Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 2, Juni 2026
https://doi.org/10.58602/jaiti.v4i2.242
Page 76-88

Implementasi Speech Recognition dan Levenshtein Distance pada Aplikasi Pembelajaran Huruf Hijaiyah Berbasis Android

Maknun Fil Safah[1*], Chamdan Mashuri[2]
[1,2]Teknik Informatika, Universitas Hasyim Asy'ari, Indonesia
[1*]maknunsafah@mhs.unhasy.ac.id, [2]chamdanmashuri@unhasy.ac.id

Abstrak: Pembelajaran huruf hijaiyah merupakan fondasi awal dalam kemampuan membaca Al-Qur'an, namun pada praktiknya proses belajar di lembaga pendidikan nonformal masih banyak bergantung pada metode konvensional yang kurang interaktif dan belum menyediakan evaluasi pelafalan secara langsung. Penelitian ini bertujuan mengembangkan aplikasi pembelajaran huruf hijaiyah berbasis Android yang mengintegrasikan teknologi *speech recognition* dan algoritma *Levenshtein Distance* sebagai *spell checker* untuk menilai kemiripan hasil pelafalan pengguna terhadap target bacaan. Metode penelitian menggunakan pendekatan pengembangan perangkat lunak yang mencakup identifikasi masalah, studi literatur, analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Perancangan sistem dilakukan menggunakan flowchart dan use case diagram. Sistem memproses suara pengguna melalui modul pengenalan suara, mengubahnya menjadi teks, lalu menghitung nilai *edit distance* dan similarity untuk menghasilkan klasifikasi benar, mendekati, atau salah. Hasil pengujian menunjukkan bahwa aplikasi mampu menjalankan fungsi utama pembelajaran, menampilkan materi, memutar audio contoh, menerima input suara, dan memberikan umpan balik otomatis. Pengujian terhadap 15 sampel menunjukkan 6 data benar, 6 data mendekati, dan 3 data salah, dengan tingkat akurasi sistem sebesar 90%. Temuan ini menunjukkan bahwa kombinasi *speech recognition* dan *Levenshtein Distance* layak diterapkan sebagai media pembelajaran interaktif untuk membantu anak-anak berlatih membaca huruf hijaiyah secara lebih mandiri, terukur, dan menarik.

Kata Kunci: Huruf Hijaiyah; Android; Speech Recognition; Levenshtein Distance; Pembelajaran Interaktif;

Abstract: Learning hijaiyah letters is the fundamental stage in Quran reading skills, yet the instructional process in non-formal Islamic education still relies heavily on conventional methods that are less interactive and provide limited direct pronunciation feedback. This study aims to develop an Android-based hijaiyah learning application that integrates speech recognition technology and the Levenshtein Distance algorithm as a spell checker to evaluate the similarity between user pronunciation and target readings. The research employed a software development approach consisting of problem identification, literature review, requirements analysis, system design, implementation, and testing. The system was designed using flowcharts, use case diagrams, and activity diagrams. User speech is processed through speech recognition, converted into

Maknun Fil Safah: * Penulis Korespondensi

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 2, Juni 2026
https://doi.org/10.58602/jaiti.v4i2.242
Page 76-88

text, and then compared with target words using edit distance and similarity calculations to classify the result as correct, nearly correct, or incorrect. The implementation results indicate that the application is capable of delivering learning materials, playing pronunciation examples, receiving voice input, and providing automatic feedback. Testing on 15 pronunciation samples produced 6 correct, 6 nearly correct, and 3 incorrect results, with an overall system accuracy of 90%. These findings indicate that the combination of speech recognition and Levenshtein Distance is feasible for interactive learning media that supports children in practicing hijaiyah pronunciation in a more independent, measurable, and engaging way.

Keywords: Hijaiyah Letters; Android; Speech Recognition; Levenshtein Distance; Interactive Learning

1. PENDAHULUAN

Pembelajaran membaca Al-Qur'an dimulai dari penguasaan huruf hijaiyah sebagai unsur paling mendasar dalam proses literasi keagamaan. Pada tahap awal, anak perlu mengenali bentuk huruf, cara pengucapan, serta perbedaan bunyi antarhuruf agar mampu melanjutkan pembelajaran ke tahap harakat, suku kata, dan bacaan yang lebih kompleks. Dalam konteks pendidikan nonformal seperti Taman Pendidikan Al-Qur'an (TPQ), pengenalan huruf hijaiyah menjadi materi inti yang menentukan kualitas pembelajaran pada tahap berikutnya.

Meskipun demikian, proses belajar huruf hijaiyah di lapangan masih sering mengandalkan buku iqra' dan penjelasan lisan guru secara berulang[1]. Pola pembelajaran seperti ini memiliki keterbatasan, terutama ketika jumlah peserta didik banyak, rentang kemampuan siswa beragam, dan waktu pendampingan guru terbatas[2]. Anak juga mudah merasa jenuh ketika pembelajaran hanya berbasis pengulangan tanpa dukungan media audio-visual yang menarik[3]. Pada saat yang sama, kemampuan anak dalam membedakan huruf yang bentuk dan bunyinya mirip, seperti ha, kha, atau "ain, memerlukan latihan yang konsisten dan umpan balik yang cepat.

Perkembangan perangkat bergerak berbasis Android membuka peluang pengembangan media pembelajaran yang lebih fleksibel, personal, dan interaktif[3]. Smartphone tidak lagi hanya berfungsi sebagai sarana komunikasi, tetapi juga dapat dimanfaatkan sebagai alat bantu belajar yang dapat diakses kapan pun. Media pembelajaran berbasis Android memungkinkan kombinasi teks, audio, ikon, dan interaksi langsung sehingga lebih sesuai dengan karakteristik belajar anak usia sekolah dasar[1].

