

Penerapan Algoritma Kalman Filter untuk Mengurangi Noise pada Sistem Pembuka Kunci Pintu dengan Pengenalan Suara

Carlos Aprillio Angger Gindaong^a, Irma Nirmala^b, Cucu Suhery^c

^{a,b,c} Universitas Tanjungpura, Pontianak, Indonesia

^{*}correspondence : h1051201083.carlos@gmail.com

Abstract— Acoustic noise refers to unwanted sound disturbances occurring during data recording and collection. This interference often mixes with desired signals, creating challenges in accurately detecting intended voice commands. This study focuses on the implementation of the Kalman Filter Algorithm to mitigate acoustic noise in a voice recognition-based door lock system. The system utilizes an INMP441 microphone module interfaced with an ESP32 microcontroller, which transmits audio data to an edge server. The Kalman Filter Algorithm is deployed on the ESP32 to denoise audio signals from environmental interference. Keyword detection is performed using six comprehensive analysis algorithms, which examine syllable structure, energy patterns, consonant-vowel differentiation, temporal patterns, spectral analysis, and voice characteristics. Experiments were conducted under various noise intensity levels and pronunciation volume variations, comparing performance both with and without the Kalman Filter. The results indicate that the Kalman Filter improved system accuracy from 69.6% to 83.2%, while effectively mitigating false positives—instances where incorrect inputs are erroneously validated as correct. The system successfully recognized the keyword "buka" (open), achieving an average response time of 6.54 seconds to unlock the door following keyword confirmation by the edge server. However, high-intensity noise, such as that produced by grinding machinery, presents significant detection challenges. Recommendations for future research include enhancing microphone sensitivity and developing adaptive keyword detection methods to manage diverse acoustic environments and noise conditions more effectively.

Keyword— Kalman Filter Algorithm, Noise, Door, Lock.

Abstrak—Noise suara merupakan gangguan suara yang tidak diinginkan yang terjadi selama perekaman dan pengumpulan data suara. Noise tersebut seringkali bercampur dengan sinyal yang diinginkan, menyebabkan kesulitan dalam mendeteksi perintah suara yang dimaksud. Penelitian ini fokus pada implementasi Algoritma Filter Kalman untuk mengurangi noise suara dalam sistem pembuka kunci pintu berbasis pengenalan suara. Sistem ini menggunakan modul mikrofon INMP441 yang terhubung ke mikrokontroler ESP32, yang mengirimkan data audio ke server edge. Algoritma Filter Kalman diterapkan pada ESP32 untuk membersihkan sinyal audio dari noise lingkungan. Deteksi kata kunci dilakukan menggunakan enam algoritma komprehensif yang menganalisis struktur suku kata, pola energi, diferensiasi konsonan-vokal, pola temporal, analisis spektral, dan karakteristik suara. Pengujian dilakukan pada berbagai tingkat kekerasan noise dan variasi volume pengucapan, baik dengan maupun tanpa Filter Kalman. Hasil pengujian menunjukkan bahwa Filter Kalman meningkatkan akurasi sistem dari 69,6% menjadi 83,2%, sekaligus secara efektif mengurangi kondisi saat kata yang diucapkan salah tetapi dinyatakan benar. Sistem berhasil mengenali kata kunci "buka" dengan waktu respons rata-rata 6,54 detik untuk membuka kunci pintu setelah kata kunci dikonfirmasi oleh edge server. Namun, kebisingan intensitas tinggi seperti suara mesin gerinda menimbulkan gangguan dalam pendeteksian. Saran penelitian di masa mendatang meliputi peningkatan sensitivitas mikrofon dan pengembangan metode deteksi kata kunci adaptif untuk menangani beragam lingkungan akustik dan kondisi kebisingan secara lebih efektif.

Kata Kunci— Algoritma Kalman Filter, Noise, Kunci, Pintu.

I. PENDAHULUAN

Sistem pembuka kunci menggunakan suara merupakan salah satu sistem keamanan yang banyak digunakan. Pengenalan suara bekerja dengan menganalisa fitur suara antara individu dikarenakan setiap orang memiliki pola suara unik yang berasal dari pola anatomi. Sistem dapat dikendalikan dengan dua mekanisme pengenalan suara yaitu *voice recognition* dan *speech recognition*. *Speech recognition* adalah teknologi yang diterapkan pada perangkat lunak untuk menerima input berupa kata yang diucapkan. *Speech*

recognition memungkinkan suatu perangkat untuk mengenali cara pengucapan kata dan mencocokkan sinyal tersebut dengan suatu pola tertentu yang tersimpan dalam suatu perangkat [1]. Dengan adanya sistem ini, pengguna hanya perlu berbicara ke sistem *input* untuk membuka pengunci pintu dengan kata kunci yang telah disesuaikan. Pada proses *input* tersebut pastinya akan mendapat gangguan suara dari lingkungan sekitar. Gangguan dari lingkungan sekitar saat pengguna melakukan *input* suara ke sistem adalah yang disebut *noise*. Untuk itu pada penelitian ini menggunakan algoritma yang dapat melakukan peredaman *noise*.

Noise adalah suatu sinyal gangguan yang bersifat akustik (suara), listrik, maupun elektronis yang hadir dalam suatu sistem dalam bentuk gangguan yang bukan merupakan sinyal yang diinginkan [2]. Suara yang terdistorsi *noise* menyebabkan terganggunya proses pengenalan suara terutama pada sistem penangkap suara. Hal tersebut terjadi karena penurunan kualitas suara yang terjadi, sehingga perlu adanya perbaikan kualitas sinyal suara agar sistem dapat mengenali suara dengan baik. Berdasarkan permasalahan tersebut, dibutuhkan suatu algoritma yang dapat meredam *noise* pada data suara. Salah satu algoritma yang dapat digunakan untuk peredaman *noise* suara adalah Algoritma Kalman Filter.

