI
13	سَيَّرَا	Ajwaf	Masdar	سَارَ	Ajwaf	Masdar	سَارَ	SESUAI
14	مَسِيرَةً	Ajwaf	Masdar	سَارَ	Ajwaf	Masdar	سَارَ	SESUAI
15	سَائِرٌ	Ajwaf	Isim Fa'il	سَارَ	Ajwaf	Isim Fa'il	سَارَ	SESUAI
16	مَسِيرٌ	Ajwaf	Isim Maf'ul	سَارَ	Ajwaf	Dhorof	سَارَ	TIDAK
17	سِرَ	Ajwaf	Fi'il Amr	سَارَ	Ajwaf	Fi'il Madhi	سَارَ	SESUAI
18	لَا تَسِيرُ	Ajwaf	Fi'il Nahyi	سَارَ	Ajwaf	Fi'il Nahyi	سَارَ	SESUAI
19	مَسِيرٌ	Ajwaf	Dhorof	سَارَ	Ajwaf	Dhorof	سَارَ	SESUAI
20	مَسِيرٌ	Ajwaf	Isim Alat	سَارَ	Ajwaf	Dhorof	سَارَ	TIDAK
21	خَافَ	Ajwaf	Fi'il Madhi	خَافَ	Ajwaf	Fi'il Madhi	خَافَ	SESUAI
22	يَخَافُ	Ajwaf	Fi'il Mudhari	خَافَ	Ajwaf	Fi'il Mudhari	خَافَ	SESUAI
23	خَوْفًا	Ajwaf	Masdar	خَافَ	Ajwaf	Masdar	خَافَ	SESUAI
24	مَخَافَةً	Ajwaf	Masdar	خَافَ	Ajwaf	Masdar	خَافَ	SESUAI
25	خَائِفٌ	Ajwaf	Isim Fa'il	خَافَ	Ajwaf	Isim Fa'il	خَافَ	SESUAI
26	مَخُوفٌ	Ajwaf	Isim Maf'ul	خَافَ	Ajwaf	Dhorof	خَافَ	TIDAK
27	خَفَ	Ajwaf	Fi'il Amr	خَافَ	Ajwaf	Fi'il Madhi	خَافَ	TIDAK
28	لَا تَخَفْ	Ajwaf	Fi'il Nahyi	خَافَ	Ajwaf	Fi'il Nahyi	خَافَ	SESUAI
29	مَخَافٌ	Ajwaf	Dhorof	خَافَ	Ajwaf	Dhorof	خَافَ	SESUAI
30	هَابٌ	Ajwaf	Fi'il Madhi	هَابٌ	Ajwaf	Fi'il Madhi	هَابٌ	SESUAI

No	Kata	Bina Kitab	Shigot	ROOT Kitab	Bina Sistem	Shigot Sistem	ROOT Sistem	Penilaian
31	يَهَابُ	<i>Ajwaf</i>	<i>Fi'il Mudhari</i>	هَابَ	<i>Ajwaf</i>	<i>Fi'il Mudhari</i>	هَابَ	SESUAI
32	هَيْبَةٌ	<i>Ajwaf</i>	<i>Masdar</i>	هَابَ	<i>Shohih</i>	<i>Masdar</i>	هَابَ	SESUAI
33	مَهَابَةٌ	<i>Ajwaf</i>	<i>Masdar</i>	هَابَ	<i>Ajwaf</i>	<i>Masdar</i>	هَابَ	SESUAI
34	هَائِبٌ	<i>Ajwaf</i>	<i>Isim Fa'il</i>	هَابَ	<i>Ajwaf</i>	<i>Isim Fa'il</i>	هَابَ	SESUAI
35	مَهِيْبٌ	<i>Ajwaf</i>	<i>Isim Maf'ul</i>	هَابَ	<i>Ajwaf</i>	<i>Dhorof</i>	هَابَ	TIDAK
36	هَبَ	<i>Ajwaf</i>	<i>Fi'il Amr</i>	هَابَ	<i>Ajwaf</i>	<i>Fi'il Madhi</i>	هَابَ	TIDAK
37	لَا تَهَبْ	<i>Ajwaf</i>	<i>Fi'il Nahyi</i>	هَابَ	<i>Ajwaf</i>	<i>Fi'il Nahyi</i>	هَابَ	SESUAI
38	مَهَابٌ	<i>Ajwaf</i>	<i>Dhorof</i>	هَابَ	<i>Ajwaf</i>	<i>Dhorof</i>	هَابَ	SESUAI
39	طَالَ	<i>Ajwaf</i>	<i>Fi'il Madhi</i>	طَالَ	<i>Ajwaf</i>	<i>Fi'il Madhi</i>	طَالَ	SESUAI
40	يَطْوُلُ	<i>Ajwaf</i>	<i>Fi'il Mudhari</i>	طَالَ	<i>Ajwaf</i>	<i>Fi'il Mudhari</i>	طَالَ	SESUAI
41	طَوَّلًا	<i>Ajwaf</i>	<i>Masdar</i>	طَالَ	<i>Ajwaf</i>	<i>Masdar</i>	طَالَ	SESUAI
42	مَطَّالًا	<i>Ajwaf</i>	<i>Masdar</i>	طَالَ	<i>Ajwaf</i>	<i>Masdar</i>	طَالَ	SESUAI
43	طَوِيلٌ	<i>Ajwaf</i>	<i>Isim Fa'il</i>	طَالَ	<i>Ajwaf</i>	<i>Isim Fa'il</i>	طَالَ	SESUAI
44	طُنَّ	<i>Ajwaf</i>	<i>Fi'il Amr</i>	طَالَ	<i>Ajwaf</i>	<i>Fi'il Madhi</i>	طَالَ	TIDAK
45	لَا تَطُنْ	<i>Ajwaf</i>	<i>Fi'il Nahyi</i>	طَالَ	<i>Ajwaf</i>	<i>Fi'il Nahyi</i>	طَالَ	SESUAI
46	مَطَّالٌ	<i>Ajwaf</i>	<i>Dhorof</i>	طَالَ	<i>Ajwaf</i>	<i>Dhorof</i>	طَالَ	SESUAI

