

**DASAR TELEKOMUNIKASI  
DENGAN PENDEKATAN  
SOFTWARE *DEFINED* RADIO**

**RUSMI AMBARWATI  
SIGIT KUSMARYANTO  
FAKHRIY HARIO P.**



# DASAR TELEKOMUNIKASI DENGAN PENDEKATAN SOFTWARE *DEFINED RADIO*

© 2020

Penulis

**RUSMI AMBARWATI**

**SIGIT KUSMARYANTO**

**FAKHRIY HARIO P.**

Desain Cover dan Penata Isi

**Tim MNC Publishing**

Cetakan I, Desember 2020

**Diterbitkan oleh :**



**Media Nusa Creative**

Anggota IKAPI (162/JTI/2015)

Bukit Cemara Tidar H5 No. 34, Malang

Telp. : 0812 3334 0088

E-mail : [mncpublishing.layout@gmail.com](mailto:mncpublishing.layout@gmail.com)

Website : [www.mncpublishing.com](http://www.mncpublishing.com)

**ISBN 978-602-462-542-9**

Hak Cipta dilindungi undang-undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ke dalam bentuk apapun, secara elektronik maupun mekanis, termasuk fotokopi, merekam, atau dengan teknik perekaman lainnya, tanpa izin tertulis dari Penerbit. Undang-Undang Nomor 19 Tahun 2000 tentang Hak Cipta, Bab XII Ketentuan Pidana, Pasal 72, Ayat (1), (2), dan (6)

# PENGANTAR PENULIS

Dengan mengucapkan syukur Alhamdulillahirobbil 'aalamiin akhirnya bahan ajar Dasar Telekomunikasi (dengan pendekatan Software Defined Radio) dapat tersusun. Buku ini disusun dengan mengacu pada buku pedoman Teknik Elektro tahun akademik 2019/2020. Mengingat sangat pentingnya ilmu komunikasi dasar bagi mahasiswa teknik elektro, buku ini diharapkan dapat membantu mahasiswa yang sedang mengikuti mata kuliah Dasar Telekomunikasi.

Buku tersusun atas beberapa bab dengan bahasan Dasar dasar Telekomunikasi yang diimplementasikan dengan perangkat lunak MATLAB dan GNU Radio. Banyak contoh program matlab dan GNU Radio pada aplikasi telekomunikasi. Diharapkan akan muncul ide dan kreativitas, setelah Mahasiswa mencoba menjalankan contoh program dalam buku ajar ini.

Sebagai bahan ajar kami menyadari buku ini masih jauh dari sempurna, masukan dan saran untuk perbaikan buku ini, kami sampaikan terima kasih

Malang, 2020

Penyusun

MNC Publishing

# DAFTAR ISI

<b>PENGANTAR PENULIS .....</b>	<b>iii</b>
<b>DAFTAR ISI .....</b>	<b>v</b>
<b>DAFTAR GAMBAR .....</b>	<b>vi</b>
<b>DAFTAR TABEL .....</b>	<b>viii</b>
<b>BAB 1 SOFTWARE DEFINED RADIO .....</b>	<b>1</b>
1.1. PENDAHULUAN .....	2
1.2. APA ITU SDR ? .....	5
1.3. ARSITEKTUR SDR .....	10
<b>BAB 2 KONSEP TELEKOMUNIKASI .....</b>	<b>17</b>
2.1. SINYAL KOMPLEKS .....	18
2.2. SISTEM TELEKOMUNIKASI .....	19
2.3. SISTEM KOMUNIKASI RADIO .....	38
2.4. GANGGUAN DALAM SISTEM TELEKOMUNIKASI .....	40
<b>BAB 3 MODULASI AM .....</b>	<b>45</b>
3.1. MODULASI DAN DEMODULASI .....	46
<b>BAB 4 MODULASI FM .....</b>	<b>59</b>
<b>BAB 5 PHASE SHIFT KEYING (PSK) .....</b>	<b>67</b>
5.1. MODULASI DIGITAL .....	68
5.2. BINARY PHASE SHIFT KEYING (BPSK) .....	71
<b>BAB 6 GNU RADIO DAN APLIKASINYA .....</b>	<b>75</b>
<b>LAMPIRAN: STUDI KASUS .....</b>	<b>93</b>
<b>DAFTAR PUSTAKA .....</b>	<b>115</b>

## DAFTAR GAMBAR

Gambar 1.1. Model OSI .....	7
Gambar 1.2. Tingkatan pada Definisi SDR menurut SDR Forum	8
Gambar 1.3. Arsitektur SDR .....	10
Gambar 1.4. Desain SDR .....	10
Gambar 1.5. USRPP beserta daughterboard .....	12
Gambar 1.6. Diagram blok USRP .....	13
Gambar 1.7. Diagram Blok Penerima Superheterodyne .....	14
Gambar 1.8. Diagram Blok Penerima SDR .....	15
Gambar 1.9. Struktur Pemancar SDR .....	16
Gambar 2.1. Komponen I dan Q Pada Sinyal Kompleks .....	18
Gambar 2.2. Pembangkitan Sinyal Kompleks .....	18
Gambar 2.3. Sinyal Pulsa $s(t)$ .....	23
Gambar 2.4. Spektrum Frekuensi $S(f)$ dari Sinyal Pulsa $S(t)$ .....	24
Gambar 2.5. Elemen Sistem Komunikasi .....	25
Gambar 2.6. Oscillator yang Menghasilkan Sinyal Sinusoida dengan frekuensi $f_0$ dan fasa $\varphi$ .....	26
Gambar 2.7. Sinyal Sinusoida Frekuensi $f = 10$ Hz dan fasa $\varphi = 0$	27
Gambar 2.8. Spektrum Sinyal Input dan Output Filter .....	29
Gambar 2.9. Proses Sampling .....	30
Gambar 2.10. Perkalian Sinyal Analog dengan Deretan Impuls .	31
Gambar 2.11. Sampling Sinyal di Kawasan Waktu .....	33
Gambar 2.12. Sampling Sinyal di Kawasan Frekuensi .....	33
Gambar 2.13. Rekoverti Sinyal di Kawasan Frekuensi .....	34
Gambar 2.14. Proses Sampling di Kawasan Waktu .....	35
Gambar 2.15. Proses Sampling di Kawasan Frekuensi .....	36
Gambar 2.16. Proses Pencampuran Sinyal (Mixer/Multiplier) ...	37
Gambar 2.17. Direct-conversion Transmitter .....	39
Gambar 2.18. Penerima Superheterodyne .....	40
Gambar 3.1. Struktur Pemancar AM .....	47
Gambar 3.2. Bentuk Sinyal Pada Modulasi AM .....	47
Gambar 3.3. Sinyal Modulasi AM-DSB-SC .....	48
Gambar 3.4. Model Simulasi DSB-SC menggunakan Simulink ...	49

Gambar 3.5. Sinyal Termodulasi DSB-SC .....	49
Gambar 3.6. Spektrum Frekuensi Sinyal Termodulasi DSB-SC ...	50
Gambar 3.7. Sinyal Modulasi AM-DSB-FC .....	51
Gambar 3.8. Model Simulasi DSB-FC menggunakan Simulink ...	51
Gambar 3.9. Sinyal Termodulasi DSB-FC .....	52
Gambar 3.10. Spektrum Frekuensi Sinyal Termodulasi DSB-FC .	52
Gambar 3.11. Penerima AM .....	53
Gambar 3.12. Synchronous Detector .....	53
Gambar 3.13. Sinyal Informasi dan Keluaran LPF .....	54
Gambar 3.14. Sinyal informasi dan keluaran demodulator .....	55
Gambar 3.15. Spektrum frekuensi sinyal keluaran demodulator	55
Gambar 3.16. Model penerima AM dengan noise .....	56
Gambar 3.17. Keluaran BPF dengan noise .....	56
Gambar 3.18. Spektrum keluaran sinyal dengan noise .....	57
Gambar 3.19. Sinyal keluaran dengan noise .....	57
Gambar 4.1. Model simulasi modulator FM .....	61
Gambar 4.2. Nilai parameter simulasi .....	61
Gambar 4.3. Sinyal termodulasi FM .....	62
Gambar 4.4. Model Simulink demodulator FM .....	62
Gambar 4.5 Parameter simulasi demodulator FM .....	62
Gambar 4.6. Sinyal keluaran demodulator FM .....	63
Gambar 4.7. Model Simulink untuk modulasi FM dengan noise	63
Gambar 4.8. keluaran demodulator FM dengan noise .....	64
Gambar 4.9. sinyal keluaran demodulator dengan noise .....	65
Gambar 5.1. Diagram blok sistem komunikasi digital .....	68
Gambar 5.2. Model Simulink untuk simulasi BPSK .....	71
Gambar 5.3. parameter simulasi .....	71
Gambar 5.4. Konstelasi sinyal BPSK .....	72
Gambar 5.5. Grafik nilai BER terhadap Eb/No .....	72
Gambar 5.6. Konstelasi sinyal BPSK dengan noise .....	73
Gambar 5.7. Grafik pola mata sinyal BPSK tanpa error .....	74
Gambar 5.8. Grafik pola mata sinyal BPSK dengan error.....	74
Gambar 6.1. Diagram blok penggunaan SDR untuk DVB-T2 .....	77
Gambar 6.2. Spektrum frekuensi kanal 22 .....	91

## DAFTAR TABEL

Tabel 1. Definisi Bertingkat dari SDR Forum .....	8
---	---

MNC Publishing





# **BAB 1**

## **SOFTWARE DEFINED RADIO**

MNC Publishing

## 1.1. PENDAHULUAN

Dalam mempelajari dasar-dasar telekomunikasi di perguruan tinggi, umumnya dilakukan melalui platform analog yaitu rangkaian dengan komponen analog. Mahasiswa bertanggung jawab merangkai/ membuat rangkaian untuk mengimplementasikan suatu teknik komunikasi tertentu. Rangkaian tersebut selanjutnya dievaluasi menggunakan perangkat seperti *function generator*, *oscilloscope*, dan *spectrum analyzer*. Mahasiswa diminta untuk melaporkan hasil pengukuran dan mengambil kesimpulan tentang eksperimen yang dibuat. Semua tugas tersebut dilakukan dalam interval waktu yang pendek serta dengan membatasi kompleksitas eksperimen.

Kegagalan yang disebabkan oleh kesalahan perakitan komponen, penggunaan komponen yang cacat, serta penggunaan perangkat yang kurang tepat dapat mengganggu keseluruhan eksperimen, menunda implementasinya, dan selanjutnya menyebabkan proses belajar menjadi sulit karena mahasiswa harus menghabiskan banyak waktu dalam menyelesaikan masalah yang timbul dan bukan menggunakan waktunya secara efektif untuk memahami konsep yang terkandung dalam eksperimen.

Contoh serupa terjadi pada pembelajaran materi rangkaian logika. Beberapa tahun yang lalu, pembelajaran rangkaian logika dilakukan melalui penggunaan protoboard dan IC logic. Setelah mendapatkan penjelasan singkat tentang eksperimen, mahasiswa akan menghabiskan banyak waktu untuk memotong dan menghubungkan kabel tembaga dari satu gerbang logika (*logic gate*) ke gerbang logika yang lain untuk mengimplementasikan proyek dalam eksperimen yang dimaksud. Dengan adanya perkembangan teknologi dalam devais logic yang dapat diprogram, penggunaan protoboard dan IC menjadi ketinggalan jaman dan semakin ditinggalkan dari agenda pengajaran. Sekarang ini, laboratorium rangkaian logika telah menggunakan development kits dengan hardware yang dapat diprogram seperti FPGA dan CPLD dari manufaktur seperti Altera dan Xilinx. Mahasiswa tidak perlu lagi menghabiskan banyak waktu untuk membuat koneksi kabel yang rentan terhadap kegagalan dan kesalahan modifikasi. Hal yang sama

dapat diterapkan pada kelas laboratorium telekomunikasi, dimana pendekatannya dilakukan dengan menggunakan software yang didukung oleh model komunikasi yang disebut *Software Defined Radio* (SDR). Pendekatan ini memungkinkan mahasiswa untuk memfokuskan usahanya pada aspek-aspek yang lebih relevan dengan desain sistem komunikasi dan mengurangi tantangan terhadap kesalahan perakitan dan validasi.

Prinsip-prinsip dasar telekomunikasi pada dasarnya tidak banyak mengalami perubahan sejak masa Shannon. Apa yang telah berubah dan masih mengalami perubahan adalah bagaimana prinsip tersebut diterapkan dalam teknologi. Salah satunya adalah adanya pergeseran dari penggunaan perangkat keras (*hardware*) menjadi teknologi berbasis perangkat lunak (*software*).

Yang dimaksud dengan 'radio' dalam teknologi SDR adalah representasi dari transmisi melalui udara, seperti televisi, telepon seluler, atau komunikasi data wireless. 'Software Defined' maksudnya bahwa elemen inti pada sistem radio tersebut diimplementasikan menggunakan software. Dalam buku ini, konsep yang mendasari transmisi radio tersebut akan dijelaskan dan ditunjukkan melalui simulasi menggunakan Matlab.

Dalam dunia pendidikan terdapat filosofi 'learning by doing', yang dimaksudkan untuk memotivasi belajar melalui eksperimen, memperkuat pemahaman matematika melalui contoh-contoh simulasi, serta mengintegrasikan konsep dengan penjelasan dan analisa pada tiap bagian sistem. Memakai filosofi tersebut, pembahasan dalam buku ini dilakukan melalui pemahaman masing-masing komponen dari sistem transmisi, kemudian menggabungkan semua komponen ke dalam satu sistem pemancar atau penerima. Buku ini mencoba menunjukkan inti dari tiap komponen dalam bentuk sesederhana mungkin. Performansi sistem akan ditunjukkan melalui analisa dan simulasi. Tujuannya untuk memberikan pemahaman mengenai suatu sistem transmisi komunikasi yang lengkap. Dengan memahami dan menguasai prinsip dasar tersebut, akan bermanfaat dalam penguasaan berbagai aplikasi sistem komunikasi selain perancangan pemancar/ penerima.

Meskipun jauh dari optimal, perancangan pemancar/penerima yang dilakukan dalam buku ini terdiri dari semua elemen yang berfungsi lengkap. Langkah-langkah yang dilakukan menyediakan cara sederhana dalam menjawab berbagai pertanyaan 'bagaimana jika'. Bagaimana jika terdapat noise dalam sistem ? Bagaimana jika frekuensi modulasinya tidak ditetapkan secara tepat ? Bagaimana jika terdapat error dalam bit yang diterima ? Bagaimana jika kecepatan datanya tidak cukup tinggi ? Bagaimana jika terdapat distorsi dalam kanal transmisi ? Teknik simulasi menyediakan fasilitas dalam menjawab berbagai pertanyaan tersebut.

Platform SDR telah digunakan secara luas dalam bidang sains. Fleksibilitasnya memungkinkan berbagai penelitian dalam bidang telekomunikasi dilakukan, sebagai contoh pada *cognitive radio*, modulasi OFDM, GSM, dan sebagainya. Karena satu hardware yang sama dapat digunakan untuk penerapan aplikasi radio yang berbeda, maka SDR menjadi satu solusi menarik untuk digunakan dalam laboratorium telekomunikasi. Berbagai devais lain yang sebelumnya diperlukan seperti diode, kapasitor, dan bahkan perangkat seperti oscilloscope dapat digantikan oleh satu platform hardware. Meskipun demikian, hal tersebut tidak berarti bahwa mempelajari bagaimana penggunaan komponen-komponen tersebut tidak penting.

Berdasarkan penjelasan di atas, terdapat dua keuntungan dalam penggunaan SDR :

1. Mahasiswa dapat menggunakan waktunya untuk lebih banyak memahami konsep dan desain dalam sistem komunikasi
2. Mahasiswa akan menghabiskan lebih sedikit waktu untuk melakukan set-up perangkat eksperimen.

Yang tidak kalah penting dari penggunaan SDR ini adalah pengurangan biaya yang diperlukan dalam menyiapkan sebuah laboratorium telekomunikasi, karena perangkat yang diperlukan minimal hanya terdiri dari sebuah PC dan board SDR yang sesuai untuk masing-masing kelompok.

## 1.2. APA ITU SDR ?

Selama beberapa puluh tahun terakhir ketika teknologi semikonduktor semakin berkembang dalam hal performansi dan kemampuannya, teknologi radio yang baru juga telah berkembang. Salah satunya adalah *Software Defined Radio*. Meskipun sudah melalui berbagai diskusi dalam forum ilmiah, definisi yang diterima semua pihak masih sulit dibuat. Hal ini disebabkan oleh sifat fleksibilitas yang dimiliki oleh SDR, yang memungkinkan SDR mempunyai berbagai macam bentuk yang berbeda sesuai dengan kebutuhan. Dapat dikatakan bahwa SDR adalah : Teknologi radio dimana beberapa atau semua fungsi pada layer fisik didefinisikan/ dilaksanakan melalui software.

Dengan kata lain, *Software Defined Radio* (SDR) merupakan teknologi komunikasi radio yang didasarkan pada protokol komunikasi wireless yang didefinisikan melalui software sebagai pengganti implementasi hardware. Fungsi dan protokol *air interface*, serta bandwidth frekuensinya dapat di-*upgrade* melalui software tanpa perlu mengganti perangkat kerasnya (hardware).

Ide utama dari SDR adalah melakukan transfer tugas yang semula dilakukan melalui perangkat keras (*hardware*) diganti melalui perangkat lunak (*software*). Berbagai karakteristik sistem seperti skema modulasi sinyal, frekuensi kerja, bandwidth, dan sebagainya tidak lagi tergantung pada rangkaian analog yang umumnya didefinisikan terlebih dahulu dalam perangkat radio konvensional. Dalam SDR, semua karakteristik tersebut tergantung pada sistem yang mengintegrasikan software dan hardware yang dapat diprogram, sehingga menyediakan fleksibilitas dalam melakukan modifikasi. Dengan radio semacam ini, memungkinkan untuk melakukan perubahan fitur sistem hanya melalui software dengan cara merubah parameter yang berkaitan dengan definisi perilaku pada fitur yang dirubah. Perubahan tersebut bahkan dapat dilakukan pada waktu sistem sedang beroperasi. Selain itu, memungkinkan juga membuat suatu sistem komunikasi yang sama sekali berbeda menggunakan perangkat SDR yang sama. Hal tersebut dapat dilakukan hanya dengan mengganti software yang dieksekusi pada hardware yang sama. Sebagai contoh, dengan

menggunakan platform yang sama misalnya USRP, kita dapat membuat berbagai aplikasi radio yang berbeda mulai dari penerima FM sampai pemancar TV digital yang kompleks.

Ide mengenai implementasi fungsi radio ke dalam bentuk perangkat lunak (software) telah dimulai sejak 1991 ketika Josep Mitola mengenalkan istilah 'software Radio'. Teknologi SDR pada awalnya digunakan sebagai cara untuk memfasilitasi komunikasi yang lebih baik antar berbagai kesatuan dalam miliar AS. SDR dilihat sebagai suatu teknologi yang dapat menjembatani komunikasi antar beberapa sistem radio yang berbeda pada beberapa kesatuan tersebut. Tujuannya adalah untuk mengurangi perbedaan sistem yang digunakan.

## Definisi

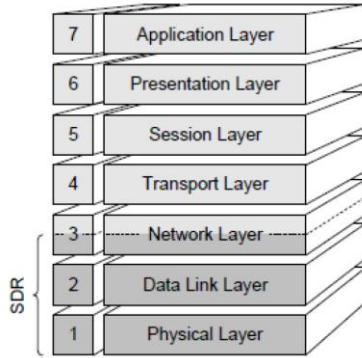
*Software Defined Radio* (SDR) merupakan teknologi komunikasi radio yang didasarkan pada protokol komunikasi dengan perangkat lunak sebagai pengganti implementasi hardware. Bandwidth frekuensi, protokol dan fungsi air interface dapat di-upgrade melalui pembaruan (update) dan download software tanpa perlu mengganti perangkat keras sistem.

Ide utamanya adalah bagaimana fleksibilitas gelombang radio dapat dirubah melalui mekanisme perubahan software tanpa perlu memodifikasi platform SDR. SDR dapat diprogram dan dikonfigurasi ulang supaya dapat beroperasi dengan bentuk gelombang dan protokol yang berbeda. Bentuk gelombang dan protokol tersebut dapat berisi komponen yang berbeda, termasuk teknik modulasi, keamanan, dan karakteristik performansi yang didefinisikan dalam perangkat lunak sebagai bagian dari bentuk gelombang itu sendiri.

SDR merupakan sistem komunikasi yang melaksanakan beberapa tugas pemrosesan sinyal yang diperlukan di dalam DSP engine. Perangkat SDR memprogram ulang segmen DSP pada physical layer untuk mengkonfigurasi berbagai parameter sistem radio.

Perhatikan model referensi Open System pada gambar 1. Secara umum SDR didefinisikan pada fungsi-fungsi di dalam

physical layer dan data link layer, serta beberapa bagian dari network layer. Sedangkan fungsi-fungsi pada layer di atasnya secara spesifik bukan bagian dari SDR dan tidak diklasifikasikan sebagai SDR.



Gambar 1.1. Model OSI

Forum SDR yang merupakan korporasi non-profit dibentuk untuk mendukung pengembangan, penerapan, dan penggunaan arsitektur terbuka dalam bidang sistem wireless. Forum SDR telah mengembangkan suatu definisi bertingkat tentang SDR. Konsep lima tingkat dirangkum seperti pada tabel 1 berikut :

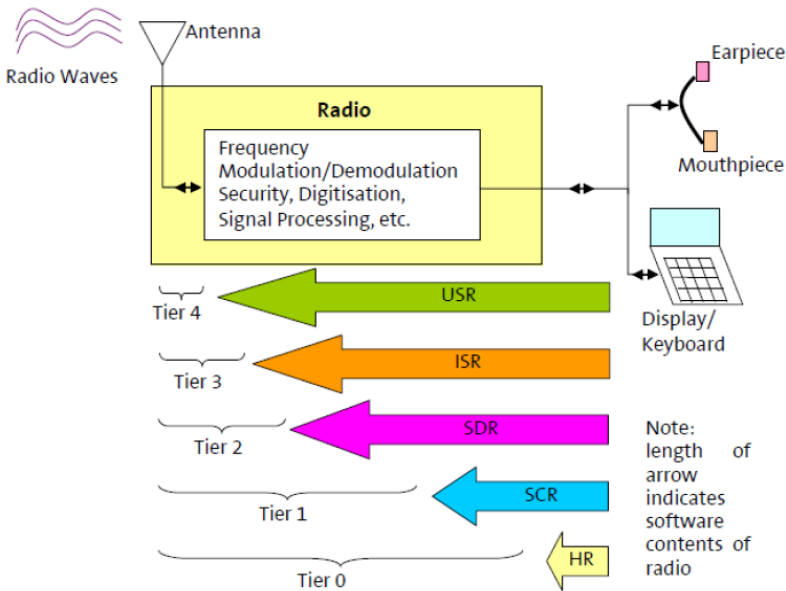
TINGKAT	NAMA	PENJELASAN
0	Hardware Radio	Baseline radio dengan fungsi-fungsi tetap (fixed)
1	Software Controlled Radio	Radio dimana beberapa atau semua fungsi dalam physical layer dikontrol menggunakan software. Jalur sinyal radio diimplementasikan menggunakan perangkat keras yang khusus untuk aplikasi tertentu ( application specific hardware)
2	Software Defined Radio	Radio dimana beberapa atau semua fungsi dalam physical layer didefinisikan menggunakan software
3	Ideal Software Radio	Radio yang mengimplementasikan lebih banyak jalur sinyal dalam domain digital

4	Ultimate Software Radio	Representasi dari visi “blu-sky” : menerima trafik dan informasi kontrol yang terprogram secara keseluruhan, mendukung operasi pada cakupan frekuensi yang lebar, dan dapat berpindah antar ‘air-interface’/ aplikasi dalam skala milliseconds.
---	-------------------------	---

Tabel 1. Definisi bertingkat dari SDR forum

Sumber : “An Evaluation of Software Defined Radio” [20].

