

Implementasi FM Transmitter Di FPGA Untuk Radio Komunitas Fakultas Ilmu Terapan

Sri Kalbu Balilayaran^{1*}, Denny Darlis¹, Dwi Andi Nurmantris¹

¹Program Studi D3 Teknologi Telekomunikasi Fakultas Ilmu Terapan, Universitas Telkom, Dayeuh Kolot, Bandung.

Informasi Artikel

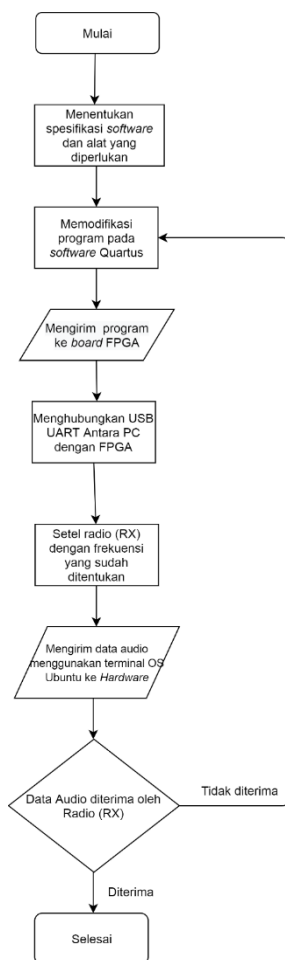
Naskah Diterima: 31 Agustus 2020

Direvisi: 21 Desember 2020

Disetujui: 21 Desember 2020

*Korespondensi Penulis :
srikalbulalilayaran@gmail.com

Graphical abstract



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 setup 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

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. Penelitian 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 radio 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 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

© 2020 Penerbit Jurusan Teknik Elektro UNTIRTA Press. All rights reserved

1. PENDAHULUAN

Frequency Modulation (FM) 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 [1]. Pada Modulasi Frekuensi ini, sinyal informasi akan mengubah frekuensi gelombang pembawanya sedangkan amplitudonya tetap selama proses modulasi [2]. Salah satu implementasi dari teknik modulasi frekuensi adalah Radio FM.

Radio FM ini ditemukan oleh Edwin Howard Amstrong 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). 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 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 penyetelan, kompromi yang kompleks antara *noise*, kecepatan dan *bandwidth loop* menggunakan nilai komponen yang dapat direalisasikan [3]. 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) [4]. DDS memiliki kelebihan yaitu frekuensi, *phase*, dan amplitudo *outputnya* dapat dimanipulasi secara tepat dan cepat di bawah kendali prosesor digital [5].

Pada penelitian ini lebih difokuskan untuk pembuatan sistem pemancar radio FM digital, menggunakan *Field Programmable Gate Array* (FPGA). Sebelumnya sudah ada beberapa penelitian dan pengembangan pembuatan sistem pemancar FM, seperti [6] membuat rancang bangun pemancar radio FM Berbasis Raspberry Pi dan [7] 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 penelitian ini adalah MAX10 yang dibuat oleh Intel. Di sisi penerima pengujian dan verifikasi desain FM Digital juga telah dilakukan pada [8].

2. METODE PENELITIAN

2.1 Metode Penelitian

Proses penelitian terbagi menjadi beberapa tahap yang dilakukan berdasarkan urutan dalam melakukan penelitian:

- a) Identifikasi masalah yaitu dengan merumuskan latar belakang hingga tujuan dalam penelitian ini.
- b) Studi literatur, yaitu mengumpulkan data-data dari buku referensi dan jurnal-jurnal sesuai dengan topik penelitian yang dilakukan yaitu tentang FM *transmitter* digital.
- c) Perancangan dan pengujian, yaitu dengan merancang model sistem FM *transmitter* digital serta menguji dan mensimulasikan menggunakan perangkatnya secara langsung, untuk mengetahui optimal penerimaan data yang dikirimkan dari perangkat (tx) ke penerima (rx) terhadap hasil yang diinginkan dan teori yang ada

