

PEMBANGKIT FREKUENSI UNTUK KESEHATAN

Disusun untuk memenuhi tugas akhir mata kuliah Pemrograman FPGA



Oleh:

- 1. Maulana Ihza Ishlahy** (225150300111012)
- 2. Maharddhika Nusapati** (225150300111013)
- 3. Daffa Raihan Dwi Ari Putra** (225150300111019)
- 4. Rihan Hidayat** (225150301111009)
- 5. Ahmad Fajar Maulana** (225150307111016)
- 6. Muhammad Fathi Radithya** (225150307111024)

Dosen Pengampu:

Dahnial Syauqy, S.T, M.T.

**PROGRAM STUDI S1 TEKNIK KOMPUTER
DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2023**

DAFTAR ISI

HALAMAN JUDUL	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	1
1.4 Manfaat	1
BAB II TINJAUAN PUSTAKA	2
BAB III METODOLOGI DAN PERANCANGAN	3
3.1 Spesifikasi sistem dan prinsip kerja	3
3.2 Blok diagram	3
3.3 Perancangan sistem	4
BAB IV PEMBAHASAN	5
4.1 <i>Flowchart</i> pemrograman	5
4.2 Implementasi dan Hasil uji	5
BAB V PENUTUP	7
5.1 Kesimpulan	7
5.2 Saran	7
Daftar Pustaka	8
Lampiran 1: Source code dan file UCF	9

BAB I

PENDAHULUAN

1.1 Latar Belakang

Frekuensi adalah kuantitas dan jumlah pada getaran yang mampu untuk menghasilkan dalam interval waktu satu detik (Mawardi, 2022). Frekuensi dalam konteks ini diketahui memiliki efek positif pada tubuh manusia, seperti meredakan stres dan meningkatkan kesejahteraan secara keseluruhan. Penerapan teknologi untuk menciptakan sistem pembangkit frekuensi yang dapat digunakan untuk tujuan kesehatan menjadi relevan dan memiliki potensi besar untuk memberikan kontribusi pada domain kesehatan.

Saat ini, banyak terapi dan pendekatan kesehatan yang melibatkan penggunaan frekuensi tertentu. Namun, kebanyakan dari mereka masih bergantung pada metode konvensional yang mungkin tidak optimal dalam hal fleksibilitas dan personalisasi. Oleh karena itu, diperlukan suatu solusi yang mampu menghasilkan frekuensi kesehatan secara lebih spesifik dan dapat disesuaikan.

Sistem-sistem konvensional yang ada mungkin belum sepenuhnya memenuhi kebutuhan ini, dan oleh karena itu, pengembangan sistem berbasis FPGA menjadi solusi yang menarik. FPGA memungkinkan pengguna untuk merancang sirkuit digital sesuai dengan kebutuhan spesifik mereka. Keunggulan FPGA terletak pada kemampuannya untuk memproses sinyal secara real-time, kemudahan dalam mengimplementasikan algoritma khusus, dan fleksibilitas dalam menyesuaikan dengan perubahan kebutuhan.

Dengan memanfaatkan FPGA, sistem ini dapat diimplementasikan secara efisien untuk menghasilkan frekuensi yang diinginkan, sementara tetap memberikan ruang untuk penyesuaian lebih lanjut. Desain sistem berbasis FPGA memiliki potensi untuk menjadi alternatif yang unggul dalam memberikan solusi kesehatan yang lebih personal dan efektif dibandingkan dengan pendekatan konvensional yang ada saat ini. Oleh karena itu, proyek ini dipilih untuk menggali potensi FPGA dalam meningkatkan kesehatan melalui pemanfaatan frekuensi tertentu.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah:

1. Apa saja sistem yang diperlukan untuk menjaga kesehatan?
2. Bagaimana merancang dan mengimplementasikan sistem pembangkit frekuensi untuk tujuan kesehatan menggunakan FPGA?
3. Bagaimana kinerja sistem dalam menghasilkan frekuensi yang diinginkan?

1.3 Tujuan

Tujuan dari tugas akhir ini adalah:

1. Mengimplementasikan sistem dengan menyediakan alternatif yang lebih fleksibel dan personal dalam penghasilan frekuensi kesehatan dibandingkan dengan metode konvensional yang ada.
2. Merancang sistem pembangkit frekuensi menggunakan FPGA.
3. Menguji kinerja sistem dalam menghasilkan frekuensi dengan dampak positif pada kesehatan.