Gambar 1.2 menunjukkan penjelasan abstrak tentang definisi bertingkat tersebut, dimana panjang panah mengindikasikan proporsi dari konten software di dalam aplikasi radio.



Gambar 1.2. Tingkatan pada definisi SDR menurut SDR forum

### Aspek Umum SDR

Beberapa karakteristik yang terdapat pada SDR :

**Multiband** : sebagian besar arsitektur radio tradisional beroperasi pada bandwidth frekuensi tunggal, tetapi terdapat beberapa aplikasi dimana diperlukan beberapa bandwidth frekuensi. Karena itu



digunakan bandwidth jamak; tiap bandwidth didesain untuk beroperasi pada bandwidth tertentu (multiband radio). Multiband radio mempunyai kemampuan untuk beroperasi pada dua atau lebih bandwidth secara sekuensial atau secara simultan, seperti dalam kasus BTS yang bisa terhubung ke perangkat mobile dari beberapa bandwidth yang berbeda.

**Multicarrier** : multicarrier/ multichannel radio mempunyai kemampuan untuk beroperasi secara simultan pada lebih dari satu frekuensi pada saat yang bersamaan. Hal ini bisa dilakukan di dalam bandwidth yang sama, atau dalam dua bandwidth yang berbeda pada waktu yang sama. Yang sering terjadi, multicarrier diaplikasikan pada base station yang bisa melayani banyak user pada saat yang sama dan juga dapat diaplikasikan pada terminal user dapat memproses voice dan data bersamaan pada frekuensi carrier yang berbeda.

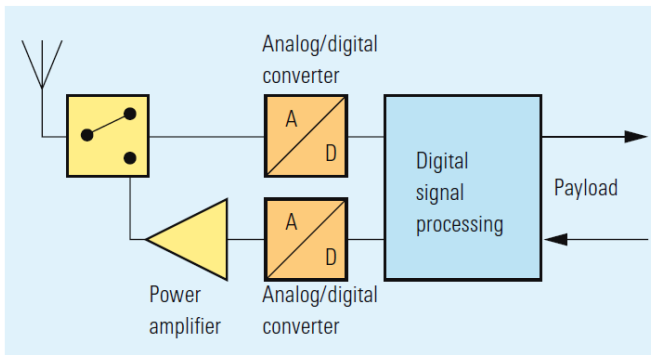
**Multimode** : multimode berarti kemampuan untuk memproses beberapa jenis standar yang berbeda ( AM, FM, GMSK, dan CDMA). Mode atau standar tersebut dapat diproses secara sekuensial atau simultan.

**Multirate** : multirate berkaitan erat dengan multimode. Suatu multirate radio memproses bagian-bagian yang berbeda pada rantai sinyal menggunakan kecepatan sampling yang berbeda-beda. Atau bisa jua berarti radio yang mempunyai kemampuan untuk memproses mode/standar yang berbeda yang memerlukan data rate yang berbeda.

**Variable bandwidth** : radio tradisional menentukan bandwidth kanal menggunakan filter analog yang tetap (fixed). Pada SDR, bandwidth kanal ditentukan menggunakan filter digital yang dapat diubah-ubah. Desain Filter digital dapat digunakan untuk mengimplementasikan sejumlah filter yang tidak mungkin diaplikasikan dalam domain analog. Filter digital juga dapat disesuaikan untuk beradaptasi pada lingkungan yang mengalami interferensi dan mengkompensasi distorsi yang terjadi pada jalur transmisi.

### 1.3. ARSITEKTUR SDR

Arsitektur ideal dari SDR seperti ditunjukkan pada gambar 1.3.

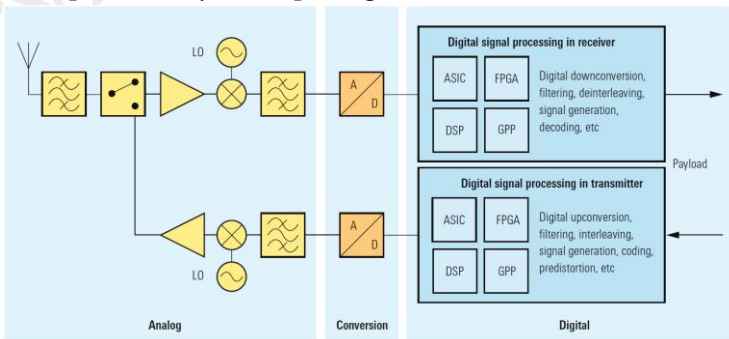


Gambar 1.3. Arsitektur SDR

Arsitektur SDR tersebut mempunyai fitur sebagai berikut :

- Skema modulasi, kanalisasi, protokol, dan equalisasi untuk pengiriman (transmit) dan penerimaan (receive) semua dilakukan secara software dalam blok DSP (digital signal processing).
- *Circulator* (duplexer) digunakan untuk memisahkan jalur antara sinyal transmit dan receive. Komponen ini mengasumsikan adanya matching ideal dengan antena dan PA (*Power Amplifier*).
- Power amplifier memastikan adanya proses modulasi RF dari DAC menjadi frekuensi tinggi yang sesuai untuk transmisi.
- Anti-aliasing filter (tidak ditunjukkan pada gambar 1.3).

Komponen SDR dapat dibagi menjadi bagian analog dan bagian digital, seperti ditunjukkan pada gambar 1.4.



Gambar 1.4. Desain SDR

Komponen analog terdiri dari antenna, prefilter, switch (*circulator*), preamplifier, dan power amplifier. Bahkan dalam era teknologi digital ini, komponen analog tersebut masih berperan penting dalam menentukan karakteristik teknis dari sistem radio yang dibuat.

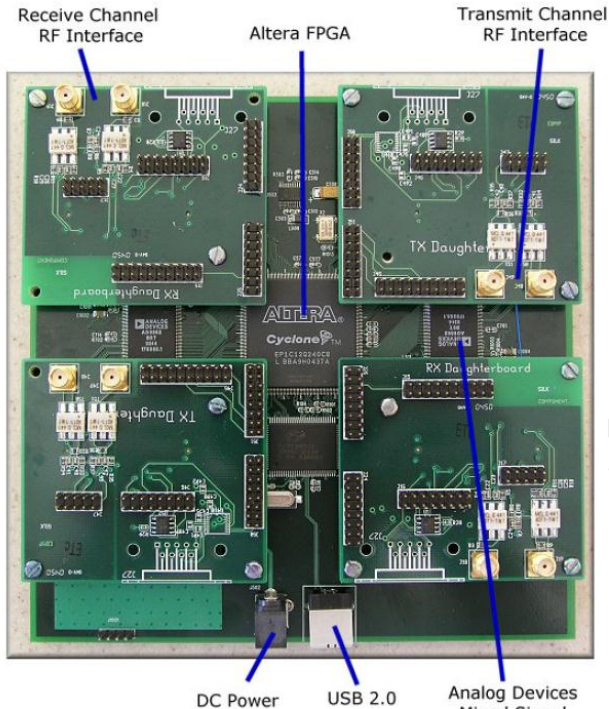
Sedangkan komponen utama pada bagian digital bisa berupa hardware berikut :

- ◆ GPPs (general-purpose processors)
- ◆ DSPs (digital signal processors)
- ◆ FPGAs (field-programmable gate arrays)
- ◆ ASICs (application-specific integrated circuits)

Komponen hardware tersebut digunakan untuk mengimplementasikan berbagai fungsi dalam sistem radio, seperti modulasi, demodulasi, pemfilteran, dan pengkodean.

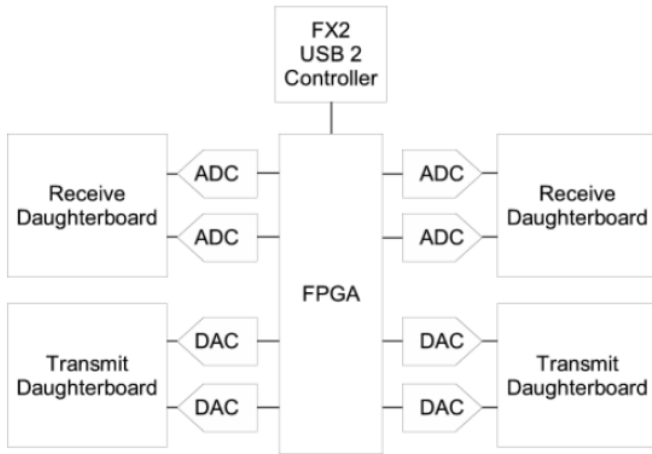
### **USRP (*Universal Software Radio Peripheral*)**

Salah satu platform yang banyak digunakan dalam implementasi SDR adalah USRP yang diproduksi oleh Ettus Research. USRP didesain untuk melakukan fungsi software radio dengan bandwidth tinggi. Pada dasarnya, USRP beroperasi pada bagian *digital baseband* dan IF dari sistem komunikasi radio. USRP dilengkapi dengan interface *RF front-end* melalui *daughterboards* yang menghubungkan antara baseband atau IF ke bandwidth RF sesuai aplikasi yang dibuat. USRP didesain untuk melakukan pemrosesan khusus seperti modulasi/demodulasi pada host CPU. Semua operasi yang bersifat umum seperti DUC (*Digital Up Converter*), DDC (*Digital Down Converter*), decimation, dan interpolation semua dilakukan pada FPGA. Bagi pengguna yang ingin melakukan proses selain proses standar tersebut, dapat melakukan penggantian pada desain FPGA standar. Gambar 1.5 berikut adalah board USRP beserta daughterboard, terdiri dari 2 port kanal *transmit* dan 2 port kanal *receive*.



Gambar 1.5. USRP beserta daughterboard

USRP mempunyai 4 buah ADC yang beroperasi pada 12 bit/sample dan 64 Msample/sec. USRP juga mempunyai 4 DAC yang beroperasi pada 14 bit/sample dan 128 Msample/sec. Empat kanal input dan empat kanal output ini dihubungkan ke FPGA Altera Cyclone EP1C12. Selanjutnya, FPGA terkoneksi ke chip interface USB2, yaitu Cypress FX2 dan seterusnya ke komputer. Sehingga USRP terkoneksi ke komputer hanya melalui interface USB2 yang berkecepatan tinggi. Diagram blok USRP ditunjukkan pada gambar 1.6.



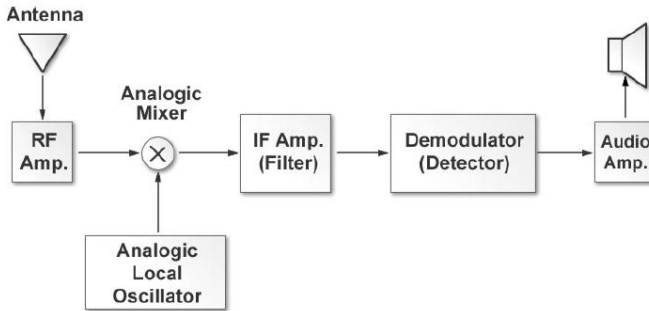
Gambar 1.6. Diagram blok USRP

USRP dapat beroperasi dalam mode *full duplex*, dimana sisi pengirim dan penerima bekerja secara independen. Satu-satunya yang harus diperhatikan adalah kecepatan gabungan dari pengirim dan penerima harus sama dengan atau kurang dari 32 Mbyte/sec. Semua kanal RX harus beroperasi pada bit rate yang sama (*decimation ratio* yang sama). Begitu juga semua kanal TX harus beroperasi pada bit rate yang sama, tetapi bisa berbeda dengan bit rate RX.

#### 1.4. SDR RECEIVER

Sebelum membahas tentang sistem penerima (*receiver*) pada SDR, mari kita bahas dulu sistem penerima pada sistem komunikasi tradisional. Selain melakukan proses demodulasi, pada receiver juga terjadi operasi : (1). Tuning frekuensi carrier untuk memilih sinyal yang diinginkan, (2). Filter untuk memisahkan sinyal yang dipilih dari sinyal lain, (3). Amplifikasi untuk mengkompensasi *losses* pada transmisi.

Sebagian besar receiver menggunakan skema penerima *superheterodyne* seperti ditunjukkan pada gambar 1.7. Pemahaman mengenai struktur penerima *superheterodyne* akan sangat membantu dalam memahami konsep penerima dari sudut pandang SDR.



Gambar 1.7. Diagram blok penerima *superheterodyne*

Dalam skema *superheterodyne*, sinyal yang masuk melalui antenna akan diperkuat oleh RF amplifier yang hanya bekerja pada frekuensi yang ditetapkan. Selanjutnya sinyal dilewatkan ke mixer yang akan merubah frekuensi sinyal ke dalam frekuensi IF (*Intermediate Frequency*). Proses ini dilakukan menggunakan *local oscillator* dengan frekuensi yang ditetapkan untuk memastikan perbedaan sinyal yang dihasilkan sama dengan frekuensi IF. Sebagai contoh, jika kita ingin menerima sinyal pada frekuensi 100,7 MHz dan frekuensi IF yang digunakan adalah 10,7 MHz, maka *local oscillator* harus ditempatkan pada frekuensi 90 MHz. Operasi ini disebut dengan *downconversion*.

Tahap berikutnya adalah *bandpass filter* yang akan melewatkan sinyal dengan spektrum frekuensi yang diinginkan. Bandwidth filter pada tahap ini akan membatasi bandwidth sinyal yang diterima. Frekuensi IF yang umum digunakan adalah 455 kHz untuk AM dan 10,7 MHz untuk FM.

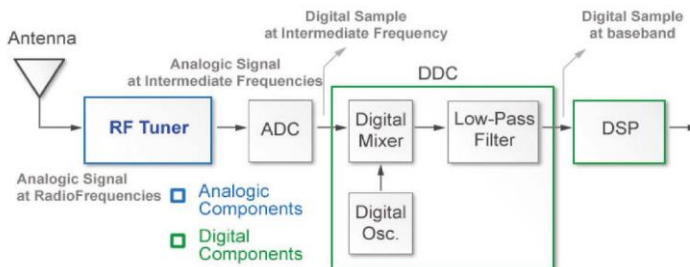
Terakhir, sinyal akan dilewatkan proses demodulator yang akan mengembalikan sinyal informasi. Misalnya pada demodulasi AM menggunakan *envelope detector*, sedangkan pada FM menggunakan *frequency discriminator*.

Berbeda dengan skema penerima *superheterodyne* di atas, struktur penerima pada SDR dapat dilihat pada gambar 1.8. Sinyal yang diterima melalui antenna akan masuk ke blok RF tuner. Blok ini melakukan proses konversi sinyal analog yang diterima menjadi sinyal dengan frekuensi IF. Pada dasarnya, RF tuner melakukan

proses seperti pada tiga blok proses di penerima *superheterodyne*. Berikutnya, sinyal IF yang dihasilkan akan dilewatkan blok ADC yang akan mengkonversi sinyal analog menjadi deretan sampel digital. Deretan sampel sinyal ini selanjutnya akan masuk ke blok DDC (*Digital Down Converter*). DDC umumnya merupakan sebuah chip monolitik yang merupakan bagian kunci dalam sistem SDR. DDC terdiri dari tiga komponen utama :

- Digital mixer
- Digital local oscillator
- FIR lowpass filter

Operasi pada komponen-komponen tersebut serupa dengan yang dilakukan pada bagian analog. Digital mixer dan local oscillator akan menggeser sampel digital frekuensi IF menjadi sinyal baseband. Sedangkan FIR lowpass filter akan membatasi bandwidth sinyal yang dihasilkan. Untuk implementasinya, di dalam DDC terdapat sejumlah besar multiplier, adder, dan shift register.



Gambar 1.8. Diagram blok penerima SDR

Prosedur lain yang disebut dengan decimation umumnya dilakukan untuk mengurangi frekuensi sampling (sample rate). Sebagai hasilnya, diperoleh frekuensi sampling yang baru pada sinyal baseband. Proses decimation dilakukan dengan membagi frekuensi sampling dengan faktor N yang disebut dengan *decimation factor*. Besarnya sample rate yang dihasilkan bisa menjadi dua kali komponen frekuensi tertinggi dari sinyal, yang sesuai dengan teorema Nyquist. Pendekatan praktis menunjukkan bahwa pengurangan sample rate tersebut dapat diaplikasikan sampai 20% tanpa mempengaruhi kualitas sinyal yang dihasilkan. Hal ini dinyatakan secara numerik melalui persamaan :

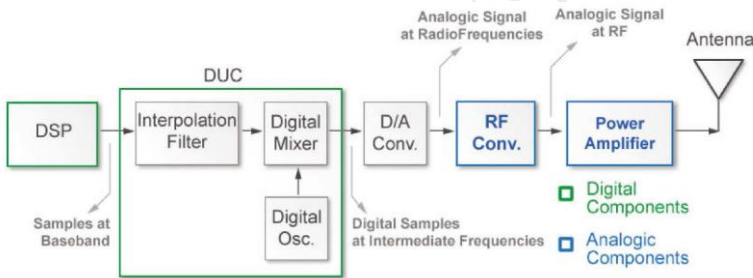
$$f_{b2} = 0,8 f_b = \frac{f_s}{N}$$

dimana  $f_b$  adalah frekuensi sinyal baseband,  $f_s$  adalah frekuensi sampling,  $N$  adalah decimation faktor, dan  $f_{b2}$  adalah frekuensi baseband baru yang dihitung setelah proses decimation.

Sebagai tahap akhir, deretan sampel baseband tersebut akan masuk ke blok DSP (*Digital Signal Processing*) yang akan melakukan proses seperti demodulasi dan decoding.

### 1.5. SDR TRANSMITTER

Sebagian besar devais SDR adalah berupa receiver. Pemancar SDR (*transmitter*) juga tersedia meskipun dengan harga yang lebih mahal. Struktur pemancar SDR dalam bentuk diagram blok ditunjukkan pada gambar 1.9.



Gambar 1.9. Struktur pemancar SDR

Pemancar menerima sebuah sinyal baseband yang dibangkitkan oleh proses pengolahan sinyal pada DSP. Sinyal baseband tersebut masuk ke blok DUC yang akan merubah frekuensi sinyal menjadi IF. Berikutnya sinyal IF tersebut masuk proses DAC yang akan mengkonversi menjadi sinyal analog. Selanjutnya sinyal analog dengan frekuensi IF akan dikonversi lagi menjadi sinyal frekuensi yang lebih tinggi oleh RF conterter, dan diarahkan ke antenna.

Di dalam DUC, filter interpolasi bertugas untuk menaikkan *sample rate* dari sinyal baseband supaya sesuai dengan frekuensi operasi dari komponen. Proses ini adalah operasi yang berkebalikan dengan *decimator* pada arsitektur receiver.





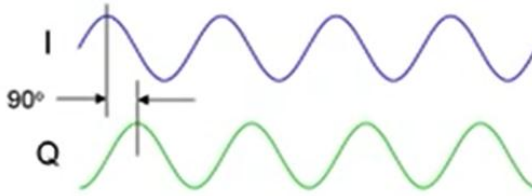
## **BAB 2**

# **KONSEP TELEKOMUNIKASI**

MNC Publishing

## 2.1. SINYAL KOMPLEKS

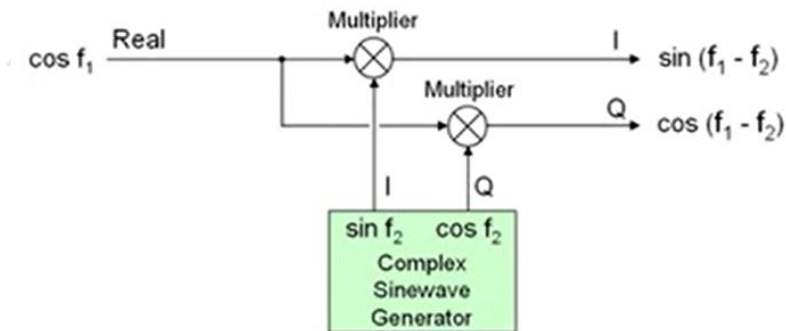
Konsep yang sangat penting dalam penggunaan *Software Defined Radio* adalah pemahaman mengenai sinyal kompleks. Jika dalam suatu sinyal real terdapat satu komponen, maka dalam sinyal kompleks terdapat dua komponen yaitu bagian real (In-phase) dan bagian imajiner (Quadrature). Komponen I dan komponen Q berbeda fasa sebesar  $90^\circ$ , seperti ditunjukkan pada gambar 2.1.



Gambar 2.1. Komponen I dan Q pada sinyal kompleks

Frekuensi pada sinyal kompleks dapat bernilai positif atau negatif. Pada sinyal kompleks dengan frekuensi positif, komponen I mendahului (*lead*) sebesar  $90^\circ$  terhadap komponen Q. Sedangkan pada sinyal kompleks dengan frekuensi negatif, komponen I tertinggal (*lag*) sebesar  $90^\circ$  terhadap komponen Q.