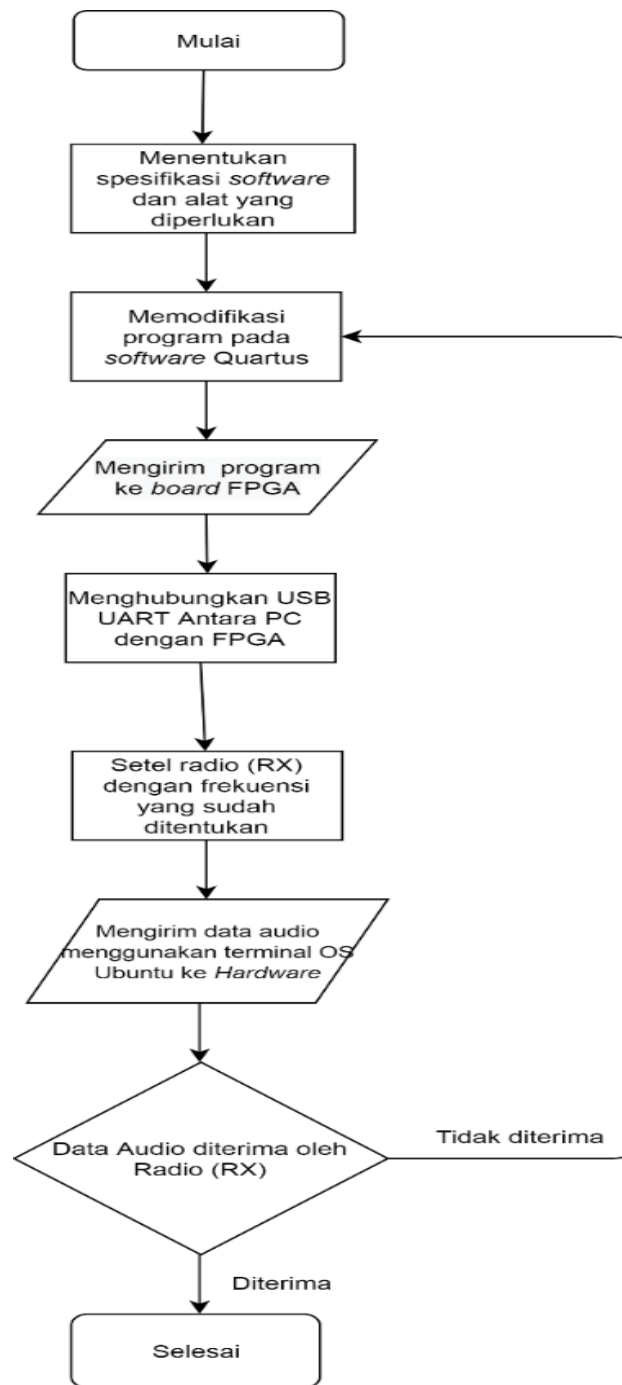
2.2 Diagram Alir Penelitian

Tahapan perancangan FM *transmitter* ini dilakukan dengan metode perancangan sistem digital untuk blok yang digunakan di sistem analog, ada beberapa tahapan yang dilakukan dalam perancangan FM *transmitter* ini, tahapan perancangan ditunjukkan oleh *flowchart* pada gambar 1.

2.3 Perancangan Penelitian

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.



Gambar 1. Diagram Alir Penelitian

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 (diambil dari penelitian [9]), namun program tersebut harus dimodifikasi kembali untuk penyesuaian dengan kebutuhan. Salah satu yang harus dimodifikasi 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. Persamaan 1 digunakan untuk perhitungan tersebut.

$$2^{32} \frac{f}{f_{in}} \quad (1)$$

Ket :
 f = Frekuensi *carrier*
 fin= Frekuensi *clock* (250MHz)

