

# IMPLEMENTASI FM TRANSMITTER DI FPGA UNTUK RADIO KOMUNITAS FAKULTAS ILMU TERAPAN

## *FM TRANSMITTER IMPLEMENTATION IN FPGA FOR COMMUNITY RADIO SCHOOL OF APPLIED SCIENCE*

Sri Kalbu Balilayaran<sup>1</sup>, Denny Darlis S.Si., M.T.<sup>2</sup>, Dwi Andi Nurmantris, S.T., M.T.<sup>3</sup>

Program Studi D3 Teknologi Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom  
Jl. Telekomunikasi No.1 Dayeuhkolot Bandung 40257 Indonesia

<sup>1</sup>[srikalbulalayaran@gmail.com](mailto:srikalbulalayaran@gmail.com), <sup>2</sup>[Denny.darlis@tass.telkomuniversity.ac.id](mailto:Denny.darlis@tass.telkomuniversity.ac.id),

<sup>3</sup>[dwiandi@tass.telkomuniversity.ac.id](mailto:dwiandi@tass.telkomuniversity.ac.id)

### Abstrak

Radio FM adalah sistem komunikasi radio yang menggunakan Frekuensi Modulasi (FM). Radio FM memiliki keunggulan seperti kualitas suara yang lebih bagus, jernih, dan bebas dari gangguan siaran dibandingkan dengan radio AM. Proyek akhir ini bermaksud untuk membuat sistem pemancar radio FM, pembuatan sistem pemancar radio FM ini diharapkan dapat dipakai untuk penyiaran di lingkungan Fakultas Ilmu Terapan Telkom University, karena nantinya sistem pemancar FM ini akan diatur untuk menyiarkan informasi pada frekuensi yang telah diatur untuk radio komunitas. Pemancar FM yang dibuat adalah sistem pemancar FM digital, sistem pemancar FM ini memiliki beberapa keunggulan dibanding dengan sistem pemancar FM analog. Sistem pemancar FM digital ini akan dibuat menggunakan board FPGA DE-10 Lite keluaran Intel. Dari proses perancangan, FM transmitter digital yang dibuat sudah sesuai dengan yang diharapkan, yaitu menggunakan FPGA dengan type board DE-10 Lite keluaran Intel dengan perancangan menggunakan bahasa Verilog. FM transmitter ini sudah dapat bekerja pada frekuensi yang diatur untuk radio komunitas yaitu 107,7MHz-107,9MHz, dan pengiriman data audionya menggunakan baudrate 230400bit/second dengan 8 bit data, 2 stop bit dan tidak menggunakan bit parity.

**Kata Kunci : FPGA, Sistem Pemancar FM Digital, Radio Komunitas**

### Abstract

FM radio is a radio communication system that uses Frequency Modulation (FM). FM radio has advantages such as better sound quality, clarity, and free of broadcast interference compared to AM radio. This final project intends to create an FM radio transmitter system, the creation of this FM radio transmitter system is expected to be used for broadcasting in the environment of Telkom University Faculty of Applied Sciences, because later this FM transmitter system will be set up to broadcast information at frequencies that have been set up for community radio. The FM transmitter is a digital FM transmitter system, this FM transmitter system has several advantages compared to analog FM transmitter systems. This digital FM transmitter system will be built using Intel's FPGA DE-10 Lite board. From the design process, FM transmitter digital made is in accordance with the expected, namely using FPGA with type board DE-10 Lite output Intel with design using Verilog language. This FM transmitter can already work on frequencies set for community radios of 107.7MHz-107.9MHz, and its audio data delivery uses a 230400bit/second baudrate with 8 bits of data, 2 stop bits and does not use parity bits.

**Keywords: FPGA, FM Digital transmitter system, Community Radio**

## 1. PENDAHULUAN

FM (*Frequency Modulation*) atau Modulasi Frekuensi adalah proses penumpangan sinyal informasi pada sinyal pembawa (*carrier*) sehingga frekuensi gelombang pembawa (*carrier*) berubah sesuai dengan perubahan tegangan (simpangan) sinyal informasi [5]. Pada Modulasi Frekuensi ini, sinyal informasi akan mengubah frekuensi gelombang pembawanya sedangkan amplitudonya tetap selama proses modulasi [6]. Salah satu implementasi dari teknik modulasi frekuensi adalah Radio FM. Radio FM adalah sistem komunikasi radio yang menggunakan Frekuensi Modulasi (FM). Radio FM ini ditemukan oleh Edwin Howard Armstrong pada tahun 1930-an setelah adanya sistem jaringan. Radio FM memiliki keunggulan seperti kualitas suara yang lebih bagus, jernih, dan bebas dari gangguan siaran. Radio broadcasting adalah salah satu sistem komunikasi, pada sistem komunikasi baik analog maupun digital, harus memiliki pemancar (TX) dan penerima (RX), begitu pula pada Radio FM. Sistem pemancar FM berawal dari sistem pemancar FM analog. Sistem pemancar ini menggunakan *Phase Locked Loop* (PLL). PLL adalah suatu sistem umpan balik dimana sinyal umpan balik digunakan untuk

mengunci frekuensi dan fasa keluaran pada suatu frekuensi serta fasa sinyal masukan [1]. Namun, sistem ini masih banyak memiliki kekurangan diantaranya seperti : masih banyak ditemukan noise seperti *thermal noise* pada *varactor* dan komponen lainnya, penuaan komponen analog yang mengarah pada kebutuhan untuk penyesuaian sesekali dan kompensasi untuk mengatasi masalah penyimpangan dan penyetyelan, kompromi yang kompleks antara noise, kecepatan dan *bandwidth loop* menggunakan nilai komponen yang dapat direalisasikan [2]. Karena hal tersebut, dikembangkanlah suatu sistem pemancar FM yang dapat mengurangi kekurangan-kekurangan pada sistem pemancar FM analog, yaitu pemancar FM Digital. Sistem pemancar FM digital menggunakan DDS. *Direct Digital Synthesiser* (DDS) adalah proses mereproduksi sinyal analog menggunakan nilai-nilai digital yang disimpan dalam tabel pencarian (LUT) [3]. DDS memiliki kelebihan yaitu frekuensi, fase, dan amplitudo outputnya dapat dimanipulasi secara tepat dan cepat di bawah kendali prosesor digital [4].

Karena hal tersebut, penelitian ini lebih difokuskan untuk pembuatan sistem pemancar radio FM digital, digitalisasi sistem pemancar FM ini akan dibuat menggunakan FPGA. Sebelumnya sudah ada beberapa penelitian dan pengembangan pembuatan sistem pemancar FM, seperti [8] membuat rancang bangun pemancar radio FM Berbasis Raspberry Pi dan [9] membuat implementasi sistem pemancar FM digital menggunakan FPGA. Namun, FPGA yang digunakan adalah jenis FPGA yang dibuat oleh Xilinx, sedangkan FPGA yang akan digunakan pada proyek akhir ini adalah jenis FPGA yang dibuat oleh Intel.