Dalam pembelajaran bacaan, aspek penting yang belum banyak difasilitasi oleh aplikasi sederhana adalah kemampuan sistem untuk menilai pelafalan pengguna. Kehadiran teknologi *speech recognition* memungkinkan sistem menerima masukan suara, mengubahnya menjadi representasi teks, lalu memprosesnya untuk kepentingan evaluasi[4]. Dalam bidang pengolahan suara, teknologi *speech* recognition telah banyak digunakan untuk mengubah sinyal suara menjadi representasi teks yang dapat diproses lebih lanjut dalam berbagai aplikasi pembelajaran berbasis Android [5]. Teknologi ini relevan untuk pembelajaran huruf hijaiyah karena memberi peluang bagi pengguna untuk memperoleh umpan balik langsung setelah melafalkan huruf atau suku kata tertentu.

Di sisi lain, dalam bidang pengolahan suara, teknologi *speech recognition* telah banyak digunakan untuk mengubah sinyal suara menjadi representasi teks yang dapat diproses lebih lanjut dalam berbagai aplikasi pembelajaran berbasis Android. Pemanfaatan teknologi ini tidak hanya terbatas pada pembelajaran bahasa umum, tetapi juga telah diterapkan dalam konteks pembelajaran Al-Qur’an untuk membantu mendeteksi kesalahan

Maknun Fil Safah: * Penulis Korespondensi

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 2, Juni 2026
https://doi.org/10.58602/jaiti.v4i2.242
Page 76-88

bacaan secara otomatis[6],[7],[8]. Pendekatan tersebut menunjukkan bahwa *speech recognition* memiliki potensi besar dalam meningkatkan interaktivitas dan efektivitas proses belajar mandiri.

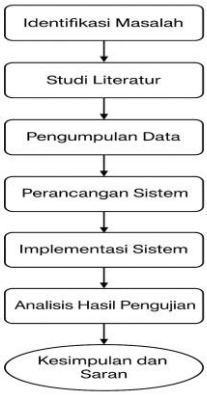
Namun, hasil pengenalan suara tidak selalu identik dengan target yang diucapkan. Variasi intonasi, kejelasan artikulasi, aksen, dan gangguan lingkungan dapat memunculkan perbedaan karakter pada hasil transkripsi. Oleh sebab itu, diperlukan mekanisme evaluasi yang tidak hanya memeriksa kecocokan mutlak, tetapi juga mampu mengukur kedekatan hasil transkripsi dengan target yang benar[5]. Pada penelitian ini, algoritma *Levenshtein Distance* dipilih karena mampu menghitung jumlah operasi edit minimum yang diperlukan untuk mengubah satu string menjadi string lain.

Sejumlah penelitian sebelumnya menunjukkan bahwa *Levenshtein Distance* efektif digunakan untuk deteksi kesalahan ejaan pada bahasa Indonesia, kamus digital, dan sistem pemeriksa kata[9]. Di sisi lain, penelitian tentang aplikasi pembelajaran huruf hijaiyah berbasis Android telah menegaskan pentingnya media interaktif untuk meningkatkan motivasi belajar anak. Celah penelitian muncul ketika kedua pendekatan tersebut belum banyak diintegrasikan secara spesifik untuk mengevaluasi pelafalan huruf hijaiyah melalui masukan suara. Karena itu, penelitian ini menempatkan *speech recognition* dan *Levenshtein Distance* dalam satu alur evaluasi bacaan yang lebih adaptif. Penerapan *Levenshtein Distance* juga telah dikembangkan pada berbagai sistem berbasis teks, seperti penerjemah bahasa dan analisis sentimen, yang menunjukkan flesksibilitas algoritma dalam berbagai konteks pengolahan bahasa alami [10].

Penelitian ini bertujuan merancang dan mengimplementasikan aplikasi pembelajaran huruf hijaiyah berbasis Android yang mampu menampilkan materi, menyediakan audio contoh pengucapan, merekam suara pengguna, serta menghitung similarity hasil pengucapan menggunakan algoritma *Levenshtein Distance*. Secara praktis, aplikasi diharapkan membantu peserta didik TPQ berlatih secara mandiri dan membantu guru memperoleh media evaluasi pelafalan yang lebih konsisten. Secara akademik, penelitian ini memperkaya kajian pada pengembangan aplikasi edukasi, pengolahan suara, dan pemanfaatan algoritma string matching pada konteks pembelajaran bacaan Arab.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan pengembangan perangkat lunak (*software development)* untuk menghasilkan aplikasi pembelajaran huruf hijaiyah berbasis Android. Lokasi penelitian berada di TPQ Al-Adnani, Kayangan, Diwek, Jombang, pada rentang September sampai Desember 2025. Pemilihan lokasi tersebut didasarkan pada kesesuaian subjek penelitian, yaitu anak-anak usia sekolah dasar yang sedang mempelajari pengenalan huruf hijaiyah.



Gambar 1. Prosedur Penelitian

Maknun Fil Safah: * Penulis Korespondensi

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 2, Juni 2026
https://doi.org/10.58602/jaiti.v4i2.242
Page 76-88

Tahapan penelitian dimulai dari identifikasi masalah, studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi aplikasi, dan pengujian. Langkah – langkah yang dilakukan adalah:

2.1 Identifikasi Masalah

Penelitian ini dimulai dengan mengidentifikasi permasalahan yang akan diangkat sebagai bahan penelitian. Berdasarkan latar belakang masalah yang telah dipaparkan sebelumnya yaitu, anak-anak masih kesulitan membedakan bentuk huruf maupun membaca dengan benar.

2.2 Studi Literatur

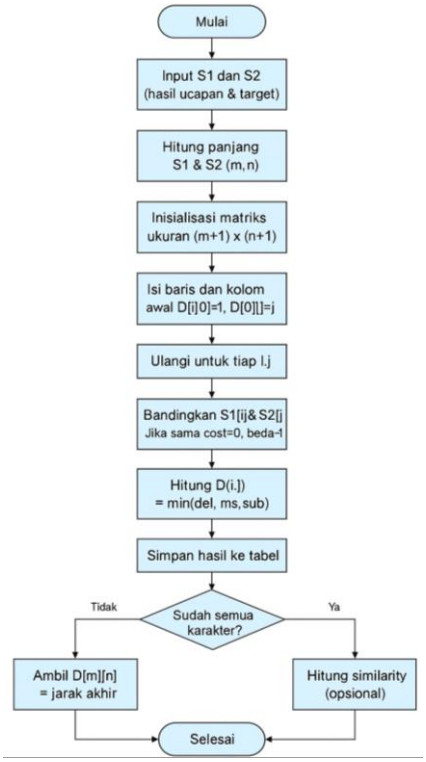
Peneliti mencari sumber referensi yang terkait dengan topik penulisan seperti artikel ilmiah, buku dan mempelajari serta mendalami fakta-fakta dan data-data serta informasi yang diperoleh berkaitan dengan penelitian.