Sinyal kompleks dapat dibuat / dibangkitkan dengan cara mengalikan suatu sinyal real dengan sinyal sinusoida kompleks. Perhatikan gambar 2.2 berikut :



Gambar 2.2. Pembangkitan sinyal kompleks

Pada gambar 2.2 tersebut, sebuah sinyal real dengan frekuensi  $f_1$  dimasukkan ke dalam dua blok mixer (*multiplier*). Blok pertama, sinyal real dikalikan dengan sinyal sinus frekuensi  $f_2$ , sedangkan pada multiplier kedua sinyal dikalikan dengan cosinus frekuensi  $f_2$ . Berdasarkan sifat identitas trigonometri kita tahu bahwa :

$$\begin{aligned}\cos f_1 \times \sin f_2 &= \frac{1}{2} \{ \sin(f_1 - f_2) + \sin(f_1 + f_2) \} \\ \cos f_1 \cdot \cos f_2 &= \frac{1}{2} \{ \cos(f_1 - f_2) + \cos(f_1 + f_2) \}\end{aligned}$$

jika kita sederhanakan Persamaan di atas dengan membuang bagian frekuensi tinggi yaitu  $(f_1 + f_2)$ , maka kita peroleh sinyal keluaran mixer adalah masing-masing komponen I dan Q seperti ditunjukkan pada gambar 2.2. Sehingga perkalian sinyal tersebut melakukan dua operasi, yaitu :

pertama, merubah frekuensi sinyal input dari  $f_1$  menjadi frekuensi yang lebih kecil yaitu  $(f_1 - f_2)$ .

Kedua, merubah sinyal real (input) menjadi sinyal kompleks (terdapat komponen I dan komponen Q)

## 2.2. SISTEM TELEKOMUNIKASI

Berikut ini penjelasan yang lebih mendalam dan detil tentang sistem telekomunikasi. Jika suatu sinyal ditransmisikan melalui udara menggunakan gelombang elektromagnetik, sinyal harus berupa bentuk gelombang kontinyu (analog). Cara memahami sinyal semacam ini adalah melalui transformasi Fourier.

Teknologi telekomunikasi yang menggunakan transmisi elektromagnetik berada di sekitar kita, seperti televisi, radio, telepon seluler memungkinkan kita untuk melihat dan mendengar apa yang terjadi di seluruh dunia. Kita selalu berharap dapat berkomunikasi setiap saat dimanapun berada. Untuk mengimplementasikan keinginan tersebut, sebuah sinyal (gelombang suara, sinyal dari kamera TV, data bit dari komputer) perlu dikodekan, disimpan, ditransmisikan, selanjutnya diterima dan dikodekan kembali ke bentuk sinyal semula. Mengapa semua proses tersebut perlu dilakukan ? Perhatikan contoh berikut, yang merupakan permasalahan dalam transmisi suara/ musik. Mentransmisikan

suara secara langsung akan sia-sia karena gelombang suara akan menghilang dengan sangat cepat di udara. Tetapi jika gelombang suara ini ditransformasikan dulu menjadi gelombang elektromagnetik, maka gelombang dapat ditransmisikan dalam jarak yang jauh secara efisien. Sama halnya dengan sinyal TV dan data komputer.

### Transmisi Gelombang Elektromagnetik

Dalam memancarkan energi elektromagnetik yang efisien melalui udara (*wireless*), diperlukan sebuah antena. Supaya dapat berfungsi secara efisien, tinggi antena harus lebih besar daripada 1/10 panjang gelombang sinyal yang akan ditransmisikan. Di penerima, diperlukan juga antena dengan ukuran yang sebanding. Kita tahu bahwa besarnya panjang gelombang sinyal ( $\lambda$ ) dan frekuensinya ( $f$ ) adalah berbanding terbalik. Pada kecepatan cahaya ( $c = 3 \times 10^8$  m/s), hubungan antara panjang gelombang dan frekuensi adalah

$$\lambda = \frac{c}{f}$$

sebagai contoh, jika frekuensi dari gelombang elektromagnetik adalah  $f = 10$  kHz, maka panjang gelombangnya adalah :

$$\lambda = \frac{3 \times 10^8 \text{ m/s}}{10^4 \text{ /s}} = 3 \times 10^4 \text{ m}$$

Untuk mentransmisikan sinyal dengan frekuensi 10 kHz tersebut, diperlukan antena yang panjangnya lebih dari  $0,1 \lambda$  yaitu sebesar 3 km ! Sinyal suara manusia yang mempunyai kandungan frekuensi kurang dari 10 kHz akan memerlukan antena yang lebih tinggi lagi. Untuk mengatasi permasalahan tersebut, maka frekuensi sinyal yang ditransmisikan perlu digeser (dinaikkan frekuensinya) menjadi frekuensi yang jauh lebih tinggi (disebut dengan frekuensi *carrier*). Dengan demikian pembuatan antena menjadi lebih mudah untuk direalisasikan. Teknik untuk merubah frekuensi dari *baseband* menjadi frekuensi *carrier* adalah melalui modulasi.

Fakta berikutnya adalah perilaku perambatan gelombang elektromagnetik di atmosfer berbeda-beda tergantung dari frekuensi gelombang tersebut:

- Di bawah 2 MHz, gelombang elektromagnetik merambat mengikuti permukaan bumi. Karena itulah kita seringkali dapat mendengar sinyal radio gelombang pendek (*shortwave*) dari jarak ratusan bahkan ribuan kilometer dari sumber pemancar.
- Antara 2 sampai 30 MHz, terjadi propagasi *sky-wave* dengan banyak pantulan sinyal dari pembiasan lapisan atmosfer
- Di atas 30 MHz, terjadi propagasi *line-of-sight* (LOS) dimana sinyal merambat secara lurus antara dua menara pemancar dan penerima. Dalam frekuensi ini, seringkali juga terjadi hamburan atmosfer (*atmospheric scattering*) yang dapat dimanfaatkan untuk komunikasi terestrial jarak jauh.

Frekuensi merupakan gagasan utama dalam proses komunikasi jarak jauh. Karena peran frekuensi sebagai *carrier* dan peranannya dalam penetapan bandwidth, maka sangat penting untuk membuat suatu tool/alat yang dapat digunakan untuk mengukur kandungan frekuensi dalam suatu sinyal. Tool untuk melakukan pekerjaan ini adalah *Transformasi Fourier* (DFT dan FFT). Transformasi Fourier sangat bermanfaat dalam mengetahui energi atau daya sinyal pada frekuensi tertentu. Transformasi Fourier dari suatu sinyal  $w(t)$  didefinisikan sebagai :

$$W(f) = \int_{-\infty}^{\infty} w(t) \cdot e^{-j2\pi ft} dt$$

Dimana  $j = \sqrt{-1}$  dan  $f$  adalah frekuensi sinyal dalam Hz (putaran/s atau 1/s)

Secara matematika,  $W(f)$  adalah besaran kompleks yang merupakan fungsi frekuensi. Untuk setiap nilai  $f$ ,  $W(f)$  dapat digambarkan dengan beberapa cara. Misalnya,  $W(f)$  digambar dalam plot bagian real dan bagian imajiner sebagai fungsi frekuensi. Atau bisa juga dibuat dalam plot bagian Real vs bagian Imajiner. Plot yang paling umum dinyatakan dalam dua bagian : grafik pertama

menggambarkan magnitudo  $|W(f)|$  sebagai fungsi  $f$  yang disebut dengan spektrum magnitudo. Grafik kedua menggambarkan sudut fasa dari  $W(f)$  sebagai fungsi  $f$  yang disebut dengan spektrum fasa.

## Bandwidth dan Spektrum Sinyal

Pada setiap frekuensi tertentu dalam suatu sinyal, jika nilai spektrum magnitudonya positif ( $|W(f_0)| > 0$ ) maka dikatakan bahwa frekuensi  $f_0$  muncul/ terkandung dalam sinyal  $w(t)$ . Himpunan semua frekuensi yang terkandung/ muncul dalam suatu sinyal disebut dengan '*frequency content*' (kandungan frekuensi). Jika kandungan frekuensi suatu sinyal terdiri dari semua frekuensi di bawah  $f$  ( yang bernilai tertentu), maka dikatakan bahwa sinyal adalah bandlimited pada  $f$ . Sehingga dikatakan bahwa bandwidth sinyal adalah  $f$ . Beberapa contoh sinyal bandlimited :

- Sinyal suara telepon : frekuensi maksimum ( $f$ ) adalah 4 kHz
- Musik audio : frekuensi maksimu adalah 20 kHz

Sinyal dalam kenyataan sehari-hari tidak pernah benar-benar bersifat bandlimited, hampir selalu terdapat energi pada setiap frekuensinya.

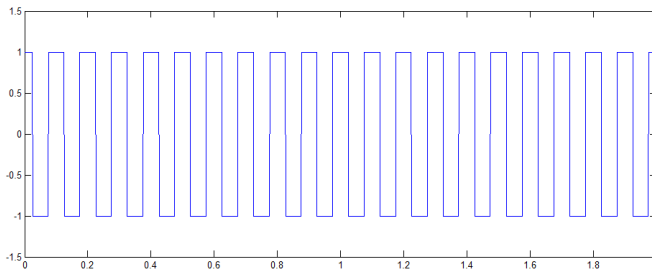
Sebuah sinyal  $s(t)$  dapat dinyatakan dalam bentuk analitis (persamaan matematika) sebagai besaran dalam fungsi waktu  $t$ . Jika  $s(t)$  ditransformasikan menggunakan transformasi Fourier akan diperoleh sinyal  $S(f)$  sebagai fungsi frekuensi.  $S(f)$  merupakan spektrum dari sinyal  $s(t)$  yang menggambarkan kandungan frekuensi dalam sinyal  $s(t)$ . Sebagai contoh, jika sinyal  $s(t)$  adalah hasil penjumlahan dari tiga buah sinyal sinusoida dengan frekuensi  $f_1$ ,  $f_2$ , dan  $f_3$ , maka spektrum frekuensinya yaitu  $S(f)$  akan terdiri dari tiga nilai pada frekuensi  $f_1$ ,  $f_2$ , dan  $f_3$ . Contoh yang lain misalkan  $s(t)$  hanya berisi frekuensi antara 100 sampai 200 Hz, maka spektrum nya akan bernilai nol untuk semua frekuensi di luar interval tersebut.

Dalam aplikasi riil, seringkali sinyal yang terjadi tidak dapat dinyatakan secara analitis dalam bentuk persamaan matematika, melainkan merupakan hasil pengukuran dari proses fisik. Sebagai contoh, sinyal yang diterima oleh receiver atau output dari filter adalah gelombang suara yang dikodekan sebagai file mp3. Dalam kasus ini, kita tidak dapat menentukan spektrum frekuensinya

melalui perhitungan transformasi fourier. Sebagai gantinya, kita gunakan DFT (*Discrete Fourier Transform*) atau FFT untuk menentukan kandungan frekuensi sinyal tersebut.

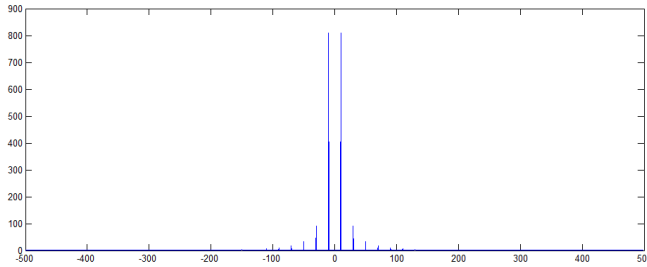
Berikut ini adalah contoh spektrum frekuensi dari sinyal pulsa yang dibangkitkan dengan menggunakan Matlab.

```
f=10;                % frekuensi sinyal pulsa
>> time=2;          % lama sinyal
>> Ts=1/1000;       % periode sampling
>> t=Ts:Ts:time;    % membuat vector waktu
>> x=sign(cos(2*pi*f*t)); % membangkitkan sinyal pulsa
                        dengan frekuensi f
>> plot(t,x)
```



Gambar 2.3. Sinyal pulsa  $s(t)$

```
>> y=fft(x);        % menentukan fungsi spektrum
                        sinyal
>> yo=fftshift(y);  % menggeser y terpusat di titik 0
>> fs=1/Ts;         % frekuensi sampling
>> f0 = (-n/2:n/2-1)*(fs/n); % range frekuensi terpusat di 0
>> power0=abs(yo).^2/n; % daya sinyal
>> plot(f0,power0)
```



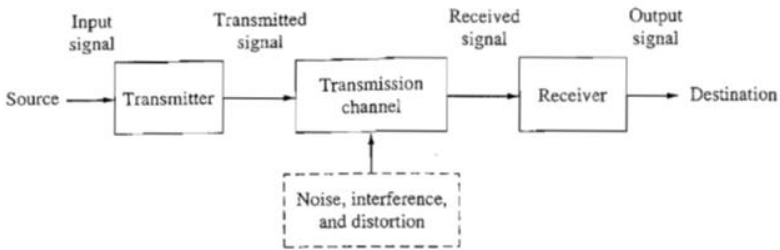
Gambar 2.4. Spektrum frekuensi  $S(f)$  dari sinyal pulsa  $s(t)$

## Elemen Sistem Komunikasi

Elemen-elemen dalam sistem komunikasi ditunjukkan dalam bentuk diagram blok seperti gambar 2.5. Dalam gambar tersebut, terdapat tiga bagian utama dari suatu sistem komunikasi, yaitu pemancar (transmitter), kanal transmisi, dan penerima (receiver). Tiap bagian mempunyai peran sebagai berikut :

- Pemancar (transmitter) berfungsi memproses sinyal input menjadi sinyal yang sesuai dengan karakteristik dari kanal transmisi. Pemrosesan sinyal untuk transmisi hampir selalu menggunakan teknik modulasi dan juga pengkodean.
- Kanal transmisi merupakan medium elektrik yang menghubungkan jarak antara sumber informasi dan tujuan. Kanal transmisi bisa berupa sepasang kabel, kabel koaksial, gelombang radio, atau laser. Dalam setiap kanal, sinyal yang ditransmisikan selalu mengalami losses/ redaman (*attenuation*), sehingga daya sinyal akan berkurang seiring dengan bertambahnya jarak.
- Penerima (receiver) berfungsi memproses sinyal yang datang dari medium (kanal transmisi). Proses ini diperlukan untuk merubah sinyal kembali ke bentuk sinyal informasi semula (baseband). Operasi yang dilakukan di penerima biasanya terdiri dari amplifikasi untuk mengkonkompensasi losses pada transmisi, serta demodulasi dan decoding untuk mengembalikan bentuk sinyal semula. Proses penting lainnya adalah pemfilteran.





Gambar 2.5. Elemen sistem komunikasi

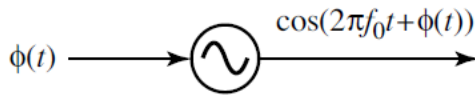
- Selama perjalanannya melalui medium (kanal transmisi), sinyal dapat mengalami distorsi, interferensi, dan noise. Distorsi merupakan gangguan bentuk gelombang yang disebabkan oleh respon sistem yang tidak sempurna. Distorsi sinyal dapat diperbaiki atau dikurangi menggunakan filter khusus yang disebut dengan *equalizer*. Interferensi adalah kontaminasi yang disebabkan oleh sinyal lain yang berasal dari pemancar lain, mesin atau jaringan listrik, rangkaian switching, dan sebagainya. Interferensi paling sering terjadi pada sistem komunikasi radio dimana antena penerima menerima beberapa sinyal pada saat yang sama. Sedangkan noise merupakan sinyal listrik yang bersifat acak dan tidak dapat diprediksi yang dihasilkan oleh proses alamiah baik dari dalam maupun dari luar sistem. Dengan menggunakan filter, noise ini dapat direduksi tetapi tidak bisa dihilangkan sama sekali. Sehingga noise akan membatasi kinerja sistem.

Sistem komunikasi seperti yang digambarkan melalui model diagram blok seperti pada gambar 2.5, dapat dibangun dengan menggunakan lima elemen sederhana :

1. Oscillator, yang membangkitkan sinyal sinusoida
2. Filter linier, yang menentukan range frekuensi dari sinyal
3. Nonlinieritas statis, yang dapat merubah kandungan frekuensi dari suatu sinyal misalnya multiplier, quantizer
4. Sampler, yang merubah sinyal analog menjadi sinyal diskrit
5. Elemen adaptif, yang akan mengikuti perubahan nilai parameter yang diinginkan

## A. Elemen Pertama : OSCILLATOR

Oscillator berasal dari bahasa Latin *oscillare* yang berarti 'mengendarai ayunan'. Dalam bahasa Inggris, *oscillate* berarti bergerak maju mundur dalam irama yang mantap (*steady*). Sehingga perangkat/ devais yang dapat membangkitkan sinyal yang bergerak maju mundur dengan irama mantap disebut dengan *oscillator*. Oscillator elektronik adalah devais yang menghasilkan sinyal elektronik berulang, biasanya berupa sinyal sinusoida.



Gambar 2.6. oscillator yang menghasilkan sinyal sinusoida dengan frekuensi  $f_0$  dan fasa  $\phi$

Sebuah oscillator dasar dapat direpresentasikan seperti pada gambar 2.6. Oscillator biasanya didesain untuk beroperasi pada suatu frekuensi tertentu  $f_0$  yang telah ditetapkan, sedangkan input oscillator menyatakan fasa dari sinyal outputnya. Output oscillator adalah sinyal sinusoida yang dinyatakan dengan persamaan :

$$s(t) = \cos(2\pi f_0 t + \phi)$$

Input bisa berupa bilangan tetap atau berubah sebagai fungsi waktu.

Terdapat beberapa cara untuk membuat oscillator dari komponen analog. Umumnya, menggunakan sebuah amplifier dan rangkaian feedback. Mekanisme yang lebih sederhana terjadi jika menggunakan oscillator digital karena dapat langsung dihitung (tidak perlu amplifier dan rangkaian feedback). Sebagai contoh, sinyal sinusoida digital dengan frekuensi  $f$  Hz dan fasa  $\phi$  radian dapat direpresentasikan secara matematika melalui persamaan :

$$s(kT_s) = \cos(2\pi f k T_s + \phi)$$

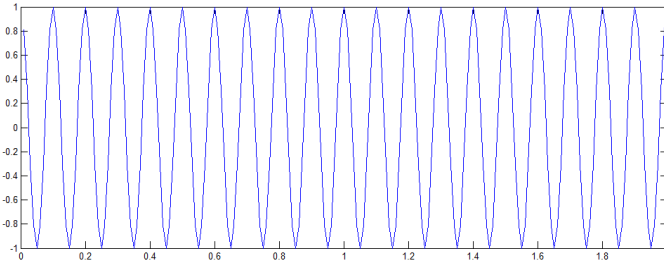
Dimana  $T_s$  adalah periode sampling dan  $k = 1, 2, 3, \dots$ . Persamaan tersebut dapat diimplementasikan secara langsung, jika menggunakan Matlab adalah sebagai berikut :

```
f=10; phi=0;           % menetapkan nilai frekuensi dan fasa
>> time = 2;          % lama simulasi
```

```

>> Ts=1/100;           % menetapkan periode sampling
>> t=Ts:Ts:time;       % membuat vector waktu
>> x=cos(2*pi*f*t+phi); % membangkitkan sinyal sinusoida

```



Gambar 2.7. Sinyal sinusoida frekuensi  $f = 10$  Hz dan fasa  $\phi = 0$

## B. Elemen Kedua : FILTER LINIER

Filter linier menentukan spektrum dari sinyal. Jika sinyal mempunyai terlalu banyak energi pada frekuensi rendah yang tidak diinginkan, energi pada frekuensi tersebut dapat dihilangkan menggunakan *highpass filter* (HPF). Sebaliknya jika sinyal mengandung terlalu banyak *noise* pada frekuensi tinggi, maka *lowpass filter* (LPF) dapat digunakan untuk menghilangkannya. Begitu pula jika sinyal yang ingin diproses adalah sinyal pada frekuensi antara  $f_1$  dan  $f_2$ , maka *bandpass filter* (BPF) dapat ditetapkan untuk melewatkan hanya frekuensi antara  $f_1$  dan  $f_2$  saja. Secara umum, filter linier dapat digunakan untuk mengkompensasi frekuensi dengan menaikkan dan/ atau menurunkan magnitudo dari spektrum frekuensinya sesuai yang diinginkan.

Berikut ini adalah kode Matlab untuk implementasi filter digital LPF, BPF, HPF. Hasilnya seperti ditunjukkan pada grafik gambar 2.8.

```

>> time=3;
>> Ts=1/10000;
>> x=randn(1,time/Ts);
>> y=fft(x);
>> n=length(x);
>> power = abs(y).^2/n;
>> fs=1/Ts;

```

```

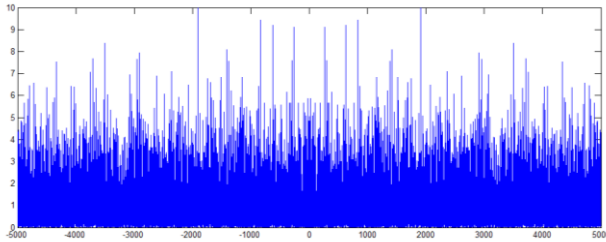
>> f = (-n/2:n/2-1)*(fs/n);
>> plot(f,power)
>> t=Ts:Ts:time;
>> figure(1); plot(t,x)
>> figure(2); plot(f,power)
>> f reqs =[0 0.2 0.21 1];

>> freqs=[0 0.2 0.21 1];
>> amps=[1 1 0 0 ] ;
>> b=firpm(100,freqs,amps);
>> ylp=filter(b,1,x);
>> ylp1=fft(ylp);
>> n1=length(ylp);
>> power1 = abs(ylp1).^2/n1;
>> figure(3); plot(f,power1)
>> freqs =[0 0.24 0.26 0.5 0.51 1];
>> amps=[0 0 1 1 0 0 ] ;
>> b=firpm(100,freqs,amps);
>> ybp=filter(b,1,x);
>> ybp1=fft(ybp);
>> n1=length(x);
>> power2=abs(ybp1).^2/n1;
>> figure(4); plot(f,power2)
>> freqs=[0 0.2 0.21 1];
amps=[1 1 0 0 ] ;
b=firpm(100,freqs,amps);
ylp=filter(b,1,x);
ylp1=fft(ylp);
>> power1 = abs(ylp1).^2/n1;
figure(3); plot(f,power1)
>> f reqs =[0 0.74 0.76 1];

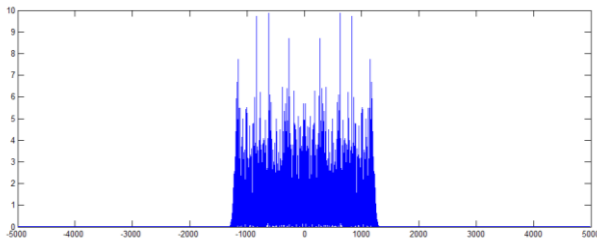
>> freqs =[0 0.74 0.76 1];
>> amps=[0 0 1 1 ] ;
>> b=firpm(100,freqs,amps);
>> yhp=filter(b,1,x);

```

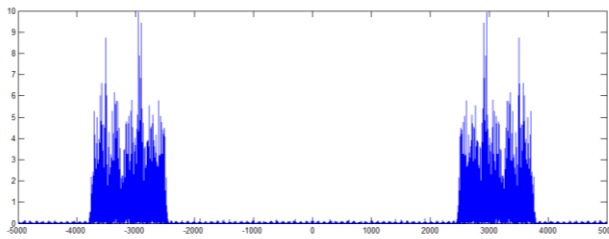
```
>> yhp1=fft(yhp);  
>> power3=abs(yhp1).^2/n1;  
>> figure(5); plot(f,power3)
```



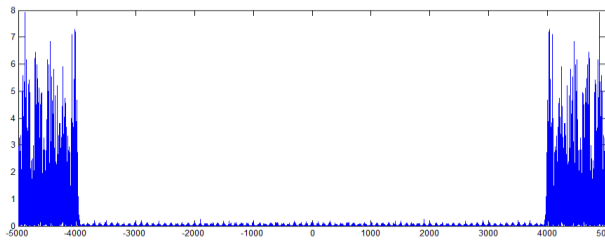
a. Spektrum magnitudo sinyal input



b. Spektrum magnitudo sinyal output LPF



c. Spektrum magnitudo sinyal output BPF

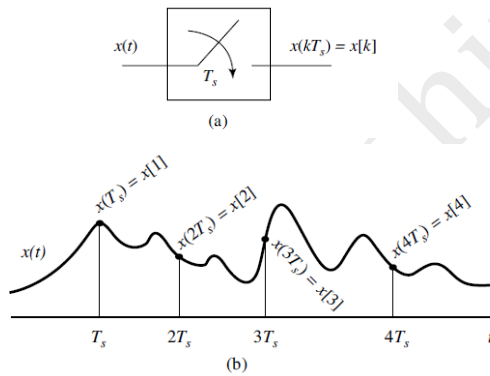


d. Spektrum magnitudo sinyal output HPF

Gambar 2.8. Spektrum sinyal input dan output filter

### C. Elemen ketiga : SAMPLER

Karena sebagian dari sistem transmisi digital adalah analog dan sebagian lagi adalah digital, maka harus dilakukan proses perubahan dari sinyal kontinyu menjadi sinyal digital. Begitu pula sebaliknya, dari sinyal digital menjadi kontinyu. Proses sampling dari suatu sinyal analog menjadi digital disebut dengan analog-to-digital conversion lebih mudah dijelaskan dalam time-domain. Gambar 2.9 menunjukkan bagaimana sinyal analog  $x(t)$  dirubah menjadi sinyal waktu diskrit  $x(kT_s)$ , dimana  $T_s$  adalah periode samplingnya.