## 2. Dasar Teori

### 2.1 Radio Komunitas

Radio Komunitas (RK) merupakan lembaga penyiaran yang bergerak di bidang pelayanan siaran yang berbentuk badan hukum Indonesia, didirikan oleh komunitas tertentu, bersifat independen, dan tidak komersial, berdaya pancar rendah, luas jangkauan wilayah terbatas, serta untuk melayani kepentingan komunitasnya [10]. Menurut Peraturan Menteri Komunikasi dan Informatika Republik Indonesia Nomor 39 tahun 2012 Tentang Tata Cara Pendirian dan Penyelenggaraan Penyiaran Lembaga Penyiaran Komunitas Pada Bagian Ketiga mengenai Cakupan Wilayah Siaran Pasal 14 ayat 1 dan 2 dimana [11]:

(1) Radius siaran Lembaga Penyiaran Komunitas dibatasi maksimum 2,5 km (dua setengah kilometer) dari lokasi pemancar atau dengan *Effective Radiated Power* (ERP) maksimum 50 (lima puluh) watt.

(2) Radius siaran sebagaimana dimaksud pada ayat (1) dapat melebihi 2,5 km (dua setengah kilometer) dari lokasi pemancar atau dengan *Effective Radiated Power* (ERP) melebihi 50 (lima puluh) watt berdasarkan kebutuhan informasi komunitas/masyarakat di daerah yang sebaran penduduknya tidak padat dan terpencil atas usulan dari KPI dalam rekomendasi kelayakan [11].

Peraturan Menteri Komunikasi dan Informatika Republik Indonesia nomer 3 Tahun 2017 tentang rencana induk (master plan) frekuensi radio untuk keperluan penyelenggaraan radio siaran FM, pada Pasal 9 ditegaskan setiap penyelenggaraan radio siaran FM wajib mengikuti pemetaan kanal frekuensi radio siaran FM sebagaimana tercantum dalam Lampiran IV PM 3/2017. Dalam Lampiran tersebut pengkalan radio komunitas adalah [12]:

Kanal 202 = 107,7 MHz

Kanal 203 = 107,8 MHz

Kanal 204 = 107,9 MHz.

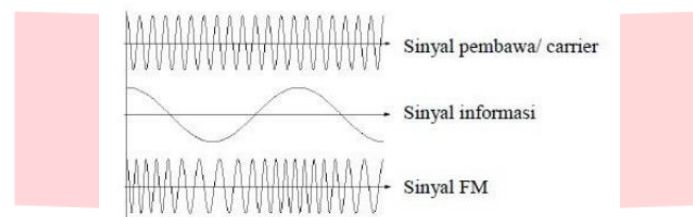
### 2.2 Radio FM

Radio *broadcasting* adalah salah satu jenis media massa satu arah yang berperan untuk menyampaikan pesan (berita, informasi dan hiburan) kepada masyarakat dengan jangkauan yang luas. Perkembangan radio *broadcasting* sebagai media massa mulai berkembang di beberapa negara. Diawali di Amerika Serikat (AS) dengan pengembangan penemuan Marconi oleh Dr. Lee De Forest pada tahun 1906, karena itu pula ia dijuluki "*The Father of Radio*". Sejak saat itu radio di AS mulai mengalami perkembangan. Sejak saat itu juga radio terus berkembang di beberapa negara seperti Inggris, Perancis, Uni Sovyet, Jepang dan RRC. Selain mengalami perkembangan, radio juga telah memasuki tahap penyempurnaan. Prof. EH Armstrong dari Universitas Columbia pada tahun 1933 memperkenalkan sistem *Frequency Modulation* (FM) sebagai penyempurnaan dari *Amplitude Modulation* (AM). Keuntungan FM dari AM antara lain [13] :

1. Dapat menghilangkan interference (gangguan/percampuran) karena cuaca.
2. Dapat menghilangkan interference yang disebabkan dua stasiun radio yang bekerja pada gelombang yang sama.
3. Menyiarkan suara sebaik-baiknya.

### 2.3 Modulasi FM

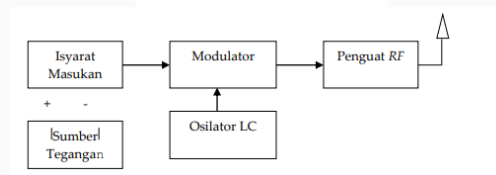
*Frequency Modulation* (FM) adalah proses penumpangan sinyal informasi pada sinyal pembawa (*carrier*) sehingga frekuensi gelombang pembawa (*carrier*) berubah sesuai dengan perubahan tegangan (simpangan) sinyal informasi. Pada modulasi frekuensi sinyal informasi mengubah-ubah frekuensi gelombang pembawa, sedangkan amplitudanya akan konstan selama proses modulasi [5]. Rentang frekuensi FM adalah 88 MHz – 108 MHz sehingga dikategorikan sebagai *Very High Frequency* (VHF), sedangkan panjang gelombangnya dibawah 1.000 KHz sehingga jangkauan sinyalnya tidak jauh. FM lebih tahan terhadap gangguan sehingga dipilih sebagai modulasi standar untuk frekuensi tinggi. Keuntungan FM antara lain, lebih tahan terhadap noise dibandingkan dengan AM [5]. Berikut adalah tabel beberapa perbedaan antara AM dan FM.



Gambar 2. 1 Bentuk Sinyal FM[5]

### 2.4 Sistem Pemancar FM

Secara umum pemancar FM memiliki alur seperti blok diagram di bawah ini :



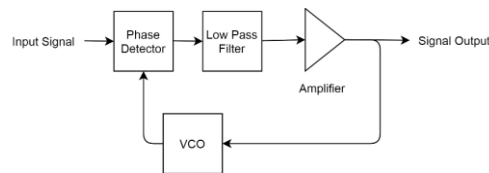
Gambar 2. 2 Blok Diagram Pemancar FM secara umum[15]

Pada prinsipnya, sebuah pembangkit gelombang radio (pemancar radio) terdiri dari sebuah osilator radio frekuensi, modulator, penguat daya, jalur transmisi, dan antenna. Sistem modulator dan penguat daya pemancar sangat ditentukan oleh jenis modulasi yang digunakan [15].

Sistem pemancar FM dibagi menjadi dua, yaitu sistem pemancar FM konvensional dan sistem pemancar FM digital. Namun, baik sistem pemancar FM konvensional maupun sistem pemancar FM digital memiliki sistem pemancar yang fungsinya hampir sama, hanya pada penamaan yang berbeda.

#### 2.4.1 Sistem Pemancar FM Konvensional

Pada sistem pemancar FM konvensional terdapat suatu sistem rangkaian bernama *Phase Locked Loop* (PLL) adalah suatu sistem umpan balik dimana sinyal umpan balik digunakan untuk mengunci frekuensi dan fasa keluaran pada suatu frekuensi serta fasa sinyal masukan. Gambar diagram blok dari PLL dapat dilihat di bawah ini [1]:



Gambar 2. 3 Blok Diagram PLL

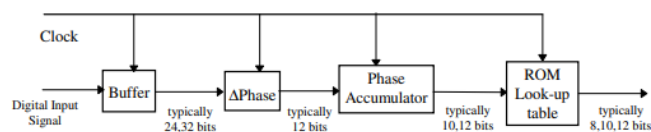
Cara kerja dari sistem PLL adalah sebagai berikut : *Phase Detector* membangkitkan sinyal keluaran yang berupa suatu fungsi beda fase kedua sinyal masukan. Keluaran detektor disaring dan tegangan DC dari perbedaan sinyal dimasukan pada suatu VCO. Sinyal umpan balik yang menuju detektor fase adalah frekuensi keluaran VCO [16]. Tegangan kontrol VCO memaksa VCO untuk mengubah frekuensi untuk mengurangi perbedaan antara frekuensi masukan dan frekuensi keluaran pembagi frekuensi. Jika kedua frekuensi tersebut cukup dekat, mekanisme umpan balik PLL memaksa kedua frekuensi masukan detektor fase menjadi sama, dan sinyal *free running* dari VCO dikunci. Ketika *loop* sudah terkunci, maka akan ada sedikit perbedaan fase antara kedua masukan detektor fase. Perbedaan ini akan menghasilkan tegangan DC pada keluaran detektor yang diperlukan oleh VCO untuk mempertahankan loop tetap terkunci. Kemampuan *self-correcting* membuat PLL mampu untuk melacak perubahan frekuensi dari sinyal masukan. Rentang frekuensi dimana PLL tetap dalam kondisi terkunci pada suatu sinyal masukan disebut *lock range*. *Capture range* adalah rentang frekuensi dimana PLL bisa melakukan penguncian [16].