Dilihat dari hasil pengujian kata *Bina Ajwaf* di atas, dapat diketahui bahwa:

$$Akurasi\ FSA = \frac{\text{Jumlah kata sesuai}}{\text{Total pengujian}} \times 100 \quad (4.3)$$

Jumlah kata sesuai = 36 kata

Total pengujian = 46 kata

Maka:

$$Akurasi\ FSA = 36/46 * 100$$

Hasil perhitungan:

Persentase Akurasi = 78%

Berdasarkan hasil pengujian yang telah dilakukan, diperoleh tingkat akurasi sebesar 78% pada pengujian kata *Bina Ajwaf* menggunakan FSA. Nilai tersebut menunjukkan bahwa FSA mampu mengenali sebagian besar pola kata *Bina Ajwaf* yang diuji. Namun, masih terdapat sejumlah data yang tidak berhasil diproses secara akurat akibat adanya variasi perubahan huruf *illat* yang memengaruhi proses pencocokan pola dan ekstraksi akar kata. Pada beberapa kasus, sistem menentukan huruf *illat* yang hilang berdasarkan keberadaan harakat tertentu pada kata masukan. Pendekatan ini dapat menghasilkan identifikasi yang kurang tepat apabila kata memiliki pola perubahan yang berbeda dari aturan normalisasi yang telah ditetapkan. Akibatnya, huruf *illat* yang direkonstruksi tidak selalu sesuai dengan akar kata sebenarnya sehingga memengaruhi hasil analisis yang diberikan oleh sistem.

4.4.3 Pengujian dan Evaluasi *Usability*

Pengujian *Usability* dilakukan setelah semua fungsi sistem sudah diuji dan berjalan sesuai dengan kebutuhan pengguna. Tujuan dari pengujian ini adalah mengenali sejauh mana sistem yang telah dibuat mudah digunakan, nyaman, dan memberikan kepuasan bagi pengguna saat menggunakan sistem. Hasil pengujian *Usability* bisa menunjukkan sejauh mana sistem sesuai dengan kebutuhan pengguna dan menjadi salah satu acuan dalam menilai tingkat keberhasilan penerapan sistem serta tingkat penerimaan pengguna akhir.

Evaluasi *Usability* dilakukan dengan menyebarkan kuisioner kepada 20 responden. Responden merupakan santri di Pondok Pesantren Miftahul Huda yang

masih mengikuti kelas Madrasah Diniyah serta telah atau sedang mempelajari ilmu shorof. Responden akan diminta untuk menggunakan sistem terlebih dahulu, kemudian mengisi kuesioner SUS berdasarkan pengalaman yang diperoleh selama menggunakan sistem. Pernyataan dalam kuisoner telah tersajikan pada tabel 3.4. Pilihan jawaban yang tersedia meliputi Sangat Tidak Setuju (STS), Tidak Setuju (TS), Ragu-ragu (RG), Setuju (S), dan Sangat Setuju (SS).

Tabel 4.5 Daftar nilai kuisoner

Responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
R1	5	2	5	1	1	3	4	1	4	3
R2	5	5	5	3	3	3	4	2	3	3
R3	3	3	3	3	3	3	3	3	3	3
R4	3	4	4	3	3	4	5	5	2	3
R5	5	2	5	2	2	3	4	2	4	3
R6	1	1	2	1	1	1	1	2	2	2
R7	4	1	5	1	1	1	5	1	5	2
R8	4	2	4	3	3	2	4	2	4	2
R9	3	3	4	4	4	1	3	1	4	3
R10	4	3	4	1	1	4	3	2	4	2
R11	5	1	5	1	1	1	5	1	5	1
R12	2	3	4	1	1	3	5	1	5	3
R13	4	2	5	4	4	3	4	2	4	4
R14	2	2	5	2	2	5	5	1	5	5
R15	1	4	1	4	4	4	1	5	2	3
R16	4	3	5	1	1	3	5	1	5	1
R17	5	2	5	2	2	3	5	2	4	1
R18	2	3	4	2	2	3	3	2	3	5
R19	3	2	4	2	2	3	5	1	5	1
R20	4	2	5	1	1	1	4	1	4	1

Setelah pengisian kuesioner, selanjutnya data tersebut dihitung sesuai aturan perhitungan skor SUS yang telah dijelaskan pada bab sebelumnya.

Tabel 4.6 Perhitungan skor SUS

Responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
R1	5-1	5-2	5-1	5-1	1-1	5-3	4-1	5-1	4-1	5-3
R2	5-1	5-5	5-1	5-3	3-1	5-3	4-1	5-2	3-1	5-3
R3	3-1	5-3	3-1	5-3	3-1	5-3	3-1	5-3	3-1	5-3
R4	3-1	5-4	4-1	5-3	3-1	5-4	5-1	5-5	2-1	5-3
R5	5-1	5-2	5-1	5-2	2-1	5-3	4-1	5-2	4-1	5-3

Responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
R6	1-1	5-1	2-1	5-1	1-1	5-1	1-1	5-2	2-1	5-2
R7	4-1	5-1	5-1	5-1	1-1	5-1	5-1	5-1	5-1	5-2
R8	4-1	5-2	4-1	5-3	3-1	5-2	4-1	5-2	4-1	5-2
R9	3-1	5-3	4-1	5-4	4-1	5-1	3-1	5-1	4-1	5-3
R10	4-1	5-3	4-1	5-1	1-1	5-4	3-1	5-2	4-1	5-2
R11	5-1	5-1	5-1	5-1	1-1	5-1	5-1	5-1	5-1	5-1
R12	2-1	5-3	4-1	5-1	1-1	5-3	5-1	5-1	5-1	5-3
R13	4-1	5-2	5-1	5-4	4-1	5-3	4-1	5-2	4-1	5-4
R14	2-1	5-2	5-1	5-2	2-1	5-5	5-1	5-1	5-1	5-5
R15	1-1	5-4	1-1	5-4	4-1	5-4	1-1	5-5	2-1	5-3
R16	4-1	5-3	5-1	5-1	1-1	5-3	5-1	5-1	5-1	5-1
R17	5-1	5-2	5-1	5-2	2-1	5-3	5-1	5-2	4-1	5-1
R18	2-1	5-3	4-1	5-2	2-1	5-3	3-1	5-2	3-1	5-5
R19	3-1	5-2	4-1	5-2	2-1	5-3	5-1	5-1	5-1	5-1
R20	4-1	5-2	5-1	5-1	1-1	5-1	4-1	5-1	4-1	5-1

Tabel 47 Total skor responden tiap responden (sebelum dikali 2,5)

Responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Total
R1	5-1	5-2	5-1	5-1	1-1	5-3	4-1	5-1	4-1	5-3	29
R2	5-1	5-5	5-1	5-3	3-1	5-3	4-1	5-2	3-1	5-3	24
R3	3-1	5-3	3-1	5-3	3-1	5-3	3-1	5-3	3-1	5-3	20
R4	3-1	5-4	4-1	5-3	3-1	5-4	5-1	5-5	2-1	5-3	18
R5	5-1	5-2	5-1	5-2	2-1	5-3	4-1	5-2	4-1	5-3	28
R6	1-1	5-1	2-1	5-1	1-1	5-1	1-1	5-2	2-1	5-2	20
R7	4-1	5-1	5-1	5-1	1-1	5-1	5-1	5-1	5-1	5-2	34
R8	4-1	5-2	4-1	5-3	3-1	5-2	4-1	5-2	4-1	5-2	28
R9	3-1	5-3	4-1	5-4	4-1	5-1	3-1	5-1	4-1	5-3	26
R10	4-1	5-3	4-1	5-1	1-1	5-4	3-1	5-2	4-1	5-2	24
R11	5-1	5-1	5-1	5-1	1-1	5-1	5-1	5-1	5-1	5-1	36
R12	2-1	5-3	4-1	5-1	1-1	5-3	5-1	5-1	5-1	5-3	26
R13	4-1	5-2	5-1	5-4	4-1	5-3	4-1	5-2	4-1	5-4	26
R14	2-1	5-2	5-1	5-2	2-1	5-5	5-1	5-1	5-1	5-5	24
R15	1-1	5-4	1-1	5-4	4-1	5-4	1-1	5-5	2-1	5-3	9
R16	4-1	5-3	5-1	5-1	1-1	5-3	5-1	5-1	5-1	5-1	31
R17	5-1	5-2	5-1	5-2	2-1	5-3	5-1	5-2	4-1	5-1	31
R18	2-1	5-3	4-1	5-2	2-1	5-3	3-1	5-2	3-1	5-5	19
R19	3-1	5-2	4-1	5-2	2-1	5-3	5-1	5-1	5-1	5-1	30
R20	4-1	5-2	5-1	5-1	1-1	5-1	4-1	5-1	4-1	5-1	32

Tabel 4.8 Skor SUS responden (Total dikali 2,5)

Responden	Skor SUS
R1	72,5
R2	60
R3	50

Responden	Skor SUS
R4	45
R5	70
R6	50
R7	85
R8	70
R9	65
R10	60
R11	90
R12	65
R13	65
R14	60
R15	22,5
R16	77,5
R17	77,5
R18	47,5
R19	75
R20	80
Total	1287,5

Setelah mendapatkan skor SUS untuk setiap responden, langkah berikutnya adalah menghitung rata-rata dari skor SUS tersebut. Perhitungan dilakukan dengan cara menjumlahkan semua skor SUS yang diperoleh dari setiap orang yang mengikuti tes, lalu hasilnya dibagi dengan jumlah orang yang terlibat dalam pengujian. Nilai rata-rata tersebut digunakan untuk menunjukkan seberapa baik sistem itu bisa digunakan secara keseluruhan. Rumus yang digunakan untuk menghitung rata-rata skor SUS adalah sebagai berikut:

$$\bar{x} = \frac{\sum x}{n} \times 100 \quad (4.4)$$

Keterangan:

$\bar{x}$ = nilai rata-rata (mean)

$\sum x$ = jumlah seluruh skor SUS responden = 1287,5

n = jumlah responden = 20

$$\text{Sehingga } \bar{x} = \frac{1287,5}{20} = 64,3$$

Berdasarkan hasil perhitungan menggunakan rumus rata-rata (*mean*), diperoleh nilai SUS sebesar 64,3 dari 20 responden dengan total skor SUS sebesar 1287,5. Nilai tersebut menunjukkan tingkat usability sistem berdasarkan persepsi pengguna setelah menggunakan sistem yang dikembangkan.