Algoritma Kalman Filter adalah sebuah algoritma untuk memproses data rekursif optimal yang sederhana dan terdiri dari dua fase yang berbeda yaitu fase prediksi dan fase *update* [3]. Metode ini memasukkan semua informasi yang disediakan untuk memproses semua pengamatan yang ada. Metode ini memperhatikan ketelitian dalam menduga nilai dari variabel penting. Kalman Filter memiliki keunggulan untuk mengestimasi/menduga suatu keadaan berdasarkan data yang terbatas. Data pengukuran terbaru menjadi suatu bagian terpenting dalam Algoritma Kalman Filter, karena data tersebut akan mengoreksi data hasil pendugaan, sehingga hasil pendugaan selalu mendekati kondisi sesungguhnya [4].

Penelitian berjudul Penerapan Algoritma Kalman Filter Untuk Mengurangi *Noise* Pada Sistem Pembuka Kunci Pintu Dengan Pengenalan Suara diharapkan dengan menerapkan Algoritma Kalman Filter dapat mengurangi *noise* suara yang terjadi saat sistem sedang bekerja yang terjadi setelah sistem menerima data suara mentah. Kemudian dapat mempermudah sistem dalam mendeteksi kata kunci yang diucapkan.

II. TINJAUAN PUSTAKA

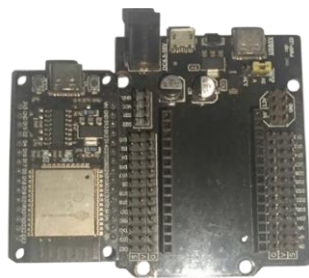
Tinjauan pustaka dimulai dengan melakukan studi literatur untuk mengumpulkan informasi mengenai Algoritma Kalman Filter yang digunakan sebagai algoritma yang dapat meredam *noise* dan mengumpulkan informasi mengenai menentukan nilai SNR sebagai nilai penentuan rasio antara *noise* dan data pada sinyal.

Beberapa penelitian terkait yang telah dilakukan yang dimana salah satu penelitian terkait tersebut memiliki tujuan untuk mengurangi *noise* menggunakan Algoritma Kalman Filter [5]. Pada penelitian tersebut dilakukan penekanan *noise* pada sinyal suara menggunakan filter Kalman dalam bentuk simulasi pada komputer dengan menggunakan bahasa pemrograman MATLAB. Simulasi dalam penelitian ini menggunakan sinyal masukan yang berupa sinyal suara dan sinyal *noise*. Setelah sinyal dimasukkan dapat dilihat kualitas dari sinyal dengan menghitung nilai PSNR, serta tampilan grafik dari sinyal dan garfik spektrogram sebelum difilter dan sinyal setelah difilter.

Penelitian terkait berikutnya yang memiliki keterkaitan dengan penggunaan Algoritma Kalman Filter dengan tujuan untuk optimasi pengukuran sensor NTC [6]. Penelitian tersebut bertujuan untuk mengurangi *noise* atau gangguan pada pembacaan sensor suhu NTC pada kompor sehingga menghasilkan pembacaan data sensor yang halus dan stabil. Hasil penelitian menunjukkan bahwa Kalman filter bekerja sangat baik untuk mengurangi *noise* dan gangguan. Pada penelitian tersebut berfokus pada penggunaan kalman filter untuk mengurangi *noise* dari pembacaan nilai pada sensor suhu NTC.

Penelitian lain yang memiliki penerapan Algoritma Kalman Filter dengan tujuan untuk melakukan optimasi pembacaan sensor *loadcell* [7]. Fokus utama dari penelitian tersebut adalah mengoptimisasi kinerja sistem penimbangan digital yang sering tidak akurat akibat gangguan *noise* pada sensor *loadcell*. Melalui implementasi Algoritma Kalman Filter pada PLC Siemens S-1200 dapat meminimalisir gangguan pembacaan sensor *loadcell* guna menghasilkan data massa yang lebih halus dan konsisten.

Berdasarkan studi literatur dan analisis kebutuhan, diperoleh komponen-komponen yang dibutuhkan sistem. Pertama, ESP32 merupakan salah satu komponen inti dalam sistem mikrokontroler yang berperan untuk menyimpan program dan sebagai sistem utama mengirim suara mentah ke *edge server* dan juga sebagai alat untuk menerima akses pembuka kunci dari *edge server* [8]. ESP32 yang digunakan pada penelitian memiliki 38 pin GPIO dan menggunakan papan ekspansi tambahan agar dapat lebih mudah saat menghubungkan ESP32 dengan perangkat lainnya. Gambar dari ESP32 dapat dilihat pada Gambar 1.



Gambar 1. ESP32 dan Papan Ekspansi

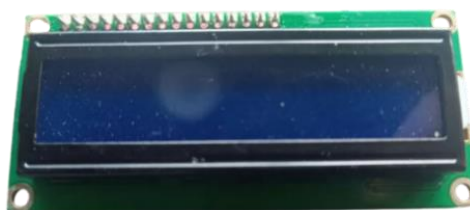


Gambar 2 INMP441

Kemudian dibutuhkan alat untuk menerima suara yaitu INMP441. Mikrofon *omnidirectional* menerima gelombang bunyi dari seluruh arah, meskipun sumber suara berpindah posisi tapi mikrofon akan tetap menunjukkan sensitivitas yang sama [9]. Nantinya setelah modul INMP441 menangkap suara, suara akan diteruskan ke ESP32 untuk dilakukan proses peredaman *noise* yang kemudian akan mengirimkan suara tersebut ke *edge server* untuk dilakukan proses mendeteksi kata kunci. Gambar dari INMP441 dapat dilihat pada Gambar 2. Sistem tersebut juga membutuhkan *solenoid* doorlock sebagai sistem pengunci pintu. Pada kondisi normal perangkat ini dalam kondisi tertutup (mengunci pintu), ketika diberi tegangan 12 volt maka kunci akan terbuka. Untuk mengendalikan *Solenoid door lock* dari arduino dibutuhkan rangkaian antarmuka atau driver. Salah satunya dapat menggunakan relay 5 volt. Dengan menggunakan relay ini maka *Solenoid door lock* dapat dikendalikan oleh *mikrokontroler* pada Arduino [10]. Gambar *solenoid doorlock* dapat dilihat pada Gambar 3.