#### 2.4.2 Sistem Pemancar FM Digital

Pada sistem pemancar FM digital terdapat suatu sistem rangkaian bernama *Direct Digital synthesizer* (DDS), DDS terdiri dari beberapa rangkaian yaitu sebagai berikut [2]:

- 1) *Numerically controlled oscillator* (NCO)  
NCO adalah generator sinyal digital yang menciptakan representasi bentuk gelombang yang sinkron, diskrit, bernilai diskrit, biasanya sinusoidal [17].
- 2) *Read-Only Memory* (ROM)  
ROM digunakan untuk menahan *look-up table* (LUT) untuk melakukan konversi fase ke amplitudo.
- 3) *Digital to Analogue Converter* (DAC)
- 4) *Filter Output Analog*  
Filter ini digunakan bertujuan sebagai anti- aliasing [2].

Untuk hubungan antar beberapa rangkaian tersebut, dapat dilihat pada blok diagram berikut :

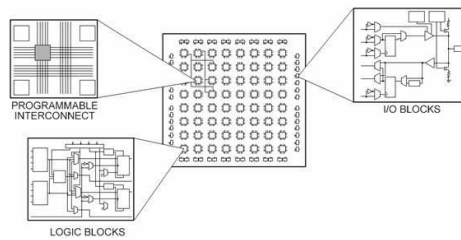


Gambar 2. 4 Blok Diagram DDS[2]

#### 2.5 Field Programmable Gate Array (FPGA)

*Field Programmable Gate Arrays* (FPGAs) adalah perangkat semikonduktor berbasis matriks blok logika yang dapat dikonfigurasi (CLB) yang terhubung melalui interkoneksi yang dapat diprogram [18]. Setiap chip FPGA terdiri dari sejumlah terbatas sumber daya yang telah ditentukan sebelumnya dengan interkoneksi yang

dapat diprogram untuk mengimplementasikan sirkuit digital yang dapat dikonfigurasi ulang dan blok I/ O untuk memungkinkan sirkuit mengakses dunia luar [19].



Gambar 2. 5 Blok Sistem FPGA[19]

### 2.6 Verilog HDL

Verilog adalah Bahasa Deskripsi Perangkat Keras; format tekstual untuk menggambarkan sirkuit dan sistem elektronik. Diterapkan untuk desain elektronik, Verilog dimaksudkan digunakan untuk verifikasi melalui simulasi, untuk analisis waktu, untuk analisis uji (analisis stabilitas dan penilaian kesalahan) dan untuk sintesis logika [20]. Berikut merupakan perbedaan arsitektur antara bahasa VHDL dan Verilog.

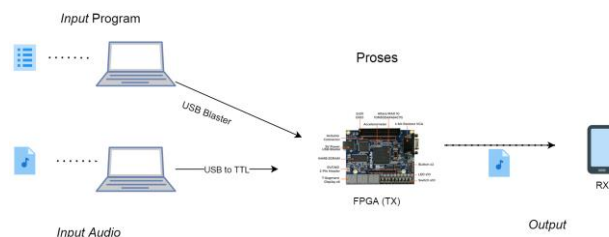
Tabel 2. 1 Perbedaan Verilog dan VHDL[20]

Verilog	VHDL
<pre> module d_flipflop(clk, d, q); input clk, d; output q;  reg q; always @(posedge clk) begin q &lt;= d; end  endmodule                     </pre>	<pre> library ieee; use ieee.std_logic_1164.all; entity d_flipflop is port( clk, d: in std_logic; q: out std_logic); end; architecture implement of d_flipflop is begin process(clk) begin if clk'EVENT and clk='1' then q &lt;= d; end if; end process; end;                     </pre>

### 3. Perancangan dan Simulasi

#### 3.1 Blok Diagram Sistem

Gambaran umum dari perancangan yang akan dibuat adalah sebagai berikut :



Gambar 3. 1 Skema Perancangan

Laptop terhubung dengan FPGA dengan dua kabel yaitu kabel USB Blaster tipe B dan USB to TTL. Kabel USB Blaster digunakan untuk mengirim program yang telah dibuat dari laptop ke FPGA, sedangkan kabel USB to TTL digunakan untuk mengirim data berupa *file audio* yang nantinya *file* tersebut akan disiarkan oleh FPGA sebagai radio TX dan akan diterima oleh perangkat RX (pada penelitian ini menggunakan *Handphone*).

### 3.2 Deskripsi *Hardware* dan *Software*

Beberapa *hardware* dan *software* yang digunakan adalah sebagai berikut :

#### 3.2.1 *Hardware*

Beberapa *hardware* yang digunakan dalam pengerjaan proyek akhir ini adalah sebagai berikut :

##### 1) FPGA

Spesifikasi FPGA yang digunakan dijabarkan pada tabel di bawah ini :

Tabel 3. 1 Spesifikasi FPGA[24]

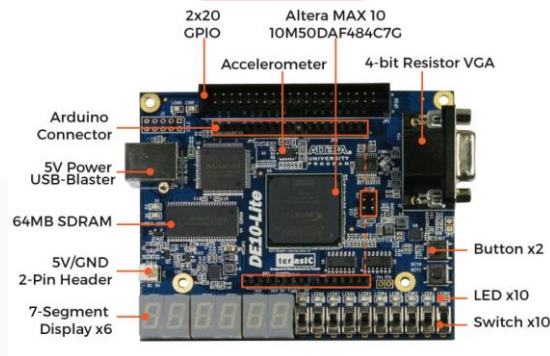
<i>FPGA Device</i>	<ul style="list-style-type: none"> <li>• <i>MAX 10 10M50DAF484C7G Device</i></li> <li>• <i>Integrated dual ADCs, each ADC supports 1 dedicated analog input and 8 dual function pins</i></li> <li>• <i>50K programmable logic elements</i></li> <li>• <i>1,638 Kbit M9K Memory</i></li> <li>• <i>144 18 × 18 Multiplier</i></li> <li>• <i>4 PLLs</i></li> </ul>
<i>Programming and Configuration</i>	<i>On-Board USB Blaster (Normal type B USB connector)</i>
<i>Memory Device</i>	<i>64MB SDRAM, x16 bits data bus</i>
<i>Sensor</i>	<i>Accelerometer</i>
<i>Expansion Connectors</i>	<ul style="list-style-type: none"> <li>• <i>One 2x20 GPIO Connector(voltage levels: 3.3V)</i></li> <li>• <i>Arduino Uno R3 Connector, including six ADC channels.</i></li> </ul>
<i>Display</i>	<i>4-bit Resistor VGA</i>
<i>Switches/Buttons/LEDs/7-Segment Display</i>	<ul style="list-style-type: none"> <li>• <i>10 LEDs</i></li> <li>• <i>10 Slide Switches</i></li> <li>• <i>2 Push Buttons</i></li> <li>• <i>Six 7-Segments Display</i></li> </ul>
<i>Power</i>	<i>5V DC input</i>



Untuk *board* FPGA, dapat di lihat pada gambar di bawah ini :



Gambar 3. 2 *Board* FPGA dan USB Blaster[24]



Gambar 3. 3 Fitur pada *board* FPGA DE-10 *Lite*[24]

2) USB to TTL

USB to TTL ini digunakan sebagai terminal penghubung antara FPGA dan laptop untuk mengirim data audio yang di inginkan.