2.3 Pengumpulan Data

Pengumpulan data dilakukan melalui observasi terhadap proses belajar di TPQ, wawancara dengan pihak terkait, serta studi pustaka dari jurnal, artikel ilmiah, dan penelitian terdahulu yang berkaitan dengan Android, *speech recognition*, dan algoritma *string matching*.

2.4 Perancangan sistem

Perancangan sistem dilakukan menggunakan *flowchart* dan use case diagram. *Flowchart* menggambarkan alur utama aplikasi dari halaman awal sampai evaluasi hasil. *Use case diagram* menunjukkan interaksi pengguna dengan fitur aplikasi, sedangkan activity diagram mendeskripsikan urutan aktivitas pengguna, sistem, dan basis data. Rancangan antarmuka disusun agar sesuai dengan karakteristik pengguna anak, yaitu menggunakan tombol yang besar, ikon yang mudah dipahami, dan alur navigasi yang sederhana.



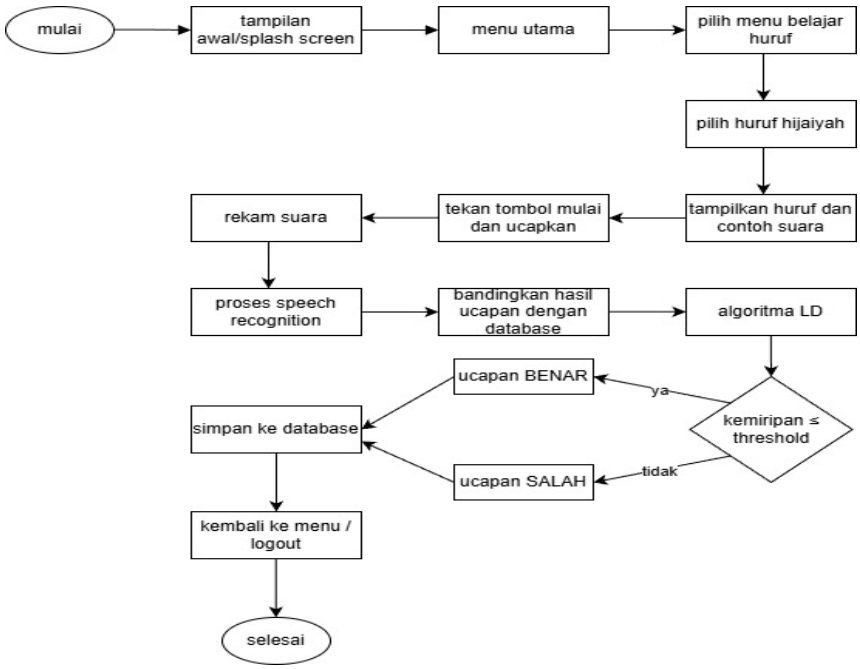
Gambar 2. *Flowchart* Algoritma *Levenshtein Distance*

Maknun Fil Safah: * Penulis Korespondensi

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 2, Juni 2026
https://doi.org/10.58602/jaiti.v4i2.242
Page 76-88

Pada gambar 2 Flowchart tersebut menggambarkan tahapan kerja algoritma *Levenshtein Distance* yang berfungsi untuk mengukur jarak edit antara dua string, yaitu jumlah minimum operasi yang dibutuhkan—meliputi penambahan, penghapusan, dan substitusi karakter—untuk mentransformasikan string pertama (S1) menjadi string kedua (S2). Proses diawali dengan memasukkan kedua string, kemudian menentukan panjang masing-masing string (m dan n). Selanjutnya dibangun sebuah matriks berukuran (m+1) × (n+1) yang digunakan sebagai media penyimpanan hasil perhitungan secara bertahap. Inisialisasi dilakukan pada baris pertama dan kolom pertama dengan nilai berurutan, yang merepresentasikan biaya dasar untuk operasi penghapusan atau penambahan karakter.

Tahap berikutnya melibatkan iterasi pada setiap pasangan karakter dari S1 dan S2. Pada setiap iterasi dilakukan perbandingan karakter; jika keduanya identik maka biaya bernilai nol, sedangkan jika berbeda diberikan biaya satu. Nilai pada setiap sel matriks kemudian ditentukan dengan memilih nilai minimum dari tiga kemungkinan operasi, yaitu deletion, insertion, dan substitution. Hasil perhitungan tersebut disimpan dalam matriks, dan proses ini berlanjut hingga seluruh elemen matriks terisi. Setelah seluruh perhitungan selesai, nilai pada posisi akhir matriks (D[m][n]) diambil sebagai hasil utama berupa jarak *Levenshtein*, yang selanjutnya dapat dimanfaatkan untuk menghitung tingkat kemiripan antara kedua string. Rumus umum algoritma *Levenshtein Distance* dapat dituliskan sebagai berikut:



Gambar 3. *Flowchart* Aplikasi Membaca Huruf Hijaiyah

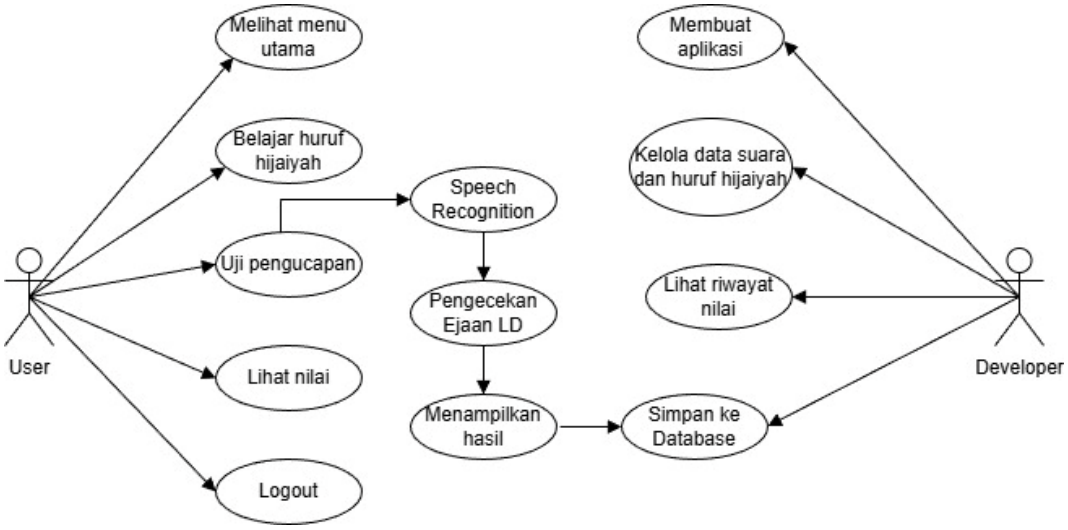
Gambar 3 menjelaskan Flowchart tersebut menunjukkan alur aplikasi pembelajaran huruf hijaiyah berbasis pengenalan suara. Proses dimulai dari *splash screen* menuju menu utama, kemudian pengguna memilih menu belajar dan menentukan huruf hijaiyah yang ingin dipelajari. Aplikasi menampilkan huruf beserta contoh pelafalannya, lalu pengguna menekan tombol mulai dan mengucapkan huruf yang dipilih. Suara pengguna direkam dan diproses menggunakan *speech recognition* untuk diubah menjadi data yang dapat dianalisis.

Selanjutnya, hasil pengucapan dibandingkan dengan data referensi menggunakan algoritma *Levenshtein Distance* untuk mengukur tingkat kemiripan. Jika nilai kemiripan

Maknun Fil Safah: * Penulis Korespondensi

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 2, Juni 2026
https://doi.org/10.58602/jaiti.v4i2.242
Page 76-88

memenuhi ambang batas yang ditentukan, maka pengucapan dinyatakan benar, sedangkan jika tidak maka dianggap salah. Hasil evaluasi tersebut disimpan ke dalam basis data, dan pengguna dapat kembali ke menu utama atau mengakhiri penggunaan aplikasi.



Gambar 4. *Use Case Diagram*

Pada gambar 4 *Use case* pengguna pada diagram tersebut menggambarkan interaksi antara user dengan sistem dalam aplikasi pembelajaran huruf hijaiyah berbasis pengenalan suara. Pengguna dapat mengakses menu utama, kemudian memilih fitur belajar huruf hijaiyah atau melakukan uji pengucapan. Pada proses ini, suara pengguna diproses melalui teknologi *speech recognition* dan dievaluasi menggunakan metode *Levenshtein Distance* untuk memeriksa kesesuaian pelafalan. Hasil evaluasi kemudian ditampilkan kepada pengguna sebagai umpan balik terhadap tingkat ketepatan pengucapan.

Selain itu, pengguna juga memiliki akses untuk melihat nilai hasil latihan yang telah dilakukan serta melakukan logout dari sistem. Seluruh hasil pengujian disimpan ke dalam basis data, sehingga dapat ditampilkan kembali sebagai riwayat pembelajaran. Dengan demikian, *use case* ini menunjukkan alur fungsional utama yang mendukung proses belajar, evaluasi, dan pemantauan perkembangan kemampuan pengguna secara terstruktur.

2.5 Algoritma *Levenshtein Distance*

Algoritma *Levensthein Distance* ditemukan oleh seorang ilmuan asal Rusia bernama Vladimir Levensthein pada tahun 1965, algoritma ini juga disebut dengan algoritma Edit Distance. *Levensthein Distance* adalah suatu matriks untuk mengukur jumlah perbedaan antara dua string. *Levensthein Distance* dua buah string adalah jumlah minimum operasi yang dibutuhkan untuk mengubah satu string *(source string*) menjadi string yang lain (target string), dimana suatu operasi melibatkan penyisipan (*insertion*), penghapusan (*deletion*), dan penggantian (*substitution*) dari suatu karakter tunggal. Perhitungan edit distance didapat dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string. Perhitungan jarak antara dua string ini ditentukan dari jumlah minimum operasi perubahan untuk membuat string A menjadi String B [5]. Rumus algoritma Levenshtein Distance dapat dituliskan sebagai berikut:

Maknun Fil Safah: * Penulis Korespondensi

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 2, Juni 2026
https://doi.org/10.58602/jaiti.v4i2.242
Page 76-88

$$D(i,j) = \begin{cases} 0, & \text{jika } i = 0 \text{ dan } j = 0 \\ i, & \text{jika } j = 0 \\ j, & \text{jika } i = 0 \\ \min \begin{cases} D(i-1,j)+1, & \text{(penghapusan)} \\ D(i,j-1)+1, & \text{(penyisipan)} \\ D(i-1,j-1)+\text{cost}, & \text{(substitusi)} \end{cases}, & \text{jika } i,j > 0 \end{cases} \quad (1)$$

$$\text{dengan cost} = \begin{cases} 0, & \text{jika } s_i = t_j \\ 1, & \text{jika } s_i \neq t_j \end{cases}$$

Algoritma *Levenshtein Distance* diterapkan setelah sistem memperoleh teks hasil *speech recognition*. Jika string hasil transkripsi dinyatakan sebagai S1 dan target pembelajaran sebagai S2, maka nilai jarak edit dihitung melalui matriks dua dimensi. Nilai akhir pada sel D(m,n) menyatakan jumlah minimum operasi edit yang dibutuhkan. Untuk mengubah nilai distance menjadi persentase kemiripan, penelitian ini menggunakan persamaan similarity berikut:

$$Similarity = 1 - \left(\frac{D}{max\big(len(string1), len(string2)\big)} \right) x100 \quad (2)$$

Nilai *similarity* kemudian ditafsirkan menggunakan tiga ambang klasifikasi, yaitu lebih dari atau sama dengan 80% dinyatakan benar, 60% sampai 79% dinyatakan mendekati, dan kurang dari 60% dinyatakan salah. Pemakaian tiga kategori ini dimaksudkan agar sistem tidak hanya memberi penilaian biner, tetapi juga menyediakan ruang bagi pengguna yang pengucapannya belum sepenuhnya tepat namun masih dekat dengan target.