Gambar 3. Solenoid Doorlock



Gambar 4. LCD

LCD merupakan alat yang digunakan untuk menampilkan informasi elektronik seperti tulisan dan gambar [12]. Layar LCD memiliki 16 kolom dan 2 baris yang dilengkapi dengan *i2c*. Gambar LCD dapat dilihat pada Gambar 5. Dibutuhkan juga *power supply switching* merupakan perangkat keras yang mampu menyalurkan tegangan listrik secara langsung dari sumber tegangan listrik PLN ke tegangan listrik yang lainnya. Peralatan *power audio* memerlukan *power supply* yang berdaya yang besar untuk menghasilkan suara yang besar dan tegangan drop yang rendah sehingga dapat memaksimalkan penggunaan energi listrik untuk konsumsi *power audio* [13]. Pada penelitian ini secara khusus akan menggunakan *power supply switching* dengan tegangan 12 volt. Gambar *power supply switching* dapat dilihat pada Gambar 6.



Gambar 5. Power Supply Switching

Visual studio code merupakan kode editor sumber yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux* dan *mac OS*. *Visual Code* memudahkan dalam penulisan code yang mendukung beberapa jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian code tersebut [14]. *Visual studio code* berfungsi untuk menjalankan *edge server* yang dimana

edge server tersebut digunakan untuk mengimplementasikan algoritma yang digunakan untuk mendeteksi kata buka.

Dibutuhkan juga Arduino IDE merupakan perangkat lunak yang memainkan peran yang sangat penting dalam pemrograman, kompilasi biner, dan unduhan memori mikrokontroler. Selain banyak modul pendukung Arduino telah menjadi platform karena telah menjadi pilihan bagi banyak profesional. Salah satu alasan Arduino memikat banyak orang adalah karena sifatnya yang *open source*, baik hardware maupun software [15]. Arduino IDE berfungsi sebagai implementasi Algoritma Kalman Filter pada ESP32 untuk peredaman *noise*.

Algoritma Kalman Filter merupakan merupakan sebuah algoritma yang digunakan untuk memprediksi hasil selanjutnya yang berdasarkan data-data sebelumnya. Filter ini tidak termasuk dalam filter umum seperti LPF (*Low Pass Filter*), HPF (*High Pass Filter*), dan BPF (*Band Pass Filter*). Filter ini merupakan sebuah estimator untuk memprediksi setiap keadaan atau bagian dalam sinyal yang mengandung sinyal [16].

Algoritma Kalman Filter terdiri dari dua fase yaitu fase prediksi dan fase *update*. Fase prediksi membutuhkan estimasi *state* awal dari suatu proses pada satu waktu yang selanjutnya menerima *feedback* atau umpan balik berupa hasil pengukuran sensor yang masih ditambah *noise* atau derau [17]. Persamaan (1) dan (2) merupakan persamaan Algoritma Kalman Filter pada fase prediksi

$$\begin{aligned}x_{t|t-1} &= x_{t-1|t-1} \\ P_{t-1} &= P_{t-1|t-1} + Q_t\end{aligned}$$

Fase kedua dari Algoritma Kalman Filter adalah fase *update*. Fase update merupakan fase pembaruan yang dimana prediksi dari sistem digabungkan dengan nilai *measurement* untuk menghasilkan estimasi keadaan yang lebih akurat. Persamaan (3), (4), dan (5) merupakan persamaan algoritma kalman filter fase *update*.

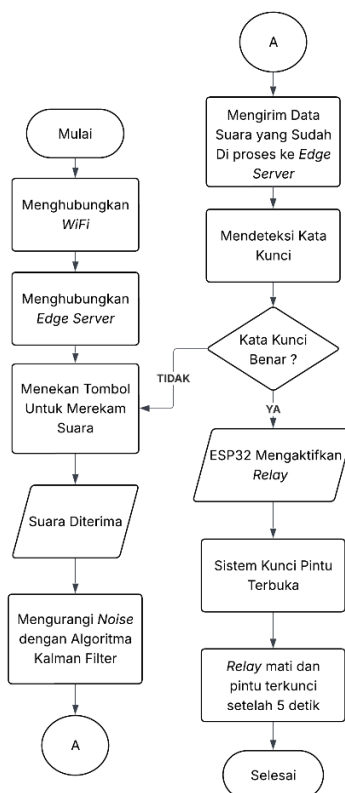
$$\begin{aligned}x_{t|t} &= x_{t|t-1} + K_t (y_t - x_{t|t-1}) \\ K_t &= P_{t|t-1} (P_{t|t-1} + R)^{-1} \\ P_{t|t} &= (1 - K_t) P_{t|t-1}\end{aligned}$$

Dibutuhkan juga nilai SNR merupakan rasio antara sinyal yang diinginkan yang bercampur dengan *noise*, dan secara luas digunakan sebagai standar ukuran kualitas sinyal untuk sistem komunikasi [18]. Suatu sinyal akan memiliki banyak gangguan oleh *noise* sehingga dapat merusak sinyal yang ada. Sinyal yang mengalami gangguan akan mengalami penurunan kualitas sinyal. Kualitas sinyal dapat ditentukan oleh nilai dari SNR yang diukur menggunakan satuan desibel (dB). Persamaan (6) merupakan persamaan umum dari SNR.