1.4 Manfaat

Manfaat dari proyek ini mencakup sejumlah aspek yang dapat memberikan dampak positif pada berbagai pihak, termasuk mahasiswa, masyarakat, dan pihak-pihak terkait. Adapun manfaatnya adalah sebagai berikut:

1. Kontribusi Teknologi Kesehatan: Pengembangan sistem frekuensi berbasis FPGA meningkatkan inovasi dalam peningkatan kesehatan masyarakat.
2. Solusi Personalisasi Kesehatan: Sistem dapat disesuaikan secara personal untuk menghasilkan frekuensi sesuai kebutuhan, memberikan solusi yang lebih terarah.
3. Dukungan pada Terapi Frekuensi: Menjadi solusi alternatif untuk terapi frekuensi dengan potensi meningkatkan kesejahteraan dan mengelola stres.
4. Potensi Implementasi Layanan Kesehatan: Dapat diimplementasikan dalam layanan kesehatan, memberikan solusi efektif bagi pasien dan penyedia layanan kesehatan.

BAB II

TINJAUAN PUSTAKA

2.1. Frekuensi dalam Terapi Kesehatan

Menurut Budi Setyawan, tubuh manusia mengandung unsur molekul senyawa yang memiliki frekuensi tersendiri. Gangguan atau ketidakharmonisan dalam frekuensi ini dapat menyebabkan penyakit. Dalam upaya penyembuhan, metode elektro therapy digunakan tanpa memasukkan sesuatu secara fisik ke dalam tubuh. Sebaliknya, metode ini menggunakan resonansi frekuensi suara untuk merestorasi harmoni frekuensi molekul dalam tubuh.

Metode ini tidak baru, karena para orang tua zaman dahulu sudah mengenalnya sebagai kekuatan "prana". Namun, kemajuan teknologi memungkinkan pendekatan ini dapat dibuktikan secara ilmiah. Kerjasama dengan PREAN melibatkan penggunaan frekuensi suara dengan gelombang tertentu untuk melemahkan atau membunuh bakteri dalam tubuh, meningkatkan imun, dan mengurangi stres.

Cara kerjanya sederhana, pasien didengarkan frekuensi dengan gelombang tertentu selama 45-60 menit, dua kali sehari. Meskipun suaranya di telinga mungkin terdengar seperti "kemresek" atau mendesis, inovasi juga diterapkan dengan menyisipkan unsur musik harmonisasi untuk memastikan kenyamanan pendengaran.

2.2. Sistem Pembangkit Frekuensi

Menggunakan metode yang digunakan oleh Krosenfeld7 pada petunjuk "FPGA Composer" kita dapat membangkitkan frekuensi dengan board FPGA menggunakan sinyal clock internal board FPGA. Perlu diketahui bahwa pada petunjuk tersebut Krosenfeld7 menggunakan board FPGA Nexys sedangkan kita menggunakan board FPGA Xilinx Spartan 3E namun hal tersebut tidak menjadi masalah karena metode yang dipakai dapat digunakan kedua Board FPGA tersebut.

Dalam petunjuk Krosenfeld7 menggunakan counter yang terus bertambah hingga kita mendapatkan sebuah nilai threshold dimana variabel “temp_out” dialihkan setiap kali nilai tersebut tercapai. Hal ini akan menghasilkan frekuensi sinyal yang lebih lambat daripada sinyal clock 50MHz untuk merepresentasikan frekuensi yang kita inginkan. Untuk mendapatkan nilai threshhold tersebut kita dapat membagi sinyal clock 50 MHz dengan frekuensi yang kita inginkan.

Setelah itu kita harus melihat apakah switch untuk frekuensi yang kita inginkan menyala atau tidak. Menggunakan varabel untuk input misalkan “note” seperti pada petunjuk kita dapat memeriksa switch mana yang sedang menyala.

Terkhir kita memastikan bahwa frekuensi dapat dibangkitkan pada saat clock rising edge. Hal ini dilakukan dengan menggunakan if statement untuk melihat apakah clock sedang rising edge.

