

# Mendeteksi Suara Menggunakan FPGA Sebagai Backend System dan MATLAB Sebagai Frontend System

1<sup>st</sup> Muhammad Ilham Jauhari

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

ilhamjauhari@student.telkomuniversity.ac.id

2<sup>nd</sup> Novi Prihatiningrum

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

nprihatiningrum@telkomuniversity.ac.id

3<sup>rd</sup> Rita Purnamasari

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

ritapurnamasari@telkomuniversity.ac.id

**Abstrak** — Penelitian ini bertujuan untuk mengembangkan sistem yang mampu mendeteksi audio menggunakan Field Programmable Gate Array (FPGA) sebagai sistem backend dan MATLAB sebagai sistem front-end. Sistem ini dirancang untuk mengatasi masalah pemrosesan dan analisis data suara secara real-time.

Menggunakan FPGA sebagai sistem tambahan menawarkan keunggulan pemrosesan data suara berkecepatan tinggi dan efisiensi energi yang baik. FPGA mampu menjalankan algoritme pemrosesan sinyal secara paralel, memungkinkan deteksi ucapan realtime dengan latensi rendah. Sementara itu, MATLAB digunakan sebagai sistem front-end untuk menyediakan antarmuka pengguna yang intuitif dan memungkinkan pemrosesan data yang lebih kompleks. .

Penelitian ini memiliki potensi aplikasi yang luas, termasuk dalam bidang pengawasan keamanan, pengolahan suara realtime, dan sistem interaksi manusia-komputer. Sistem yang dikembangkan dapat membantu meningkatkan keandalan dan efisiensi dalam mendeteksi suara, dengan menggunakan FPGA sebagai backend system dan MATLAB sebagai frontend system.

**Kata kunci** — MATLAB, FPGA, UART, Euclidean Distance, GPIO, K-Nearest Neighbor, KNN.

## I. PENDAHULUAN

Penggunaan FPGA pada penelitian penulis menjadi peran yang paling utama dalam program riset ini. FPGA dapat melakukan pemrosesan data dengan cara paralel. Proses paralel ini membuat FPGA sebagai alat proses yang bisa dikatakan paling baik untuk memproses sebuah data. Untuk FPGA yang penulis gunakan sendiri yaitu FPGA DE1, dimana FPGA tersebut dipabrikasi untuk kebutuhan pembelajaran universitas. Penulis Menggunakan MATLAB sebagai tempat penyimpanan data *trainer* dan sebagai pemanggilan data suara yang akan di uji coba nantinya.

Pengujian dilakukan dengan langkah-langkah yang sudah ada. Pengujian dilakukan didalam ruangan akustik yang jauh dari *noise*. MATLAB mengambil 10 suara sampel data untuk menjadi *data Trainer* yang akan dipanggil melalui matlab dan akan diproses oleh FPGA. Pengiriman data dari MATLAB ke FPGA menggunakan UART yang sudah tersedia di FPGA DE1, UART FPGA akan menerima data dari MATLAB dan *data trainer* akan di potong menjadi 1000

bagian yang akan di proses oleh FPGA. Selanjutnya akan dikeluarkan output melalui 7 *segment* yang tersedia di FPGA.

## II. KAJIAN TEORI

### A. FPGA

FPGA (Field-Programmable Gate Array) adalah jenis perangkat semikonduktor yang memungkinkan pengguna untuk merancang sirkuit digital yang dapat diprogram ulang setelah produksi. FPGA memiliki keunggulan dalam fleksibilitas dan rekonfigurasi yang membedakannya dari perangkat semikonduktor lainnya seperti ASIC (Application-Specific Integrated Circuit). Dalam kajian teori ini, kita akan membahas konsep dasar, arsitektur, dan aplikasi FPGA.

FPGA terdiri dari blok logika yang terhubung melalui jaringan sirkuit yang dapat diatur ulang. Setiap blok logika terdiri dari gerbang logika, flip-flop, dan elemen logika lainnya. FPGA juga memiliki sumber daya yang dapat diprogram ulang, seperti penghubung antarblok, pengaturan switch, dan pengaturan arsitektur.

Blok Logika merupakan unit dasar FPGA yang terdiri dari gerbang logika, LUT (Lookup Tables), flip-flop, dan elemen logika lainnya. Blok logika ini dapat diprogram ulang untuk mengimplementasikan fungsi logika yang diinginkan. Penghubung Antarblok merupakan jalur yang menghubungkan blok logika satu dengan yang lain. Penghubung antarblok dapat dikonfigurasi untuk membentuk jalur data yang diinginkan. Switch pada FPGA digunakan untuk mengatur penghubung antarblok dan mengonfigurasi jalur data yang menghubungkan blok logika.