$$SNR = 10 \log_{10} \left(\frac{\rho}{1 - \rho} \right) dB$$

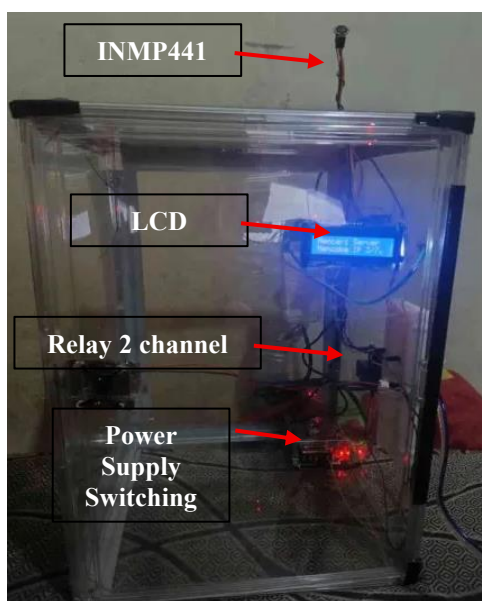
III. METODOLOGI PENELITIAN

Metodologi penelitian dilakukan dengan menentukan analisis kebutuhan, yang dilakukan untuk menunjang sistem yang terdiri dari perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras meliputi perancangan ESP32 yang berperan dalam menerima data suara yang kemudian diproses dengan Algoritma Kalman Filter sebagai peredaman *noise*, INMP441 untuk menerima suara penguapan dan suara yang ada di lingkungan sekitar, LCD untuk menampilkan pesan-pesan seperti *timer* dan status apakah kata kunci terdeteksi atau tidak, *relay* yang berperan sebagai penghubung antara ESP32, *solenoid doorlock*, dan *power supply switching* sebagai sistem pembuka kunci pintu. Sedangkan perancangan perangkat lunak meliputi perancangan program penerima suara menggunakan Arduino IDE sebagai implementasi Algoritma Kalman Filter yang kemudian ESP32 akan mengirim data suara tersebut ke *edge server* yang berperan dalam mendeteksi kata kunci dan menentukan SNR sebagai nilai rasio antara *noise* dan sinyal kata kunci. Untuk diagram alir pembuka kunci pintu dengan pengenalan suara dapat dilihat pada Gambar 7.



Gambar 6. Diagram Alir Pembuka Kunci Pintu Dengan Pengenalan Suara

Setelah melakukan perancangan maka dilakukan dengan implementasi. Implementasi perangkat keras untuk memastikan bahwa sistem secara keseluruhan perangkat keras dapat berfungsi secara optimal. Sistem dirancang untuk menerima suara dari pengucapan pengguna yang secara langsung menerima suara dari lingkungan sekitar melalui mikrofon INMP441 yang terhubung ke ESP32. LCD yang berperan menampilkan pesan terhubung ke ESP32. Sistem pengunci pintu saling terhubung antara *relay* yang terhubung ke ESP32, *power supply switching*, dan *solenoid doorlock*. Implementasi perangkat keras dapat dilihat pada Gambar 8.



Gambar 7. Implementasi Perangkat Keras

Implementasi komunikasi antara ESP32 dan *edge server* dirancang untuk memastikan data yang diterima dan sudah diproses pada ESP32 dapat terkirim dengan lancar ke *edge server*. ESP32 berperan sebagai perangkat utama yang menerima data suara dari INMP441 yang kemudian di proses untuk peredaman

noise. Kemudian data tersebut dikirimkan ke *edge server* dengan protokol HTTP, yang dimana mikrokontroler mengirimkan data tersebut dengan metode POST ke *endpoint* pada *edge server* yang telah ditentukan. Setelah data berhasil diterima oleh *edge server*, data tersebut diproses untuk mendeteksi kata kunci dan menentukan nilai rasio antara sinyal dan *noise*. Data tersebut akan dapat diakses pada *dashboard* dari *edge server* dan tersimpan ke dalam format file .csv. Hasil implementasi komunikasi antara ESP32 dan *edge server* dapat dilihat pada Gambar 9.

```
Server dimulai di http://0.0.0.0:5000
Dasbor DENGAN KALMAN: http://0.0.0.0:5000/dashboard
Endpoint Arduino: POST /arduino/voice_check
Endpoint Tes: GET /test
Endpoint Status: GET /arduino/status
```

Gambar 9. Hasil Implementasi Komunikasi Antara ESP32 dengan *Edge Server*

Implementasi Algoritma Kalman Filter dalam sistem bertujuan untuk melakukan peredaman *noise* suara berdasarkan data suara yang diterima oleh INMP441 dengan tujuan agar nilai sinyal suara memiliki nilai SNR lebih tinggi daripada *noise*. Algoritma ini menentukan prediksi dan *update* yang dimana menghasilkan nilai *update* estimasi dan *update* prediksi yang semakin mengecil seiring berjalannya iterasi.

Sistem dari Algoritma Kalman Filter dijalankan saat setelah suara diterima oleh INMP441. Data suara yang masih mengandung *noise* akan diproses pada ESP32 untuk dilakukannya peredaman *noise*. Kemudian suara tersebut dikirimkan ke *edge server* untuk diproses mencari kata kunci yang ada pada data suara tersebut. Nantinya jika ada kata kunci yang sudah disesuaikan terdeteksi maka sistem akan meminta ESP32 mengaktifkan *relay* untuk membuka pengunci pintu tersebut.

Setelah sistem selesai dibangun, dilakukan pengujian untuk memastikan setiap komponen bekerja sesuai harapan. Pengujian dilakukan dengan menguji INMP441 dengan tujuan menentukan jarak optimal antara mulut dengan INMP441. Pengujian dilakukan dengan variasi *volume* pengucapan berbeda yaitu pengucapan pelan, sedang, dan kuat. Pengujian berikutnya adalah pengujian pengenalan kata kunci dengan tujuan untuk memastikan sistem hanya dapat mengenali kata kunci yang sudah disesuaikan pada *edge server*. Kemudian pengujian berikutnya adalah pengujian sistem pengunci pintu dengan tujuan untuk mengetahui waktu sistem membuka sistem pengunci setelah kata kunci terdeteksi. Pengujian berikutnya dilakukan untuk menguji kinerja sistem yang sudah diimplementasikan Algoritma Kalman Filter dengan cara mengucapkan kata kunci yang telah ditentukan dengan variasi *noise* yang berbeda. Pengujian terakhir adalah pengujian keseluruhan sistem dengan tujuan untuk mengetahui akurasi sistem dalam mendeteksi kata kunci.