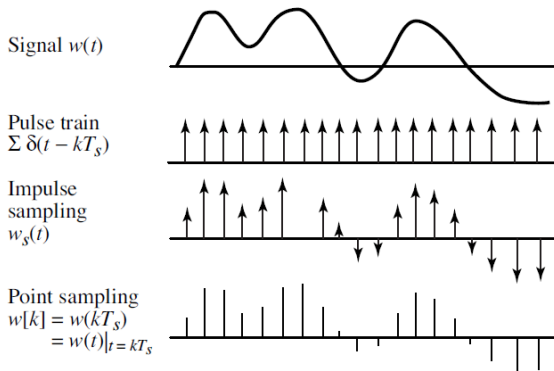


Gambar 2.9. Proses sampling

Salah satu teori utama dalam pemrosesan sinyal adalah Deret Fourier. Teori tersebut menyatakan bahwa sebuah sinyal dikatakan periodik jika dan hanya jika spektrum sinyalnya dapat ditulis dalam bentuk penjumlahan sinusoida kompleks dengan komponen frekuensi yang merupakan kelipatan dari frekuensi dasarnya ( $f$ ). Frekuensi dasar ini dapat ditulis sebagai  $f = 1/T$ , dimana  $T$  adalah periode perulangan sinyal. Sehingga, jika suatu sinyal berulang 100 kali dalam setiap detik ( $T = 0,01$  detik) maka spektrumnya akan terdiri dari jumlahan sinusoida dengan frekuensi 100, 200, 300, ... Hz. Sebaliknya, jika suatu spektrum tersusun dari jumlahan sinusoida dengan frekuensi 100, 200, 300, ... Hz, maka sinyal tersebut adalah sinyal periodik dengan periode  $T = 0,01$  detik.

## Teorema Sampling

Sampling dapat dimodelkan sebagai perkalian yang dilakukan dalam kawasan waktu antara sinyal analog dengan deretan sinyal impuls. Gambar 2.10 menunjukkan proses perkalian tersebut. Sinyal analog  $w(t)$  akan berubah menjadi deretan sinyal impuls yang amplitudonya sesuai dengan amplitude  $w(t)$  pada saat titik perkalian. Sinyal yang dihasilkan berupa sinyal diskrit. Efek yang lebih jelas bisa diamati bila sampling diamati dalam kawasan frekuensi. Bila sampling dilakukan dengan benar, maka tidak akan ada informasi yang hilang. Tetapi apabila dilakukan terlalu lambat (frekuensi sampling kecil), maka dapat menimbulkan aliasing.



Gambar 2.10. Perkalian sinyal analog dengan deretan impuls

Mari kita perhatikan sinyal  $w(t)$  pada gambar 2.10. Misalkan  $w(t)$  akan disampling pada setiap  $T_s$  detik untuk menghasilkan sinyal diskrit  $w(kT_s)$ , dimana  $k$  bernilai integer dan disebut dengan titik sampling. Pada setiap titik sampling ke- $k$ , nilai sinyal diskrit yang diperoleh adalah  $w(kT_s)$  yang merupakan nilai  $w(t)$  pada saat  $t = kT_s$ .

Pada saat kita akan melakukan proses sampling terhadap suatu sinyal, kita harus mengetahui informasi tentang kandungan frekuensi dari sinyal tersebut. Sebagai contoh, sinyal suara untuk percakapan telepon berisi informasi dengan kandungan frekuensi maksimal sampai 3,4 kHz. Sinyal suara harus difilter untuk membuang kandungan frekuensi yang berada di atas 3.4 kHz untuk memastikan bahwa bandwidth sinyal berada di dalam 3.4 kHz yang diperlukan.

Teorema sampling menyatakan bahwa :

Setiap sinyal bandlimited dengan bandwidth  $W$  yang disampling menggunakan frekuensi sampling  $\geq 2W$ , maka sinyal hasil sampling akan dapat direkonstruksi kembali menjadi sinyal semula.

Kecepatan sampling (frekuensi sampling) sebesar  $2W$  disebut dengan *Nyquist rate* (frekuensi Nyquist). Jika frekuensi sampling yang dipilih dinyatakan sebagai  $F_s$ , maka frekuensi Nyquistnya adalah sebesar :

$$F_N = \frac{F_s}{2}$$

Faktor paling penting pada saat proses sampling sinyal analog adalah pemilihan frekuensi sampling. Frekuensi sampling dikatakan sudah tepat/sesuai jika kita dapat merekonstruksi kembali sinyal semula dari deretena sampelnya. Jika kita tidak bisa mengembalikan ke sinyal semula, maka frekuensi sampling belum tepat dan perlu dikoreksi.

Bagaimana cara kita menentukan frekuensi Nyquist ?

#### ➤ **Kasus I**

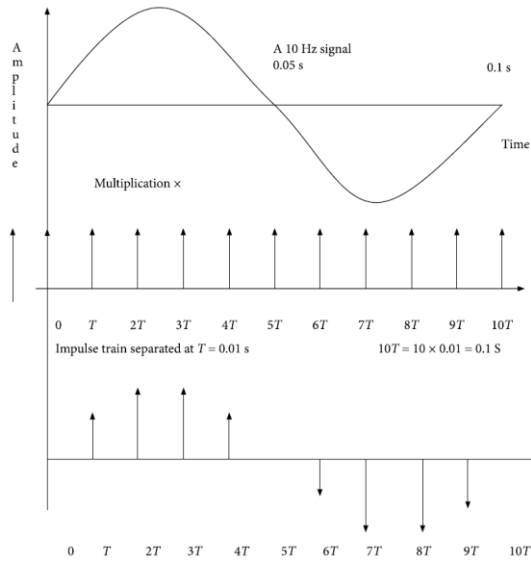
##### *Proses Sampling*

Sebuah sinyal  $x(t)$  mempunyai bandwidth sebesar 10 Hz. Sinyal tersebut akan disampling dengan frekuensi  $F_s = 100$  Hz, sehingga  $F_s > 2W$ .

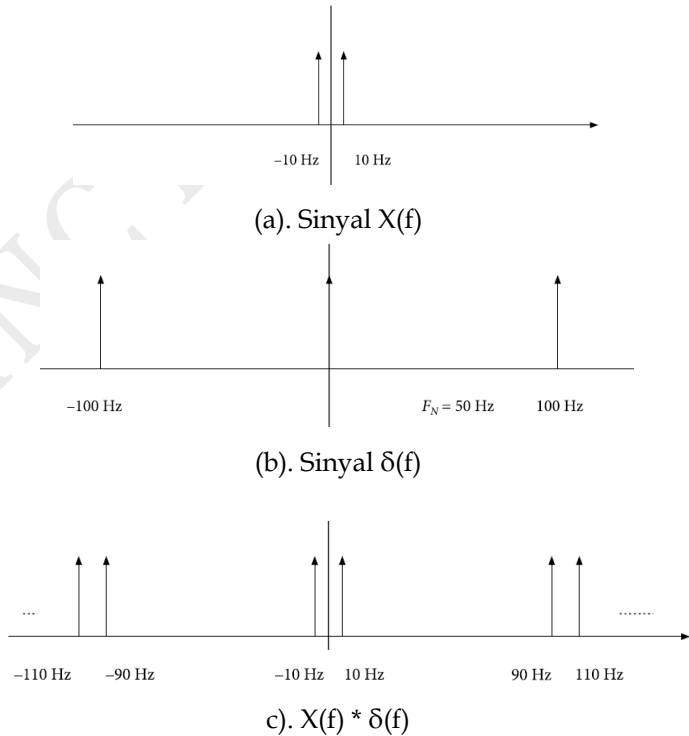
Seperti dijelaskan sebelumnya, dalam kawasan waktu sinyal sampling diperoleh melalui perkalian sinyal dengan deretan sinyal impuls  $\delta(t)$  yang terpisah sejauh  $T_s = 1/F_s = 1/100$  detik. Gambar 2.11 mengilustrasikan sinyal tersebut.

Berdasarkan sifat transformasi Fourier, perkalian dua buah sinyal dalam kawasan waktu adalah ekuivalen dengan operasi konvolusi di kawasan frekuensi, begitu juga sebaliknya. Pada saat sinyal  $x(t)$  disampling dengan frekuensi sampling 100 Hz (sepuluh kali frekuensi sinyal), maka sinyal dikalikan dengan deretan impuls yang masing-masing terpisah sejauh  $1/100$  detik. Dalam kawasan frekuensi, transformasi fourier dari sinyal  $x(t)$  dan  $\delta(t)$  akan dikonvolusikan. Jika  $x(t)$  adalah sinusoida dengan frekuensi 10 hz, maka transformasi fouriernya adalah spektrum garis yang letaknya di frekuensi 10 Hz seperti ditunjukkan pada gambar 2.12a.





Gambar 2.11. Sampling sinyal di kawasan waktu

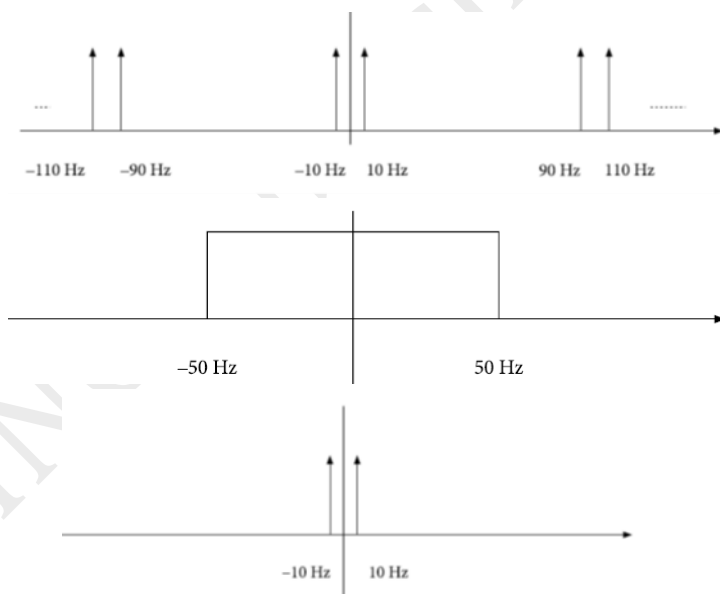


Gambar 2.12. Sampling sinyal di kawasan frekuensi

Gambar 2.12 menjelaskan proses sampling yang terjadi di kawasan frekuensi. Bagian (a) adalah spektrum sinyal  $x(t)$ , bagian (b) adalah spektrum sinyal impuls, sedangkan bagian (c) adalah spektrum sinyal hasil sampling yang merupakan konvolusi antara  $X(f)$  dan  $\delta(f)$ . Dapat dilihat pada gambar bagian (c) bahwa sinyal hasil sampling mempunyai spektrum garis yang merupakan replika dari  $X(f)$  dan terletak pada frekuensi-frekuensi yang merupakan kelipatan dari frekuensi samplingnya yaitu 100 Hz.

**Rekoveri sinyal dari deretan sampelnya**

Untuk mengembalikan/ rekoveri sinyal dari deretan sampelnya, dilakukan proses yang berkebalikan dengan proses sampling.



Gambar 2.13. Rekoveri sinyal di kawasan frekuensi

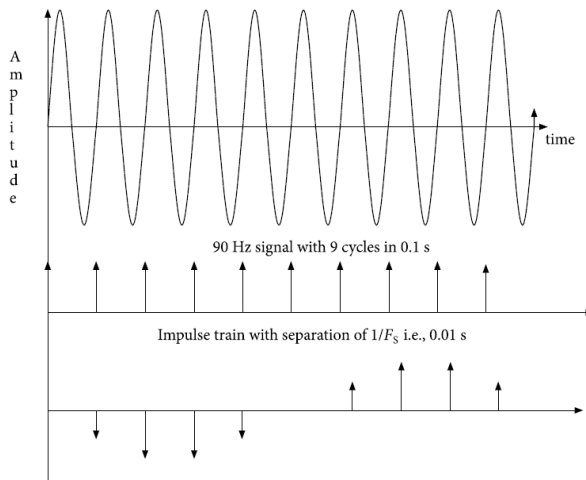
Gambar 2.13 menunjukkan proses yang terjadi. Bagian (a) menunjukkan replica sinyal yang merupakan deretan sampel yang dihasilkan dari proses sampling sebelumnya. Untuk mengembalikan sinyal tersebut ke bentuk  $X(f)$  semula, deretan sampel dikalikan dengan sinyal pada bagian (b) yang merupakan

sinyal dengan spektrum frekuensi 1 mulai dari frekuensi -50 Hz sampai 50 Hz (filter lowpass dengan *rectangular window*). Sehingga hasil akhirnya adalah sinyal yang berada di antara frekuensi -50 Hz – 50 Hz ( $-F_N$  sampai  $F_N$ ), yang merupakan sinyal semulay  $X(f)$ .

➤ **Kasus II**

**Proses Sampling**

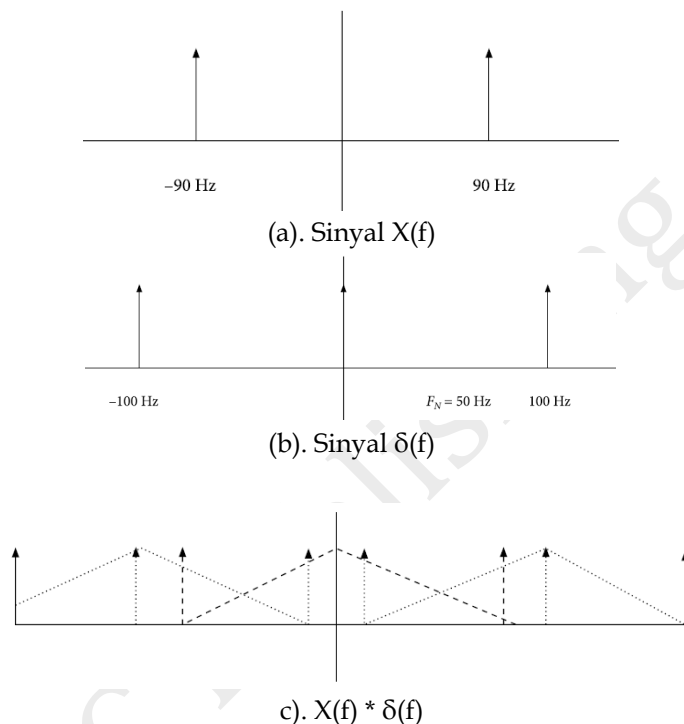
Sebagai contoh kasus kedua, mari kita perhatikan sinyal  $x(t)$  dengan frekuensi 90 Hz. Sinyal akan kita sampling menggunakan frekuensi sampling sebesar  $F_s = 100$  Hz. Kita tahu bahwa nilai  $F_s$  tersebut masih di bawah Frekuensi Nyquist ( $2 \times \text{bandwidth} = 180$  Hz). Gambar 2.14 menunjukkan proses sampling di kawasan waktu. Deretan sinyal hasil sampling diperoleh dengan mengalikan sinyal  $x(t)$  dengan deretan impuls dengan periode  $T_s = 1/F_s = 0.01$  detik.



Gambar 2.14. Proses sampling di kawasan waktu

Sedangkan proses sampling di kawasan frekuensi ditunjukkan pada gambar 2.15. Dapat diamati pada gambar tersebut bahwa sinyal hasil sampling menunjukkan adanya *overlap/ tumpang tindih* pada spektrum frekuensinya. Masing-masing replika dalam deretan sinyal tersebut tidak terpisah

dengan baik. Kondisi tersebut dapat menyebabkan error pada saat rekonstruksi sinyal, karena adanya interferensi dari replika sinyal di sebelahnya yaitu spektrum dari replika di sebelahnya ikut lewat dalam bandwidth filter yang digunakan.



Gambar 2.15. Proses sampling di kawasan frekuensi

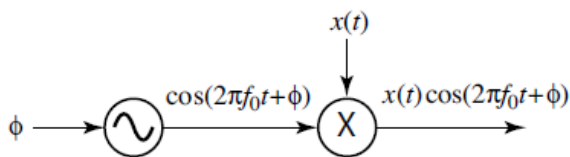
### ***Rekoveri sinyal dari deretan sampelnya***

Karena replika sinyal hasil sampling tidak terpisah dengan baik, maka terjadi percampuran antara spektrum replika yang bersebelahan. Replika sinyal yang berpusat di 0 berada pada frekuensi -90 Hz dan 90 Hz. Sedangkan replika sinyal yang terpusat di 100 Hz berada pada frekuensi 10 Hz dan 190 Hz. Bila deretan sampel tersebut kita lewatkan filter dengan frekuensi dari -50 Hz sampai 50 Hz, maka sinyal yang dilewatkan adalah komponen sinyal pada frekuensi -10 Hz dan 10 Hz dan bukannya pada -90 Hz dan 90 Hz (*aliasing*). Sehingga sinyal  $x(t)$  yang harusnya mempunyai frekuensi 90 Hz berubah menjadi sinyal dengan frekuensi 10 Hz.

#### D. Elemen keempat : Nonlinearitas Statis

Sistem linier seperti filter tidak dapat menambah komponen frekuensi baru pada suatu sinyal, tetapi hanya dapat menghilangkan komponen frekuensi yang tidak diinginkan. Nonlinearitas seperti multiplier dapat menambahkan komponen frekuensi pada suatu sinyal. Proses ini sangat bermanfaat dalam sistem komunikasi.

Salah satu sifat yang diperlukan dalam sistem komunikasi adalah kemampuan untuk merubah frekuensi sinyal tanpa harus merubah kandungan informasi dalam sinyal tersebut. Sebagai contoh, suara manusia berada pada range frekuensi di bawah 8 kHz. Supaya dapat mentransmisikan sinyal suara dalam jarak yang jauh, frekuensi ini harus dinaikkan (*upconversion*) menjadi frekuensi radio dimana energi sinyal dapat merambat dengan mudah sampai jarak yang jauh. Pada receiver, frekuensi sinyal tersebut harus diturunkan kembali (*downconversion*) ke dalam frekuensi aslinya. Sehingga terjadi dua kali pergeseran frekuensi. Suatu cara untuk melaksanakan proses pergeseran frekuensi tersebut adalah dengan mengalikan sinyal dengan suatu gelombang sinusoida seperti ditunjukkan pada gambar 2.10.



Gambar 2.16. Proses percampuran sinyal (mixer / multiplier)

Dalam sistem komunikasi, proses seperti ditunjukkan pada gambar 2.10 disebut dengan modulasi. Teknik modulasi dapat dikelompokkan menjadi modulasi analog dan modulasi digital, tergantung dari jenis sinyal informasi (sinyal  $x(t)$ ) sebagai pemodulasi.

#### E. Elemen kelima : Adaptasi

Adaptasi merupakan bentuk pembelajaran yang paling dasar. Elemen adaptif dalam sistem telekomunikasi berfungsi untuk

menentukan nilai pendekatan dari suatu parameter yang nilainya tidak diketahui. Tujuannya adalah untuk mengkompensasi perubahan kondisi atau untuk meningkatkan performansi sistem. Strategi yang umum digunakan adalah memperkirakan nilai yang dicari, kemudian menganalisa seberapa bagus nilai perkiraan tersebut, dan terus melakukan perbaikan berulang sampai konvergen pada satu nilai yang diinginkan.

Beberapa contoh penggunaan elemen adaptasi dalam sistem komunikasi adalah sebagai berikut :

- Dalam Automatic Gain Control (AGC), dimana suatu sistem melakukan proses perkalian dengan konstanta  $a$ . Selanjutnya suatu mekanisme feedback akan menganalisa apakah daya pada output AGC terlalu besar atau terlalu kecil, dan selanjutnya membuat penyesuaian terhadap nilai  $a$ .
- Dalam phase-lock loop (PLL), sistem adaptif berisi sinusoida dengan pergeseran fase sebesar  $a$  yang nilainya tidak diketahui. Mekanisme adaptasi akan menyesuaikan nilai  $a$  dengan nilai maksimum dari sinyal keluaran mixer.
- Dalam Equalizer, sistem adaptif berupa sebuah filter linier yang dengan sejumlah parameter  $a$ . Mekanisme adaptasi akan memonitor penyimpangan output sistem dari nilai targetnya dan menyesuaikan parameter  $a$  untuk memperbaiki sesuai target.
- Power control pada komunikasi bergerak, dan lain sebagainya.