Sehingga apabila dimasukan angka, menjadi sebagai berikut :

$$2^{32} \frac{107900000}{250000000} = 1853707885$$

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 program ditunjukkan oleh gambar 2.

```

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'b00000000000000000000;
21     end
22     else if (sw == 2'b10) begin
23         inifrek = 32'h6E7D_566C;
24         ledoutput = 19'b000000000000000010000;
25     end
26 end
27
    
```

Gambar 2. 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 3. Setup FPGA dan Laptop

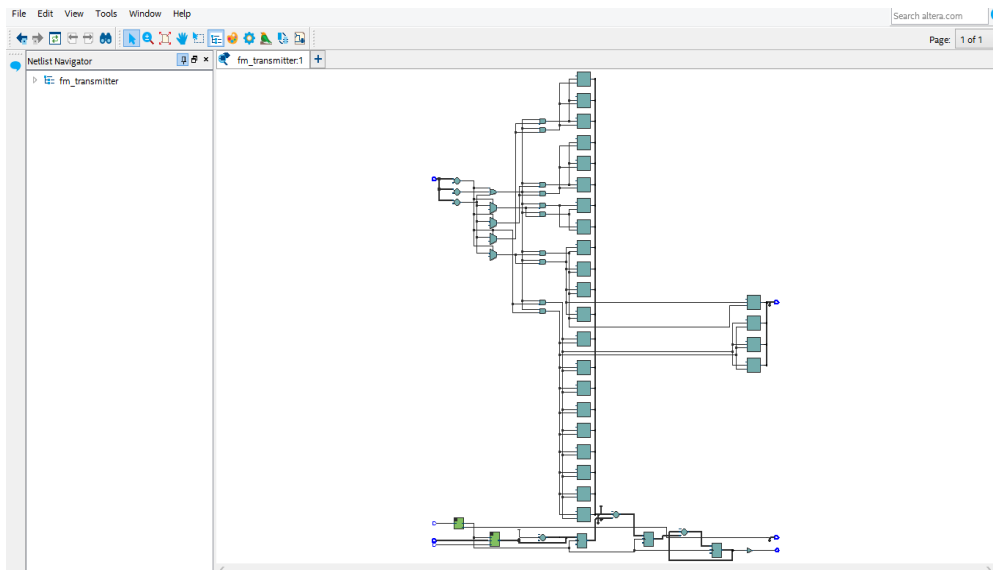
3. HASIL DAN PEMBAHASAN

3.1 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 :

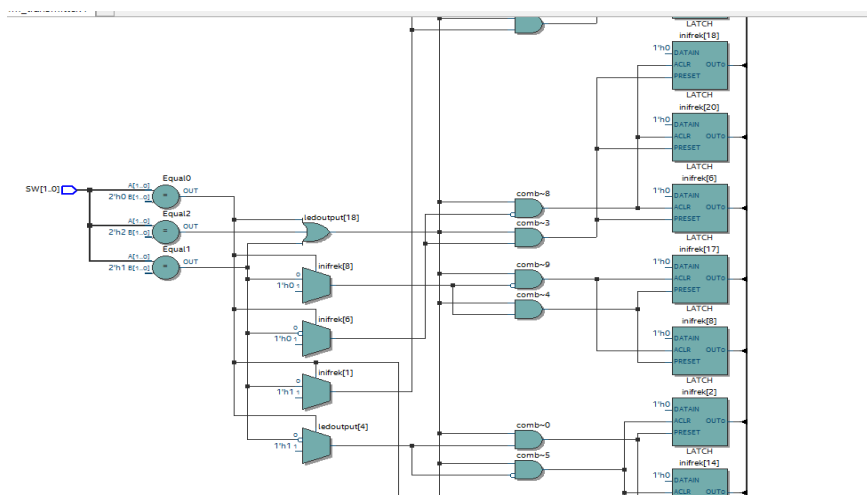
3.1.1 Hasil Simulasi Program

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 ditunjukkan pada gambar 4.



Gambar 4. 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 pada gambar 5.