Setelah pengujian sistem telah dilakukan, penelitian diakhiri dengan perumusan kesimpulan untuk menjawab rumusan masalah serta memberikan saran untuk penelitian selanjutnya.

IV. HASIL DAN PEMBAHASAN

A. Pengujian Penerima Suara INMP441

Pengujian pertama dilakukan dengan cara menguji kemampuan INMP441 menangkap sinyal suara pengucapan berdasarkan variasi *volume* suara yang diucapkan dan perubahan jarak antara mulut dengan INMP441. Pengujian dilakukan sebanyak 5 kali setiap jarak dan setiap variasi *volume* pengucapan.

Tabel 1. Persentase Hasil Pengujian INMP441

No	Jarak	Variasi Volume Pengucapan		
		Pelan	Sedang	Kuat
1	± 5 cm	60%	80%	80%
2	± 10 cm	60%	60%	80%
3	± 15 cm	60%	40%	20%
4	± 20 cm	40%	40%	20%
5	± 25 cm	20%	0%	0%

Pada pengujian Tabel 1 dilakukan dengan variasi *volume* pengucapan yang diantaranya *volume* pelan kurang lebih 62dB, *volume* sedang kurang lebih 70dB, dan *volume* kuat kurang lebih 80dB. Dari persentase tersebut didapatkan bahwa ketika mengucapkan kata kunci dengan *volume* sedang dengan jarak kurang lebih 5 cm kemudian dengan pengucapan *volume* kuat dengan jarak kurang lebih 5 dan 10 cm, sistem dapat

mengenali kata kunci tersebut dengan lebih baik yaitu sebesar 80%. Kemudian untuk tingkat keberhasilan terkecil yaitu sebesar 20% dapat terlihat pada jarak kurang lebih 15 cm dan 20 cm pada kondisi *volume* pengucapan kuat. Kemudian pada jarak kurang lebih 25 cm, sistem tidak dapat mengenali kata kunci yang diucapkan saat *volume* pengucapan sedang dan kuat, tetapi dapat mendeteksi kata kunci saat *volume* pengucapan pelan yaitu sebesar 20%.

B. Pengujian Pengenalan Kata Kunci

Pengujian kedua dilakukan dengan cara mengucapkan kata kunci yang ditentukan dengan variasi tempo yang berbeda seperti “bu” dan “ka” panjang, “bu” panjang dan “ka” pendek, “bu” pendek dan “ka” panjang, dan “bu” dan “ka” pendek. Kemudian pengujian juga dilakukan dengan mengucapkan kata kunci selain buka seperti duka, luka, suka, muka, bukan, dan pula.

Tabel II. Pengujian Pengenalan Kata Kunci

No	Kata Yang Diucapkan	Percobaan Membuka		Persentase	
		Terbuka	Tidak Terbuka	Terbuka	Tidak Terbuka
1	Buu-Kaa	3	2	60%	40%
2	Bu-Kaa	3	2	60%	40%
3	Buu-Ka	3	2	60%	40%
4	Bu-Ka	5	0	100%	0%
5	Du-Ka	1	4	20%	80%
6	Lu-Ka	0	5	0%	100%
7	Su-Ka	0	5	0%	100%
8	Mu-Ka	0	5	0%	100%
9	Bu-Kan	0	5	0%	100%
10	Pu-La	0	5	0%	100%

Pengujian pada proses pengenalan kata kunci dapat dilihat pada Tabel 2 dilakukan sebanyak 5 kali setiap kata kunci dengan hasil bahwa variasi pengucapan kata “bu” panjang dan “ka” pendek, kemudian pengucapan kata “bu” pendek dan “ka” panjang, pengucapan kata “bu” panjang dan “ka” panjang, dan variasi pengucapan “bu” pendek dan “ka” pendek dapat dikenali oleh sistem yang dimana saat menggunakan pengucapan kata “bu” pendek dan “ka” pendek mendapat persentase keberhasilan sebesar 100%. Sedangkan dengan kata yang diucap “bu” pendek “ka” panjang, “bu” panjang “ka” panjang, dan “bu” panjang “ka” pendek hanya mendapat nilai persentase keberhasilan 60%. Sedangkan kata kunci yang memiliki pola pengucapan yang hampir sama dengan kata “buka” yaitu duka menghasilkan persentase kondisi dimana kata kunci salah tetapi dinyatakan benar sebesar 20%. Untuk pengujian dengan kata luka, suka, muka, bukan, dan pula tidak terjadinya kondisi dimana kata kunci salah tetapi dinyatakan benar yang mengakibatkan *solenoid* tidak akan terbuka.

C. Pengujian Sistem Pengunci Pintu

Pengujian ketiga merupakan pengujian yang melibatkan sistem utama pengunci pintu *solenoid doorlock*. Sistem pengunci pintu ini bekerja pada saat sistem sudah mendeteksi kata kunci yang benar dan akan menampilkan pesan “sukses” pada LCD. Pada saat sistem berhasil menampilkan pesan “sukses” maka akan butuh waktu untuk sistem membuka sistem pengunci pintu. *Solenoid* akan terbuka saat layar LCD menampilkan “BU-KA 6s Berhasil”.

Tabel III. Pengujian Waktu Respon Sistem Membuka Kunci Selenoid

No	Percobaan	Status	Waktu (detik)
1	Pertama	Sukses	5,26
2	Kedua	Sukses	6,76
3	Ketiga	Sukses	6,76
4	Keempat	Sukses	5,9
5	Kelima	Sukses	6,94
6	Keenam	Sukses	6,61
7	Ketujuh	Sukses	6,5
8	Kedelapan	Sukses	6,93
9	Kesembilan	Sukses	6,4
10	Kesepuluh	Sukses	5,63
11	Kesebelas	Sukses	6,75
12	Keduabelas	Sukses	6,99

Tabel III. Pengujian Waktu Respon Sistem Membuka Kunci Selenoid (Lanjutan)

No	Percobaan	Status	Waktu (detik)
13	Ketigabelas	Sukses	6,27
14	Keempatbelas	Sukses	7,64
15	Kelimabelas	Sukses	6,72
Total			98,06
Rata-Rata			6,54
Waktu Respon Tercepat			5,26
Waktu Respon Terlambat			7,64