2.3. Metode Konvensional dalam Terapi Frekuensi

Terapi frekuensi adalah pengobatan alternatif yang memanfaatkan gelombang elektromagnetik untuk menstimulasi sel-sel tubuh. Metode konvensional dalam terapi frekuensi memiliki beberapa macam seperti terapi gelombang radio dan terapi laser

Terapi Gelombang Radio merupakan metode yang menggunakan gelombang radio berfrekuensi rendah untuk mempercepat penyembuhan jaringan lunak seperti ligamen, tendon, dan otot. Metode ini biasa digunakan untuk penyembuhan cedera olahraga.

Terapi Laser merupakan metode yang menggunakan cahaya laser untuk merangsang produksi ATP dan sintesis protein di dalam sel tubuh. Metode ini sering digunakan untuk mengurangi nyeri, peradangan, dan mempercepat penyembuhan luka. Stimulasi Listrik merupakan metode yang menerapkan arus listrik kecil dan terkontrol ke area tubuh tertentu untuk merangsang saraf dan otot. Metode ini biasa digunakan untuk mengurangi nyeri dan meningkatkan pergerakan sendi atau otot.

BAB III

METODOLOGI DAN PERANCANGAN

3.1 Spesifikasi sistem dan prinsip kerja

Dalam projek ini kami menggunakan board FPGA Xilinx Spartan 3E sebagai pembangkit frekuensi dan speaker kecil sebagai alat output kami. Board FPGA Xilinx Spartan 3E membangkitkan frekuensi dengan menggunakan signal *clock* 50 Mhz pada board FPGA Xilinx Spartan 3E.

Kami menggunakan metode yang sama seperti Krosenfeld7 pada petunjuknya “*FPGA Composer*” untuk membangkit frekuensi kesehatan. Kami menggunakan *counter* yang terus bertambah hingga mencapai nilai *threshold* tertentu setelah itu mengalihkan “*temp_out*” untuk menghasilkan frekuensi yang kita inginkan. Frekuensi yang kita pilih adalah frekuensi bila susah tidur (3Hz), frekuensi bila kurang stamina (417Hz), frekuensi bila stres (432 Hz) dan frekuensi bila kurang enak badan (528 Hz). Karena frekuensi bila susah tidur hanyalah 3 Hz sehingga susah di dengan melalui speaker, kami menggunakan LED untuk menandai bahwa sebuah switch sedang menyala.

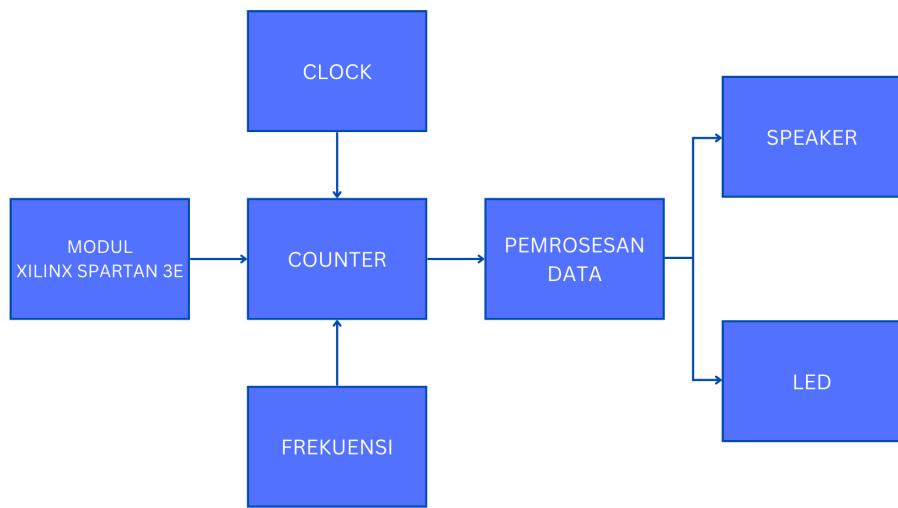
Untuk mendapatkan nilai *threshold* kami bagi sinyal *clock* 50MHz dengan frekuensi tersebut. Hasil dari pembagian tersebut kita bulatkan untuk ditaruh pada program menjadi :

- Nilai *threshold* frekuensi bila susah tidur = 16666667
- Nilai *threshold* frekuensi bila kurang stamina = 199904
- Nilai *threshold* frekuensi bila stres = 155741
- Nilai *threshold* frekuensi bila kurang enak badan = 94697

Pada saat *counter* mencapai nilai-nilai tersebut maka “*temp_out*” akan dialihkan sehingga frekuensi dikeluarkan pada speaker.