”Kinerja CPU adalah 1/12 - 1/7 dari FPGA, yang mana berarti CPU dengan quad-core dapat mengeksekusi sekitar 1/10 operasi FPGA dalam satuan waktu (algoritma yang sama adalah dieksekusi pada CPU dan FPGA). Performa FPGA dibatasi oleh ukuran FPGA dan bandwidth memori. Dengan papan FPGA terbaru dengan DRAM DDR-II dan lebih besar FPGA, dimungkinkan untuk menggandakan kinerja dengan memproses dua kali jumlah piksel secara paralel.” (*Performance comparison of FPGA, GPU and CPU in image processing*, 2009)

FPGA dilengkapi dengan konfigurasi untuk mengatur jumlah dan jenis blok logika yang ada, ukuran penghubung antarblok, dan sumber daya lain yang tersedia. Desain FPGA dilakukan menggunakan bahasa deskripsi perangkat keras (HDL) seperti Verilog atau VHDL. HDL digunakan untuk menggambarkan perilaku sirkuit dan logika yang diinginkan. Setelah desain selesai, alat desain FPGA (FPGA design tools) digunakan untuk mengimplementasikan desain HDL ke dalam konfigurasi FPGA yang sesuai.

### III. METODE

Metode yang kami gunakan pada penelitian kami saat ini adalah menggunakan metode komunikasi data antar system. Mulai dari pengambilan suara yang diproses dalam FPGA. *Voice recognition* menggunakan 2 bahasa pemrograman dikarenakan, FPGA tidak bisa mengubah sinyal analog menjadi digital atau tidak memiliki ADC. Sehingga, waktu pengujian menjadi cukup lama. Sistem *voice recognition* yang telah dibangun seperti metode *K-Nearest Neighbour* (KNN). Dimana nantinya akan ada nilai yang sudah diset melalui FPGA untuk mendeteksi suara "Zero". Dimana, nilai ini akan dibandingkan dengan data yang diambil dari MATLAB menggunakan FPGA.

Desain sistem ini menggunakan 2 jenis *code* untuk melakukan akuisisi data suara manusia. *Code trainer* sebagai pelatihan sistem, dan *code recorder* sebagai pemanggilan data yang akan dikirimkan dan dibandingkan oleh FPGA melalui teknologi UART. Lalu, FPGA akan menampilkan keluaran melalui layar *7segment* dan akan memberikan sinyal ke sistem GPIO yang sudah tersedia pada FPGA DE1.

#### A. Singkatan dan Akronim

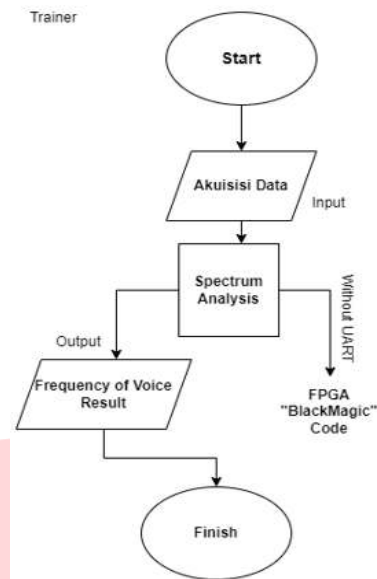
FPGA	: Field-Programmable Gate Array
GPIO	: General Purpose Input/Output
UART	: Universal Asynchronous Receiver Transmitter
RX/TX	: Receive/Transmit
KNN	: K-Nearest Neighbor

### IV. HASIL DAN PEMBAHASAN

Hasil dari keseluruhan sistem *voice recognition* ini menunjukkan bahwa FPGA mampu mendeteksi suara "zero" dengan tingkat akurasi 80% pada ruangan akustik, seperti pada Tabel 1. Sementara pada ruangan non-akustik, sistem ini hanya mampu mendeteksi sebesar 20% tingkat akurasi seperti pada Tabel 2. Percobaan dilakukan dengan pengambilan sampel data *trainer* dan data *recorder* dengan ruangan yang berbeda. Percobaan juga dilakukan sebanyak 5 kali disetiap ruangan yang berbeda.