Berdasarkan rentang interpretasi skor SUS pada Gambar 2.4, nilai rata rata yang dihasilkan dari hasil perhitungan SUS yaitu 64,3 berada pada kategori *Marginal Acceptable* dengan grade D dan termasuk dalam penilaian kualitatif OK. Hasil ini menunjukkan bahwa sistem yang dikembangkan telah dapat digunakan oleh pengguna, namun tingkat kegunaannya masih berada pada batas menengah sehingga masih terdapat beberapa aspek yang perlu diperbaiki dan dikembangkan lebih lanjut. Perbaikan tersebut diharapkan dapat meningkatkan kemudahan penggunaan, efisiensi, serta kepuasan pengguna sehingga pada pengujian berikutnya sistem dapat mencapai kategori *Acceptable* dengan nilai usability yang lebih baik.

4.5 Integrasi Islam

Penelitian ini tidak hanya bertujuan untuk menghasilkan teknologi yang dapat membantu proses analisis bahasa Arab, tetapi juga menjadi sarana dalam mendukung pembelajaran bahasa Al-Qur'an sehingga memiliki nilai ibadah dan manfaat bagi orang yang mempelajari bahasa Arab.

4.5.1 *Muamalah Ma'a Allah*

Hubungan manusia dengan Allah SWT dapat diwujudkan melalui upaya mempelajari dan memahami bahasa Arab sebagai bahasa Al-Qur'an. Dalam penelitian ini, sistem yang dikembangkan bertujuan membantu proses analisis kata bahasa Arab sehingga dapat mempermudah pembelajaran ilmu sharaf dan pemahaman teks Arab. Hal tersebut menjadi bagian dari bentuk ikhtiar dalam mendukung pembelajaran ilmu agama. Sebagaimana Firman Allah SWT dalam QS. Yusuf ayat 2:

إِنَّا أَنْزَلْنَاهُ قُرْآنًا عَرَبِيًّا لَعَلَّكُمْ تَعْقِلُونَ

"Sesungguhnya Kami menurunkannya berupa Al-Qur'an dengan berbahasa Arab, agar kamu memahaminya".(Q.S Yusuf:2)

Imam Ibnu Katsir seorang ulama besar ahli tafsir, dalam kitabnya yaitu *Tafsir Al-Quran Al-Azhim* memberikan penjelasan pada ayat diatas sebagai berikut "Demikian itu karena bahasa Arab adalah bahasa yang paling jelas, paling terang, paling luas, dan paling banyak perbendaharaan kata-katanya untuk mengungkapkan berbagai pengertian guna meluruskan jiwa manusia. Karena itulah Allah menurunkan Kitab-Nya yang paling mulia dengan bahasa yang paling mulia di antara bahasa-bahasa lainnya yang disampaikan-Nya kepada rasul yang paling mulia melalui perantaraan malaikat yang paling mulia." (*Imam Ibnu Katsir*).

4.5.2 *Mu'amalah Ma'annas*

Penelitian ini juga berkaitan dengan muamalah ma'an nas karena sistem yang dibangun diharapkan dapat memberikan manfaat kepada pengguna. Dengan

adanya sistem analisis bahasa arab berbasis FSA, proses identifikasi pola kata dan akar kata dapat dilakukan lebih cepat dan sistematis sehingga membantu proses pembelajaran. Sebagaimana Hadits Nabi Muhammad SAW

خَيْرُ النَّاسِ أَنْفَعُهُمْ لِلنَّاسِ

Artinya: “Sebaik-baik manusia adalah yang paling bermanfaat bagi orang lain.” (Hadits Riwayat ath-Thabrani, Al-Mu’jam al-Ausath, juz VII, hal. 58, dari Jabir bin Abdullah r.a.. Dishahihkan Muhammad Nashiruddin al-Albani dalam kitab: As-Silsilah Ash-Shahîhah).

Menjadi pribadi yang bermanfaat adalah salah satu karakter yang harus dimiliki oleh seorang Muslim. Seorang Muslim lebih diperintahkan untuk memberikan manfaat bagi orang lain, bukan hanya mencari manfaat dari orang atau memanfaatkan orang lain. Ini adalah bagian dari implementasi konsep Islam yang penuh cinta, yaitu memberi.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, *Arabic Root Analysis Sistem* (ARAS) berhasil dibangun dengan menggunakan metode *Finite State Automata* (FSA) dan pendekatan pengembangan perangkat lunak *Incremental*. Sehingga setiap fitur yang dibutuhkan dapat diimplementasikan dalam setiap tahap yang telah ditentukan.

Hasil uji fungsional menggunakan *blackbox testing* menunjukkan semua fitur yang ada dalam sistem ARAS berjalan sesuai dengan kebutuhan yang sudah ditentukan. Dari 9 skenario pengujian yang dilakukan, semua berhasil dijalankan dengan baik, sehingga mencapai tingkat keberhasilan 100%. Hasil tersebut menunjukkan bahwa sistem sudah memenuhi semua fungsi yang direncanakan dan bisa menjalankan setiap proses, mulai dari memasukkan kata, menganalisis bentuk kata, hingga menampilkan akar kata sesuai dengan cara kerja yang telah ditentukan.

Implementasi metode FSA dalam sistem ARAS juga berhasil digunakan untuk menganalisis struktur kata dalam bahasa Arab, terutama untuk menentukan akar kata pada *fi'il tsulasi mujarrad bina mudhoaf* dan *fi'il tsulasi mujarrad bina ajwaf* yang terdapat dalam buku *Amtsilatu Tashrifiyyah*. Hasil pengujian menunjukkan metode FSA mencapai tingkat akurasi 82% dalam menguji kata *Bina Mudhoaf*, di mana 24 dari 29 data uji berhasil dikenali dengan benar. Sementara itu, dalam pengujian kata *Bina Ajwaf*, diperoleh tingkat akurasi sebesar 78%, di mana

36 kata berhasil dikenali dengan benar dari total 46 data uji. Hasil tersebut menunjukkan bahwa FSA mampu mengenali pola kata dan mengambil akar kata sesuai dengan aturan yang sudah dibuat.