3.2 Blok diagram

Berikut merupakan gambaran dari *block diagram* dalam program yang telah kami buat untuk projek ini.



Gambar 3.1. Blok diagram sistem

3.3 Perancangan sistem

Pada projek ini kita menggunakan keempat switch yang terdapat pada board FPGA Spartan 3E sebagai input. Untuk switch menggunakan port L13, L14, H18, N17. Tentu karena kita menggunakan switch pulltype diatur menjadi pullup.

Selain itu kita juga menggunakan internal clock pada board FPGA sebagai input dengan menggunakan port C9 yang merupakan salah satu port internal clock pada board FPGA Spartan 3E. Untuk output kita menggunakan speaker dengan menggunakan port B4 untuk menyalurkan frekuensi yang dihasilkan oleh program ke speaker.

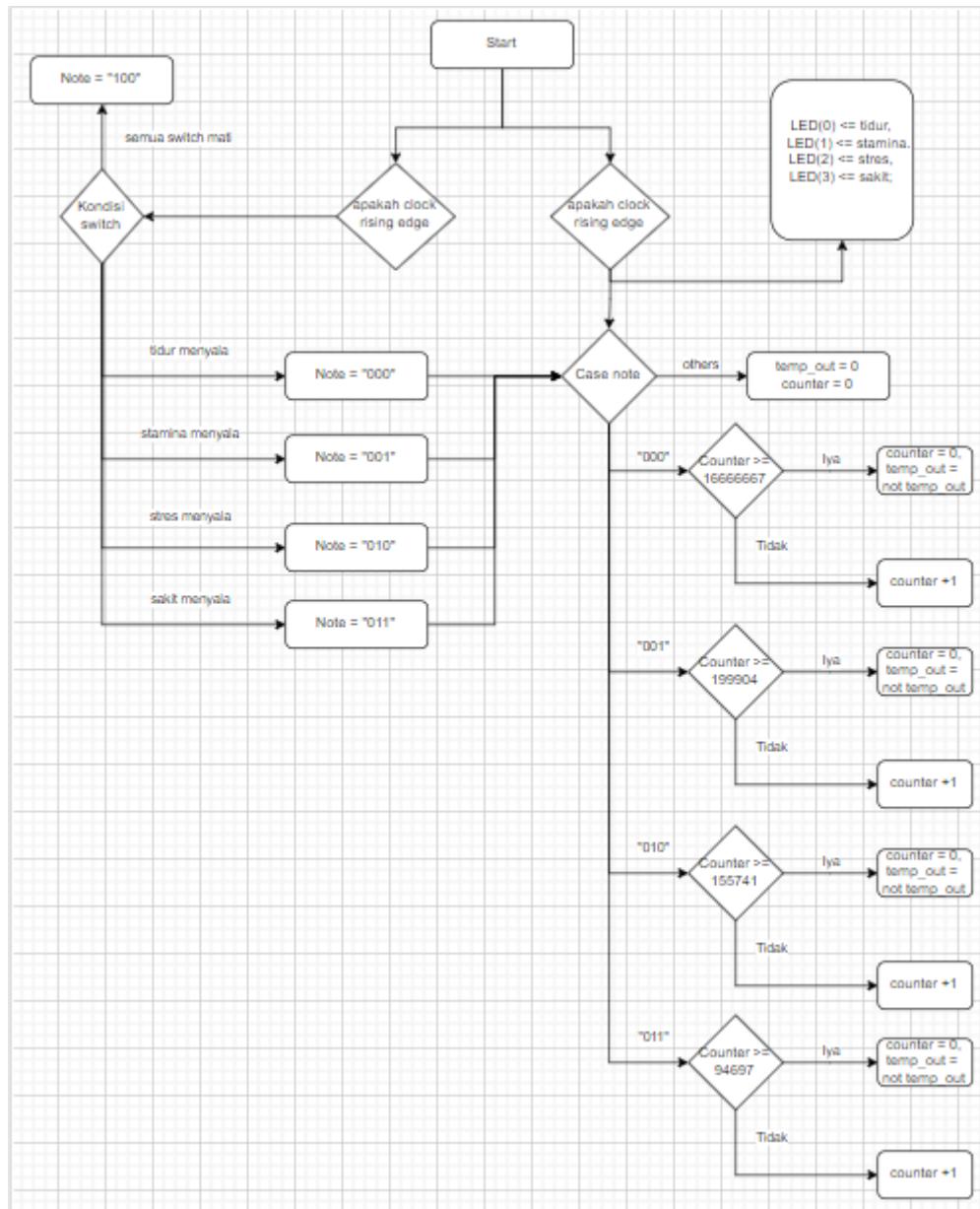
Untuk drive strength sendiri semua memakai drive strength yang sama yaitu 12 karena tidak ada spesifikasi khusus untuk drive strength dalam projek ini.