### 2.3. SISTEM KOMUNIKASI RADIO

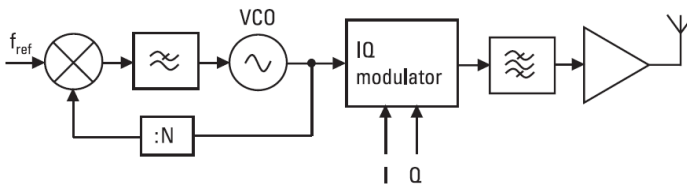
Sistem komunikasi radio merupakan sistem komunikasi yang menggunakan gelombang radio, seperti komunikasi seluler dan sistem navigasi/ radar. Jalur komunikasi radio antara dua stasiun terdiri dari transmitter, jalur transmisi, dan receiver. Performansi sistem tergantung pada karakteristik dari masing-masing komponen penyusunnya, serta karakteristik propagasi dari gelombang radio yang digunakan. Jika daya yang ditransmisikan serta *gain/loss* pada tiap bagian sistem diketahui, maka daya yang diterima pada receiver dapat dihitung. Yang menjadi masalah adalah adanya faktor lain yang mempengaruhi daya terima tersebut, seperti

modulasi sinyal, kestabilan frekuensi, interferensi dengan sistem radio yang lain, noise, dispersi kanal, dan sebagainya.

### Transmitter dan Receiver

Transmitter harus menghasilkan suatu sinyal yang mempunyai daya yang cukup dan frekuensi yang sangat akurat. Hal tersebut diperlukan supaya tidak mengganggu pengguna sistem radio yang lain. Informasi yang akan ditransmisikan yaitu sinyal baseband, ditumpangkan ke dalam sinyal carrier melalui proses modulasi. Untuk mendapatkan sinyal yang akurat, frekuensi oscillator harus stabil dan sinyal dilewatkan melalui bandpass filter sebelum ditransmisikan. Oscillator dapat distabilkan menggunakan sebuah resonator dengan faktor kualitas yang tinggi.

Gambar 2.11 menunjukkan sebuah transmitter dengan konversi langsung (*direct-conversion transmitter*).

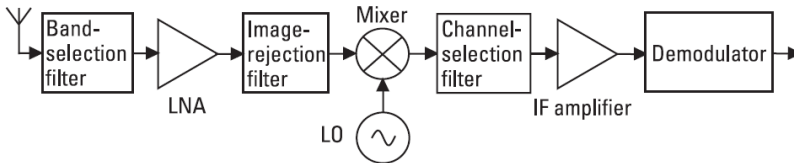


Gambar 2.17. Direct-conversion transmitter

Sinyal baseband digital akan memodulasi sinyal carrier pada modulator. Sinyal termodulasi yang keluar dari modulator selanjutnya difilter dan dikuatkan. Dalam pemancar superheterodyne sinyal akan dimasukkan upconverter untuk menaikkan ke frekuensi akhir. Frekuensi carrier distabilkan menggunakan PLL yang merupakan sistem feedback, terdiri dari VCO, detektor fasa, dan lowpass filter. Dalam gambar 2.11 juga terdapat jalur feedback yang membagi frekuensi VCO dengan N. Hasilnya kemudian dibandingkan dengan sinyal yang datang dari oscillator referensi pada detektor fasa. Selisih dari nilai perbandingan tersebut (dalam fasa) dirubah menjadi tegangan yang sebanding dengan perbedaan fasa, selanjutnya dimasukkan lowpass filter. Keluaran lowpass filter adalah tegangan yang digunakan untuk

mengontrol frekuensi VCO supaya selisihnya dengan frekuensi referensi menjadi nol.

Pada bagian penerima, receiver harus sensitif dan selektif. Receiver harus dapat mendeteksi sinyal terlemah yang datang di antara sinyal lain yang lebih kuat. Karena itu receiver yang baik harus mempunyai filter yang baik, frekuensi local oscillator yang akurat, dan komponen yang rendah noise.



Gambar 2.18. Penerima superheterodyne

Pada umumnya, receiver yang banyak digunakan adalah superheterodyne. Gambar 2.12 menunjukkan diagram blok penerima superheterodyne, dimana sinyal yang datang dari antena pertama kali akan difilter kemudian dikuatkan sebesar 10 sampai 20 dB menggunakan LNA (Low Noise Amplifier). Sebuah frequency synthesizer akan membangkitkan sinyal oscillator yang akurat. Selanjutnya mixer akan melakukan proses downconversion sinyal menjadi frekuensi IF. Hasilnya dimasukkan lagi ke dalam filter dan dikuatkan kembali sebesar 70 sampai 100 dB sebelum proses demodulasi. Selanjutnya demodulator akan mengembalikan sinyal analog menjadi sinyal informasi baseband kembali. Jika variasi level daya sinyal cukup signifikan, maka gain IF harus dapat dikontrol menggunakan rangkaian AGC untuk menghindari saturasi.

## 2.4. GANGGUAN DALAM SISTEM TELEKOMUNIKASI

Tiga jenis gangguan yang terjadi dalam sistem transmisi telekomunikasi adalah :

1. Distorsi amplitudo (*attenuation*)
2. Distorsi fasa
3. Noise



## Distorsi Amplitudo

IEEE mendefinisikan distorsi amplitudo (*attenuation*) sebagai perubahan amplitudo sinyal pada setiap frekuensi dibandingkan terhadap suatu frekuensi referensi. Sumber distorsi amplitudo diantaranya adalah noise dan interferensi, tetapi sumber utamanya adalah proses penguatan nonlinier (*nonlinear amplification*). Sumber lain adalah dari intermodulation distortion, yaitu ketika terjadi percampuran antara dua buah sinyal, misalnya perkalian, penjumlahan, dan sebagainya.

## Distorsi Fasa

Dalam sistem transmisi voice, kanal transmisi dapat dipandang sebagai sebuah bandpass filter. Sinyal yang ditransmisikan memerlukan waktu untuk tiba dari pemancar ke penerima. Lamanya waktu yang diperlukan tergantung dari kecepatan propagasi pada medium serta panjang medium (jarak). Kecepatan propagasi sinyal juga tergantung pada frekuensi yang digunakan. Pada kanal transmisi suara, kecepatan propagasi cenderung meningkat pada frekuensi tengah, dan berkurang pada frekuensi tepi.

Waktu yang diperlukan sinyal untuk sampai ke tujuan disebut dengan delay. Pada saat melalui kanal transmisi, sinyal termodulasi tidak akan mengalami distorsi jika pergeseran fasanya berubah secara uniform terhadap besarnya frekuensi. Jika pergeseran fasa adalah nonlinier terhadap frekuensi, maka sinyal output akan terdistorsi tergantung pada nilai frekuensinya.

Pada dasarnya, jika hubungan antara fasa dan frekuensi yang melewati kanal passband tidak linier, maka akan terjadi distorsi fasa pada sinyal yang ditransmisikan. Distorsi fasa diukur menggunakan parameter yang disebut *envelope delay distortion* (EDD). Secara matematika, EDD merupakan turunan (*derivative*) dari pergeseran fasa. Variasi maksimum dalam envelope sinyal sepanjang bandwidth frekuensi disebut dengan envelope delay distortion. EDD disebut juga dengan *group delay*, yang merupakan rasio perubahan pergeseran fasa antara dua titik dalam jaringan.

## Noise

Noise merupakan sinyal yang tidak diinginkan yang terjadi dalam suatu sistem komunikasi. Sehingga topik tentang reduksi noise menjadi satu topik yang sangat penting dalam teknik transmisi. Hal tersebut merupakan faktor utama dalam menentukan performansi sistem keseluruhan. Noise dikelompokkan dalam beberapa kategori :

### 1. Thermal noise

Thermal noise terjadi dalam semua medium transmisi dan perangkat komunikasi. Noise ini muncul sebagai akibat adanya pergerakan elektron secara acak, yang dicirikan dengan distribusi energi secara uniform pada seluruh spektrum frekuensi. Dimodelkan sebagai distribusi Gaussian.

Setiap perangkat dan medium transmisi akan berkontribusi terhadap munculnya thermal noise bila berada pada temperatur di atas nol mutlak (pada skala Kelvin). Noise ini merupakan faktor yang membatasi sensitifitas pada receiver.

Besarnya nilai thermal noise berbanding lurus dengan bandwidth dan temperatur noise. Berdasarkan konstanta Boltzmann, kita dapat menurunkan hubungan antara level thermal noise ( $P_n$ ) dalam setiap 1 Hz bandwidth pada temperatur nol mutlak (dalam skala Kelvin) sebagai berikut :

Pada receiver ideal berlaku :

$$P_n = -228,6 \text{ dBW} / \text{Hz pada temperatur nol}$$

Pada temperatur ruang ( 290 K atau 17° C ) :

$$P_n = -204 \text{ dBW} / \text{Hz}$$

atau  $P_n = -174 \text{ dBm} / \text{Hz}$

Receiver ideal adalah devais penerima yang tidak menimbulkan thermal noise pada kanal komunikasi. Tentu saja hal ini hanya ada pada kondisi ideal dan bukan pada sistem riil. Meskipun demikian, kondisi ideal tersebut dapat digunakan sebagai referensi dalam menentukan nilai kondisi riil, dimana :

$$P_n = -204 \text{ dBW} / \text{Hz} + NF_{dB} + 10 \log B$$

dimana B adalah bandwidth dari receiver (dinyatakan dalam satuan Hz).

NF adalah Noise Figure dari receiver. NF merupakan besaran yang digunakan untuk menyatakan jumlah thermal noise yang diberikan oleh receiver ke dalam kanal komunikasi, dinyatakan dalam satuan dB.

Sebagai contoh perhitungan level thermal noise, misalkan sebuah receiver mempunyai *noise figure* sebesar 3 dB dan bandwidth 10 MHz. Maka level thermal noise dari receiver tersebut adalah :

$$\begin{aligned} P_n &= -204 \frac{dBW}{Hz} + 3dB + 10 \log(10 \times 10^6) \\ &= -204 \frac{dBW}{Hz} + 3dB + 70 dB \\ &= -131 dBW \end{aligned}$$

## 2. Intermodulation noise

Intermodulation noise (IM) hasil dari adanya intermodulasi. Jika dua buah sinyal dengan frekuensi F1 dan F2 dilewatkan melalui medium atau devais nonlinear, hasilnya akan mengandung komponen energi pada frekuensi palsu (bukan kandungan frekuensi sinyal). Komponen tersebut dapat muncul baik di dalam maupun di luar bandwidth sistem/ devais. IM dapat dihasilkan dari harmonisa sinyal, baik karena perkalian antar harmonisa, atau antar komponen sinyal dasar dan harmonisanya.

## 3. Impulse noise

Impulse noise merupakan suatu loncatan sinyal noise dengan durasi yang singkat dan tidak kontinyu. Loncatan sinyal ini sering disebut dengan hits, dan masing-masing mempunyai spektrum yang luas (bandwidth frekuensi lebar). Impulse noise dapat menyebabkan penurunan kinerja transmisi telepon, dan bisa menyebabkan error yang serius pada rangkaian digital. Sumber impuls noise adalah petir, switch mekanik, lampu neon, dan sebagainya.

## 4. Crosstalk

MNC Publishing



## **BAB 3**

### **MODULASI AM**

MNC Publishing

### 3.1. MODULASI DAN DEMODULASI

Modulasi AM (*Amplitude Modulation*) merupakan tipe sistem modulasi yang paling sederhana. Modulasi ini juga paling mudah dipahami karena perkalian analog dari dua buah sinyal akan menghasilkan spektrum yang berisi penjumlahan dan selisih (pengurangan) dari kedua frekuensi sinyal. Contoh berikut ini menggambarkan proses perkalian sinyal sederhana :

Contoh :

Sinyal  $y(t)$  diperoleh dengan mengalikan sinyal informasi  $m(t)$  dan sinyal carrier  $x(t)$ . Persamaan sinyal adalah :

$$m(t) = 2 \cos(10.000 t)$$

$$x(t) = 50 \cos(1.000.000 t)$$

berdasarkan identitas trigonometri, diperoleh :

$$\begin{aligned} y(t) &= 50 [\cos(1000000 t + 10000 t) + \cos(1000000 t - 10000 t)] \\ &= 50 [\cos(1,010,000t) + \cos(990,000t)] \end{aligned}$$

Dari hasil perhitungan, diperoleh bahwa sinyal  $y(t)$  mengandung dua komponen frekuensi yang masing-masing diperoleh dari penjumlahan dan pengurangan frekuensi sinyal  $m(t)$  dan  $x(t)$ , yaitu sebesar 1.010.000 rad/sec dan 990.000 rad/sec.

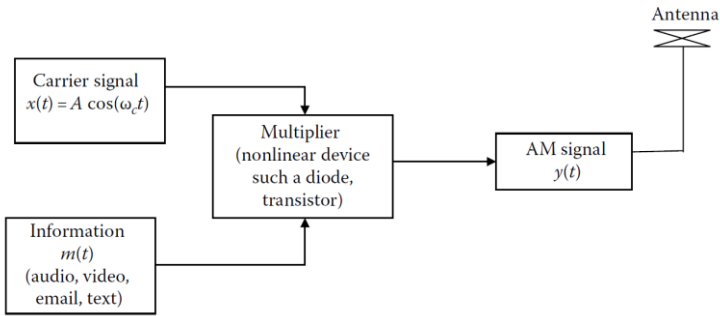
Bandwidth dari sinyal  $y(t)$  adalah selisih antara frekuensi tertinggi dan frekuensi terendah dari sinyal, yaitu :

$$\begin{aligned} \text{Bandwidth} &= 1.010.000 - 990.000 \text{ rad/sec.} \\ &= 20.000 \text{ rad/sec} \end{aligned}$$

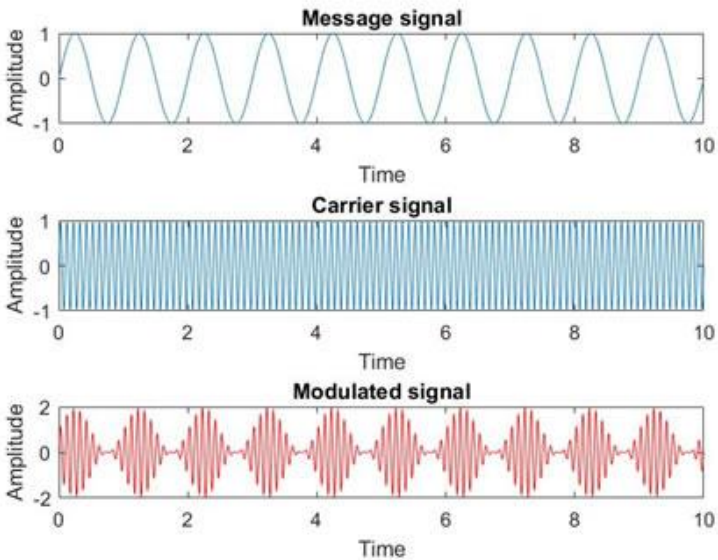
Dalam bentuk diagram blok, sistem pemancar modulasi AM ditunjukkan seperti pada gambar 3.1. Diagram tersebut menunjukkan proses perkalian antara sinyal informasi dengan sinyal carrier.

Untuk melihat bentuk sinyal informasi, sinyal carrier, dan sinyal hasil modulasinya, kita dapat membuat simulasi menggunakan matlab. Secara umum, bentuk ketiga sinyal tersebut dapat dilihat seperti pada gambar 3.2. Sinyal termodulasi

(modulated signal) bentuknya bisa berbeda tergantung pada skema modulasi yang digunakan, apakah DSB-SC, DSB-FC, ataukah SSB. Contoh pembangkitan sinyal dengan menggunakan ketiga skema tersebut juga diberikan pada dan ditunjukkan pada gambar 3.3, gambar 3.4, dan gambar 3.5.



Gambar 3.1. Struktur pemancar AM



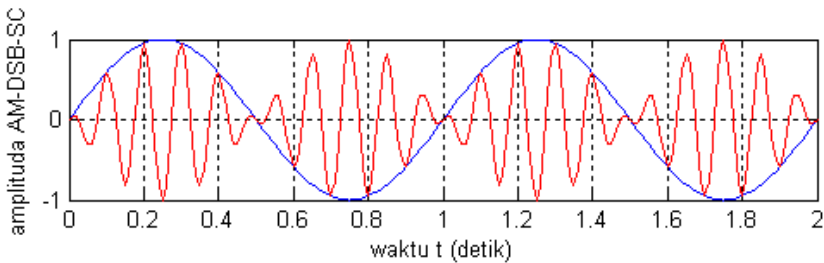
Gambar 3.2. Bentuk sinyal pada modulasi AM

## Modulasi AM-DSB-SC

Untuk membangkitkan sinyal AM dengan modulasi DSB-SC menggunakan Matlab, kita dapat mengetikkan perintah sebagai berikut :

```
ydouble = amod(x,Fc,Fs,'amdsb-sc'); %modulasi am-dsb-sc
plot(t,ydouble,'r')                %menampilkan sinyal ydouble
                                    %sebagai fungsi waktu t
xlabel('waktu t (detik)');          %memberikan komentar pada
                                    %sumbu datar %gambar
ylabel('amplituda');                %memberikan komentar pada
                                    %sumbu tegak gambar
axis([0 2 -1 1]);                  %membatasi sumbu datar dengan
                                    %nilai 0 sampai 2 dan sumbu tegak
                                    %dengan nilai -1 sampai 1
```

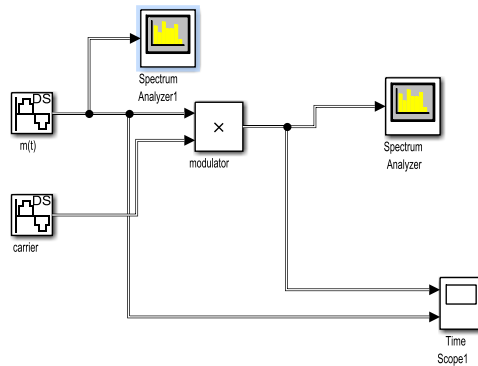
Sinyal yang dihasilkan ditunjukkan pada gambar 3.2, merupakan hasil perkalian antar sinyal informasi dengan frekuensi  $F_s$  dan sinyal carrier dengan frekuensi  $F_C$  menggunakan skema modulasi DSB-SC.



Gambar 3.3. Sinyal modulasi AM-DSB-SC

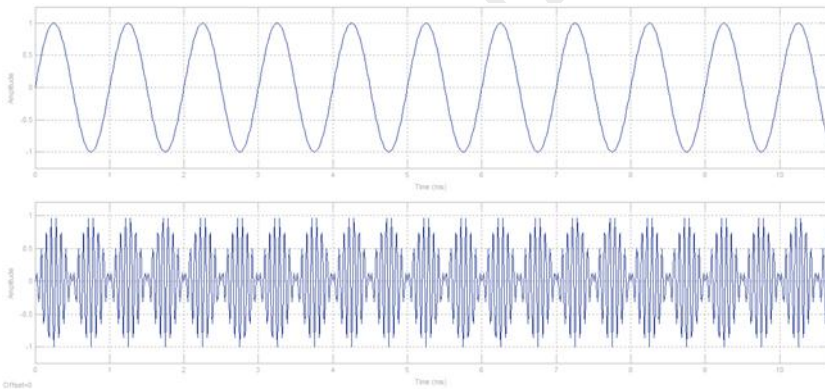
Skema modulasi AM DSB-SC juga dapat disimulasikan menggunakan Simulink seperti ditunjukkan pada gambar 3.4.





Gambar 3.4. Model simulasi DSB-SC menggunakan Simulink

Hasil simulasi ditunjukkan pada gambar 3.5. Dalam gambar tersebut, sinyal pesan  $m(t)$  dan sinyal keluaran modulator ditunjukkan .



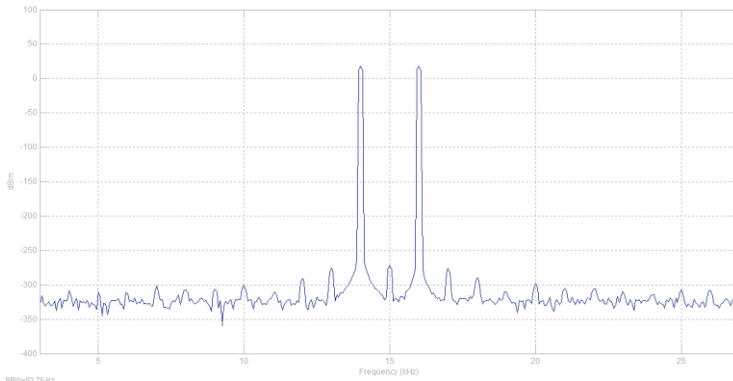
Gambar 3.5. Sinyal termulasi DSB-SC

Dengan mengalikan sinyal pesan  $m(t)$  frekuensi 1 kHz dan sinyal carrier frekuensi 15 kHz, diperoleh sinyal termulasi  $y(t)$  dengan persamaan :

$$y(t) = [km \cos (2\pi 1000t) * \cos (2\pi 15000t)]$$

sinyal  $y(t)$  mengandung dua komponen frekuensi, satu frekuensi berada di sebelah kiri (bawah) sinyal carrier dan satu komponen frekuensi berada di sebelah kanan (atas) sinyal carrier. Seperti bisa

kita lihat pada gambar spektrum frekuensi pada gambar 3.6, komponen frekuensi di kiri carrier adalah sebesar  $(f_c - f_m) = 14$  kHz. Sedangkan komponen frekuensi di kanan carrier sebesar  $(f_c + f_m) = 16$  kHz. Karena itulah skema modulasi seperti ini disebut dengan double-side band, dimana semua daya yang ditransmisikan berada di kedua sisi frekuensi carriernya. Sedangkan frekuensi carrier sendiri tidak ditransmisikan (*suppress-carrier*).