Selain melakukan pengujian fungsional dan FSA, sistem juga dinilai dari segi kemudahan penggunaan dengan metode *Sistem Usability Scale* (SUS) yang melibatkan 20 orang responden. Dari hasil perhitungan didapatkan rata-rata nilai SUS sebesar 64,3. Hasil tersebut menunjukkan tingkat usability sistem berdasarkan persepsi pengguna setelah menggunakan sistem yang dikembangkan.

5.2 Saran

Berdasarkan hasil penelitian yang sudah dilakukan, berikut beberapa saran untuk penelitian lebih lanjut:

1. Menggunakan metode lain dalam analisis morfologi, seperti *Natural Language Processing* (NLP), agar sistem lebih mampu mengenali dan memahami bentuk kata yang lebih rumit dalam bahasa Arab.
2. Menggunakan kamus atau *database morfologi* bahasa Arab, sehingga hasil analisis bisa diperiksa dengan data yang sudah ada sebagai acuan dan memperbaiki ketepatan dalam menentukan akar kata.
3. Menggunakan pendekatan lain yang lebih terstruktur dalam proses pengembangan perangkat lunak, sehingga memudahkan proses perawatan dan pengembangan yang akan datang.

DAFTAR PUSTAKA

- Abidin zainal, F. (2015). Pembangkitan Otomatis Bentuk Kata Arab Berbasis Morfologi. *Repository.Uin-Malang.Ac.Id*.
- Amalia, N. (2024). *Evaluating the Regular Sharaf Learning Program at the Foundation of Islamic and Arabic Language Learning In Indonesia*. 2(2), 101–110.
- Ana;, T., Ari;, D., & Agus; Zainal; (2019). Autonomy Stemmer Algorithm for Legal and Illegal Affix Detection Use Finite-State Automata Method. *EPI International Journal of Engineering*, 2(1), 46–55. <https://doi.org/10.25042/epi-ije.022019.09>
- Aqil, M., & Hadi, S. (2019). Morfologi Bahasa Arab : Reformulasi Sistem Derivasi dan Infleksi. *Alsina : Journal of Arabic Studies*, 1(1), 1–22.
- Azhar Muhammad, et. al. (2025). Revolusi Pembelajaran Bahasa Arab Di Era Digital: Analisis Peran Artificial Intelligence Dalam Pengembangan Kompetensi Berbahasa. *Al-Ma'lumat*, 3(2), 58–81.
- Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean; adding an adjective rating. *Journal of Usability Studies*, 4(3), 114–123.
- Brooke, J. (2020). SUS: A “Quick and Dirty” Usability Scale. *Usability Evaluation In Industry*, 207–212. <https://doi.org/10.1201/9781498710411-35>
- Cahya Edi Setyawan, K. A. (2025). PERAN BAHASA ARAB DALAM PENDIDIKAN ISLAM SEBAGAI URGENSITAS MENGHADAPI REVOLUSI INDUSTRI 4.0 Cahya. *Lahjah Arabiyah: Jurnal Bahasa Arab Dan Pendidikan Bahasa Arab*, 22(1), 11–19.
- Daffa, G. M. (2024). IMPLEMENTASI FINITE STATEAUTOMATA (FSA) PADAAPLIKASI SIMULASI VENDINGMACHINEMINUMAN BOBA. *Scientica*, 3, 65–73.
- Dandy, D., Suprpto, A., Informatika, P. S., & Nasional, U. (2020). IMPLEMENTASI FINITE STATE AUTOMATA PADA MESIN. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 5(1).
- Evi Dwi Wahyuni, Firdatul Nurul Ramadha, Fatimatus Zahroh, Ayesha Fahrelia Ningrum, & Radhiya Astifa. (2025). Analisis Komparatif metode Spiral dan Incremental Berdasarkan Manajemen Resiko dan Fokus Utama Pengembangan. *JEKIN - Jurnal Teknik Informatika*, 5(2), 836–844.