BAB IV

PEMBAHASAN

4.1 Flowchart pemrograman

Berikut merupakan gambaran dari *flowchart* dalam program yang telah kami buat untuk projek ini.



Gambar 4.1. Flowchart sistem

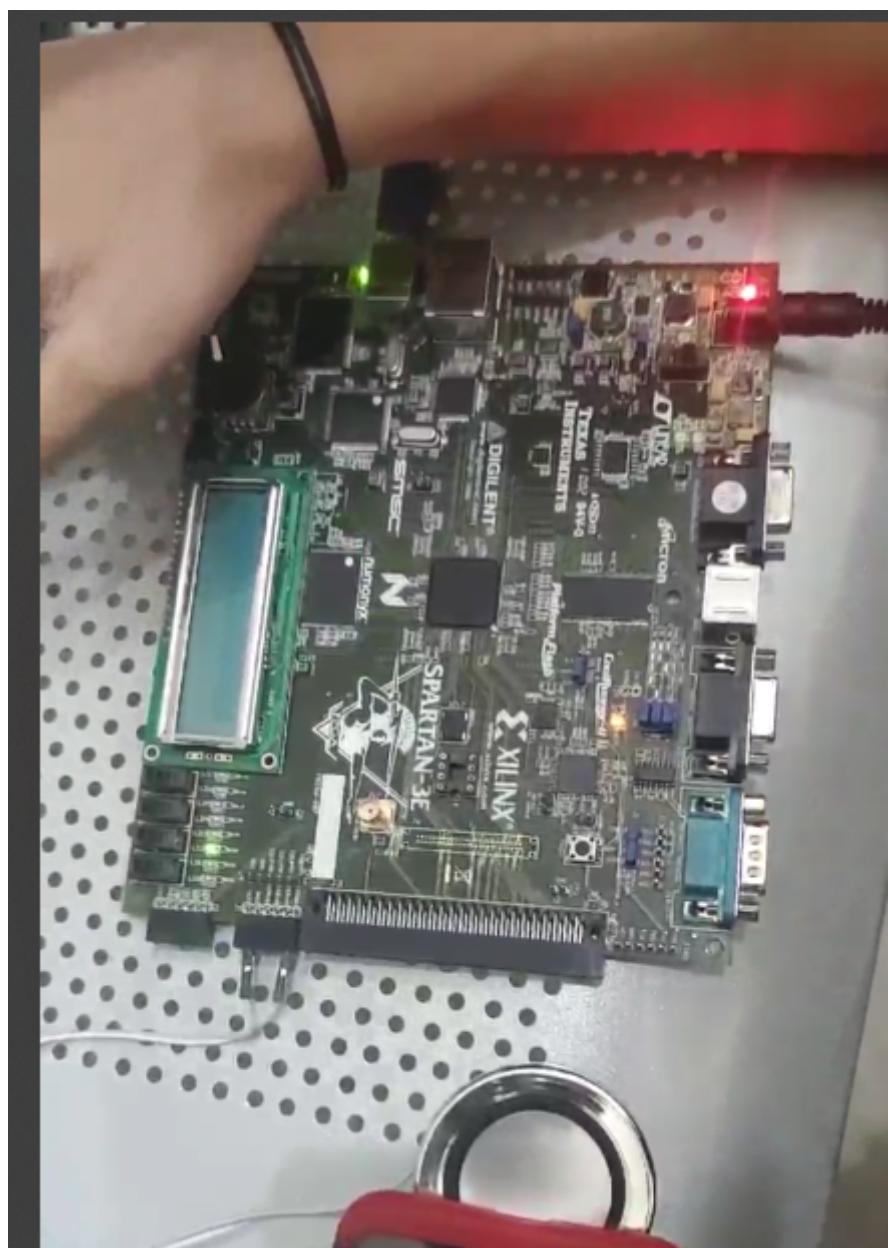
Program ini menggunakan 2 proses. Proses pertama merupakan case dengan variabel ‘note’. Bila note sama dengan nilai tertentu maka akan dilihat apakah counter lebih besar sama dengan suatu nilai. Nilai tersebut merupakan pembagian nilai frekuensi internal clock pada Spartan 3E yaitu 50MHz dengan frekuensi yang diinginkan. Misalkan pada case “note” sama dengan “000” maka dilihat apakah “counter” lebih besar dari “16666667” yang merupakan hasil pembagian dari 50MHz dengan 3Hz. bila benar lebih besar atau sama dengan nilai tersebut maka menjadikan nilai “counter” dijadikan 0 dan temp_out dijadikan kebalikan dari nilainya pada saat itu. Misal nilai temp_out adalah 0 maka akan menjadi 1. Bila note nilainya diluar “000”, “001”, “010”, dan “011” maka “temp_out” dijadikan 0 dan “counter” dijadikan 0. temp_out mengontrol apakah terdapat output atau tidak bila sama dengan 1 maka terdapat output bila 0 maka tidak ada output. Selain itu terdapat juga pengaturan agar LED menyala saat switch tertentu diaktifkan dengan memasukan nilai pada input switch masing-masing.