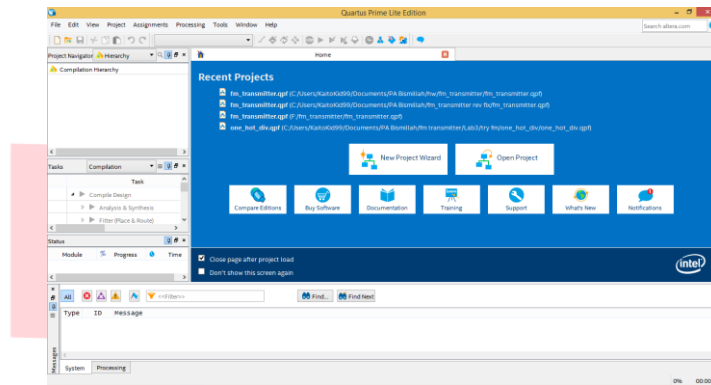
Gambar 3. 4 USB to TTL/ Converter USB UART[21]

### 3.2.2 Software

Beberapa *software* yang digunakan adalah sebagai berikut :

#### a) Software Application

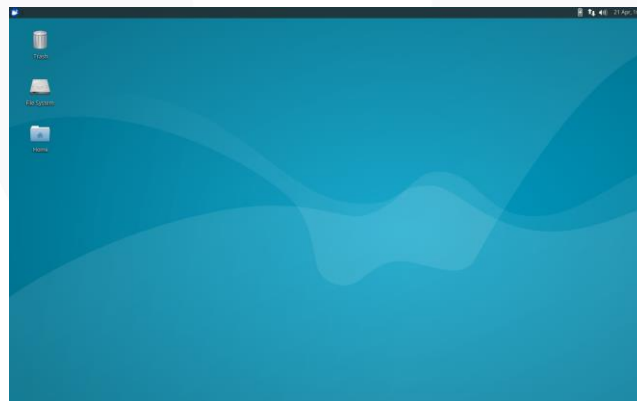
*Software* aplikasi yang digunakan adalah Quartus Altera versi 18.1 *Lite*. *Software* ini digunakan karena *compatible* dengan *board* FPGA yang digunakan. Tampilan utama dari *software* Quartus Altera versi 18.1 *Lite* adalah seperti pada gambar di bawah ini :



Gambar 3. 5 Tampilan Awal *Software* Quartus versi 18.1 *Lite*

#### b) Software Operating System

*Software* OS yang digunakan pada proyek akhir ini adalah Linux versi Xubuntu. OS ini digunakan untuk mengirim data berupa audio dari PC/ Laptop ke FPGA via kabel USB to TTL.

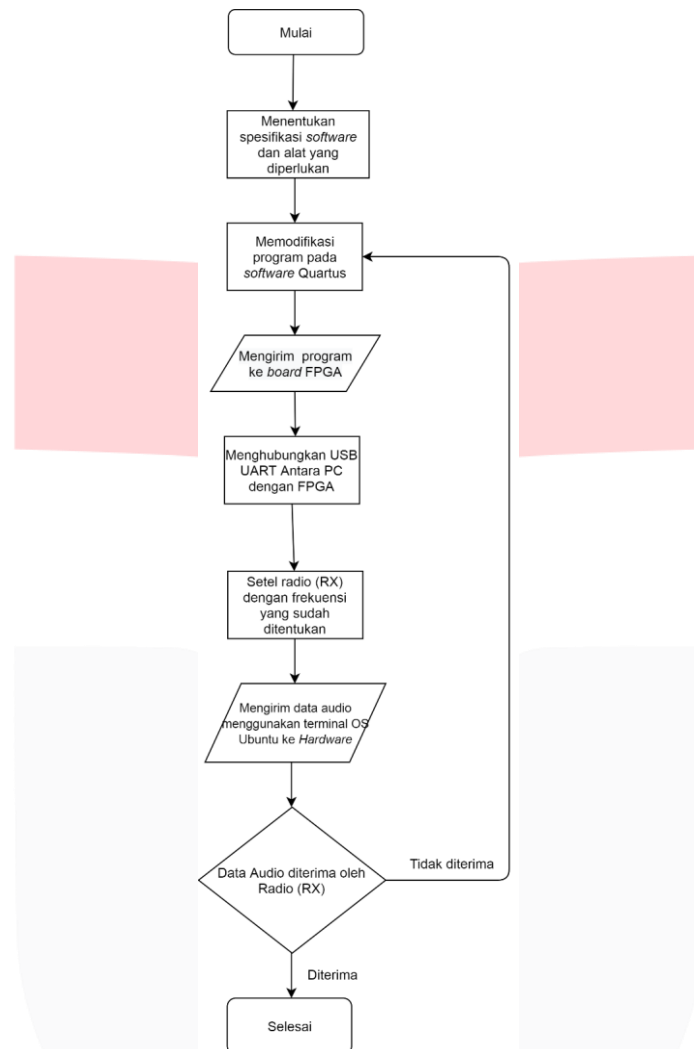


Gambar 3. 6 Tampilan *Desktop* Xubuntu



### 3.3 Tahapan Perancangan dan Simulasi Alat

Tahapan perancangan FM *transmitter* ini dilakukan dengan metode *try and error*, ada beberapa tahapan yang dilakukan dalam perancangan FM *transmitter* ini, tahapan perancangan dapat dilihat pada *flowchart* berikut :



Gambar 3. 7 Diagram Alir Perancangan Alat

Beberapa tahapan yang dilakukan dalam perancangan alat ini adalah sebagai berikut:

1. Tahapan awal perancangan alat ini dimulai dengan menentukan spesifikasi alat dan *software* yang digunakan, dimana alat yang digunakan adalah *board* FPGA DE-10 *Lite*, kabel *converter* USB UART, kabel jumper, laptop. Lalu, *software* yang digunakan adalah *Quartus Lite* 18.1 yang digunakan untuk memodifikasi program sebelum dikirim ke *board* FPGA dan terminal pada OS Xubuntu untuk mengirim data audio melalui USB UART ke *board* FPGA.
2. Tahap kedua, setelah penentuan spesifikasi alat dan *software* selesai tahapan selanjutnya adalah memodifikasi program pada *software* Quartus, dimana program yang dipakai adalah program yang sudah ada (didapat dari referensi acuan [21] ), namun program tersebut harus dimodifikasi kembali untuk penyesuaian dengan kebutuhan. Salah satu yang harus di

modifikasi adalah frekuensi yang ingin digunakan, pada program frekuensi yang digunakan direpresentasikan ke dalam nilai *hexadecimal*. Untuk mendapatkan frekuensi yang diinginkan maka harus melakukan perhitungan terlebih dahulu pada bentuk *decimal*. Setelah mendapatkan hasil *decimal* angka-angka tersebut di *convert* pada bentuk *hexadecimal*, baru dapat digunakan pada program. Rumus yang digunakan untuk perhitungan tersebut adalah :

$$2^{32} \frac{f}{f_{in}} \quad (3.1)$$

Ket :

f = Frekuensi *carrier*

f<sub>in</sub> = Frekuensi *clock* (250MHz)

Sehingga apabila dimasukan angka, menjadi sebagai berikut :

$$2^{32} \frac{107900000}{250000000} \quad (3.1)$$

$$=1853707885$$

Setelah didapatkan hasil bilangan *decimal* lalu bilangan tersebut di *convert* ke dalam bentuk bilangan *hexadecimal*. Sehingga hasil yang didapatkan nilai akhir dalam bentuk *hexa* adalah 6E7D566D. Angka *hexadecimal* tersebut dapat dimasukan ke program. dimasukan ke program. Untuk pengaturan pada programnya adalah sebagai berikut :

```

9
10 reg [31:0] inifrek;
11
12 always@(sw) begin
13     if (sw == 2'b00) begin
14         inifrek = 32'h6E48_E8A7
15         ;
16         ledoutput = 19'b0000000000001111000;
17     end
18     else if (sw == 2'b01) begin
19         inifrek = 32'h6E63_1F8A;
20         ledoutput = 19'b0000000000000000000;
21     end
22     else if (sw == 2'b10) begin
23         inifrek = 32'h6E7D_566C;
24         ledoutput = 19'b000000000000010000;
25     end
26 end
27

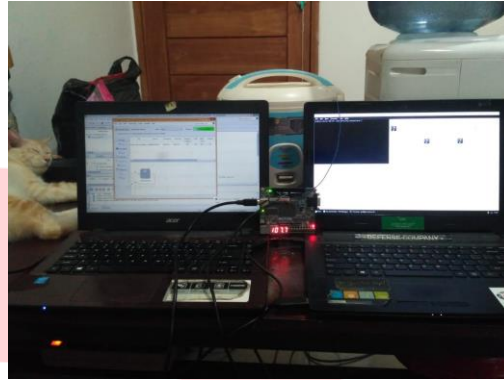
```

Gambar 3. 8 Pengaturan program untuk pengaturan frekuensi

3. Tahap ketiga, setelah program selesai dimodifikasi dan sudah tidak terdapat *error* pada program, tahapan selanjutnya adalah program dimasukan atau di *upload* ke *board* FPGA. Setelah *upload* program sukses,
4. Tahap keempat, adalah menghubungkan kabel *converter* USB UART antara PC atau laptop dengan *board* FPGA. Setelah USB UART terhubung antara laptop dengan *board* FPGA.
5. Tahap kelima, yaitu menyetel radio (RX) pada frekuensi yang ditentukan atau yang dipakai (frekuensi yang dipakai adalah frekuensi yang di tetapkan untuk radio Komunitas).
6. Tahap keenam, setelah radio (RX) siap, langkah selanjutnya yaitu mengirim data audio dengan ekstensi file *.wav* menggunakan terminal yang terdapat pada OS Xubuntu melalui USB UART yang terhubung dari laptop ke *board* FPGA.

7. Tahapan terakhir adalah mendengar atau mengamati pada radio (RX) apakah data audio yang dikirim sudah diterima oleh radio (RX) atau belum, apabila sudah diterima maka alat yang dibuat sudah berfungsi dan berhasil, apabila belum maka kemungkinannya adalah ada kesalahan pada program yang sudah dimodifikasi sehingga harus mengecek kembali program.

Gambar untuk *setup* antara FPGA dan laptop yang terhubung seperti di bawah ini :



Gambar 3. 8 *Setup* FPGA dan Laptop

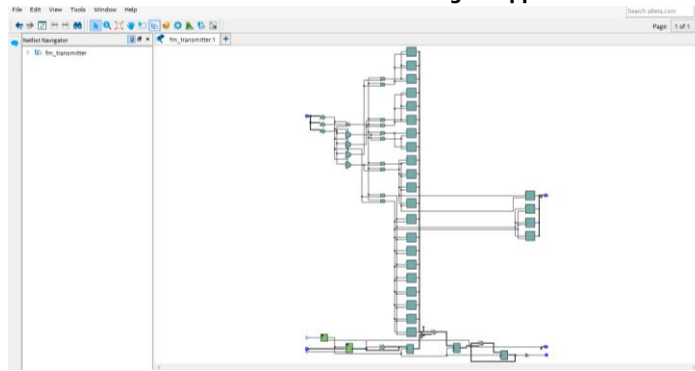
#### 4. Analisis simulasi dan pengujian alat

##### 4.1 Hasil Simulasi dan Pengujian Alat

Simulasi dan pengujian alat dimaksudkan untuk mengetahui apakah alat yang telah dibuat sudah memenuhi hasil yang diharapkan atau tidak. Setelah melakukan simulasi dan pengujian, didapatkan beberapa hasil sebagai berikut :

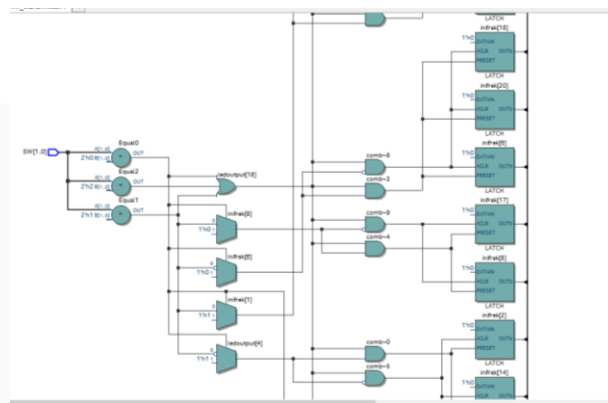
##### 4.1.1 Hasil Simulasi Program

Sebelum melakukan uji coba pada alat, hal yang perlu dilakukan adalah melihat hasil simulasi program. Simulasi program dapat dilihat setelah program selesai dibuat dan dilakukan *compiling* pada program yang nantinya akan dikirim ke FPGA. Hal ini dilakukan juga untuk mengecek bagaimana nanti program bekerja pada FPGA setelah program dikirim. Hasil simulasi program ini disajikan dalam bentuk blok diagram RTL, dimana hasilnya sebagai berikut :



Gambar 4. 1 Blok Diagram RTL

Hasil dari blok diagram RTL tersebut secara garis besar sudah sesuai dengan program yang dibuat, dimana program dibuat agar bisa memilih beberapa frekuensi radio komunitas yang sudah ditetapkan oleh pemerintah dan memodulasi frekuensi tersebut dengan data *audio* yang dikirimkan. Untuk blok-blok yang menunjukkan pemilihan frekuensi pada blok diagram RTL di atas dapat dilihat di bawah ini :