Algoritma *Levenshtein Distance* cukup baik digunakan dalam penghitungan tingkat *similarity* antara dua buah string, khususnya pada penelitian ini. Terlihat dari nilai korelasi sebesar 89.47%, dimana nilai 100% bernilai sangat baik dan -100% bernilai sangat buruk. Secara keseluruhan, penerapan algoritma *Levenshtein Distance* dalam aplikasi pembelajaran huruf hijaiyah berbasis Android ini terbukti mampu memberikan evaluasi bacaan yang objektif dan terukur. Dengan tingkat akurasi mencapai 90%, algoritma ini dapat dijadikan metode yang efektif untuk memeriksa kesesuaian hasil pengucapan huruf hijaiyah oleh pengguna. Hasil tersebut mendukung fungsi utama aplikasi sebagai media edukatif yang interaktif dan adaptif dalam membantu anak-anak belajar membaca huruf hijaiyah dengan cara yang menarik dan berbasis teknologi modern[11].

Selain itu, dalam implementasi sistem berbasis *speech-to-text*, integrasi dengan teknik pengolahan bahasa alami (*Natural Language Processing*) menjadi aspek penting dalam meningkatkan kualitas hasil evaluasi. Pendekatan ini memungkinkan sistem tidak hanya mentranskripsi suara, tetapi juga menganalisis kesesuaian hasil transkripsi terhadap target pembelajaran secara lebih kontekstual[12].

Penelitian ini memberikan kontribusi dalam pengembangan media pembelajaran berbasis teknologi dengan pendekatan evaluasi pelafalan yang lebih adaptif. Secara umum, algoritma Levenshtein Distance banyak dimanfaatkan dalam pengolahan teks, khususnya pada sistem *spell checker* dan koreksi kesalahan ejaan kata untuk meningkatkan kualitas penulisan [13], [14]. Namun, pemanfaatannya dalam konteks evaluasi hasil *speech recognition* masih relatif terbatas dan umumnya hanya digunakan sebagai metrik pengukuran kesalahan transkripsi. Oleh karena itu, penelitian ini mengusulkan penerapan algoritma *Levenshtein Distance* untuk mengukur tingkat kemiripan antara hasil transkripsi suara (*speech-to-text*) dan target bacaan huruf hijaiyah. Selain itu, penelitian ini juga mengembangkan aplikasi pembelajaran berbasis Android yang mengintegrasikan elemen audio, visual, serta input suara pengguna secara langsung. Melalui pendekatan ini, sistem tidak hanya melakukan transkripsi suara, tetapi juga memberikan evaluasi pelafalan

Maknun Fil Safah: * Penulis Korespondensi

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 2, Juni 2026
https://doi.org/10.58602/jaiti.v4i2.242
Page 76-88

berbasis nilai *similarity* dengan klasifikasi bertingkat, yaitu benar, mendekati, dan salah. Dengan demikian, penelitian ini memperluas pemanfaatan algoritma *Levenshtein Distance* dari domain koreksi teks menuju evaluasi pelafalan berbasis suara secara lebih kontekstual dan aplikatif.

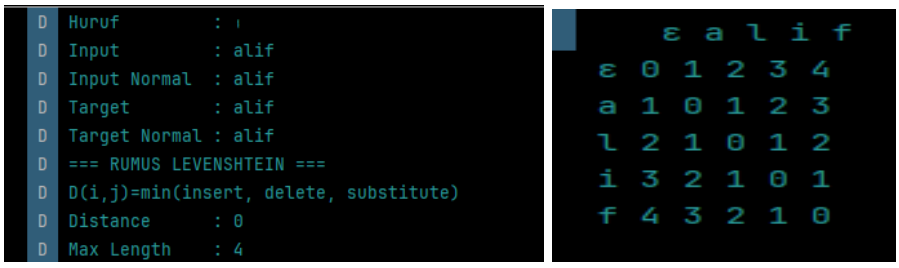
3. HASIL DAN PEMBAHASAN

Hasil penelitian mencakup implementasi aplikasi, pengujian fungsi utama, serta analisis terhadap keluaran *speech recognition* dan perhitungan *similarity*. Aplikasi yang dibangun memiliki beberapa tampilan inti, yaitu splash screen, menu utama, menu materi, menu play, konfirmasi perekaman, hasil evaluasi, dan riwayat hasil belajar. Struktur ini dirancang untuk mendukung proses belajar bertahap: pengguna memilih huruf, mendengarkan contoh suara, menirukan, lalu menerima penilaian otomatis.

Algoritma *Levenshtein Distance* pada penelitian ini diimplementasikan sebagai metode utama dalam melakukan evaluasi kesesuaian antara hasil pengucapan pengguna dan huruf hijaiyah yang menjadi target pembelajaran. Berdasarkan implementasi yang telah dilakukan, algoritma *Levenshtein Distance* terbukti efektif dalam mendeteksi kesalahan kecil pada pengucapan, seperti penambahan atau pengurangan karakter.

```
private fun levenshteinDistance(a: String, b: String): Int {
    val dp = Array( size = a.length + 1) { IntArray( size = b.length + 1) }

    for (i in 0 ≤ .. ≤ a.length) dp[i][0] = i
    for (j in 0 ≤ .. ≤ b.length) dp[0][j] = j

    for (i in 1 ≤ .. ≤ a.length) {
        for (j in 1 ≤ .. ≤ b.length) {
            val cost = if (a[i - 1] == b[j - 1]) 0 else 1

            dp[i][j] = minOf(
                a = dp[i - 1][j] + 1,
                b = dp[i][j - 1] + 1,
                c = dp[i - 1][j - 1] + cost
            )
        }
    }

    return dp[a.length][b.length]
}
```