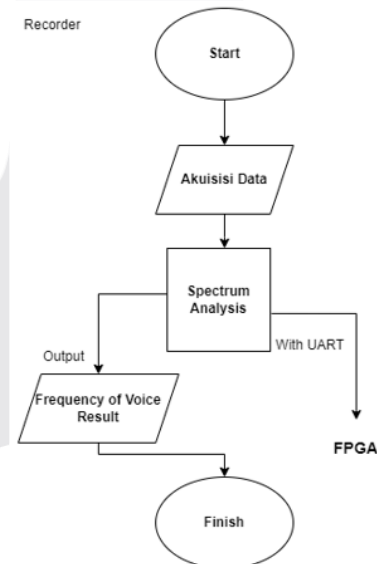
Hasil pada sistem MATLAB adalah berhasil menunjukkan hasil plot suara manusia yang sudah diambil sebelumnya pada *code trainer* dan *code recorder*. sistem ini juga berhasil melakukan serial data melalui UART. MATLAB mampu mengirimkan 1000 data dari suara yang sudah diambil sebelumnya. *Code injection* yang dilakukan berhasil dan memunculkan keluaran seperti pada Gambar 4. Hasil waktu untuk melakukan proses menjalankan code pada aplikasi Quartus II adalah sekitar 2 menit 30 detik. Hasil dari *code VHDL* yang sudah ada menghasilkan 17,445 *Logic element* dan menggunakan 47 Pin pada FPGA.

#### A. Gambar



GAMBAR 1  
(Flowchart trainer MATLAB)

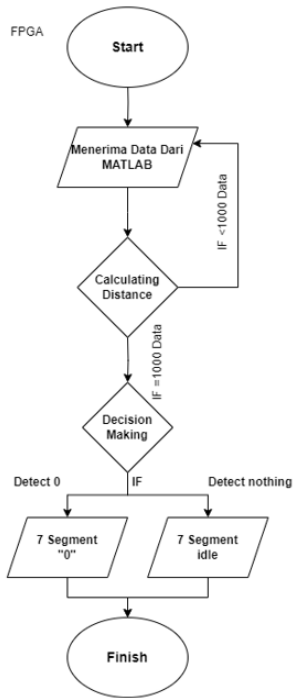
Pada *Flowchart source code trainer* terdapat penyimpanan data dalam *double precision array*, menggunakan *Fast Fourier Transformation* sebagai analisa spektrum, dan *plot voice file* dalam *frequency domain*. Selanjutnya sistem diatas akan menunjukkan plot yang memiliki 10 subplot, yang akan menunjukkan 10 data yang sudah diambil. Setelah, melakukan hal dilakukan maka, data yang ada pada "mean\_x" yang berada pada *workspace* MATLAB dimasukkan ke *code Blackmagic* pada FPGA.



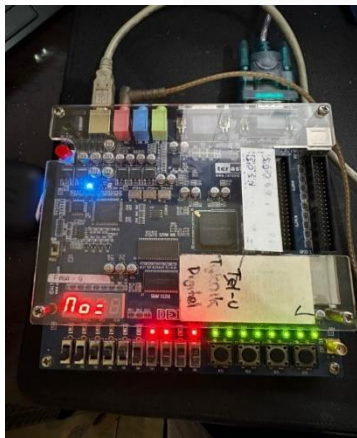
GAMBAR 2  
(Flowchart Recorder MATLAB)

Pada GAMBAR 2, yaitu gambar dari *Flowchart code recorder* memiliki kemiripan dengan *code trainer* yang ada pada GAMBAR 1. Tetapi yang membedakan hal ini adalah, hasil dari sistem yang dijalankan tidak dimasukkan secara manual. Tetapi sistem ini akan dimasukkan secara serial melalui teknologi UART dan satu hal yang membedakannya

adalah, sistem ini hanya membutuhkan satu kali pengambilan sampel suara. Dengan tujuan, sebagai masukan yang akan dibandingkan oleh FPGA.

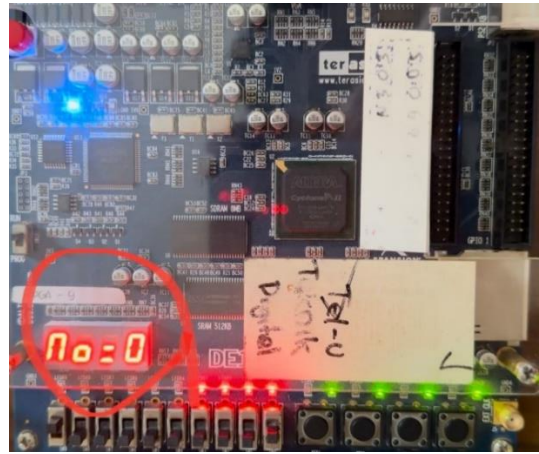


GAMBAR 3 (Flowchart Sistem FPGA)



GAMBAR 4 (FPGA setelah Code Injection)

Pada GAMBAR 4 menunjukkan bahwa, FPGA dalam idle mode. Idle mode adalah kondisi dimana, FPGA sudah berhasil melakukan *code injection* dari aplikasi QUARTUS II, dan kondisi ini menunjukkan bahwa, FPGA sudah siap digunakan.