Gambar 4. 2 Blok RTL pemilihan frekuensi

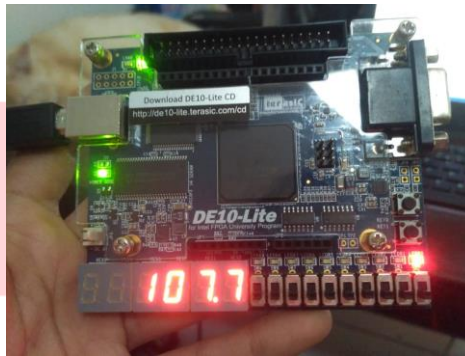
Tampilan blok RTL di atas berdasarkan dari program yang dibuat agar pada FPGA dapat memilih beberapa frekuensi yang diinginkan dan menampilkannya pada *Seven segment*. Secara garis besar bagian program tersebut adalah seperti gambar di bawah :

```

12 always@(sw) begin
13     if (sw == 2'b00) begin
14         inifrek = 32'h6E48_E8A7;
15         ledoutput = 19'b0000000000001111000;
16     end
17     else if (sw == 2'b01) begin
18         inifrek = 32'h6E63_1F8A;
19         ledoutput = 19'b0000000000000000000;
20     end
21     else if (sw == 2'b10) begin
22         inifrek = 32'h6E7D_566C;
23         ledoutput = 19'b000000000000000010000;
24     end
25 end
26
    
```

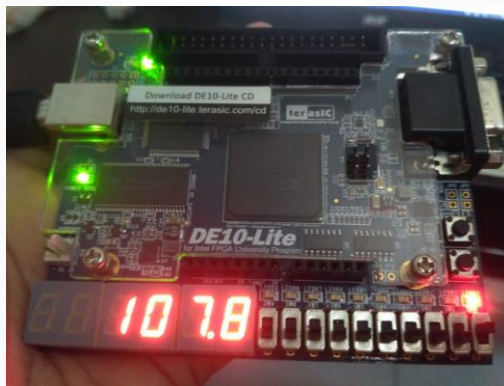
Gambar 4. 3 Part pemrograman pemilihan frekuensi

*Input* yang digunakan adalah dua buah *switch* yang di terjemahkan menjadi dua bit 0 dan 1 sebagai *trigger* untuk pemilihan frekuensi. Terdapat tiga frekuensi yang dapat dipilih yaitu 107,7MHz – 107,9 MHz, frekuensi ini adalah frekuensi yang telah ditetapkan sebagai kanal frekuensi untuk radio komunitas. *Default* ketika program sudah di masukan ke dalam FPGA frekuensi yang terlihat pada FPGA adalah 107,7MHz, karena pada program diatur apabila kedua *switch* berada dibawah, *switch* menunjukkan bit 00. *Trigger* 00 dari *switch* pada program dimaksudkan untuk memilih FPGA bekerja pada frekuensi 107,7MHz, tampilan pada FPGA untuk frekuensi 107,7MHz seperti gambar berikut :



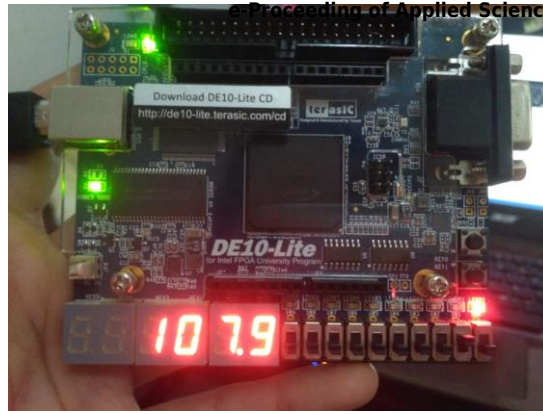
Gambar 4. 4 Tampilan FPGA untuk frekuensi 107,7MHz

Apabila *trigger* dari *switch* adalah 01 maka itu dimaksudkan untuk memilih FPGA bekerja pada frekuensi 107,8MHz, tampilan pada FPGA untuk frekuensi 107,8MHz seperti gambar berikut:



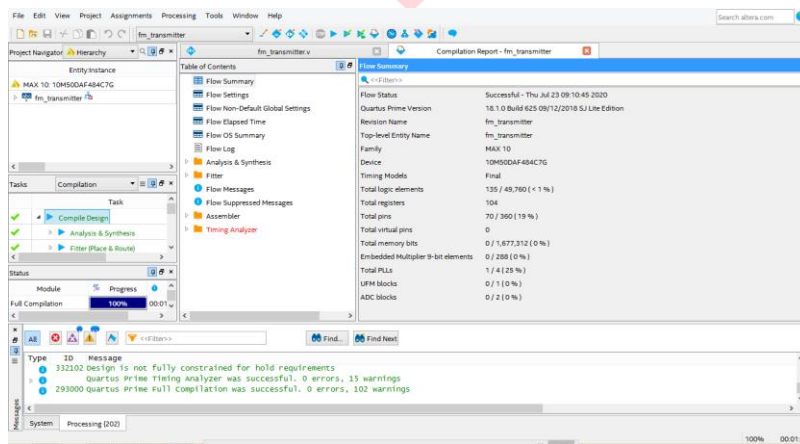
Gambar 4. 5 Tampilan FPGA untuk frekuensi 107,8MHz

sedangkan apabila *trigger* dari *switch* adalah 10 itu dimaksudkan untuk memilih FPGA bekerja pada frekuensi 107,9MHz. tampilan pada FPGA untuk frekuensi 107,8MHz seperti gambar berikut:



Gambar 4. 6 Tampilan FPGA untuk frekuensi 107,9MHz

Sehingga setelah program dimasukan ke FPGA, pada FPGA tersebut dapat diatur frekuensi berapa yang di inginkan hanya dengan merubah *switch* tanpa harus merubah komponen khusus seperti pada FM *transmitter* Analog. Selain blok diagram RTL, data yang didapatkan dari simulasi adalah data hasil *synthesis* dari program, dimana hasil *synthesis* ini melaporkan penggunaan sumber (*resource*) Pada FPGA. Data hasil *synthesis* dari program dapat dilihat pada gambar di bawah ini :



Gambar 4. 7 Hasil *synthesis* pada software Quartus Lite

Dari hasil *Syntesis* tersebut, secara keseluruhan dihasilkan utilisasi FPGA sebagai berikut :

Tabel 4. 1 utilisasi FPGA

Total pin	70 dari 360	19%
104	104	-
Total <i>logic element</i>	135 dari 49,760	<1%
Total PLL	1 dari 4	25%