Gambar 5. *Source Code* Implementasi Algoritma *Levenshtein Distance*



Gambar 6. Hasil Perhitungan Algoritma *Levenshtein Distance*

Pada gambar 5 dan gambar 6 hasil pengujian memperlihatkan bahwa algoritma *Levenshtein Distance* mampu memberikan evaluasi bacaan secara otomatis, terukur, dan informatif. Sistem tidak hanya menghasilkan keputusan benar atau salah, tetapi juga memberikan kategori mendekati yang penting dalam proses pembelajaran. Dengan

Maknun Fil Safah: * Penulis Korespondensi

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 2, Juni 2026
https://doi.org/10.58602/jaiti.v4i2.242
Page 76-88

demikian, aplikasi yang dikembangkan telah mampu menjalankan fungsi utamanya sebagai media pembelajaran interaktif yang membantu pengguna melatih pengucapan huruf hijaiyah secara mandiri.

```
private fun mulaiRekamSuara() {
    val intent = Intent( action = RecognizerIntent.ACTION_RECOGNIZE_SPEECH).apply {
        putExtra(
            name = RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            value = RecognizerIntent.LANGUAGE_MODEL_FREE_FORM
        )
        putExtra( name = RecognizerIntent.EXTRA_LANGUAGE, value = "id-ID")
        putExtra(
            name = RecognizerIntent.EXTRA_PROMPT,
            value = "Ucapkan huruf hijaiyah dengan jelas ya"
        )
    }

    try {
        speechLauncher.launch( input = intent)
```

Gambar 7. *Source Code* Implementasi *Speech Recognition*

Pada gambar 7 level implementasi, *speech recognition* digunakan untuk menangkap suara pengguna ketika tombol perekaman diaktifkan. Hasil pengenalan suara kemudian dipetakan menjadi teks yang selanjutnya dibandingkan dengan target huruf atau suku kata. Ketika hasil transkripsi identik dengan target, sistem memberi nilai *similarity* 100% dan mengelompokkannya sebagai benar. Sebaliknya, ketika terjadi penambahan, pengurangan, atau penggantian karakter, nilai *similarity* turun sesuai besarnya *distance* yang dihasilkan.

```
private fun hitungSimilarity(target: String, inputUser: String): Int {
    val a = inputUser.lowercase().trim()
    val b = target.lowercase().trim()

    if (a.isEmpty() && b.isEmpty()) return 100
    if (a.isEmpty() || b.isEmpty()) return 0

    val d = levenshteinDistance(a, b)
    val maxLen = maxOf( a = a.length, b = b.length)

    val similarity = (1.0 - d.toDouble() / maxLen.toDouble()) * 100.0
    return similarity.toInt().coerceIn(0, 100)
}
```

Gambar 8. *Source Code* Akurasi Perhitungan *Similarity*

```
D  === RUMUS SIMILARITY ===
D  (1 - distance/maxLen) x 100
D  (1 - 0/4) x 100
D  Similarity     : 100%
D  Kategori       : Benar
```

Gambar 9. Hasil Perhitungan *Similarity*

Berdasarkan gambar 8 dan gambar 9 implementasi sistem, nilai *similarity* digunakan sebagai dasar dalam mengklasifikasikan hasil pengucapan ke dalam tiga kategori, yaitu: **Benar**, jika nilai similarity ≥ 80%. **Mendekati**, jika nilai *similarity* berada pada rentang 60% – 79%. **Salah**, jika nilai *similarity* < 60%. Hasil ini menunjukkan bahwa metode *similarity* berbasis *Levenshtein Distance* mampu memberikan evaluasi yang proporsional dan terukur terhadap kesalahan pengucapan.

Maknun Fil Safah: * Penulis Korespondensi

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 2, Juni 2026
https://doi.org/10.58602/jaiti.v4i2.242
Page 76-88

Untuk menilai kinerja sistem, pengujian dilakukan terhadap 15 sampel bacaan. Setiap sampel berisi target huruf atau suku kata, hasil speech recognition, nilai edit distance, panjang maksimum string, nilai similarity, dan kategori akhir. Data tersebut menunjukkan variasi keluaran sistem dalam menghadapi hasil pelafalan yang tepat, hampir tepat, maupun jauh dari target.

Tabel 1. Hasil Pengujian Similarity Pada 15 Sampel Bacaan

No	Target	Hasil Speech Recognition	Distance	Panjang Maks.	Similarity	Kategori
1	Ba	Ba	0	2	100%	Benar
2	Ta	Ta	0	2	100%	Benar
3	Tsa	Tsa	0	3	100%	Benar
4	Jim	Jim	0	3	100%	Benar
5	Dal	Dal	0	3	100%	Benar
6	Sin	Sin	0	3	100%	Benar
7	Ha	Ka	1	2	50%	Salah
8	Sin	Sn	1	3	66,67%	Mendekati
9	Sin	Sina	1	4	75%	Mendekati
10	Dal	Darl	1	4	75%	Mendekati
11	Mim	Mi	1	3	66,67%	Mendekati
12	Lam	Lla	2	3	33,33%	Salah
13	Nun	Run	1	3	66,67%	Mendekati
14	Kaf	Kah	1	3	66,67%	Mendekati
15	Dal	Sendal	3	6	50%	Salah

Berdasarkan Tabel 1, hasil pengujian terhadap 15 sampel bacaan menunjukkan bahwa sebanyak 6 data (40%) termasuk dalam kategori benar, 6 data (40%) berada pada kategori mendekati, dan 3 data (20%) masuk dalam kategori salah. Distribusi tersebut mengindikasikan bahwa sistem memiliki kemampuan yang cukup baik dalam mengenali serta mengevaluasi pelafalan pengguna, di mana sebagian besar hasil berada pada kategori benar dan mendekati. Hal ini menunjukkan bahwa pendekatan berbasis *similarity* yang diterapkan mampu memberikan penilaian yang lebih adaptif terhadap variasi pengucapan, sehingga tidak terbatas pada klasifikasi biner, melainkan juga mempertimbangkan tingkat kedekatan hasil pelafalan terhadap target bacaan. Karakteristik ini penting dalam konteks pembelajaran, karena pengguna tidak sekadar memperoleh respons hitam-putih, melainkan juga penilaian bertingkat yang lebih edukatif.