Proses kedua melihat apakah ada switch yang menyala. Misalkan switch “tidur” menyala maka “note” sama dengan “000”. Bila semua switch tidak menyala maka note menjadi “100”. Bila dilihat tidak ada case untuk “note” sama dengan “100” membuat tidak terdapat output suara.

4.2 Implementasi dan Hasil uji

Untuk implementasi kita menggunakan speaker sebagai output eksternal. Untuk speaker sendiri disambungkan pada Ground dan port B4. Selain itu untuk switch kita menggunakan switch 1-4 dan menamainya “tidur”, “stamina”, “stres”, dan “sakit”. Kita juga memakai LED sebagai penanda bahwa frekuensi sedang berjalan karena terdapat 1 frekuensi yang bila didengar dengan speaker tidak terlalu terdengar.

Setelah diuji board FPGA berhasil menghasilkan frekuensi yang diinginkan melalui speaker dan LED menyala sesuai switch yang menyala.



Gambar 4.2. Bentuk fisik sistem

BAB V

PENUTUP

5.1 Kesimpulan

Dalam penutup proyek ini, dapat disimpulkan bahwa pengembangan sistem pembangkit frekuensi berbasis FPGA memiliki potensi besar dalam meningkatkan kesehatan melalui aplikasi frekuensi tertentu. Implementasi elektro therapy dengan memanfaatkan resonansi frekuensi suara menunjukkan hasil positif dalam merestorasi harmoni frekuensi molekul dalam tubuh. Proyek ini mengusung pendekatan inovatif yang dapat menjadi alternatif efektif dalam terapi kesehatan.

5.2 Saran

Sebagai saran pengembangan lanjut, proyek ini dapat ditingkatkan melalui beberapa langkah berikut:

1. Perluasan Penelitian:

Melakukan penelitian lebih lanjut untuk memperdalam pemahaman tentang jenis frekuensi tertentu yang dapat memberikan manfaat spesifik pada kondisi kesehatan tertentu.

2. Validasi Klinis:

Melibatkan uji klinis lebih lanjut untuk mengonfirmasi efektivitas dan keamanan penggunaan elektro therapy berbasis frekuensi dalam konteks kesehatan manusia.

3. Pengoptimalan Sistem:

Melakukan pengoptimalan pada desain sistem FPGA untuk meningkatkan efisiensi dan respons sistem dalam menghasilkan frekuensi yang diinginkan.

4. Pengembangan Antarmuka Pengguna:

Menambahkan antarmuka pengguna yang lebih intuitif dan informatif untuk memudahkan penggunaan sistem oleh tenaga medis atau individu yang membutuhkan terapi frekuensi.

5. Penelitian Lebih Lanjut tentang Pengaruh Frekuensi pada Virus dan Bakteri:

Menggali lebih dalam dampak frekuensi pada tingkat mikroorganisme, khususnya dalam konteks menangani virus dan bakteri yang mungkin relevan, seperti penelitian pada pengurangan stres.