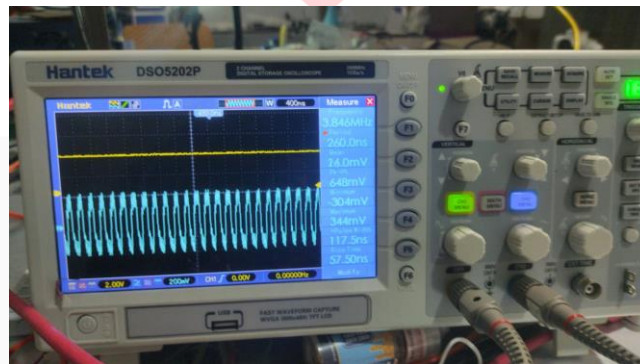
#### 4.1.2 Hasil Pengujian Alat

Pengujian alat ini dimaksudkan untuk melihat apakah alat yang dibuat sudah sesuai dengan yang diharapkan atau tidak. Beberapa langkah yang dilakukan dalam pengujian alat ini adalah dengan mengukur menggunakan osiloskop untuk domain waktu dan RTL-SDR untuk domain frekuensi, Dua

pengukuran ini menggunakan cara yang tidak sama. Pada pengukuran menggunakan osiloskop, kaki-kaki pin yang digunakan pada FPGA yaitu GPIO 0 dan GPIO 35 dihubungkan dengan kabel *probe* osiloskop untuk melihat sinyal yang dikirimkan, sedang untuk RTL-SDR menggunakan *dongle* yang ditancapkan pada laptop untuk bertindak sebagai *receiver* FM, dan *software* RTL-SDR untuk melihat tampilan sinyal yang diterima oleh *dongle*. Dari FPGA data yang dikirimkan adalah data digital *file* audio .wav 8 bit dengan komunikasi uart memakai *baudrate* 230400bit/second dengan 8 bit data, 2 stop bit, dan tidak menggunakan bit *parity*. Untuk tampilan hasil yang didapatkan adalah sebagai berikut :

#### 1) Hasil Pengukuran Osiloskop

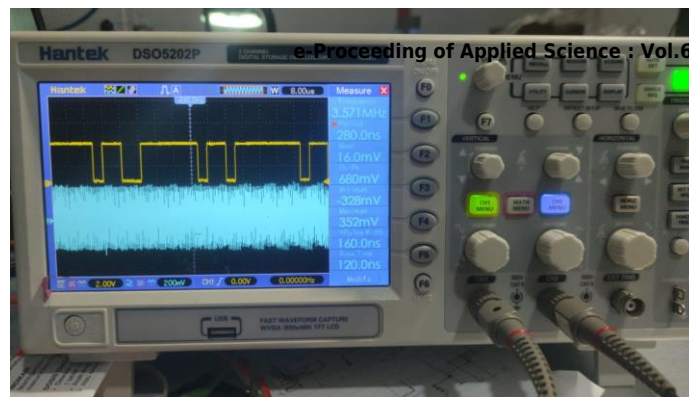
Pengukuran menggunakan osiloskop ini dimaksudkan untuk melihat tampilan sinyal dalam domain waktu. Pada pengukuran ini menggunakan dua *Channel*, hal ini dimaksudkan untuk mendapatkan dua sinyal di kaki pin berbeda pada FPGA, *channel* 1 dihubungkan ke pin GPIO 0 untuk mendapatkan sinyal data serial yang dikirimkan, *channel* 2 dihubungkan ke pin GPIO 35 yang di set sebagai antena untuk melihat sinyal FM (gambar catuan dapat dilihat pada lampiran A). Dari proses tersebut, didapatkan hasil sebagai berikut :



Gambar 4. 8 cuplikan tampilan pada Osiloskop ketika tidak ada data

Gambar di atas menunjukkan tampilan osiloskop ketika tidak ada data yang dikirimkan, garis berwarna kuning adalah tampilan sinyal pada osiloskop untuk *channel* 1 (menampilkan sinyal data serial), di bawah garis kuning terdapat gelombang berwarna biru muda. Gelombang biru muda tersebut adalah tampilan pada osiloskop untuk *channel* 2 (menampilka sinyal FM). Ketika tidak ada data yang dikirimkan, tampilan sinyal untuk data serial pada *channel* 1 *flat* (tidak ada data yang dikirimkan), sedangkan untuk sinyal FM pada *channel* 2 dalam keadaan diam, ini menunjukkan tidak adanya data yang masuk sehingga tidak ada pergerakan ataupun perubahan sinyal.



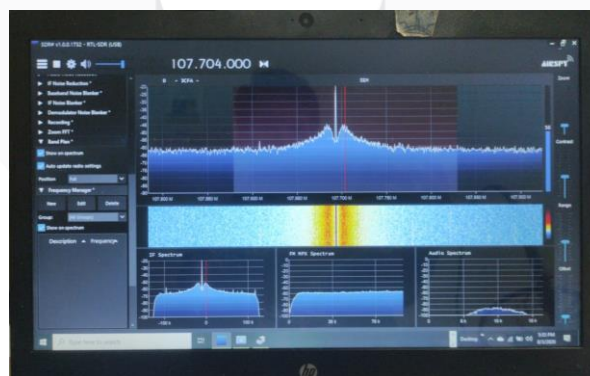


Gambar 4. 9 Cuplikan Sinyal pada osiloskop ketika data masuk

Pada gambar 4.9 di atas terlihat perbedaan dengan gambar sebelumnya 4.8, yaitu sudah ada data yang masuk sehingga tampilan sinyal pada *channel 1* tidak lagi *flat* seperti sebelumnya, dan tampilan sinyal pada *channel 2* menunjukkan perubahan, ini menandakan adanya sinyal yang berubah ketika data dikirim, walaupun rapat-renggang pada tampilan sinyal FM tidak terlalu kelihatan. Hal ini dikarenakan osiloskop menerima data perbit, sedangkan agar pergeseran sinyalnya terlihat data yang diterima seharusnya per delapan bit karena sesuai dengan data yang dikirim yaitu file audio .wav yang berjumlah 8 bit.

## 2) Hasil Pengukuran RTL-SDR

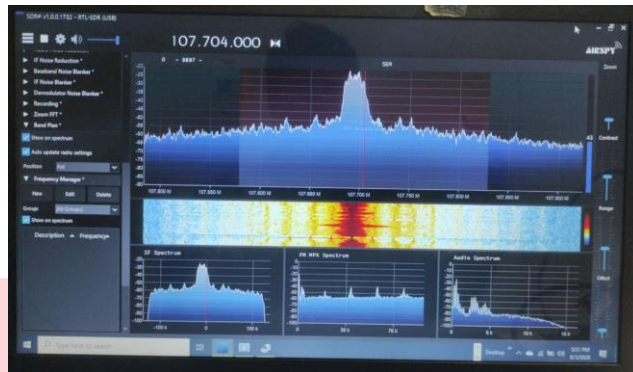
RTL-SDR adalah *dongle* yang dapat digunakan sebagai pemindai radio berbasis komputer untuk menerima sinyal radio langsung tanpa perlu internet [24] (gambar *dongle* seperti pada lampiran B). *Dongle* ini tidak akan terlihat fungsinya tanpa bantuan sebuah *software* untuk melengkapinya. SDR# saat ini adalah program SDR paling populer yang digunakan dengan RTL-SDR. RTL-SDR ini dapat digunakan untuk beberapa pengaplikasian terkait radio pita lebar, salah satunya adalah dapat digunakan sebagai *Spektrum Analyzer*. Pada pengukuran ini, RTL-SDR digunakan sebagai penerima FM dan *Spectrum Analyzer* untuk melihat sinyal FM dalam domain frekuensi. Hasil yang didapatkan adalah sebagai berikut :



Gambar 4. 10 Tampilan Spektrum pada software SDR# ketika tidak ada data yang dikirim

Pada tampilan gambar di atas menunjukkan spektrum yang stabil tidak ada perubahan, hanya terdapat *floor noise*, pada grafik *audio* spektrum yang terletak di kanan bawah pun terlihat tidak terjadi perubahan cenderung sedikit spektrum yang tertangkap oleh RTL-

SDR, ini menandakan belum adanya data yang dikirim dari FPGA yang bertindak sebagai FM transmitter digital. Perbedaan spektrum terlihat ketika ada data yang dikirimkan dari FPGA, perbedaan tersebut dapat dilihat pada gambar di bawah ini :



Gambar 4. 11 Tampilan spektrum pada saat pengiriman data

Pada gambar di atas terlihat perbedaan dengan gambar sebelumnya yaitu 4.10. Dimana, pada gambar 4.11 terlihat perubahan spektrum yang melebar dari sebelumnya pada gambar 4.10, ini menandakan adanya data yang diterima sehingga terjadi perubahan spektrum yang semula meruncing karena tidak ada data yang diterima menjadi melebar setelah adanya penerimaan data.