<https://doi.org/10.58794/jekin.v5i2.1520>

- Fitriana, L. A., Mustopa, A., Firdaus, M. R., & Dahlia, R. (2024). Application of the Finite State Automata (FSA) Method in Indonesian Stemming using the Nazief & Adriani Algorithm. *Sistemasi*, 13(3), 1125. <https://doi.org/10.32520/stmsi.v13i3.4038>
- Halimah, H., Arsa, D., & Hutabarat, B. F. (2024). Penerapan Model Incremental Pada Implementasi Dan Pengujian Sistem Informasi Pendanaan Usaha Mikro Kecil (Pumk). *Jurnal Sistem Informasi Dan Bisnis Cerdas*, 17(2), 11–20. <https://doi.org/10.33005/sibc.v17i2.284>
- Hasyim, N. A., Industri, F. R., Telkom, U., Musnansyah, A., Industri, F. R., Telkom, U., Azzahra, Z. F., Industri, F. R., Telkom, U., & Bandung, K. (2025). Pengembangan Back-End Aplikasi Mobile Web Untuk Pedagang Keliling Di Lingkungan Rt 04 Palem Kota Bandung. *E-Proceeding of Engineering*, 12(1), 1582–1589.
- Ihwan, M., Mawardi, S., & Ni'mah, U. (2022). Pengaruh Penguasaan Ilmu Nahwu Dan Sharaf Terhadap Kemampuan Membaca Kitab Fathul Qarib. *TADRIS AL-ARABIYAT: Jurnal Kajian Ilmu Pendidikan Bahasa Arab*, 2, 61–77. <https://doi.org/10.30739/arabiyat.v2i1.1422>
- Musa, & Ghofur, M. A. (2024). Analysis of Twelfth Grade Students' Difficulties in Understanding Nahwu and Sharaf. *International Journal of Islamic Thought and Humanities*, 3(2), 268–276. <https://doi.org/10.54298/ijith.v3i2.387>
- Postiglione, A. (2024). Finite State Automata on Multi-Word Units for Efficient Text-Mining †. *Mathematics*, 12(4). <https://doi.org/10.3390/math12040506>
- Pressman R.S. (2010). *Software process*. Raghothaman Srinivasan. <https://doi.org/10.1145/336512.336521>
- Purbaya, A. M., Fauzi, R., & Pramesti, D. (2023). Penerapan Metode Iterative Incremental Dalam Pengembangan Website Tripinaja Untuk Meningkatkan Pelayanan Dan Proses Bisnis. *E-Proceeding of Engineering*, 10(5), 4820–4825.
- Rahman, T., & Ramdani, M. F. (2026). Pengembangan Aplikasi Penerimaan Peserta Didik Baru Berbasis Web Menggunakan Metode Agile. *Jurnal Ilmiah Sinus (JIS)*, 1, 13–24.
- Sandhiyasa, I. M. S., Pande, N. K. N. N., & Fibriyanthini, L. (2024). Transliteration of Latin Letters to Bali Characters Based on Unicode for Mobile Devices using Finite State Automata and Levenshtein Distance. *Sinkron: Jurnal Dan Penelitian Teknik Informatika*, 8(3), 1976–1986.

<https://doi.org/10.33395/sinkron.v8i3.13856>

- Saputera, S. A., Handayani, S., & Awaludin, E. (2023). Implementasi Metode Incremental Pada Sistem Pengelolaan Arsip Berbasis Web. *Journal of Technopreneurship and Information Sistem (JTIS)*, 6, 31–36. <http://jurnal.umb.ac.id/index.php/JTIS/article/view/5442>
- Sekarsari, A., Kurniawan Siagian, J., & Amelia, P. (2025). Basics of Arabic Morphology: Word Structure in Arabic. *Dalam Journal Of Education, Arabic And Islamic Studies*, 3(1), 4–5.
- Sholehurrohman, R., Qurrota A'yuni, S., Sakethi, D., Sabda Ilman, I., Muhaqiqin, M., & Taufik, R. (2025). Implementation of Black-Box Testing on the Information Sistem for the Smart Indonesia Card College Recommendation. *JESII: Journal of Elektronik Sistem InformasI*, 3(1), 1–11. <https://doi.org/10.31848/jesii.v3i1.4038>
- Thalji, N. J. (2024). A Survey of Arabic Root Extraction Algorithms. *Iraqi Journal of Science*, 65(4), 2304–2321. <https://doi.org/10.24996/ijs.2024.65.4.43>
- Yudi, I. C., & Kosdiana, M. R. (2023). Penerapan Model Iterative Incremental Dalam Membangun Aplikasi E-Commerce Di Toko Kopi Rarukuh Luas Berbasis Web Mobile. *Seminar Nasional Teknologi Informasi Dan Komunikasi STI&K (SeNTIK)*, 7(1), 179–190.

LAMPIRAN

LAMPIRAN 1

DOKUMENTASI KEGIATAN PROSES MEMBANGUN *ARABIC ROOT ANALYSIS SYSTEM* (ARAS) BERBASIS *WEBSITE* MENGGUNAKAN METODE *INCREMENTAL*

Bulan	Waktu	Nama kegiatan	Hasil
M A R E T	Minggu ke-1	Analisis kebutuhan sistem ARAS	Identifikasi kebutuhan pengguna dan fitur utama sistem
		Perancangan antarmuka sistem	Desain tampilan awal halaman input, proses, dan output sistem
	Minggu ke-2	Pembuatan struktur navigasi dan layout	Struktur halaman dan alur penggunaan sistem berhasil dibuat
		Implementasi antarmuka sistem	Antarmuka sistem ARAS berhasil diimplementasikan berbasis website
	Minggu ke-3	Perancangan FSA mulai dari prepare hingga root	Logika FSA berhasil namun belum spesifik antara mudhoaf dan ajwaf
	Minggu ke-4	Perancangan FSA untuk fi'il ajwaf	Penyusunan state dan transisi pola kata ajwaf
A P R I L	Minggu ke-1	Implementasi FSA untuk ajwaf dan pengujian	Sistem belum mampu mengenali pola dan root kata ajwaf sepenuhnya
	Minggu ke-2	Implementasi FSA untuk ajwaf dan pengujian	Hasil pengujian menunjukkan Sistem dapat mengenali aturan ajwaf dan menghasilkan root
	Minggu ke-3	Perancangan FSA untuk fi'il mudhoaf	Penyusunan state dan transisi pola kata mudhoaf
		Implementasi FSA untuk mudhoaf dan pengujian	Sistem berhasil mengenali pola mudhoaf dan menghasilkan root
	Minggu ke-4	Merubah tampilan interface	Sistem berhasil menampilkan interface baru
M E I	Minggu ke-1	Pengujian Seluruh kata	Menguji seluruh kata mudhoaf dan ajwaf.
	Minggu ke-2	Validasi sistem dengan Ahli	Ahli menilai output yang dihasilkan oleh sistem

LAMPIRAN 2

TABEL PENILAIAN OUTPUT HASIL SISTEM OLEH AHLI (Bina Ajwaf)