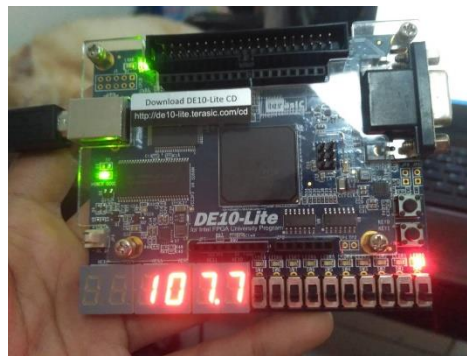
Gambar 5. Blok RTL pemilihan frekuensi

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'b00000000000000010000;
24   end
25 end
26
```

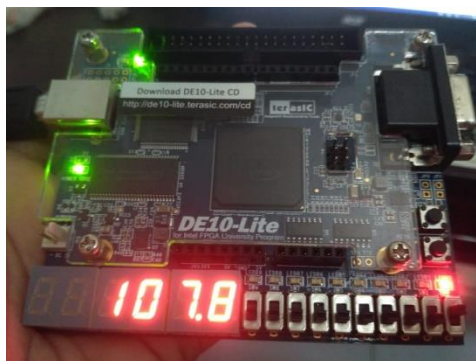
Gambar 6. Part pemrograman pemilihan frekuensi

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* menunjukan 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 7.



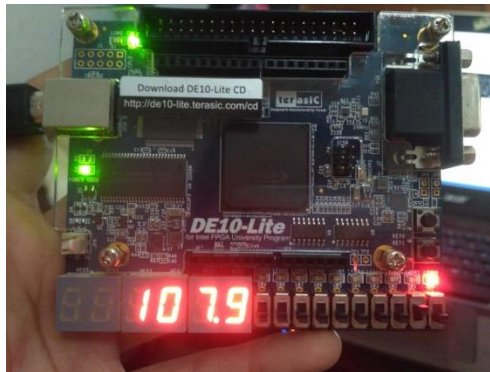
Gambar 7. Tampilan FPGA untuk frekuensi 107,7MHz

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 8.



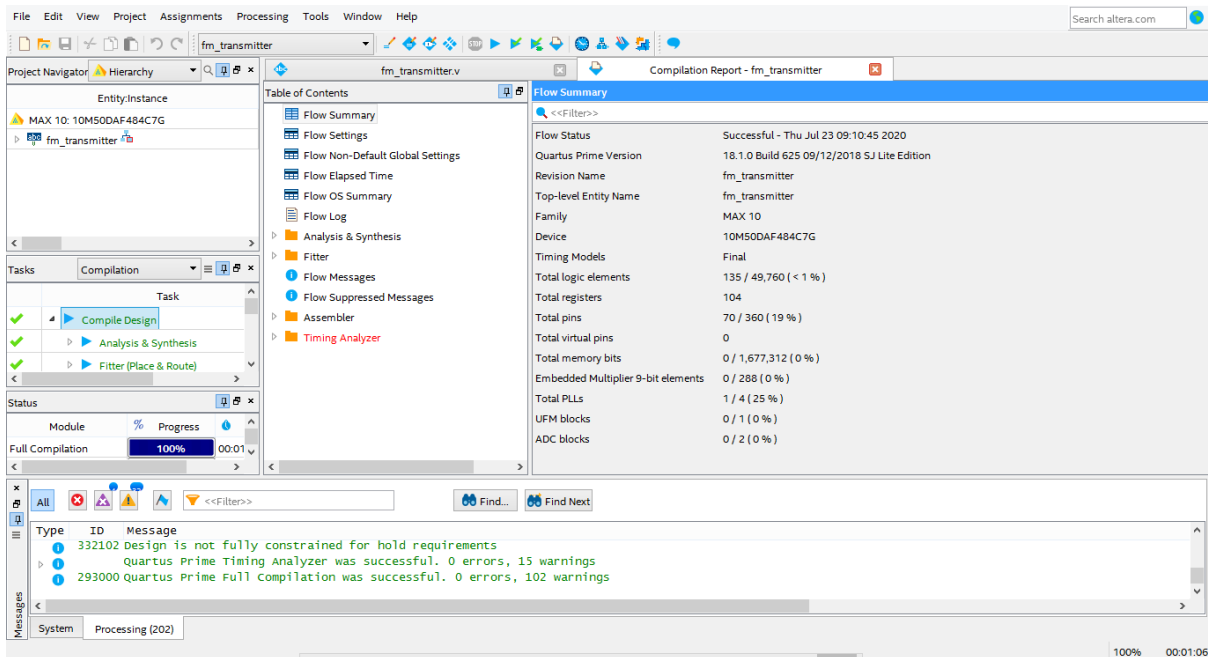
Gambar 8. Tampilan FPGA untuk frekuensi 107,8MHz

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



Gambar 9. Tampilan FPGA untuk frekuensi 107,9MHz

Sehingga setelah program dimasukkan 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 10.



Gambar 10. Hasil *synthesis* pada *software* Quartus Lite

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

Tabel 1. utilisasi FPGA

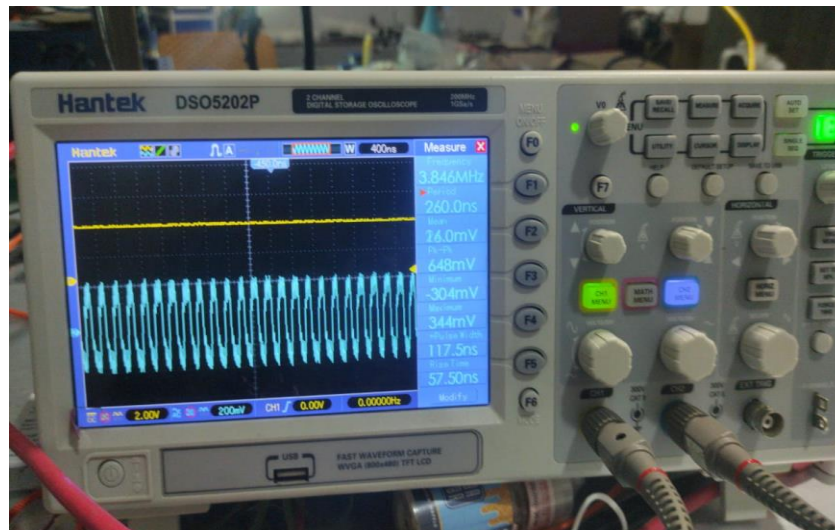
Total pin	70 dari 360	19%
Total <i>logic element</i>	135 dari 49,760	<1%
Total PLL	1 dari 4	25%

3.1.2 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

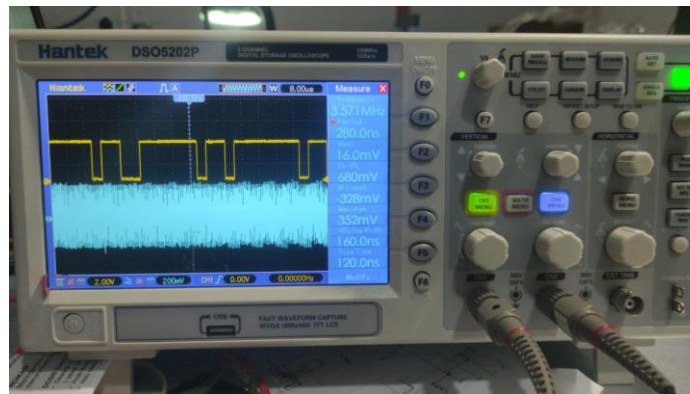
Pengukuran menggunakan osiloskop ini dimaksudkan untuk melihat tampilan sinyal dalam domain waktu. Pada pengukuran ini menggunakan dua *channel*, hal ini dimaksudkan untuk mendapatkan dua sinyal di kaki pin berbeda pada FPGA, *channel* 1 dihubungkan ke pin GPIO 0 untuk mendapatkan sinyal data serial yang dikirimkan, *channel* 2 dihubungkan ke pin GPIO 35 yang di set sebagai antenna untuk melihat sinyal FM. Dari proses tersebut, ditunjukkan pada gambar 11.