Dengan menggali saran-saran ini, diharapkan proyek ini dapat terus berkembang dan memberikan kontribusi lebih besar dalam bidang kesehatan.

DAFTAR PUSTAKA

- Mawardi, Rafi Aufa. “*Pengertian Frekuensi Beserta Jenisnya, Lengkap dengan Contoh Soal*”. [online] (<https://www.detik.com/edu/detikpedia/d-6131532/pengertian-frekuensi-beserta-jenisnya-lengkap-dengan-contoh-soal>) diakses pada 6 Desember 2023.
- Hariyanto, Didik. “*analog to digital conversion*”. [online] (<https://staffnew.uny.ac.id/upload/132304810/pendidikan/Teknik+Antarmuka+-+ADC.pdf>) diakses pada 12 Desember 2023.
- Resonansi Frekuensi Suara untuk Tingkatkan Kesehatan Tubuh. (8 Juni, 2020). unika.ac.id (<https://www.unika.ac.id/news/media-massa/online/resonansi-frekuensi-suara-untuk-tingkatkan-kesehatan-tubuh/>) diakses pada 6 Desember 2023.
- Braz J. “*Studi Audiometri Frekuensi Tinggi pada Pasien Sembuh Kanker yang Diobati dengan Cisplatin*” [online] (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9442129/>) diakses pada 6 Desember 2023.
- Kosenfeld7. “*FPGA Composer*”. [online] (<https://www.instructables.com/FPGA-Composer/>) diakses pada 7 Desember 2023

LAMPIRAN I: SOURCE CODE dan FILE UCF

A. Source Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
```

```

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity main is
    Port ( tidur : in STD_LOGIC;
           stres : in STD_LOGIC;
           sakit : in STD_LOGIC;
           stamina : in STD_LOGIC;
           clk : in STD_LOGIC;
           LED : out STD_LOGIC_VECTOR (3 downto 0);
           audioout : out STD_LOGIC);
end main;

architecture Behavioral of main is
signal counter : integer := 0;
signal temp_out : STD_LOGIC := '0';
signal note : STD_LOGIC_VECTOR(2 downto 0);

begin
    process(clk, note)
    begin
        if rising_edge(clk) then
            case note is
                when "000" => -- tidur
                    if (counter >= 16666667) then
                        counter <= 0;
                        temp_out <= not temp_out;
                    else
                        counter <= counter + 1;
                    end if;
                when "001" => -- stamina
                    if (counter >= 199904) then
                        counter <= 0;
                        temp_out <= not temp_out;
                    else
                        counter <= counter + 1;
                    end if;
                when "010" => -- stres
                    if (counter >= 155741) then
                        counter <= 0;
                        temp_out <= not temp_out;
                    else
                        counter <= counter + 1;
                    end if;
                when "011" => -- sakit
                    if (counter >= 94697) then
                        counter <= 0;
                        temp_out <= not temp_out;
                    else
                        counter <= counter + 1;
                    end if;
                when others =>

```

```

        temp_out <= '0';
        counter <= 0;
    end case;

        LED(0) <= tidur;
        LED(1) <= stamina;
        LED(2) <= stres;
        LED(3) <= sakit;
    end if;
end process;

audioout <= temp_out;

process(clk, tidur, stres, sakit, stamina)
begin
    if rising_edge(clk) then
        if (tidur = '0' and stamina= '0' and stres = '0' and sakit
= '0') then
            note <= "100";
        else
            if tidur = '1' then
                note <= "000";
            elsif stamina = '1' then
                note <= "001";
            elsif stres = '1' then
                note <= "010";
            elsif sakit = '1' then
                note <= "011";
            end if;
        end if;
    end if;
end process;

end Behavioral;

```

B. UCF code

```

NET "audioout" LOC = B4;
NET "sakit" LOC = N17;
NET "stamina" LOC = L14;
NET "stres" LOC = H18;

NET "clk" LOC = C9;

NET "tidur" LOC = L13;

NET "sakit" PULLUP;
NET "stamina" PULLUP;
NET "stres" PULLUP;
NET "tidur" PULLUP;

NET "audioout" DRIVE = 12;
NET "clk" DRIVE = 12;

```

```
NET "sakit" DRIVE = 12;
```

```
NET "LED[0]" LOC = F12;  
NET "LED[1]" LOC = E12;  
NET "LED[2]" LOC = E11;
```

```
NET "LED[3]" LOC = F11;
```