No	Input	Shigot	Bina	Root	Ketepatan Shigot	Ketepata Bina	Ketepatan Root
1	صَانَ	Fi'il Madhi	Ajwaf	صان	S	S	S
2	يَصُونُ	Fi'il Mudhari	Ajwaf	صان	S	S	S
3	صَوْنًا	Masdar	Ajwaf	صان	S	S	S
4	مَصَانًا	Masdar	Ajwaf	صان	S	S	S
5	صَائِنٌ	Isim Fa'il	Ajwaf	صان	S	S	S
6	مَصُونٌ	Dhorof	Ajwaf	صان	TS	S	S
7	صُنْ	Fi'il Madhi	Ajwaf	صان	TS	S	S
8	لَا تَصُنْ	Fi'il Nahyi	Ajwaf	صان	S	S	S
9	مَصَانٌ	Dhorof	Ajwaf	صان	S	S	S
10	مِصْنُونٌ	Dhorof	Ajwaf	صان	TS	S	S
11	سَارَ	Fi'il Madhi	Ajwaf	سار	S	S	S
12	يَسِيرُ	Fi'il Mudhari	Ajwaf	سار	S	S	S
13	سَيْرًا	Masdar	Ajwaf	سار	S	S	S
14	مَسِيرَةً	Masdar	Ajwaf	سار	S	S	S
15	سَائِرٌ	Isim Fa'il	Ajwaf	سار	S	S	S
16	مَسِيرٌ	Dhorof	Ajwaf	سار	TS	S	S
17	سِرْ	Fi'il Madhi	Ajwaf	سار	TS	S	S
18	لَا تَسِرْ	Fi'il Nahyi	Ajwaf	سار	S	S	S
19	مَسِيرٌ	Dhorof	Ajwaf	سار	S	S	S
20	مِسِيرٌ	Dhorof	Ajwaf	سار	TS	S	S
21	خَافَ	Fi'il Madhi	Ajwaf	خاف	S	S	S
22	يَخَافُ	Fi'il Mudhari	Ajwaf	خاف	S	S	S

23	خَوْفًا	Masdar	Ajwaf	خاف	S	S	S
24	مَخَافَةً	Masdar	Ajwaf	خاف	S	S	S
25	خَائِفٌ	Isim Fa'il	Ajwaf	خاف	S	S	S
26	مُخَوِّفٌ	Dhorof	Ajwaf	خاف	TS	S	S
27	خَفَ	Fi'il Madhi	Ajwaf	خاف	TS	S	S
28	لَا تَخَفْ	Fi'il Nahyi	Ajwaf	خاف	S	S	S
29	مَخَافٌ	Dhorof	Ajwaf	خاف	S	S	S
30	هَابٌ	Fi'il Madhi	Ajwaf	هاب	S	S	S
31	يَهَابُ	Fi'il Mudhari	Ajwaf	هاب	S	S	S
32	هَيْبَةً	Masdar	Shahih	هاب	S	S	S
33	مَهَابَةً	Masdar	Ajwaf	هاب	S	S	S
34	هَائِبٌ	Isim Fa'il	Ajwaf	هاب	S	S	S
35	مَهِيبٌ	Dhorof	Ajwaf	هاب	TS	S	S
36	هَبَ	Fi'il Madhi	Ajwaf	هاب	TS	S	S
37	لَا تَهَبْ	Fi'il Nahyi	Ajwaf	هاب	S	S	S
38	مَهَابٌ	Dhorof	Ajwaf	هاب	S	S	S
39	طَالَ	Fi'il Madhi	Ajwaf	طال	S	S	S
40	يَطُولُ	Fi'il Mudhari	Ajwaf	طال	S	S	S
41	طَوَّلًا	Masdar	Ajwaf	طال	S	S	S
42	مَطَالًا	Masdar	Ajwaf	طال	S	S	S
43	طَوِيلٌ	Isim Fa'il	Ajwaf	طال	S	S	S
44	طُلَّ	Fi'il Madhi	Ajwaf	طال	TS	S	S
45	لَا تَطُلْ	Fi'il Nahyi	Ajwaf	طال	S	S	S
46	مَطَالٌ	Dhorof	Ajwaf	طال	S	S	S

NB : S = Sesuai

TS = Tidak Sesuai

Evaluator

A handwritten signature in black ink, appearing to be 'Jauhar Fuadi', with a horizontal line extending to the right.

Ustad Jauhar Fuadi S.Pd

LAMPIRAN 3

TABEL PENILAIAN OUTPUT HASIL SISTEM OLEH AHLI (Bina Mudho'af)

No	Input	Shigot	Bina	Root	Ketepatan Shigot	Ketepata Bina	Ketepatan Root
1	مَدَّ	Fi'il Madhi	Mudhoaf	مَدَّ	S	S	S
2	يَمُدُّ	Fi'il Mudhari	Mudhoaf	مَدَّ	S	S	S
3	مَدًّا	Masdar	Mudhoaf	مَدَّ	S	S	S
4	مَمْدًا	Masdar	Mudhoaf	مَدَّ	S	S	S
5	مَادٌّ	Isim Fa'il	Mudhoaf	مَدَّ	S	S	S
6	مَمْدُودٌ	Isim Maful	Mudhoaf	مَدَّ	S	S	S
7	مُدَّ	Fi'il Madhi	Mudhoaf	مَدَّ	TS	S	S
8	لَا يَمُدُّ	Fi'il Nahyi	Mudhoaf	مَدَّ	S	S	S
9	مَمْدٌ	Dhorof	Mudhoaf	مَدَّ	S	S	S
10	مِمْدٌ	Dhorof	Mudhoaf	مَدَّ	TS	S	S
11	فَرَّ	Fi'il Madhi	Mudhoaf	فَرَّ	S	S	S
12	يَفِرُّ	Fi'il Mudhari	Mudhoaf	فَرَّ	S	S	S
13	فَرًّا	Masdar	Mudhoaf	فَرَّ	S	S	S
14	مَفَرًّا	Masdar	Mudhoaf	فَرَّ	S	S	S
15	فَارٌّ	Isim Fa'il	Mudhoaf	فَرَّ	S	S	S
16	مَفْرُورٌ	Isim Maful	Mudhoaf	فَرَّ	S	S	S
17	فِرٌّ	Fi'il Madhi	Mudhoaf	فَرَّ	TS	S	S
18	لَا يَفِرُّ	Fi'il Nahyi	Mudhoaf	فَرَّ	S	S	S
19	مَفَرٌّ	Dhorof	Mudhoaf	فَرَّ	S	S	S