Gambar 11. 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 seperti ditunjukkan pada gambar 12..

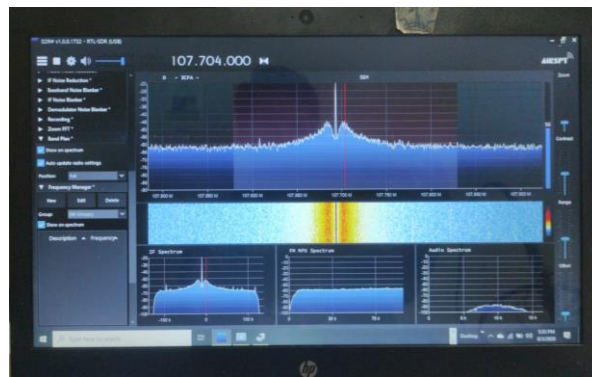
Pada gambar 12 di atas terlihat perbedaan dengan gambar sebelumnya 11, yaitu sudah ada data yang masuk sehingga tampilan sinyal pada *channel* 1 tidak lagi *flat* seperti sebelumnya, dan tampilan sinyal pada *channel* 2 menunjukkan perubahan, ini menandakan adanya sinyal yang berubah ketika data dikirim, walaupun rapat-renggang pada tampilan sinyal FM tidak terlalu terlihat. 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 .



Gambar 12. Cuplikan Sinyal pada osiloskop ketika data masuk

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 [9]. *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 *Spektrum Analyzer* untuk melihat sinyal FM dalam domain frekuensi. Hasil yang didapatkan ditunjukkan pada gambar 13.



Gambar 13. Tampilan Spektrum 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 14.



Gambar 14. Tampilan spektrum pada saat pengiriman data

Pada gambar di atas terlihat perbedaan dengan gambar sebelumnya yaitu 13. Dimana, pada gambar 14 terlihat perubahan spektrum yang melebar dari sebelumnya pada gambar 13, 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.

4. KESIMPULAN

4.1 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*.

4.2 Saran