Kasus pada data "Ha" yang dikenali sebagai "Ka" menghasilkan *similarity* 50%. Hasil ini menunjukkan bahwa satu penggantian karakter pada string yang sangat pendek dapat menurunkan *similarity* secara signifikan. Sebaliknya, kasus "Sin" yang dikenali sebagai "Sn" atau "Sina" tetap memperoleh kategori mendekati karena hanya terjadi satu operasi *insertion* atau *deletion* pada string dengan panjang lebih besar. Dengan demikian, panjang string maksimum berpengaruh terhadap proporsi *similarity* yang dihasilkan.

Dari sudut pandang implementasi algoritma, *Levenshtein Distance* terbukti sesuai untuk data target yang pendek seperti huruf hijaiyah dan suku kata sederhana. Setiap perubahan satu karakter dapat segera dipetakan sebagai *insertion, deletion*, atau *substitution*. Keunggulan ini memudahkan sistem menghasilkan evaluasi yang terukur, sederhana, dan cepat dieksekusi pada perangkat Android. Hal tersebut sejalan dengan kebutuhan aplikasi edukasi yang membutuhkan respons instan agar interaksi belajar tidak terputus.

Maknun Fil Safah: * Penulis Korespondensi

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 2, Juni 2026
https://doi.org/10.58602/jaiti.v4i2.242
Page 76-88

Hasil pengujian menunjukkan bahwa sistem mampu mengklasifikasikan tingkat kemiripan pelafalan ke dalam kategori benar, mendekati, dan salah secara konsisten. Temuan ini sejalan dengan penelitian sebelumnya yang memanfaatkan teknologi *speech recognition* dalam evaluasi pembelajaran berbasis suara, di mana sistem mampu memberikan umpan balik otomatis dengan tingkat akurasi yang cukup baik [12],[15]. Perbedaan utama pada penelitian ini terletak pada penggunaan algoritma Levenshtein Distance sebagai mekanisme evaluasi berbasis *similarity*, sehingga penilaian tidak hanya bersifat benar atau salah, tetapi juga mempertimbangkan tingkat kedekatan hasil pengucapan pengguna terhadap target bacaan.

Namun, hasil pengujian juga memperlihatkan keterbatasan penting. Pertama, keluaran *speech recognition* dipengaruhi oleh kejelasan suara, kebisingan lingkungan, dan perbedaan artikulasi pengguna. Kedua, *Levenshtein Distance* bekerja pada representasi teks, bukan pada ciri fonetik yang sebenarnya. Akibatnya, huruf-huruf dengan kemiripan bunyi dapat menghasilkan keluaran teks yang kurang konsisten. Kasus ini mengindikasikan bahwa peningkatan di masa mendatang dapat diarahkan pada penggabungan pendekatan fonetik atau model pembelajaran mesin yang lebih sensitif terhadap karakteristik bunyi Arab.

Jika dibandingkan dengan penelitian terdahulu, hasil penelitian ini memberikan kontribusi pada dua sisi sekaligus. Dari sisi pengembangan media, aplikasi menyediakan pengalaman belajar yang lebih menarik dibanding metode yang hanya mengandalkan buku. Dari sisi evaluasi, sistem tidak hanya memanfaatkan speech recognition sebagai alat transkripsi, tetapi juga menambahkan mekanisme pengukuran *similarity* untuk menghasilkan umpan balik otomatis yang lebih bermakna. Dengan demikian, aplikasi tidak sekadar menyajikan materi, melainkan juga berfungsi sebagai alat latihan dan penilaian dasar.

Secara praktis, aplikasi dapat dimanfaatkan sebagai media pendamping pembelajaran di TPQ maupun di rumah. Anak dapat mengulangi latihan tanpa harus selalu menunggu koreksi manual dari guru. Guru atau orang tua juga memperoleh sarana bantu untuk memonitor kecenderungan hasil belajar melalui riwayat latihan. Keunggulan utama penelitian ini terletak pada penyederhanaan proses evaluasi bacaan menjadi alur yang ringan, mudah dipahami, dan sesuai dengan perangkat yang umum digunakan masyarakat.

Walaupun hasilnya menunjukkan kinerja yang baik, penelitian ini masih memiliki sejumlah keterbatasan. Materi yang dibahas baru mencakup pengenalan huruf hijaiyah dasar dan belum mencakup tajwid, harakat, atau pembacaan rangkaian ayat. Selain itu, pengujian dilakukan pada jumlah sampel yang terbatas sehingga generalisasi hasil perlu dilakukan secara hati-hati. Oleh sebab itu, penelitian lanjutan dapat diarahkan pada perluasan dataset, integrasi analisis fonetik, pemakaian model pengenalan suara yang lebih spesifik terhadap bacaan Arab, dan penambahan fitur gamifikasi agar motivasi belajar anak semakin meningkat.

4. KESIMPULAN

Penelitian ini berhasil menghasilkan aplikasi pembelajaran huruf hijaiyah berbasis Android yang memadukan materi visual, audio pelafalan, input suara, dan evaluasi otomatis. Perancangan sistem yang berbasis analisis kebutuhan, flowchart dan use case diagram mampu diterjemahkan menjadi aplikasi yang berjalan sesuai tujuan penelitian.

Implementasi *speech recognition* memungkinkan sistem menerima dan mentranskripsi suara pengguna secara *real-time*. Selanjutnya, algoritma *Levenshtein Distance* mampu mengukur tingkat kemiripan antara hasil transkripsi dan target pembelajaran melalui nilai *edit distance* dan *similarity*. Hasil pengujian menunjukkan

Maknun Fil Safah: * Penulis Korespondensi

akurasi sistem sebesar 90%, dengan distribusi 6 data benar, 6 data mendekati, dan 3 data salah.