## 5. Penutup

### 5.1 Kesimpulan

Berdasarkan hasil perancangan, pengujian dan analisa yang telah dilakukan maka dapat diambil kesimpulan bahwa FM transmitter digital yang dibuat sudah sesuai dengan yang diharapkan, yaitu menggunakan FPGA dengan *type board DE-10 Lite* keluaran Intel dengan perancangan menggunakan bahasa verilog. FM transmitter ini sudah dapat bekerja pada frekuensi yang diatur untuk radio komunitas yaitu 107,7MHz-107,9MHz, dan pengiriman data audionya menggunakan *baudrate* 230400bit/second dengan 8 bit data, 2 stop bit dan tidak menggunakan bit *parity*.

### 5.2 Saran

Berdasarkan hasil pembuatan Proyek Akhir ini, dapat disampaikan beberapa saran untuk pengembangan selanjutnya yaitu :

- 1) Tambahkan format file audio yang dapat dikirimkan seperti Mp3, RAW.
- 2) Gunakan antenna modifikasi yang dapat mencakup jarak yang jauh atau sesuai ketentuan jarak untuk radio komunitas yaitu 2,5 Km.
- 3) Kantongi izin sebelum melakukan pengujian, agar tidak terjadi hal-hal yang tidak diinginkan,terkait penggunaan frekuensi dan cakupan jarak yang diharapkan.

**DAFTAR PUSTAKA**

- [1] A. L. Siskom, Modul Praktikum Sistem Komunikasi, Bandung: Fakultas Ilmu Terapan, Telkom University, 2018.
- [2] U. L. FIT, "Jenis-jenis Modulasi Sinyal," Fakultas Ilmu terapan, telkom University, 12 Mei 2017. [Online]. Available: <https://fit.labs.telkomuniversity.ac.id/jenis-jenis-modulasi-sinyal/>. [Diakses 20 Mei 2020].
- [3] R. M. D. C. Martins, "Rancang bangun Pemancar FM Portable," 2013.
- [4] S. Twelves, "A Digital Stereo FM Modulator for Broadcasting," *The Open University (UK)*, July 1997.
- [5] K. Harrison dan S. L. R. R. S. Warren Fletcher, "Direct Digital Synthesis," EEE4084F, South Africa, 2017.
- [6] J. Vankka, "Direct Digital Synthesizers : Theory, Design and Applications," *Helsinki University of Technology*, November 2000.
- [7] R. Sopian dan S. ., D. D. Iswahyudi Hidayat, "Pengujian dan verifikasi Desain Penerima FM Digital Pada FPGA," Telkom University, Bandung, 2012.
- [8] Q. K. Aditya dan S. M. A. Dr.F. Yudi Limpraptono, "Rancang Bangun Pemancar Radio FM berbasis raspberry Pi," Institut Teknologi malang, Malang, 2019.
- [9] H. Bostrom, "An FPGA Implementation of a Digital FM Modulator," Linkoping University, Linkoping, Swedia, 2011.
- [10] Admin, "Radio Komunitas : pengertian, Karakter, Izin, dan Program," 20 Oktober 2012. [Online]. Available: <https://romeltea.com/radio-komunitas-karakter-dan-program/>.
- [11] Menkominfo, "Tata Cara Pendirian dan Penyelenggaraan Penyiaran Lembaga Penyiaran Komunitas No.39," Menkominfo, Jakarta, 2012.
- [12] Menkominfo, "Rencana Induk (Master Plan) Frekuensi Radio untuk Keperluan Penyelenggaraan Radio Siaran FM," Menkominfo, Jakarta, 2017.
- [13] RG, "Sejarah Singkat Perkembangan radio," Komisi Penyiaran Indonesia, 03 Januari 2018. [Online]. Available: <http://www.kpi.go.id/index.php/id/umum/38-dalam-negeri/34250-sejarah-perkembangan-radio?detail5=5290> . [Diakses 25 November 2019].
- [14] Admin, "AM vs. FM," Diffen, 2012. [Online]. Available: [https://www.diffen.com/difference/AM\\_vs\\_FM](https://www.diffen.com/difference/AM_vs_FM). [Diakses August 2020].
- [15] A. Junior, "Modulasi Analog, Modulasi Frekuensi (FM) Modulasi Phae (PM)," Academia Edu, Mataram.
- [16] B. Murtianta, "Pemancar dan Penerima FM," *Techné Jurnal Ilmiah Elektroteknika*, vol. 16 , no. 2, pp. 65-78, 2017.

- [17] R. S dan T. W. Endah Setyaningsih, "Alat Interkom Melalui Jala-jala Listrik Pada Jurusan Teknik Elektro Universitas Taruma Negara," TESLA, vol. 15, 2013.
- [18] R. Dhoke dan P. V. M. Minal Kharbikar, "Design of CMOS Based Numerical Control Oscillator with Better Performance Parameter 45nm CMOS Process," IJIRST- International Jurnal for Innovative Research in Science & Technology, vol. 2, no. 09, 2016.
- [19] Admin, "Field Programmer Gate Array," Xilinx, [Online]. Available: <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html> . [Diakses 25 November 2019].
- [20] Admin, "FPGA Fundamentals," National Instrument, 05 November 2019. [Online]. Available: <https://www.ni.com/en-id/innovations/white-papers/08/fpga-fundamentals.html>. [Diakses 02 Januari 2020].
- [21] Admin, "What is Verilog?," Doulos, [Online]. Available: [https://www.doulos.com/knowhow/verilog\\_designers\\_guide/what\\_is\\_verilog/](https://www.doulos.com/knowhow/verilog_designers_guide/what_is_verilog/) . [Diakses 25 November 2019].
- [22] Terasic, DE-10 Lite User Manual, [www.terasic.com](http://www.terasic.com), 2018.
- [23] K.Levgen, "Лабораторна робота №3 Схеми ділення і підвищення частоти, таймінг аналіз, цифровий FM передавач," Github, Ukraina, 2018.
- [24] Dzale, "FPGA Design Entry," Ndownare, 19 November 2009. [Online]. Available: <https://ndoware.com/fpga-design-entry.html> . [Diakses 25 November 2019].
- [25] S. Roy, "FPGA IMPLEMENTATION – Step By Step," Digital System Design, 21 Mei 2019. [Online]. Available: <https://digitalsystemdesign.in/fpga-implementation-step-by-step/>. [Diakses 29 Juli 2020].
- [26] Admin, "RTL-SDR.com," 2016. [Online]. Available: <https://www.rtl-sdr.com/about-rtl-sdr/>. [Diakses Agustus 2020].