Gambar 3.6. Spektrum frekuensi sinyal termulasi DSB-SC

### Modulasi AM-DSB-FC

Untuk membangkitkan sinyal AM dengan modulasi DSB-FC menggunakan Matlab, kita dapat menyetikkan perintah sebagai berikut :

```

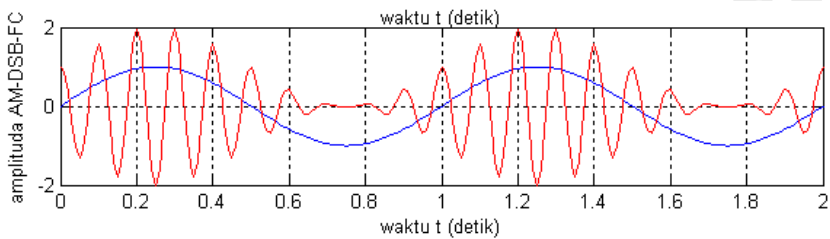
ydoubletc = amod(x,Fc,Fs,'amdsb-fc',1);%memodulasi x dengan
am-dsb-fc
plot(t,x); %memplot sinyal x terhadap t
hold on %menampilkan sinyal x sebagai
fungsi waktu t
xlabel('waktu t (detik)'); %memberikan komentar pada sumbu
datar %gambar
ylabel('amplituda AM-DSB-FC'); %memberikan komentar pada
sumbu tegak gambar
plot(t,ydoubletc,'r'); %menampilkan sinyal ydoubletc
sebagai fungsi waktu t, berwarna
merah
    
```

```

axis([0 2 -2 2]);           %membatasi sumbu datar dengan
                             nilai 0 sampai 2 dan sumbu tegak
                             dengan nilai -1 sampai 1
grid on;                    %menampilkan garis putus-putus
                             grafik

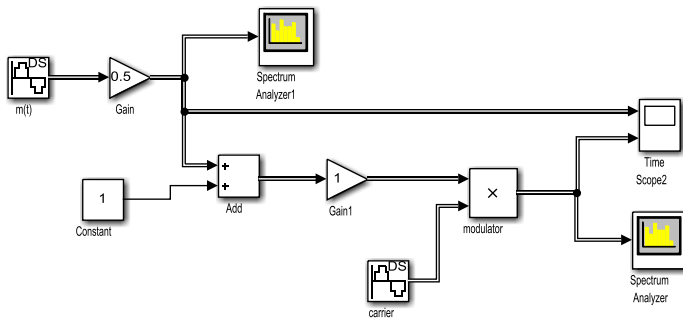
```

Sinyal yang dihasilkan ditunjukkan pada gambar 3.7, merupakan hasil perkalian antar sinyal informasi dengan frekuensi  $F_s$  dan sinyal carrier dengan frekuensi  $F_c$  menggunakan skema modulasi DSB-FC.



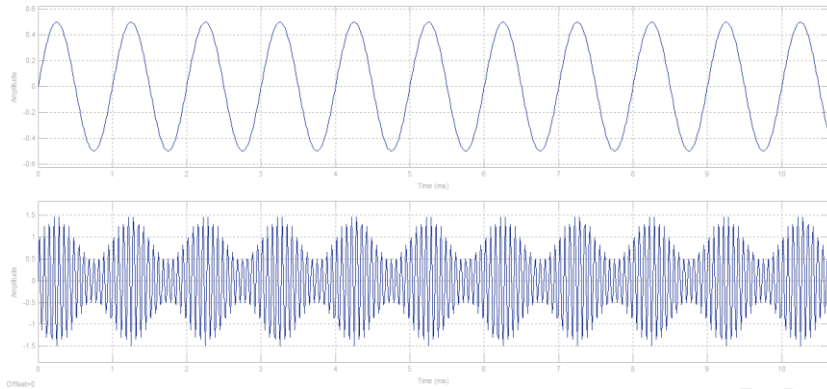
Gambar 3.7 Sinyal Modulasi AM-DSB-FC

Skema modulasi AM DSB-SC juga dapat disimulasikan menggunakan Simulink seperti ditunjukkan pada gambar 3.8.



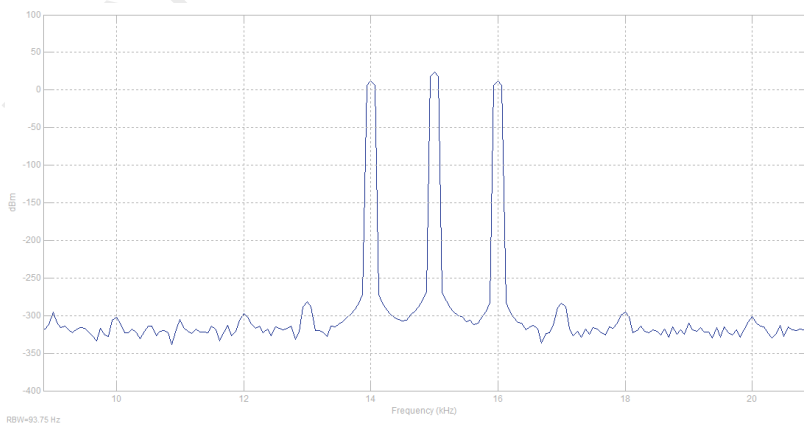
Gambar 3.8. Model simulasi DSB-FC menggunakan Simulink

Hasil simulasi ditunjukkan pada gambar 3.9. Dalam gambar tersebut, sinyal pesan  $m(t)$  dan sinyal keluaran modulator ditunjukkan.



Gambar 3.9. Sinyal termodulasi DSB-FC

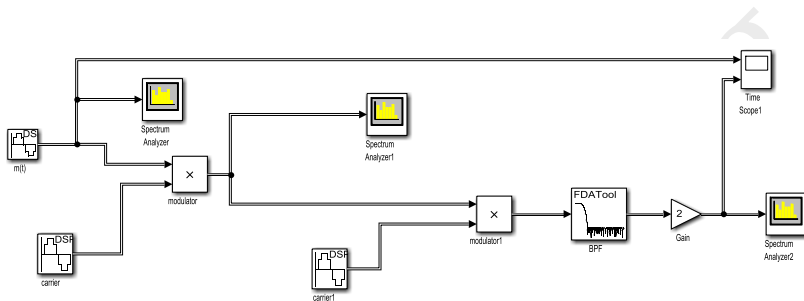
Kalau kita perhatikan sinyal termodulasi pada DSB-FC, terlihat bahwa amplitude sinyal pada envelope-nya lebih tinggi dibandingkan dengan pada DSB-SC. Hal ini menunjukkan bahwa daya transmisi pada DSB-FC lebih besar. Sesuai dengan spektrum frekuensi yang ditunjukkan pada gambar 3.10, terlihat bahwa terdapat tiga komponen frekuensi pada sinyal termodulasi. Satu komponen frekuensi carrier (yaitu 15 kHz), satu komponen di sebelah kiri carrier (14 kHz), dan satu lagi di sebelah kanan carrier (16 kHz). Sesuai dengan namanya yaitu DSB-FC (*full carrier*), daya sinyal yang ditransmisikan termasuk pada frekuensi carrier nya. Sehingga daya transmisi lebih besar dibandingkan dengan DSB-SC.



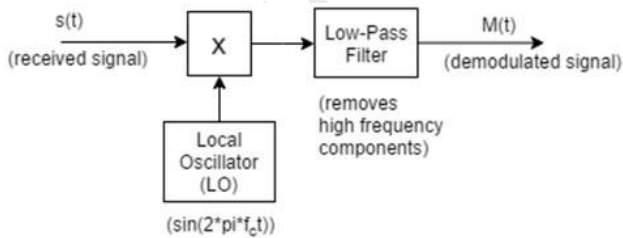
Gambar 3.10. Spektrum frekuensi sinyal termodulasi DSB-FC

## Demodulasi AM-DSB-SC

Untuk melihat proses demodulasi sinyal AM pada receiver, kita dapat menggunakan simulasi melalui Simulink seperti model sistem yang ditunjukkan pada gambar 3.11. Metode demodulasi yang digunakan di sini adalah menggunakan *Synchronous Detector/Demodulator*. Local oscillator pada demodulator harus sinkron dengan sinyal carrier. Dalam bentuk diagram blok, metode ini ditunjukkan seperti pada gambar 3.12.



Gambar 3.11. Penerima AM



Gambar 3.12. Synchronous Detector

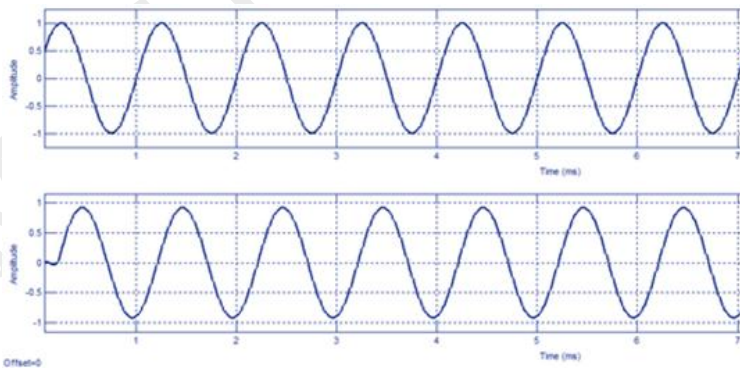
Supaya filter dapat mendeteksi envelope dari sinyal informasi, maka frekuensi carrier harus dibuat setinggi mungkin. Tetapi harus diperhatikan bahwa adanya noise selama propagasi sinyal tidak dapat difilter/ dibuang dengan sempurna dalam sistem semacam itu.

Selanjutnya simulasi dijalankan menggunakan parameter sebagai berikut :

- Sinyal informasi  $m(t)$  yang dikirimkan mempunyai frekuensi 1 kHz, dengan sample time :  $1/100$  kHz
- Sinyal carrier dengan frekuensi 10 KHz, fase  $\pi/2$  dan sample time :  $1/100$  kHz
- Local oscillator di demodulator sama seperti carrier
- Filter yang digunakan : lowpass,  $F_s = 100$  kHz,  $F_{pass} = 6$ ,  $F_{stop} = 12$
- Waktu simulasi 0,02

Setelah dijalankan, diperoleh hasil sebagai berikut :

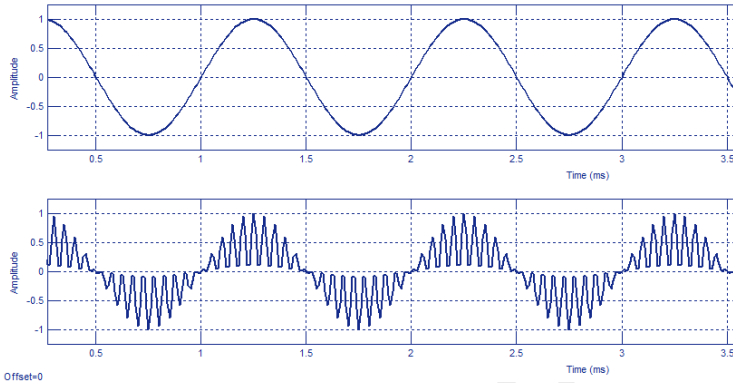
- Gambar 3.13 menunjukkan dua buah sinyal, yang di atas adalah sinyal informasi yang dikirimkan. Yang di bagian bawah adalah sinyal keluaran dari lowpass filter setelah proses demodulasi. Jika kita perhatikan, amplitudo sinyal keluaran LPF sedikit lebih kecil dibandingkan dengan sinyal informasi aslinya. Hal ini sesuai dengan teori bahwa sinyal mengalami peredaman selama transmisi. Sinyal juga mengalami penundaan (ada beda fasa)



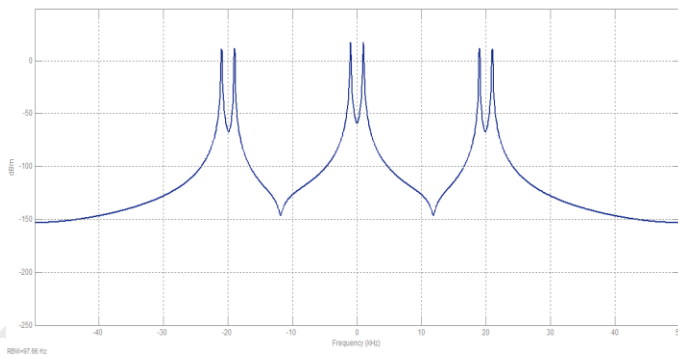
Gambar 3.13. Sinyal informasi dan keluaran LPF

- Gambar 3.14 menunjukkan bentuk sinyal keluaran dari demodulator sebelum masuk lowpass filter. Bisa diperhatikan bahwa sinyal tersebut adalah frekuensi carrier dengan envelope frekuensi informasi. Jika digambarkan spektrum

frekuensinya, terlihat seperti ditunjukkan pada gambar 3.15. Dengan melewati sinyal tersebut pada lowpass filter yang mempunyai bandwidth 12 kHz, maka akan diperoleh sinyal baseband yang merupakan sinyal informasi.



Gambar 3.14. Sinyal informasi dan keluaran demodulator



Gambar 3.15. Spektrum frekuensi sinyal keluaran demodulator

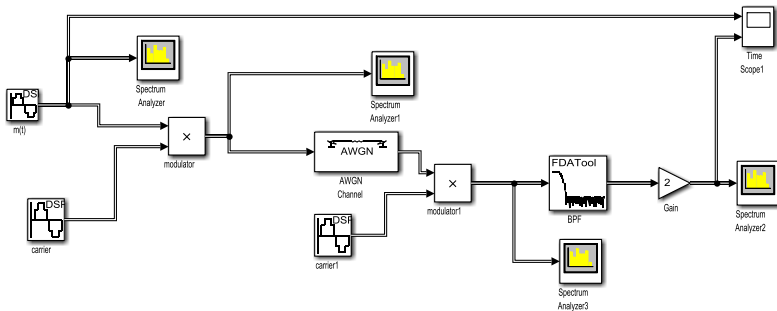
### Pengaruh noise pada sinyal

Dalam suatu sistem komunikasi, noise selalu muncul selama proses transmisi. Untuk melihat pengaruh noise terhadap sinyal yang diterima, kita akan melakukan simulasi seperti pada gambar 3.11 dengan menambahkan blok kanal AWGN di antara modulator dan demodulator. Kanal AWGN akan memberikan pola noise yang bersifat acak. Model simulasinya seperti ditunjukkan pada gambar 3.16. Untuk melihat pengaruh noise terhadap sinyal yang diterima, parameter yang diubah dalam blok AWGN adalah :

Mode : variance from mask

Variance : 0.01; 0.05; 0.5

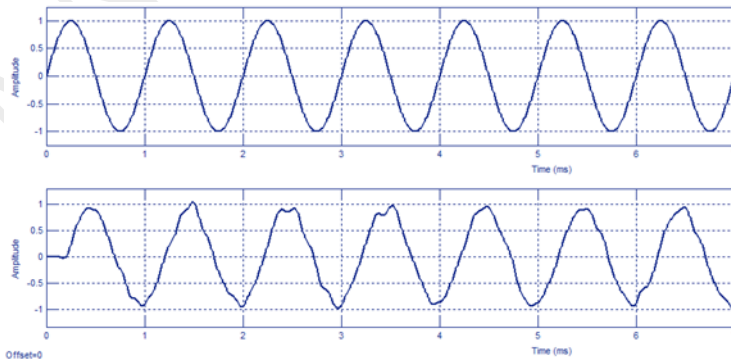
Merubah nilai *variance* dari sinyal noise acak yang dibangkitkan akan berpengaruh pada nilai S/N (*signal to noise ratio*). Semakin besar *variance* akan mengakibatkan nilai S/N yang semakin kecil.



Gambar 3.16. Model penerima AM dengan noise

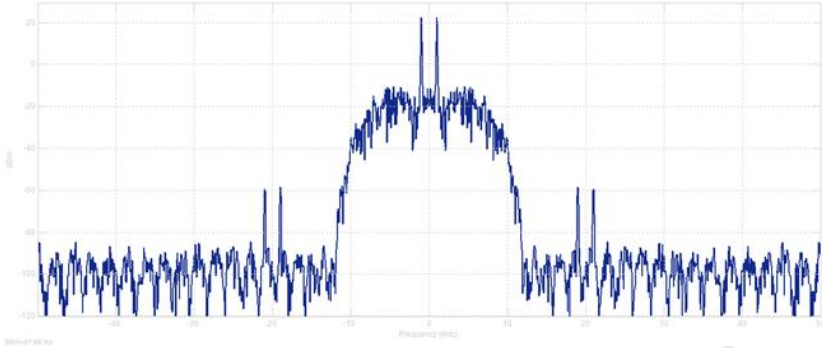
Hasil simulasi menunjukkan hal-hal sebagai berikut :

- Pada nilai variance 0,01 sinyal keluaran dari BPF ditunjukkan seperti pada gambar 3.17. Terlihat pada gambar bahwa bentuk sinyal tidak sinusoida sempurna karena adanya noise. Jika di plot bentuk spektrum frekuensinya, terlihat bahwa ada energi sinyal di semua frekuensi sebagai akibat adanya energi dari sinyal noise. Frekuensi spektrum ditunjukkan pada gambar 3.18



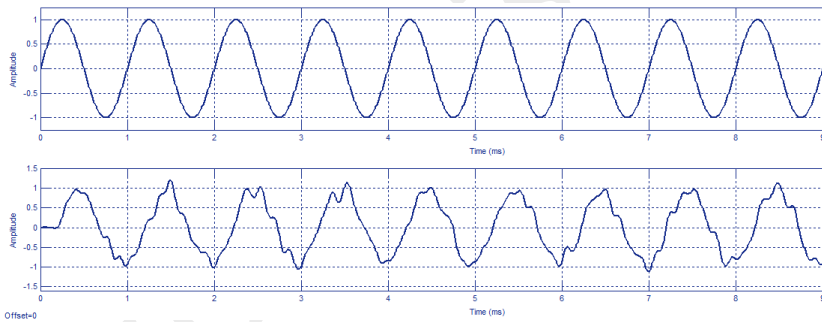
Gambar 3.17. Keluaran BPF dengan noise





Gambar 3.18. Spektrum keluaran sinyal dengan noise

- Pada nilai variance 0,05 sinyal keluaran dari BPF ditunjukkan seperti pada gambar 3.19. Terlihat pada gambar bahwa bentuk sinyal mengalami semakin banyak gangguan karena ada sinyal noise yang lebih besar.



Gambar 3.19. Sinyal keluaran dengan noise

MNC Publishing



## **BAB 4**

### **MODULASI FM**

MNC Publishing

Pada bab sebelumnya kita telah membahas tentang modulasi AM. Pada bab ini kita akan mempelajari tentang Modulasi FM yang merupakan salah satu jenis dari modulasi sudut. Dibandingkan dengan AM, modulasi FM dapat memberikan perbaikan kinerja sinyal terhadap adanya noise.

Dalam modulasi FM, frekuensi sinyal carrier dibuat bervariasi mengikuti frekuensi pada sinyal informasi. Jika sinyal carrier dinyatakan dengan  $s(t) = A \cos(\omega t)$ , nilai frekuensi sesaat dari sinyal carrier dapat dinyatakan dengan Persamaan :

$$\omega = \omega_c + k_f m(t)$$

Dimana  $k_f$  adalah konstanta FM dengan satuan Hz/V.

Sedangkan hubungan antara frekuensi dan fasa adalah :

$$\omega = \frac{d\theta}{dt}$$

Selanjutnya kita bisa menurunkan persamaan untuk menentukan nilai  $\theta$  sebagai berikut :

$$\begin{aligned} \theta &= \int_{-\infty}^t \omega dt \\ &= \int_{-\infty}^t [\omega_c + k_f m(t)] dt \\ &= \omega_c t + k_f \int_{-\infty}^t m(t) dt \end{aligned}$$

Sinyal FM yang dihasilkan oleh modulator dinyatakan dengan persamaan berikut :

$$\begin{aligned} y(t) &= A \cos(\theta) \\ &= A \cos \left[ \omega_c t + k_f \int_{-\infty}^t m(t) dt \right] \end{aligned}$$

Dari persamaan sinyal FM di atas, dapat kita lihat bahwa nilai frekuensi sinyal bervariasi mengikuti persamaan  $k_f \int_{-\infty}^t m(t) dt$  yang kalau di-diferensialkan menjadi :

$$\begin{aligned} \Delta\omega &= \omega - \omega_c \\ \Delta\omega &= k_f m(t) \\ \Delta\omega &= k_f |m(t)|_{max} \end{aligned}$$

Atau

$$\Delta f = \frac{k_f |m(t)|_{max}}{2\pi}$$

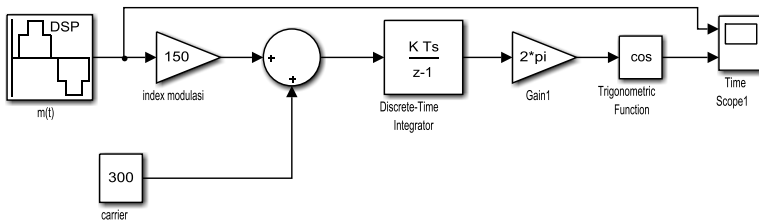
$\Delta\omega$  atau  $\Delta f$  disebut dengan maksimum deviasi frekuensi, dapat digunakan untuk menentukan nilai index modulasi dalam modulasi FM sebagai berikut :

$$\beta = \frac{\Delta f}{B}$$

dimana B adalah besarnya bandwidth sinyal informasi.

## Modulator

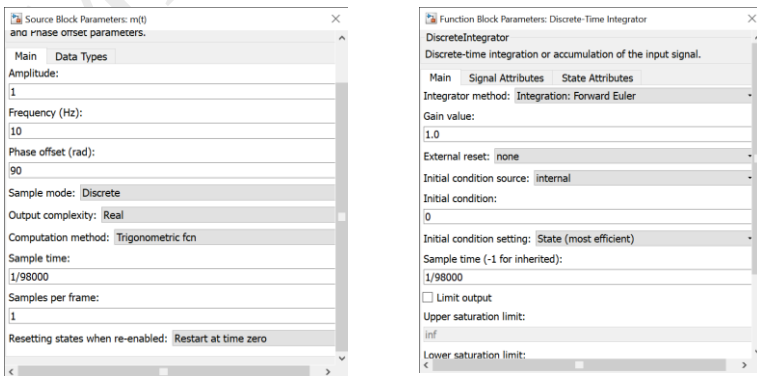
Untuk menganalisa proses modulasi FM, kita dapat melakukan simulasi menggunakan Simulink seperti ditunjukkan pada gambar 4.1 berdasarkan persamaan sinyal termodulasi  $y(t)$  seperti telah dijelaskan di atas.



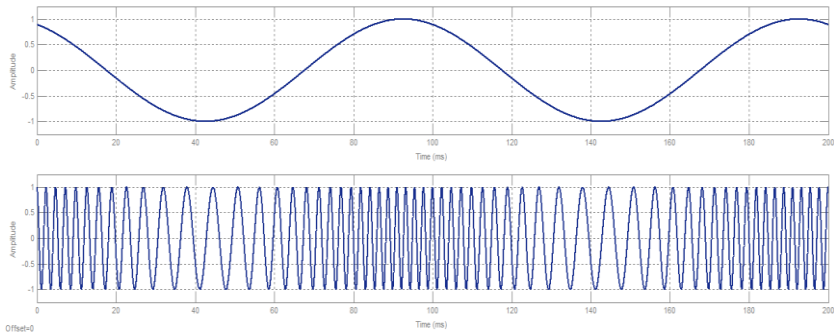
Gambar 4.1. Model simulasi modulator FM

Nilai parameter yang digunakan dalam simulasi ini adalah seperti ditunjukkan pada gambar 4.2.