20	مِفْرٌ	Dhorof	Mudhoaf	فَرّ	TS	S	S
21	عَضّ	Fi'il Madhi	Mudhoaf	عَضّ	S	S	S
22	يَعَضُّ	Fi'il Mudhari	Mudhoaf	عَضّ	S	S	S
23	عَضًّا	Masdar	Mudhoaf	عَضّ	S	S	S
24	مَعَضًّا	Masdar	Mudhoaf	عَضّ	S	S	S
25	عَاضٌ	Isim Fa'il	Mudhoaf	عَضّ	S	S	S
26	مَعْمُوضٌ	Isim Maf'ul	Mudhoaf	عَضّ	S	S	S
27	عَضّ	Fi'il Madhi	Mudhoaf	عَضّ	TS	S	S
28	لَا تَعَضّ	Fi'il Nahyi	Mudhoaf	عَضّ	S	S	S
29	مَعَضّ	Dhorof	Mudhoaf	عَضّ	S	S	S

NB : S = Sesuai

TS = Tidak Sesuai

Evaluator



Ustad Jauhar Fuadi S.Pd

LAMPIRAN 4

Form penilaian usability menggunakan metode System Usability Scale SUS

1. Saya ingin menggunakan sistem ini secara sering.

	1	2	3	4	5	
Sangat tidak setuju	☐	☐	☐	☐	☐	Sangat setuju

2. Saya merasa sistem ini terlalu rumit.

	1	2	3	4	5	
Sangat tidak setuju	☐	☐	☐	☐	☐	Sangat setuju

3. Sistem ini mudah digunakan.

	1	2	3	4	5	
Sangat tidak setuju	☐	☐	☐	☐	☐	Sangat setuju

4. Saya memerlukan bantuan orang teknis untuk menggunakan sistem ini.

	1	2	3	4	5	
Sangat tidak setuju	☐	☐	☐	☐	☐	Sangat setuju

5. Fitur-fitur dalam sistem ini terintegrasi dengan baik.

	1	2	3	4	5	
Sangat tidak setuju	☐	☐	☐	☐	☐	Sangat setuju

6. Saya menemukan banyak ketidakkonsistenan dalam sistem ini.

	1	2	3	4	5	
Sangat tidak setuju	☐	☐	☐	☐	☐	Sangat setuju

7. Saya merasa kebanyakan orang akan belajar menggunakan sistem ini dengan sangat cepat.

	1	2	3	4	5	
Sangat tidak setuju	☐	☐	☐	☐	☐	Sangat setuju

8. Saya merasa sistem ini sangat sulit digunakan.

	1	2	3	4	5	
Sangat tidak setuju	☐	☐	☐	☐	☐	Sangat setuju

☐ ☐ ☐ ☐ ☐

9. Saya merasa percaya diri saat menggunakan sistem ini.

	1	2	3	4	5	
Sangat tidak setuju	☐	☐	☐	☐	☐	Sangat setuju

10. Saya perlu mempelajari banyak hal sebelum dapat menggunakan sistem ini.

	1	2	3	4	5	
Sangat tidak setuju	☐	☐	☐	☐	☐	Sangat setuju

LAMPIRAN 5

DAFTAR RIWAYAT HIDUP EVALUATOR

	Nama lengkap : Jauhar Fuadi TTL : Pasuruan , 10 Oktober 2001 Alamat : Jl. Urip sumoharjo rt 3, rw 1 Profesi : Pengajar Bahasa Arab di MA Muallimat Kota Malang
Riwayat Pendidikan	
Pendidikan Formal	- TK khoirur rohman 2006-2007 - SDN bukir 2008 - 2013 - SMPN 4 pasuruan 2014-2016 - SMAN 2 pasuruan 2017 - 2019 - UIN Maulana Malik Ibrahim Malang 2020 - 2024
Pendidikan Non Formal	- TPQ Khoirur Rohman 2006 - 2012 - TPQHidayatul Quran 2012 - 2014 - Madrasah Sunniyah Salafiyah 2014 - 2022 - Madrasah Assidqiyah 2021 - 2022 - Pondok Pesantren Miftahul Huda 2022 - sekarang
Pengalaman	
- Pelatihan Al-Qur'an Metoda Wafa 2021 - Kursus Bahasa Arab El Kaffah Center Malang 2022 - Kursus Bahasa Arab Pare Al Azhar 2023 - Tutor Bahasa Arab Lembaga Kursus Bahasa Asing (LKBA) PP. Miftahul Huda 2023 - Kursus bahasa Arab Pare Natiq International Arabic 2024 - Anggota Lembaga Bahtsul Masail PP miftahul Huda 2024 - sekarang - Pengajar Bahasa Arab di MA Muallimat Kota Malang 2024- Sekarang - Pembina Ekskul Bahasa Arab MA Muallimat 2024 - sekarang.	