Dalam penelitian ini masih banyak sekali kekurangan dan ketidaksempurnaan. Untuk itu, perlu dilakukan pengembangan agar ke depannya menjadi sempurna ataupun lebih baik lagi sehingga memiliki beberapa saran, diantaranya:

- 1) Tambahkan format *file audio* yang dapat dikirimkan seperti Mp3, RAW.
- 2) Gunakan antena 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.

REFERENSI

- [1] K. Harrison dan S. L. R. R. S. Warren Fletcher, "Direct Digital Synthesis," EEE4084F, South Africa, 2017.
- [2] J. Vankka, "Direct Digital Synthesizers : Theory, Design and Applications," *Helsinki University of Technology*, November 2000.
- [3] 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].
- [4] R. M. D. C. Martins, "Rancang bangun Pemancar FM Portable," 2013.
- [5] S. Twelves, "A Digital Stereo FM Modulator for Broadcasting," *The Open University (UK)*, July 1997.
- [6] Q. K. Aditya dan S. M. A. Dr.F. Yudi Limpraptono, "Rancang Bangun Pemancar Radio FM berbasis raspberry Pi," Institut Teknologi malang, Malang, 2019.
- [7] H. Bostrom, "An FPGA Implementation of a Digital FM Modulator," Linkoping University, Linkoping, Swedia, 2011.
- [8] R. Sopian, I. Hidayat, D. Darlis, "Pengujian Dan Verifikasi Desain Penerima FM Digital Pada FPGA," Institut Teknologi Telkom, Bandung, 2012.
- [9] K. Levgen, "Лабораторна робота №3 Схеми ділення і підвищення частоти, таймінг аналіз, цифровий FM передавач," Github, Ukraina, 2018.

- [10] Dzale, "FPGA Design Entry," Ndoware, 19 November 2009. [Online]. Available: <https://ndoware.com/fpga-design-entry.html> . [Diakses 25 November 2019].