Setelah dijalankan, sinyal keluaran modulator FM (sinyal termodulasi FM) seperti ditunjukkan pada gambar 4.3. Sinyal pada baris atas adalah sinyal informasi dengan frekuensi 10 Hz, sedangkan sinyal pada baris di bawah adalah sinyal termodulasi dengan indeks modulasi sebesar 150.



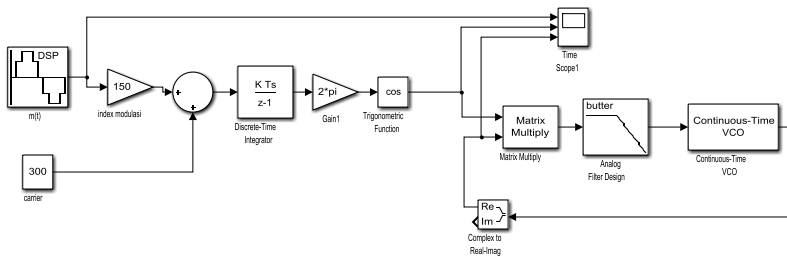
Gambar 4.2. Nilai parameter simulasi



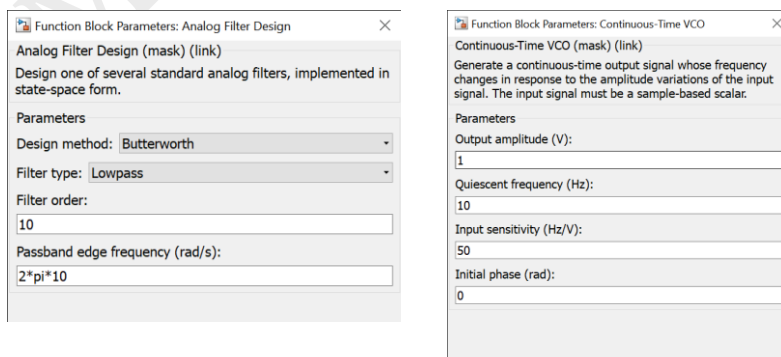
Gambar 4.3. Sinyal termodulasi FM

## Demodulator

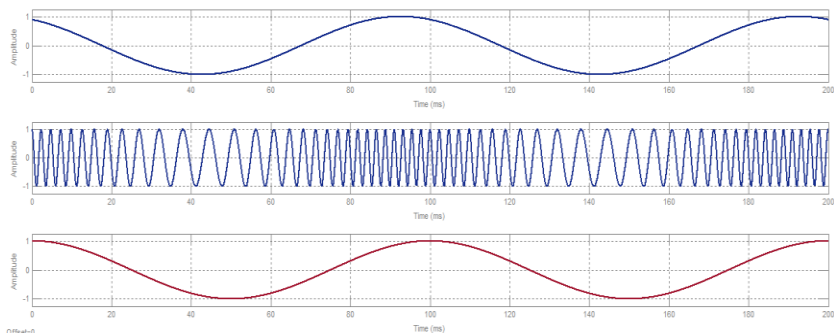
Berikutnya, marilah kita pelajari proses yang terjadi pada demodulator. Setelah sinyal termodulasi FM ditransmisikan melalui medium, selanjutnya diterima di receiver dan masuk proses demodulasi. Proses ini dapat disimulasikan menggunakan Simulink seperti ditunjukkan pada gambar 4.4. Sedangkan parameter yang digunakan ditunjukkan pada gambar 4.5.



Gambar 4.4. Model Simulink demodulator FM



Gambar 4.5 Parameter simulasi demodulator FM

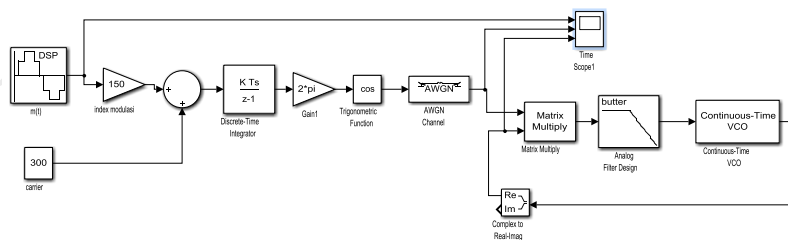


Gambar 4.6. Sinyal keluaran demodulator FM

Simulasi dijalankan selama 0,2 sec, menghasilkan sinyal seperti ditunjukkan pada gambar 4.6. Sinyal pada baris paling atas adalah sinyal informasi, baris tengah adalah sinyal termodulasi FM, dan pada baris terakhir adalah sinyal keluaran demodulator pada receiver. Seperti terlihat pada gambar, proses demodulator yang dilakukan berhasil merekonstruksi sinyal informasi sesuai dengan aslinya. Hanya terjadi delay karena adanya proses transmisi.

### Pengaruh Noise

Analisa berikutnya, kita akan melihat pengaruh noise pada sinyal termodulasi FM. Sama seperti pada simulasi pada modulasi AM pada bab sebelumnya, kita akan menambahkan blok kanal AWGN untuk membangkitkan variasi noise pada sinyal. Model simulasi ditunjukkan pada gambar 4.7.



Gambar 4.7. Model Simulink untuk modulasi FM dengan noise

Untuk melihat pengaruh noise terhadap sinyal yang diterima, parameter yang diubah dalam blok AWGN adalah :

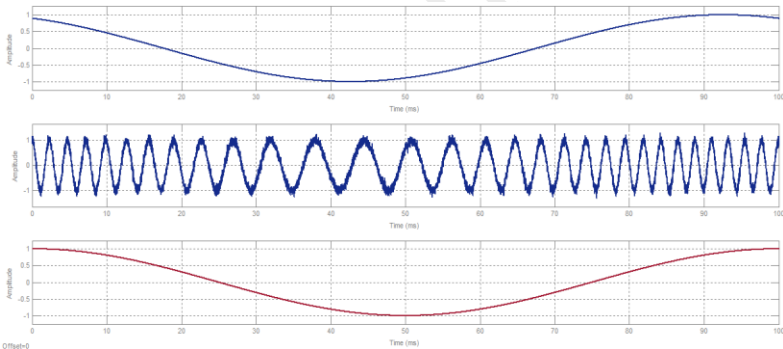
Mode : variance from mask

Variance : 0.01; 0.05; 0.5

Merubah nilai variance dari sinyal noise acak yang dibangkitkan akan berpengaruh pada nilai S/N (*signal to noise ratio*). Semakin besar variance akan mengakibatkan nilai S/N yang semakin kecil.

Hasil simulasi menunjukkan hal-hal sebagai berikut :

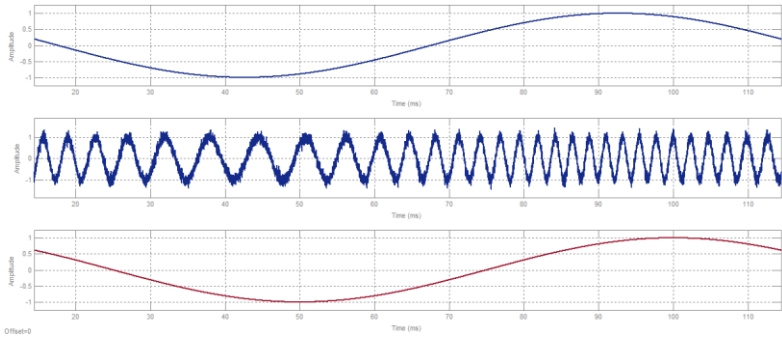
- Pada nilai variance 0,01 sinyal keluaran demodulator ditunjukkan seperti pada gambar 4.8. Terlihat pada gambar bahwa bentuk sinyal keluaran demodulator dapat dihasilkan kembali seperti sinyal informasi semula. Berbeda dengan yang terjadi pada modulasi AM dimana noise dapat mempengaruhi bentuk sinyal sinusoida. Sehingga bisa dikatakan bahwa modulasi FM lebih tahan terhadap noise dibandingkan dengan AM. Pada gambar 4.8 terlihat bahwa noise yang terjadi mempengaruhi bentuk sinyal termodulasi (ada ripple) pada bentuk sinyalnya, tetapi frekuensinya tetap (tidak terpengaruh).



Gambar 4.8. keluaran demodulator FM dengan noise

- Selanjutnya kalau kita coba lagi dengan menaikkan nilai sinyal noise sebesar dua kalinya (nilai variance 0.02), kita peroleh bentuk sinyal seperti ditunjukkan pada gambar 4.9. Kalau kita perhatikan gambar tersebut, kita lihat bahwa bentuk sinyal keluaran demodulator tetap bisa direkonstruksi dengan baik. Hanya pada bagian sinyal termodulasi yang menunjukkan adanya noise (*ripple*) yang lebih besar dibandingkan sebelumnya.





Gambar 4.9. sinyal keluaran demodulator dengan noise

MNC Publishing

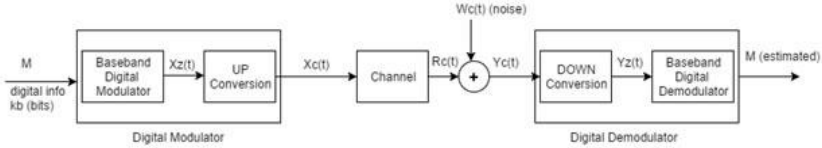


## **BAB 5**

### **PHASE SHIFT KEYING (PSK)**

## 5.1. MODULASI DIGITAL

Diagram blok model sistem komunikasi digital dapat ditunjukkan seperti pada gambar 5.1.



Gambar 5.1. Diagram blok sistem komunikasi digital

Komunikasi digital juga mempunyai blok modulator dan demodulator sama seperti pada komunikasi analog. Modulator menghasilkan sinyal analog yang merepresentasi informasi dalam bentuk data digital. Sedangkan demodulator menerima sinyal dari medium dan merekonstruksi menjadi informasi (data digital).

Pada jenis modulasi digital, sinyal pembawa berupa sinyal sinusoidal dan sinyal pemodulasi berupa sinyal digital. Berikut akan dibahas berbagai jenis modulasi digital, yaitu:

- FSK (*Frequency Shift Keying*)
- PSK (*Phase Shift Keying*)
- ASK (*Amplitude Shift Keying*)

Juga akan dibahas beberapa pengembangan dari masing-masing jenis modulasi tersebut.

### FSK (*Frequency Shift Keying*)

Pada modulasi FSK (diterjemahkan sebagai penguncian penggeseran frekuensi), sinyal pemodulasi yang berupa sinyal digital digunakan untuk memodulasi frekuensi sinyal pembawa sinusoidal. Jika sinyal informasi mempunyai logika "0" maka sistem akan mentransmisikan suatu frekuensi tertentu misalnya  $f_1$ , sedangkan jika sinyal informasi mempunyai logika "1" maka sistem akan mentransmisikan suatu frekuensi yang lain, misalnya  $f_2$ . Dengan demikian, sinyal FSK yang ditransmisikan adalah sinyal sinusoidal dengan amplitudo konstan, tetapi dengan frekuensi berbeda sesuai dengan arus data (sinyal pemodulasi). Berikut ilustrasi jenis modulasi FSK, dengan mentransmisikan frekuensi

tinggi untuk arus data logika "1" dan frekuensi rendah untuk arus data logika "0"

Bit "1" → frekuensi tinggi ( $f_2$ )

Bit "0" → frekuensi rendah ( $f_1$ )

FSK seperti yang telah dibahas di atas biasa disebut *Binary FSK*.

Dalam perkembangannya, FSK berkembang menjadi *M-ary FSK*.

Dalam *M-ary FSK*, ada sejumlah  $M$  sandi, setiap sandi tersusun atas

$n$  bit ( $M = 2^n$ ) dan dinyatakan dengan sebuah frekuensi tertentu, sehingga ada sejumlah  $M$  frekuensi yang berbeda yang digunakan.

Misal

$n = 2$  maka jumlah sandi =  $M = 2^2 = 4$

Maka

arus data "00" dinyatakan dengan frekuensi  $a$

arus data "01" dinyatakan dengan frekuensi  $b$

arus data "10" dinyatakan dengan frekuensi  $c$

arus data "11" dinyatakan dengan frekuensi  $d$

### **PSK (*Phase Shift Keying*)**

Pada modulasi PSK (diterjemahkan sebagai penguncian penggeseran fase), sinyal pemodulasi yang berupa sinyal digital digunakan untuk memodulasi fase sinyal pembawa sinusoidal. Jika sinyal informasi mempunyai logika "1" maka sistem akan mentransmisikan sinyal pembawa dengan suatu fase tertentu misalnya fase 0, sedangkan jika sinyal informasi mempunyai logika "0" maka sistem akan mentransmisikan sinyal pembawa dengan suatu fase yang lain, misalnya fase  $180^\circ$ . Dengan demikian, maka sinyal PSK yang ditransmisikan adalah sinyal sinusoidal dengan amplitudo konstan dengan fase yang sesuai dengan arus data pada sinyal informasi. Berikut ilustrasi jenis modulasi PSK, dengan mentransmisikan fase 0 untuk arus data logika "1" dan frekuensi rendah untuk arus data logika "0". Jenis modulasi PSK seperti ini disebut *Binary Phase Shift Keying* (BPSK)

Bit "1" → dinyatakan dengan fase 0

Bit "0" → dinyatakan dengan fase  $180^\circ$

Modulasi PSK juga berkembang menjadi beberapa bentuk, yaitu:

- a) DPSK (*Differential PSK*)
- b) DEPSK (*Differential Encoded PSK*)
- c) *M-ary PSK*

Pada *M-ary PSK*, ada sejumlah  $M$  sandi, setiap sandi tersusun atas  $n$  bit ( $M = 2^n$ ) dan dinyatakan dengan sebuah fase tertentu, sehingga ada sejumlah  $M$  fase yang berbeda yang digunakan.

Misal:

QPSK (*Quadrature PSK*)

$n = 2$  maka jumlah sandi =  $M = 4$

Maka:

arus data "00" dinyatakan dengan fase  $45^\circ$

arus data "01" dinyatakan dengan fase  $135^\circ$

arus data "10" dinyatakan dengan fase  $225^\circ$

arus data "11" dinyatakan dengan fase  $315^\circ$

### **ASK (*Amplitude Shift Keying*)**

Pada modulasi ASK (diterjemahkan sebagai penguncian penggeseran amplitude), sinyal pemodulasi yang berupa sinyal digital digunakan untuk memodulasi amplitude sinyal pembawa sinusoidal. Jika sinyal informasi mempunyai logika "1" maka sistem akan mentransmisikan sinyal pembawa dengan suatu amplitude, sedangkan jika sinyal informasi mempunyai logika "0" maka sistem akan mentransmisikan sinyal pembawa dengan suatu amplitude yang lain. Dengan demikian, maka sinyal ASK yang ditransmisikan adalah sinyal sinusoidal dengan frekuensi dan fase konstan namun dengan amplitude yang berubah-ubah sesuai dengan arus data pada sinyal informasi. Berikut ilustrasi jenis modulasi ASK, dengan mentransmisikan sinyal pembawa dengan amplitude  $A$  untuk arus data logika "1" dan sinyal pembawa dengan amplitude  $B$  untuk arus data logika "0"

Bit "1"  $\rightarrow$  dinyatakan dengan amplitudo  $A$

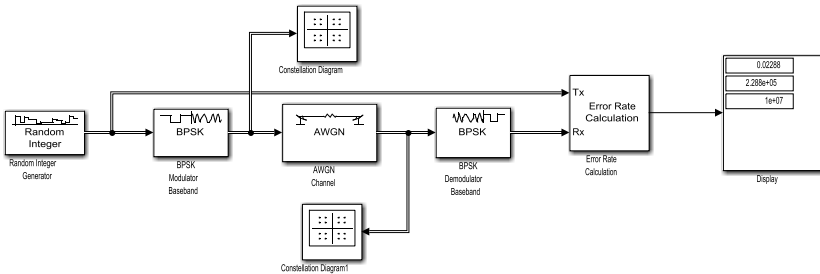
Bit "0"  $\rightarrow$  dinyatakan dengan amplitudo  $B \neq A$

PSK dan ASK berkembang menjadi QASK (*Quadrature Amplitude Shift Keying*). Pada QASK, sinyal yang

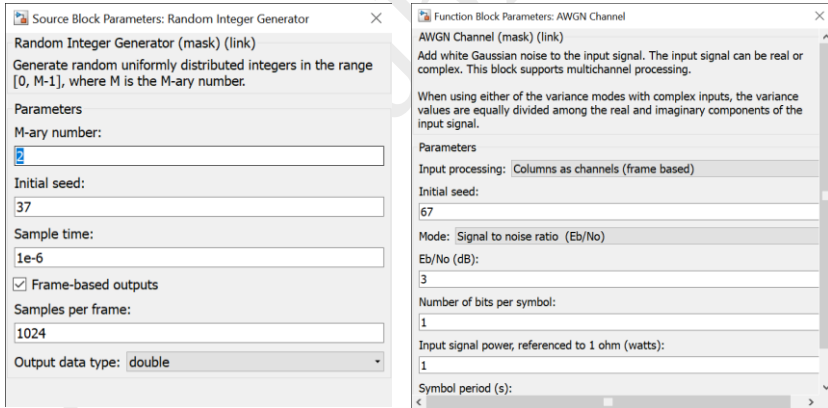
ditransmisikan adalah sinyal sinusoidal dengan fase dan amplitudo yang berbeda-beda sesuai dengan arus data.

## 5.2. BINARY PHASE SHIFT KEYING ( BPSK )

Berikut ini akan kita bahas simulasi BPSK untuk menghitung nilai BER (*bit error rate*). Model simulasi ditunjukkan seperti pada gambar 5.2.



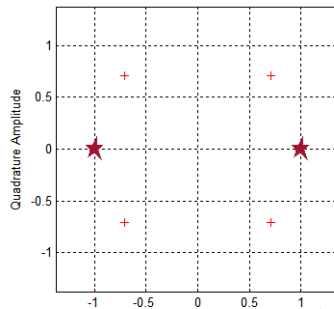
Gambar 5.2. Model Simulink untuk simulasi BPSK



Gambar 5.3. parameter simulasi

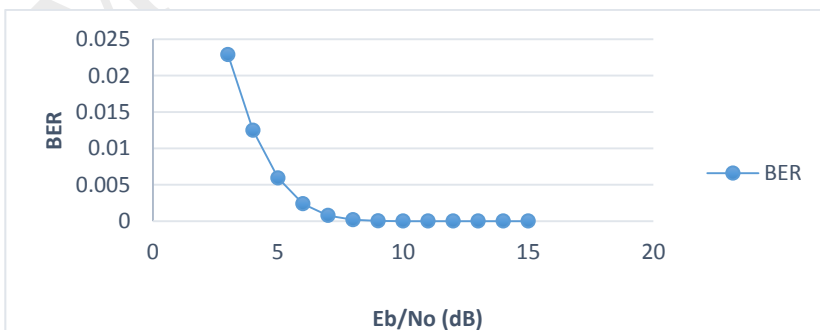
Blok *random integer generator* berfungsi untuk membangkitkan sinyal biner yang merupakan sinyal informasi yang akan ditransmisikan menggunakan modulasi BPSK. Parameter yang digunakan pada simulasi ditunjukkan pada gambar 5.3. Sinyal biner yang dibangkitkan kemudian dimodulasi menggunakan skema BPSK. Keluaran modulator akan dilihat konstelasi sinyalnya. Seperti kita ketahui, dalam modulator BPSK hanya mempunyai dua simbol,

masing-masing membawa satu bit data yaitu  $s_0$  untuk bit 0 dan  $s_1$  untuk bit 1. Sehingga konstelasi sinyalnya akan terdiri dari dua simbol pada bidang kompleks (IQ) dengan masing-masing simbol berada di sumbu *In-phase* (real) dan berbeda fasa  $180^\circ$ . Gambar konstelasi sinyal ditunjukkan seperti pada gambar 5.4.



Gambar 5.4. Konstelasi sinyal BPSK

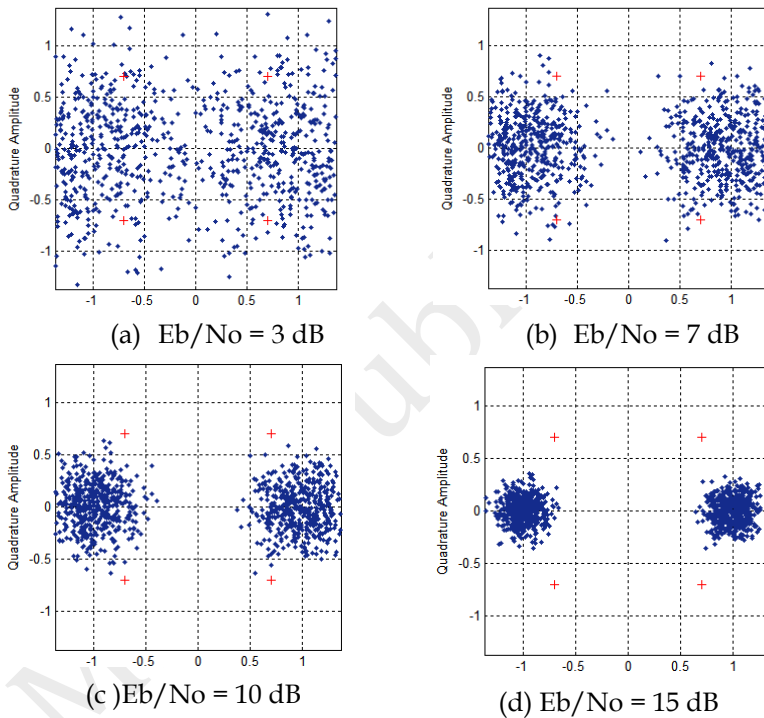
Selanjutnya untuk mengetahui nilai BER terhadap perubahan nilai  $E_b/N_0$ , nilai  $E_b/N_0$  dibuat bervariasi mulai 3 sampai 15 dB dengan kenaikan setiap 1 dB. Setiap perubahan nilai dicatat BER yang terjadi, kemudian digambar grafik BER fungsi  $E_b/N_0$ . Grafik BER ditunjukkan pada gambar 5.5. Nilai  $E_b/N_0$  yang semakin besar menunjukkan bahwa noise yang terjadi semakin kecil. Sehingga error yang terjadi juga semakin kecil, pada data tersebut untuk  $E_b/N_0$  mulai dari 12 dB sudah tidak menyebabkan error pada data yang diterima.