GAMBAR 5 (FPGA setelah mendeteksi)

GAMBAR 5 menunjukkan FPGA yang sudah berhasil mendeteksi suara yang sudah diakuisisi. FPGA berhasil mengeluarkan keluaran yang berada pada layar 7 Segment yang tersedia. Selanjutnya, jika FPGA ingin digunakan kembali. Pengguna harus mematikan dan melakukan *code injection* kembali. Setelah *code injection* dilakukan, tampilan akan kembali seperti pada GAMBAR 4.

B. Tabel

TABEL 1 (Pengujian Pada Ruangan Akustik)

Pengujian	Alat yang digunakan	
	Waktu	Pengujian
1	3.51 Menit	Terdeteksi
2	4.01 Menit	Terdeteksi
3	3.48 Menit	Terdeteksi
4	3.42 Menit	Tidak Terdeteksi
5	3.55 Menit	Terdeteksi
Tingkat Akurasi = 80%		

TABEL 1 menunjukkan waktu pengujian dan tingkat keberhasilan sistem ini untuk mendeteksi angka yang akan disebutkan. Tabel pengujian ini menunjukkan bahwa tingkat akurasi sistem pada ruangan yang memiliki kadar *noise* rendah, memiliki akurasi yang cukup tinggi. Yaitu, berada pada angka 80% dari 5 kali percobaan.

TABEL 2 (Pengujian pada ruangan Non-akustik)

Pengujian	Alat yang digunakan	
	Waktu	Pengujian
1	3.52 Menit	Tidak Terdeteksi
2	4.05 Menit	Tidak Terdeteksi
3	3.59 Menit	Tidak Terdeteksi
4	3.44 Menit	Terdeteksi
5	3.56 Menit	Tidak Terdeteksi
Tingkat Akurasi = 20%		

TABEL 2 menunjukkan waktu pengujian dan tingkat keberhasilan sistem ini untuk mendeteksi angka yang akan disebutkan. Tabel pengujian ini menunjukkan bahwa tingkat akurasi sistem pada ruangan yang memiliki kadar *noise* tinggi, atau berada pada tempat yang memiliki kondisi *industrial noise*. Memiliki akurasi yang rendah, Yaitu berada pada angka 20% dalam 5 kali percobaan.

## V. KESIMPULAN

Hasil yang ditunjukkan adalah FPGA mampu mendeteksi suara "Zero" dengan tingkat akurasi 80% pada ruangan akustik. Proses pengujian sistem ini harus dilakukan diruangan akustik (*acoustic chamber*) agar tidak ada *noise* yang mengganggu frekuensi. MATLAB dan FPGA mampu melakukan komunikasi serial tanpa adanya hambatan dengan teknologi UART. FPGA juga mampu menampilkan hasil dari layar *7segment* dan mampu memberikan sinyal "high" untuk keluaran yang berada pada GPIO. Alat ini memiliki keterbatasan, jika berada dilokasi yang memiliki *industrial noise*.

Sistem ini memiliki tingkat kompleksitas yang cukup rumit. Hal ini dikarenakan, masih banyaknya *device* yang tersambung ke FPGA. Hal ini dipengaruhi juga dengan adanya keterbatasan alat yang digunakan. FPGA yang digunakan sudah cukup tua. Sehingga, banyak teknologi tambahan yang tidak tersedia didalam FPGA jenis DE1.

## REFERENSI

- [1] Dubey, R. Introduction to Embedded System Design Using Field Programmable Gate Arrays. London: Springer-Verlag, 2009, pp.
- [2] Rita Purnamasari. (2012, Nov.). "Desain Arsitektur dan Implementasi Teknik Pengkodean LDPC pada FPGA." *Telkom University Repository*. [Online].1, pp. 5-67. Available: <https://repository.telkomuniversity.ac.id> [Dec. 29, 2022].
- [3] Asano, Shuichi & Maruyama, Tsutomu & Yamaguchi, Yoshiki. (2009). Performance comparison of FPGA, GPU and CPU in image processing. FPL 09: 19th International Conference on Field Programmable Logic and Applications. 126-131. 10.1109/FPL.2009.5272532.
- [4] Asgar, M. R. G., Hidayat, R., & Bejo, A. (2023). Comparison Euclidean Distance and Manhattan Distance as Classification in Speech Recognition System. In *Proceedings of the International Conference on Educational Management and Technology (ICEMT 2022)* (pp. 454–463). Atlantis Press SARL. [https://doi.org/10.2991/978-2-494069-95-4\\_54](https://doi.org/10.2991/978-2-494069-95-4_54).
- [5] Jingjiao Li, Dong An\*, Lili Lang, Dan Yang., Embedded Speaker Recognition System Design and Implementation Based on FPGA, School of Information Science & Engineering, Northeastern University, Shenyang 110004, China.