Dengan demikian, kombinasi *speech recognition* dan *Levenshtein Distance* terbukti layak digunakan sebagai dasar evaluasi pengucapan pada pembelajaran huruf hijaiyah. Aplikasi ini berpotensi menjadi media belajar pendamping yang interaktif, membantu pengguna berlatih secara mandiri, serta memberikan umpan balik yang lebih cepat dan konsisten. Penelitian lanjutan disarankan untuk menambahkan analisis fonetik, memperluas cakupan materi, dan meningkatkan kualitas pengujian dengan jumlah responden yang lebih besar.

5. UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada Program Studi Teknik Informatika Universitas Hasyim Asy'ari, pembimbing penelitian, serta TPQ Al-Adnani Kayangan Diwek Jombang yang telah memberikan dukungan, kesempatan, dan data yang diperlukan selama proses penelitian dan penyusunan artikel ini.

6. REFERENCES

[1] I. Bahroni dan R. Purwanto, "Aplikasi Pembelajaran (E-learning) Mengenal Huruf Hijaiyah bagi Anak-anak Berbasis Mobile untuk Mendukung Pembelajaran Secara Mandiri," *J. Edukasi dan Penelit. Inform.*, vol. 4, no. 2, hal. 163, 2023, doi: 10.26418/jp.v4i2.25566.

[2] N. Hidayat, D. Kurniawan, A. H. Prabawa, R. Rusnoto, dan A. N. Syafiq, "Peningkatan Kemampuan Membaca Anak Menggunakan Aplikasi Android Belajar Membaca di Dusun Kentengsari Kaliwungu Semarang," *J. Ilm. Kampus Mengajar*, no. 2013, hal. 72–79, 2022, doi: 10.56972/jikm.v2i2.45.

[3] Nofaa, Hapsari, dan Putri, "Aplikasi Pembelajaran Huruf Hijaiyah Berbasis Android," *J. Ilm. Tek.*, vol. 2, no. 1, hal. 11–19, 2023, doi: 10.56127/juit.v2i1.473.

[4] Evanti, "Penerapan Speech Recognition Pada Aplikasi Cek Pengucapan Bahasa Inggris Berbasis Android," vol. 12, no. 1, hal. 57–64, 2023.

[5] Y. P. Sari, G. A. Pradnyana, dan I. M. A. Wirawan, "Pengembangan Aplikasi Kamus Bahasa Bima - Bahasa Indonesia Menggunakan Algoritma Levenshtein Distance Sebagai Spell Checker Berbasis Android," *Kumpul. Artik. Mhs. Pendidik. Tek. Inform.*, vol. 8, no. 2, hal. 86, 2024, doi: 10.23887/karmapati.v8i2.17964.

[6] M. I. Syaifullah, "Penerapan Teknologi Automatic Speech Recognition Menggunakan Model Wav2vec2 . 0 Sebagai Alat Bantu Untuk Mendeteksi Kesalahan Dalam Membaca Al- Qur ' an Berbasis Mobile," vol. 16, no. 2, hal. 349–359, 2024.

[7] N. A. Hudayana, A. Haris, dan A. R. Zakaria, "Pengembangan Aplikasi Penyimak Al-Qur ' an Menggunakan Teknologi AI dengan Metode Speech Recognition pada Platform Android," vol. 2, no. November, hal. 83–93, 2024.

[8] A. Sujjada, G. P. Insany, dan M. F. Nugraha, "Implementasi Automatic Speech Recognition Pada Penilaian Hafalan Al-Quran Dengan Metode Muroja ' ah Berbasis Android," vol. 10, no. 2, hal. 216–228, doi: 10.33050/cices.v10i2.3247.

[9] M. Aswin, "Pengoreksi Kesalahan Ejaan Bahasa Indonesia Menggunakan Metode Levenshtein Distance," *J. Mhs. TEUB*, vol. 1, hal. 1–6, 2024, [Daring]. Tersedia pada: http://elektro.studentjournal.ub.ac.id/index.php/teub/article/view/70

[10] I. Bagus, N. Wijana, G. R. Dantes, dan G. Indrawan, "Analisis Sentimen Data Provider Layanan Internet Pada Twitter Menggunakan Support Vector Machine

Maknun Fil Safah: * Penulis Korespondensi

(SVM) Dengan Penambahan Algoritma Levenshtein Distance," vol. 5, 2022.

[11] E. Putri, "Pembangunan Sistem Pencarian Ayat Al- Qur ' an Berbasis Mobile Menggunakan Algoritma Metaphone Berdasarkan Kemiripan Pengucapan dan Algoritma Levenshtein Distance Untuk Mengukur Tingkat Similarity Pendahuluan Studi Terkait Al- Qur ' an," vol. 5, no. 3, hal. 7832–7844, 2021.

[12] I. P. Siti Ayu Hermaliah, Shinta Puspasari, "APLIKASI ANDROID PENCARIAN FASILITAS KAMPUS BERBASIS SPEECH-TO-TEXT DAN NATURAL LANGUAGE PROCESSING (STUDI KASUS: UNIVERSITAS INDO GLOBAL MANDIRI)," vol. 10, no. 2, 2025, doi: https://doi.org/10.24252/instek.v10i2.60822.

[13] A. Bu, S. Paul, dan M. C. Mih, "Spell Checker Application Based on Levenshtein Automaton," hal. 45–53, 2021, doi: 10.1007/978-3-030-91608-4.

[14] A. Tirta Adi Kusuma dan C. Indah Ratnasari, "Perbandingan Metode Peter Norvig, Levenshtein distance, dan Damerau-Levenshtein distance : Tinjauan Literatur," *Automata*, vol. 4, no. 2, 2023, [Daring]. Tersedia pada: https://journal.uii.ac.id/AUTOMATA/article/view/28633

[15] M. Assisi, A. Septiarini, A. H. Kridalaksana, dan M. Wati, "Rancang Bangun Aplikasi Hafalan Al-Quran dengan Google Speech API Berbasis Android," vol. 6, no. 1, hal. 26–35, 2022.

Maknun Fil Safah: * Penulis Korespondensi