Setelah dilakukan pengujian, dihitung lama (*delay*) sistem dalam membuka sistem pengunci pintu *selenoid*, terlihat pada Tabel 3 dilakukan percobaan sebanyak 15 kali dengan tujuan menentukan waktu respon sistem setelah kata kunci terdeteksi. Hasil dari lima belas percobaan didapat waktu respon sistem untuk membuka *selenoid*. Hasil dari waktu respon tersebut terlihat pada percobaan pertama, waktu respon yang dibutuhkan selama 5,26 detik, sedangkan percobaan lainnya selain percobaan kedua dan kedelapan membutuhkan waktu lebih dari enam detik. Dari data tersebut didapatkan bahwa rata-rata waktu yang dibutuhkan sistem untuk membuka sistem pengunci *selenoid doorlock* adalah 6,54 detik.

D. Pengujian Algoritma Kalman Filter

Pengujian keempat adalah pengujian Algoritma Kalman Filter yang merupakan metode utama dalam penelitian untuk peredaman *noise*. Proses pengujian Algoritma Kalman Filter dilakukan dengan cara melakukan percobaan terhadap alat dengan cara mengucapkan kata kunci dengan *volume* pengucapan kurang lebih 55 dB hingga 65 dB dan jarak pengucapan antara INMP441 dengan mulut kurang lebih 15 cm. Pengujian ini menggunakan dua kode program dari ESP32 yang berbeda, yang dimana salah satunya menggunakan Algoritma Kalman Filter dan untuk salah satunya lagi tidak menggunakan Algoritma Kalman Filter. Selain itu pengujian akan dilakukan juga dengan tambahan variasi *noise* yang berbeda. Variasi *noise* tersebut dimulai dari suasana hening yang perlahan akan ditingkatkan hingga *noise* tingkat tinggi.

Tabel IV. Pengujian Dengan dan Tanpa Algoritma Kalman Filter

No	Jenis & Tingkat Noise	Kondisi Sistem Pengunci		Rata-Rata SNR (dB)	
		Tanpa Algoritma Kalman Filter	Dengan Algoritma Kalman Filter	Tanpa Algoritma Kalman Filter	Dengan Algoritma Kalman Filter
I	Hening (32 dB)	2 / 5	5 / 5	1,78	18,1
II	Kipas Angin (55 dB)	1 / 5	5 / 5	0,98	12,88
III	Bor Listrik (66 dB)	1 / 5	3 / 5	1,22	9,06
IV	Motor Listrik Kecil (77 dB)	0 / 5	5 / 5	0,86	8,28
V	Gerinda (89 dB)	0 / 5	3 / 5	0,66	3,54
Total SNR				5,50	51,86
Rata-Rata SNR				1,10	10,37
Tingkat Akurasi Sistem				16,00%	84,00%
Error / Kata Kunci Tidak Terdeteksi				84,00%	16,00%

Setelah melakukan pengujian, hasil pengujian dapat dilihat melalui Tabel 4. Pengujian dilakukan dengan cara mengucapkan kata kunci “buka” sebanyak 5 kali pada setiap pengujian terhadap *noise*. Pengujian dimulai dari *noise* lingkungan yang bersifat hening. Hening yang dimaksud adalah suasana ruangan saat adanya *noise* yang sangat rendah. Kemudian pengujian dilakukan secara bertahap untuk meningkatkan *noise* suara mulai dari kondisi hening, kemudian kondisi dinaikan menjadi *noise* kipas angin, setelah itu tingkat *noise* dinaikkan sedikit lebih tinggi menggunakan bor listrik, kemudian *noise* ditingkatkan menggunakan motor listrik kecil, *noise* ditingkatkan lagi menggunakan mesin gerinda. Pada pengujian Algoritma Kalman Filter, suara *noise* dari mesin gerinda merupakan tingkat *noise* yang tertinggi yang memunculkan nilai *error* saat mencoba mengucapkan kata kunci menggunakan Algoritma Kalman Filter. Dari hasil pengujian tersebut terdapat hasil bahwa dengan tidak menggunakan Algoritma Kalman Filter, diketahui tingkat akurasi sistem dalam mendeteksi kata kunci hanya bernilai 16% dan dengan tingkat *error*

sebesar 84%. Saat menggunakan Algoritma Kalman Filter nilai akurasi sistem dalam mendeteksi kata kunci meningkat yaitu menjadi 84% dan dengan nilai *error* menjadi lebih kecil yaitu sebesar 16%. Rata-rata SNR yang didapat ketika menggunakan Algoritma Kalman Filter sebesar 10,37 dB, sedangkan tanpa Algoritma Kalman Filter rata-rata SNR yang didapat sebesar 1,10 dB.

E. Pengujian Keseluruhan Sistem

Pengujian kelima merupakan pengujian keseluruhan sistem yang dimana pengujian ini akan mencakup seluruh sistem mulai dari perangkat lunak hingga ke perangkat keras. Untuk melakukan pengujian akan dimulai dengan menggunakan kata kunci yang benar dan *volume* suara yang tepat. Pengujian juga akan dilakukan menggunakan *noise* yang bervariasi. Kemudian dengan *noise* yang sama akan dilakukan pengujian jika kata kunci memiliki suku kata atau pengucapan yang hampir sama dengan kata “buka”. Pengujian keseluruhan sistem juga akan dilakukan tanpa menggunakan Algoritma Kalman Filter yang akan dilakukan dengan kata kunci benar dan kata kunci salah. Selain itu pengujian keseluruhan sistem juga akan dilakukan menggunakan variasi *noise* suara yang sama dengan pengujian keseluruhan sistem menggunakan Algoritma Kalman Filter.

Tabel V. Pengujian Keseluruhan Sistem Dengan dan Tanpa Algoritma Kalman Filter.

No	Jenis & Tingkat Noise	Tanpa Kalman Filter			Dengan Kalman Filter		
		Kata yang Diucapkan & Hasil	SNR	Nilai Kepercayaan	Kata yang Diucapkan & Hasil	SNR	Nilai Kepercayaan
1	Hening (32 dB)	Buka (5/5)	16,74	0,67	Buka (5/5)	14,32	0,65
		Suka (3/5)	3,90	0,50	Suka (1/5)	1,08	0,30
		Luka (2/5)	5,94	0,40	Luka (0/5)	0,72	0,22
		Kuda (3/5)	1,54	0,44	Kuda (0/5)	0,88	0,22
		Muda (5/5)	4,82	0,61	Muda (0/5)	0,86	0,22
2	Kipas Angin (55 dB)	Buka (2/5)	5,54	0,39	Buka (5/5)	28,02	0,72
		Suka (0/5)	0,52	0,22	Suka (0/5)	0,82	0,22
		Luka (1/5)	0,72	0,27	Luka (0/5)	1,30	0,25
		Kuda (2/5)	1,12	0,40	Kuda (0/5)	1,48	0,31
		Muda (2/5)	0,76	0,29	Muda (0/5)	1,18	0,26
3	Bor Listrik (66 dB)	Buka (1/5)	2,56	0,31	Buka (4/5)	9,56	0,62
		Suka (1/5)	3,78	0,34	Suka (0/5)	1,16	0,24
		Luka (1/5)	1,32	0,32	Luka (0/5)	1,24	0,22
		Kuda (1/5)	0,92	0,29	Kuda (0/5)	1,20	0,22
		Muda (1/5)	1,42	0,29	Muda (0/5)	1,04	0,22
4	Dinamo (77 dB)	Buka (1/5)	1,76	0,28	Buka (5/5)	8,70	0,64
		Suka (1/5)	0,60	0,23	Suka (1/5)	1,24	0,29
		Luka (2/5)	1,68	0,33	Luka (0/5)	0,78	0,22
		Kuda (1/5)	1,56	0,29	Kuda (0/5)	0,98	0,22
		Muda (0/5)	1,30	0,24	Muda (0/5)	1,06	0,22
5	Gerinda (89 dB)	Buka (0/5)	0,54	0,23	Buka (4/5)	3,64	0,52
		Suka (0/5)	0,60	0,23	Suka (0/5)	0,94	0,22
		Luka (0/5)	0,58	0,26	Luka (0/5)	0,92	0,22
		Kuda (0/5)	0,64	0,25	Kuda (0/5)	0,64	0,22
		Muda (0/5)	0,46	0,23	Muda (0/5)	0,72	0,22
Tingkat Akurasi		69,60%			83,20%		

Terlihat pada Tabel 5 merupakan hasil pengujian keseluruhan sistem yang dimana dilakukan sebanyak 5 kali pada setiap kata kunci yang diucapkan. Pada Tabel 5 merupakan hasil pengujian keseluruhan sistem tanpa dan dengan menggunakan Algoritma Kalman Filter. Pengujian dilakukan dengan jarak antara mulut ke INMP441 kurang lebih 15cm dan *volume* pengucapan kurang lebih 55 – 65 dB. Dalam pengujian keseluruhan sistem tanpa Algoritma Kalman Filter terlihat pada Tabel 5, memiliki tingkat akurasi secara keseluruhan lebih rendah daripada sistem yang menggunakan Algoritma Kalman Filter. Pada pengujian tanpa menggunakan Algoritma Kalman Filter menghasilkan nilai akurasi keseluruhan sebesar 69,6% yang dimana nilai tersebut sangat rendah sehingga mengakibatkan sistem mendapatkan data yang masih memiliki *noise* suara di dalamnya. Untuk sistem yang menggunakan Algoritma Kalman Filter, tingkat akurasi keseluruhan mencapai 83,2%. Nilai tersebut sangat tinggi dan dapat membuktikan bahwa ketika sistem menggunakan implementasi Algoritma Kalman Filter dalam peredaman *noise* suara dapat mendeteksi kata kunci lebih baik.

Dalam pengujian keseluruhan sistem, dapat diketahui pada Tabel 5 memiliki nilai rata-rata SNR terendah sebesar 0,46 dB. Nilai SNR tersebut memiliki perbedaan pada nilai SNR terendah yang dimana data tersebut memiliki nilai rata-rata SNR terendah sebesar 0,64 dB. Untuk nilai SNR tertinggi, sistem yang menggunakan Algoritma Kalman Filter memiliki rata-rata SNR tertinggi sebesar 28,2 dB. Sedangkan pada sistem yang tidak menggunakan Algoritma Kalman Filter memiliki rata-rata SNR sebesar 16,74 dB. Dari hasil pengujian tersebut didapatkan hasil bahwa sistem yang menggunakan Algoritma Kalman Filter memiliki kemampuan untuk mendeteksi kata kunci dengan lebih baik dikarenakan memiliki nilai SNR yang lebih tinggi daripada sistem yang tidak menggunakan Algoritma Kalman Filter.

Dalam penelitian tersebut terdapat nilai kepercayaan yang merupakan angka yang menunjukkan seberapa yakin bahwa suara yang masuk adalah suara yang mengandung kata kunci benar. Penentuan nilai kepercayaan ditetapkan secara empiris melalui serangkaian tahap uji coba sebagai langkah mengoptimasi kinerja sistem. Berdasarkan hasil observasi pada fase pra-penelitian, nilai minimum untuk kepercayaan merupakan titik temu optimal yang menyeimbangkan antara sensitivitas dan spesifitas algoritma pendeteksi kata kunci. Jika nilai kepercayaan <0,38 mengakibatkan sistem akan sangat sensitif sehingga kesalahan deteksi kata kunci akan lebih tinggi. Jika nilai kepercayaan >0,38 maka sistem akan sulit mengenali ucapan saat *volume* suara pengucapan diantara 45 dB hingga 65 dB.

V. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan Algoritma Kalman Filter untuk mengurangi *noise* suara membuat sistem dapat mengurangi *noise* melalui implementasi langsung ke ESP32 sehingga dapat mendeteksi kata kunci “buka” yang dilakukan pada *edge server* dengan lebih baik daripada sistem yang tidak menggunakan Algoritma Kalman Filter.

Penerapan Algoritma Kalman Filter dapat meningkatkan kinerja sistem pendeteksi kata kunci secara signifikan. Hal ini ditunjukkan oleh peningkatan akurasi keseluruhan dari yang sebelumnya 69,60% menjadi 83,20%.

Penelitian selanjutnya disarankan dapat menggunakan modul mikrofon yang lebih baik serta metode untuk mendeteksi kata kunci yang lebih baik. Selain itu disarankan juga untuk membuat mekanisme penyimpanan data atau membuat *database* hasil pengujian yang optimal. Kemudian pada penelitian berikutnya disarankan menyesuaikan tahapan dari pendeteksi kata kunci. Secara khusus saat implementasi analisis struktur suku kata dilakukan penyesuaian agar saat mengucapkan kata kunci tidak wajib ada jeda pengucapan.

VI. REFERENCE

- [1] S. Paturusi, M. M. I. Putra, dan S. R. U. A. Sompie, “Implementasi Speech Recognition pada Aplikasi Pembelajaran Bahasa Inggris untuk Anak,” *Jurnal Teknik Informatika*, vol. 15, no. 4, hlm. 247–256, Okt 2020.
- [2] S. Wirawan dan E. Prasetyo, “Implementasi Metode Noise Gate, Low Pass Filter dan Silent Removal untuk Menghilangkan Noise pada File Suara Menggunakan Parameter Dinamis,” *Jurnal Teknologi Rekayasa*, vol. 21, no. 3, hlm. 152–162, Des 2016.
- [3] E. T. Herdiani, S. A. Thamrin, Nuravia, dan Amran, “Aplikasi Kalman Filter pada Data Survival,” Jan 2010.
- [4] B. A. Tengger dan Ropiudin, “Pemanfaatan Metode Kalman Filter Diskrit untuk Menduga Suhu Udara,” *Square : Journal of Mathematics and Mathematics Education*, vol. 1, no. 2, hlm. 127–132, Des 2019, doi: 10.21580/square.2019.1.2.4202.

- [5] M. Iqbal, H. Walidainy, dan D. E. Elizar, “Analisis Filter Kalman untuk Menghapus Noise pada Sinyal Suara,” Aceh, 2019.
- [6] I. Novianto, L. Kurniasar, A. O. Pristisahida, B. K. Prasaja, dan Amanda, “Implementasi Filter Kalman Untuk Optimasi Pengukuran Sensor Suhu NTC pada Kompor Listrik Malam Berbasis Fuzzy,” Yogyakarta, Des 2022.
- [7] J. S. Fambudi, M. Syai’in, dan R. Y. Aditya, “Penerapan Kalman Filter Pada Pembacaan Sensor Loadcell Berbasis PLC Siemens S7-1200,” *Jurnal Elektronika dan Otomasi Industri*, vol. 11, no. 3, hlm. 700–707, Sep 2024, doi: 10.33795/elkolind.v11i3.6158.
- [8] R. Rohmatul Hidayah, S. Nurcahyo, D. Dewatama, P. N. Malang, dan P. Korespondensi, “Implementasi Pengaturan Suhu Menggunakan Mikrokontroler ESP32,” *Journal of Mechanical and Electrical Technology*, vol. 3, no. 3, 2024.
- [9] M. A. Shah, I. A. Shah, D. G. Lee, dan S. Hur, “Design approaches of MEMS microphones for enhanced performance,” 2019, *Hindawi Limited*. doi: 10.1155/2019/9294528.
- [10] A. Jufri, “Rancang Bangun dan Implementasi Kunci Pintu Elektronik Menggunakan Arduino dan Android,” Malang, 2016.
- [11] R. D. Risanty dan L. Arianto, “Rancang Bangun Sistem Pengendalian Listrik Ruangan dengan Menggunakan ATMEGA 328 dan Sms Gateway Sebagai Media Informasi,” 2017.
- [12] D. Sugiarto, M. Baehaqi, dan E. Subiyanta, “Rancang Bangun Sistem Absensi Menggunakan RFID Berbasis Web,” *MESTRO JURNAL*, vol. 4, no. 01, hlm. 25–31, Jun 2022.
- [13] F. A. Yaqin, A. F. Ibadillah, D. Rahmawati, dan K. A. Wibisono, “Perancangan Power Supply Switching Dengan Power Factor Correction (PFC) Untuk Mengoptimalkan Daya Output dan Pengaman Proteksi Hubung Singkat,” *Jurnal Arus Elektro Indonesia*, hlm. 42–50, 2021.
- [14] J. Firnando, B. Franko, S. Pratama Tanzil, N. Wilyanto, H. Christianto Tan, dan E. M. Hartati Kom, “Pembuatan Website Menggunakan Visual Studio Code di SMA Xaverius 3 Palembang,” Palembang, 2023.
- [15] S. P. Santoso dan F. Wijayanto, “Rancang Bangun Akses Pintu dengan Sensor Suhu Dan Handsanitizer Otomatis Berbasis Arduino,” Jan 2022.
- [16] A. Ma’arif, I. Iswanto, A. A. Nuryono, dan R. I. Alfian, “Kalman Filter for Noise Reducer on Sensor Readings,” *Signal and Image Processing Letters*, vol. 1, no. 2, hlm. 11–22, Jul 2019, doi: 10.31763/simple.v1i2.2.
- [17] E. S. A. Nugroho, N. D. Resty, dan I. Hudati, “Implementasi Filter Kalman Pada Sensor Jarak Berbasis Ultrasonik,” *Jurnal Listrik, Instrumentasi dan Elektronika Terapan*, vol. 2, no. 2, hlm. 20–24, Okt 2021.
- [18] A. D. Haq, I. Santoso, A. Ajulian, dan Z. Macrina, “Estimasi Signal To Noise Ratio (SNR) Menggunakan Metode Korelasi,” Semarang, Des 2012.