Gambar 5.5. Grafik nilai BER terhadap  $E_b/N_0$



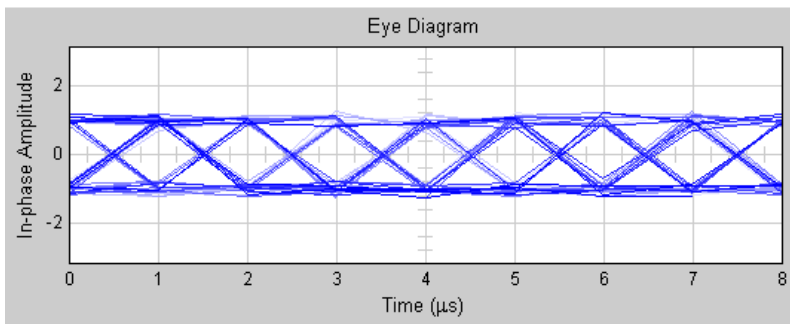
Bila diamati pengaruh noise terhadap bentuk konstelasi sinyal BPSK, bisa kita lihat pada gambar 5.6. Dalam gambar tersebut, bisa kita lihat bahwa pada daya noise yang besar, konstelasi sinyalnya menunjukkan letak titik-titik yang menyebar. Hal tersebut menunjukkan probabilitas error yang besar. Konstelasi yang ideal adalah seperti ditunjukkan pada gambar 5.4 sebelumnya. Dengan bertambahnya nilai  $E_b/N_0$  yang mengindikasikan pengaruh noise yang semakin kecil, terlihat bahwa letak titik-titik pada konstelasi semakin terpusat di dua lokasi seperti pada konstelasi tanpa noise.



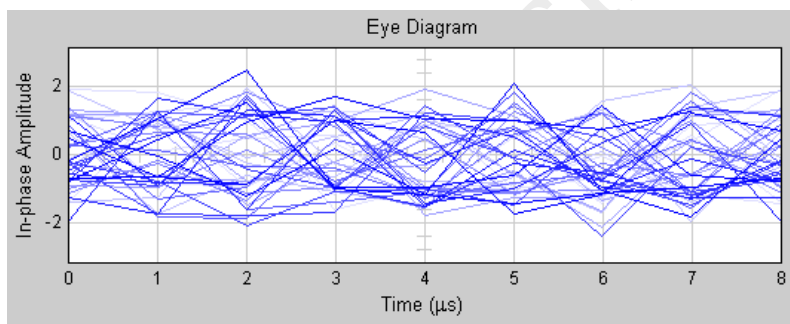
Gambar 5.6. Konstelasi sinyal BPSK dengan noise

Cara lain untuk melihat bentuk sinyal termodulasi digital adalah melalui grafik pola mata (*eye pattern*). Sinyal yang tidak mengalami error/ error kecil, grafik pola mata akan terbuka di tengah. Gambar 5.7 menunjukkan grafik pola mata untuk sinyal BPSK dengan  $E_b/N_0 = 15$  dB, dimana pada hasil simulasi tidak terjadi error. Terlihat pada gambar bahwa pola yang dihasilkan mendekati ideal. Hal ini berbeda pada grafik yang ditunjukkan pada

gambar 5.8 yang merupakan pola mata untuk sinyal BPSK dengan noise yang besar ( $E_b/N_0 = 3 \text{ dB}$ ). BER yang dihasilkan sebesar 0.02288, atau sekitar 2 bit dalam setiap 100 bit yang ditransmisikan.



Gambar 5.7. Grafik pola mata sinyal BPSK tanpa error



Gambar 5.8. Grafik pola mata sinyal BPSK dengan error



## **BAB 6**

### **GNU RADIO DAN APLIKASINYA**

GNU Radio adalah *software* gratis berbasis *open source* yang digunakan untuk memproses sinyal di udara yang ditangkap antenna. GNU Radio merupakan aplikasi yang digunakan untuk menimplementasikan sistem SDR (*Software Defined Radio*). GNU radio menggunakan sistem operasi Linux. GNU radio banyak digunakan pada sistem komunikasi radio, contohnya radar, satelit, jaringan GSM dan CDMA, dan lain-lain. GNU radio bekerja pada komputer yang mempunyai RF eksternal seperti USRP.

GNU radio memungkinkan penggunaanya untuk melakukan perancangan, simulasi, pengukuran pada USRP. Pada umumnya GNU radio menggunakan *software python* untuk bahasa pemrogramannya, namun juga bisa dikombinasikan dengan menggunakan bahasa pemrograman dev C++ untuk hasil yang lebih akurat.

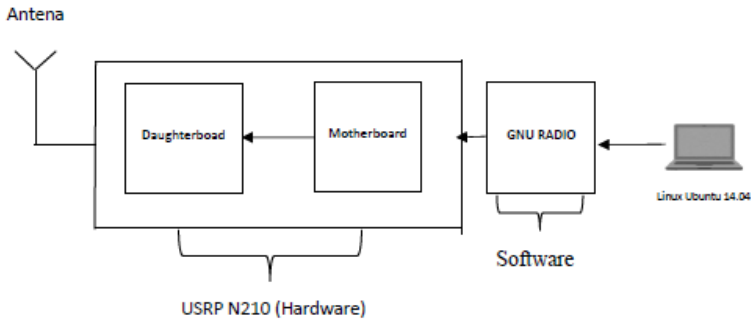
### **Studi kasus Aplikasi GNU Radio**

Pada bagian ini, kita akan membahas tentang salah satu aplikasi SDR menggunakan platform USRP N210. Aplikasi yang dibuat adalah perancangan pemancar TV digital standar T2 (DVB-T2). Perancangan alat ini meliputi identifikasi alat dengan membaca spesifikasi yang tertera dari alat yang digunakan. Selanjutnya, dilakukan proses unduh dan instalasi perangkat lunak SDR. Setelah proses unduh dan instalasi selesai, melakukan pemodelan sistem pemancar TV digital. Terakhir, mengeksplor program ke USRP N210.

Dalam membangun perangkat Pemancar TV Digital diperlukan beberapa komponen sebagai berikut:

1. Sistem operasi pada *personal computer* yang digunakan adalah Linux Ubuntu 14.04 sebagai pengimplementasian SDR (*Software Define Radio*).
2. Pada *Motherboard* terjadi pemrosesan algoritma yang sudah di modelkan pada GNU Radio. *Motherboard* ini juga berperan dalam proses konversi kecepatan data serta mengontrol dan menyiapkan sinyal yang diinginkan untuk dikirim yang dimana pada proses ini yaitu mengubah sinyal digital menjadi sinyal analog.

3. Daughterboard disini berfungsi untuk merubah gelombang radio yang akan dipancarkan oleh antena menjadi sinyal *IF (Intermediate Frequency)*.



Gambar 6.1. Diagram blok penggunaan SDR untuk DVB-T2

### Proses Instalasi Linux Ubuntu 18.04

Instalasi sistem operasi linux ini menggunakan teknik dual boot karena awalnya pada *host computer* sudah terdapat sistem operasi windows. Penjelasan rinci langkah-langkah sistem operasi linux ubuntu 18.04 adalah:

1. Memastikan *host computer* memenuhi spesifikasi minimum dari ubuntu. Berikut adalah persyaratannya:
  - *Dual Core Processor*: 2 GHz (64-bit) *BIOS Legacy*
  - *RAM*: 4 GB
  - *Space pada hard disk*: 25 GB
  - Skema partisi *hard disk*: *MBR (Master Boot Record)*
  - Resolusi layar: 1024×768

Catatan: *Host Computer* sudah mendukung arsitektur 64-bit

2. Menyediakan perangkat dan unduh beberapa aplikasi pendukung yang akan digunakan, yaitu :
  - Akses Internet
  - Berkas ISO Ubuntu 18.04 LTS
  - EasyBCD
  - Rufus
  - USB Flashdisk dengan kapasitas minimum 4 GB

3. Melakukan partisi *hard disk* untuk memisahkan sistem operasi windows 10 dan linux ubuntu 18.04. Adapun langkah-langkahnya sebagai berikut:
  - Ketik **diskmgmt.msc** pada pencarian Windows untuk membuka *Disk Management*
  - Pada partisi (D:) klik kanan lalu pilih Shrink Volume
  - Masukkan kapasitas yang diinginkan untuk partisi baru tersebut lalu klik Shrink. Setelah selesai akan tampil sebuah partisi kosong (*unallocated*) sesuai kapasitas yang diinginkan.
4. Membuat media install pada flashdisk dengan cara menyambungkan flashdisk ke komputer lalu buka aplikasi Rufus. Setelah terbuka pilih menu Select lalu pilih berkas ISO Ubuntu 18.04 LTS yang telah di unduh. Biarkan semua pengaturan dalam keadaan 'default'. Klik Start untuk mulai proses. Setelah proses selesai akan terlihat flashdisk berisi data-data dari berkas ISO Ubuntu 18.04 LTS. Ini menandakan bahwa flashdisk telah berubah menjadi media instalasi dan siap digunakan. Untuk tahapan selanjutnya membutuhkan proses *restart*.
5. Saat menjalani proses *restart* lalu tekan tombol **Esc** untuk membuka *Boot Device Options*. Setelah itu lakukan proses booting menggunakan *flashdisk* untuk masuk ke sistem operasi Linux Ubuntu 18.04. Setelah berhasil boot dari flashdisk, akan tampil wizard dengan opsi 'Try Ubuntu' dan 'Install Ubuntu'. Pilih "Try Ubuntu" karena partisi kosong yang sudah dibuat tadi belum diatur.
6. Mengatur partisi dengan cara Klik ikon **Menu** di kiri bawah lalu buka aplikasi **GParted**. Pilih partisi '*Unallocated*' lalu klik ikon **Create New Partition** di kiri atas (atau bisa juga dengan klik kanan pada partisi lalu pilih **New**).
7. Setelah berubah menjadi partisi '*Extended*', Sekarang buat partisi baru di dalamnya. Lakukan cara yang sama seperti di atas untuk

membuat tiga partisi baru. Adapun pengaturan partisi yang diperlukan adalah sebagai berikut:

- Pada partisi pertama, isi kapasitas sebesar 2 GB lalu beri nama 'Boot' pada label. Partisi ini nantinya digunakan untuk menyimpan berkas *GRUB* (berkas untuk melakukan proses *boot*).
  - Pada partisi kedua, isi kapasitas sebesar 8 GB lalu ubah bagian 'File system' menjadi *linux-swap* dan beri nama 'Swap' pada label. Partisi ini nantinya digunakan untuk membantu kinerja RAM. Jumlah kapasitas ditentukan berdasarkan 2x RAM *host computer* yang digunakan.
  - Pada partisi ketiga, isi kapasitas yang masih tersisa lalu beri nama 'Home' pada label. Partisi ini nantinya digunakan untuk menyimpan berkas-berkas pribadi. Setelah pembagian ketiga partisi telah selesai silahkan *apply*.
8. Memastikan computer telah terhubung ke internet dan lakukan instalasi ubuntu 18.04 dengan langkah-langkah sebagai berikut :
- Klik *Install Ubuntu 18.04 LTS* pada *desktop*.
  - Pilih bahasa yang ingin digunakan saat proses instalasi lalu klik *Continue*.
  - Pilih tata letak *keyboard* yang ingin digunakan lalu klik *Continue*.
  - Pilih *Normal installation* dan ceklis kedua opsi yang ada lalu klik *Continue*.
  - Pada 'Installation type' pilih *Something else* lalu klik *Continue*.
  - Pada partisi yang sudah diatur menjadi 'Boot' klik *Change*. Ubah 'Use as' menjadi *Ext4*, ceklis *Format the partition* dan ubah 'Mount point' menjadi */boot*, lalu klik *OK*.
  - Pada partisi yang sudah diatur menjadi 'Home' klik *Change*. Ubah 'Use as' menjadi *Ext4*, ceklis *Format the partition* dan ubah 'Mount point' menjadi */home*, lalu klik *OK*.
  - Ubah 'Device for boot loader installation' menjadi pada partisi 'Boot' (pastikan kembali bagian ini dipilih pada partisi 'Boot' karena kalau tidak diubah akan menimpa *boot OS Windows*). Setelah semua sudah benar klik *Install Now*.

- Pilih kota lalu klik *Continue*.
  - Isi semua kolom yang tersedia lalu klik *Continue*.
  - Sekarang hanya perlu menunggu proses instalasi selesai. Lama proses instalasi juga dipengaruhi oleh kecepatan internet karena proses instalasi ikut mengunduh pembaruan. Setelah proses instalasi selesai akan tampil dua opsi, yaitu '*Continue Testing*' dan '*Restart Now*'. Pilih *Restart Now* karena masih ada satu tahap lagi yang perlu dilakukan.
9. Tahapan terakhir adalah menambahkan pilihan boot untuk Linux Ubuntu 18.04 LTS melalui aplikasi EasyBCD. Saat restart computer akan langsung masuk ke Windows tanpa ada opsi untuk memilih sistem operasi yang ingin digunakan. Hal tersebut terjadi karena sebenarnya Linux Ubuntu 18.04 LTS telah terinstalasi dan hanya perlu menambahkan opsi boot-nya. Silakan buka aplikasi EasyBCD untuk menambahkan boot Ubuntu 18.04 LTS. Adapun Langkah-langkah yang harus dilakukan adalah sebagai berikut:
- Klik *Add New Entry* dan pada bagian '*Operating Systems*' pilih Linux/BSD. Beri nama '*Ubuntu 18.04 LTS*' dan ubah bagian '*Drive*' menjadi *Partition 4 (Linux - 1GB)* lalu klik (+) *Add Entry* sekali saja.
  - Klik *Edit menu boot* untuk memeriksa apakah *boot* Ubuntu sudah berhasil ditambahkan atau belum.

## 10. Restart Host Computer

### Instalasi UHD

UHD merupakan driver untuk perangkat USRP. Proses instalasinya adalah sebagai berikut :

- Melakukan perbaruan dan instalasi terhadap *dependencies* yang dibutuhkan oleh *UHD* dan *GNU Radio*

Sebelum membangun *UHD* dan *GNU Radio*, terlebih dahulu harus dilakukan perbaruan terhadap paket *file* yang sudah terdapat didalam *host computer* serta instalasi terhadap komponen-komponen yang dibutuhkan oleh *driver* dan aplikasi



tersebut (*dependencies*).

### 1. Perbaruan

Catatan: Perintah untuk melakukan perbaruan pada sistem atau paket file berbeda pada tiap sistem operasi. Berikut adalah perintah untuk Linux Ubuntu :

```
sudo apt-get update
```

### 2. Instalasi *dependencies* yang dibutuhkan oleh UHD dan GNU Radio

### 3. Setelah proses instalasi pada *dependencies* selesai, lakukan reboot.

#### a) Melakukan instalasi terhadap UHD melalui kode sumber

Ada beberapa cara untuk melakukan instalasi terhadap UHD, yaitu dapat melalui binaries, source code, dan pybombs. Akan tetapi penulis menyarankan untuk melakukan instalasi dengan melalui source code karena kelemahan jika melalui binaries adalah paket biner tersebut tidak tersedia pada semua platform dan jika melalui pybombs kita tidak memiliki kontrol penuh terhadap apa yang kita instalasikan. Untuk dapat membangun UHD dari kode sumber, kita dapat melakukan kloning pada github. Pastikan pada tahap ini perangkat USRP belum terpasang.

#### 1) Buat *folder* untuk tempat penyimpanan

```
cd $HOME mkdir  
workarea cd  
workarea
```

#### 2) Lakukan kloning pada tempat penyimpanan di github dan masuk kepadadirektori hasil kloning.

```
git clone https://github.com/EttusResearch/uhd  
cd uhd
```

#### 3) Lakukan pengecekan terhadap versi UHD yang tersedia. Daftar lengkapnya bisa didapatkan dengan melakukan perintah dibawah ini.

```
git tag -l
```

#### 4) Setelah melakukan pengecekan dan menemukan versi UHD yang diinginkan, jalankan perintah berikut

```
Example: For UHD 3.15.0.0
```

git checkout v3.15.0.0

- 5) Buat *folder* dengan tempat penampungan telah tersedia didalamnya serta gunakan *CMake* dan bangun *UHD*

```
cd host
mkdir build
cd build
cmake ../
make
```
- 6) Jalankan beberapa tes dasar untuk memastikan bahwa proses pembangunan *UHD* telah selesai dengan tepat

```
make test
```
- 7) gsh Lakukan instalasi *UHD*

```
sudo make install
```
- 8) Lakukan perbaruan pada *lybrary* sistem

```
sudo ldconfig
```

pastikan bahwa variabel lingkungan `LD_LIBRARY_PATH` sudah didefinisikan dan termasuk di dalam folder dimana *UHD* diinstal. Paling umum, kita dapat menambahkan baris di bawah ini ke akhir file `$HOME / .bashrc`

```
export LD_LIBRARY_PATH=/usr/local/lib
```
- 9) Jika `LD_LIBRARY_PATH` telah tersedia, tambahkan baris dibawah ini pada akhir file `$HOME / .bashrc`

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```
- 10) Jika semua langkah telah selesai, tutup terminal untuk membiarkan perubahan tersebut berpengaruh pada sistem dan buka terminal baru.
- 11) *UHD* telah terinstalasi. Lakukan cek terakhir dengan melakukan perintah dibawah ini tanpa perlu ada perangkat *USRP* terhubung

```
uhd_find_devices
```

## Instalasi GNU Radio

Langkah-langkah untuk instalasi GNU radio adalah sebagai berikut

1. Buat folder untuk menampung tempat penyimpanan dengan melakukan command:

```
cd $ HOME cd workarea
```

2. Lakukan *clone* pada tempat penyimpanan dengan melakukan command:

```
git clone --recursive  
https://github.com/gnuradio/gnuradio
```

3. Masuk ke tempat penyimpanan dan periksa versi GNU Radio yang diinginkan

```
cd gnuradio
```

Untuk memeriksa gnuradio versi 3.8.0.0

```
git checkout v3.8.0.0
```

Atau untuk memeriksa versi maint-3.8

```
git checkout maint-3.8
```

4. Perbarui *submodules*

```
git submodule update --init --recursive
```

5. Lakukan beberapa tes dasar untuk memverifikasi bahwa proses pembuatan selesai dengan benar

```
make test
```

6. Install GNU Radio, menggunakan default install sebagai awalan, yang akan menginstal GNU Radio di bawah folder /usr/local/lib. Kita perlu menjalankan ini sebagai root sebagai izin pada folder itu

```
sudo make install
```

7. Perbarui cache pada library bersamaan dengan sistem

```
sudo ldconfig
```

8. Pada langkah ini, GNU Radio harus diinstall dan siap digunakan. Lakukan tes berikut:

```
gnuradio-config-info --version
```

```
gnuradio-config-info --prefix
```

```
gnuradio-config-info --enabled-components
```

9. Jalankan GNU Radio dengan melakukan command:

```
gnuradio-companion
```

## Implementasi Sistem

Penerapan sistem pada pemancar TV Digital standar T2 (*DVB-T2*) ini menggunakan aplikasi GNU Radio dengan rancangan diagram blok yang sesuai, sebagai implementasi dari fungsi yang dibutuhkan oleh Pemancar TV Digital standar T2 (*DVB-T2*).

Adapun diagram blok beserta fungsinya dan parameter-parameter di dalamnya yang dibutuhkan pada penerapan sistem pemancar TV digital standar T2 (*DVB-T2*) menggunakan GNU Radio:

### 1. *Options*

- ID: TV\_Transmitter
- Generate Options: WX GUI

Blok ini berfungsi untuk menerapkan parameter khusus untuk *flowgraph*, mengontrol jenis kode yang dihasilkan, dan menentukan nama *file* yang dihasilkan.

### 2. *Variable*

- ID: samp\_rate
- Value: 10M

Blok ini berfungsi untuk memetakan nilai variabel unik. Variabel pada parameter blok lain dapat menggunakan ID pada *variable block*. Parameter *value* yang terdapat pada blok ini adalah nilai yang dapat diubah secara *realtime*.

### 3. *File Source*

Blok ini berfungsi untuk membaca nilai data mentah dalam format biner dari file yang ingin ditentukan. File ini dapat berupa file yang ditangkap menggunakan blok file sink.

### 4. *BBHeader*

- Standard: DVB-T2
- FECFRAME Size: Normal
- Code Rate: 4/5
- Baseband Framing Mode: Normal
- In-band signaling: off

Blok ini berfungsi untuk memformat paket *transport stream* MPEG-2 kedalam kerangka pita dasar FEC dan menambahkan header 10-byte.

#### 5. *BBScrambler*

- Standard: DVB-T2
- FECFRAME Size: Normal
- Code Rate: 4/5

Blok ini berfungsi untuk mengacak frame baseband FEC dengan encoder PRBS. Input dari blok ini adalah panjang dari *frame baseband*. Output dari blok ini adalah variabel panjang dari *frame baseband* secara acak.

#### 6. *BCH Encoder*

- Standard: DVB-T2
- FECFRAME Size: Normal
- Code Rate: 4/5

Blok ini berfungsi untuk mengkodekan BCH (*Bose, Chaudhuri, Hocquenghem*). Input dari blok ini adalah variabel panjang dari FEC *baseband frames*. Output dari blok ini adalah variabel panjang dari FEC *baseband frames* yang sudah ditambahkan dengan BCH.

#### 7. *LDPC Encoder*

- Standard: DVB-T2
- FECFRAME Size: Normal
- Code Rate: 4/5

Blok ini berfungsi untuk mengkodekan LDPC FEC (*Low Density Parity Check*). Input dari blok ini adalah variabel panjang dari FEC *baseband frames* yang sudah ditambahkan dengan BCH. Output dari blok ini adalah *frame baseband* FEC yang sudah ditambahkan dengan LDPC.

#### 8. *Bit Interleaver*

- FECFRAME Size: Normal
- Code Rate: 4/5
- Constellation: 64QAM

Input dari blok ini adalah frame baseband FEC yang sudah ditambahkan dengan LDPC. Output dari blok ini adalah *bit interleaved cells*.

#### 9. DVB-T2 Modulator

- FECFRAME Size: Normal
- Constellation: 64QAM
- Constellation Rotation: On

Blok ini berfungsi untuk memodulasi DVB-T2 *cells*. Input dari blok ini adalah *bit interleaved cells*. Output dari blok ini adalah *cells* termodulasi sesuai dengan konstellasi yang ditentukan yaitu 64QAM.

#### 10. Cell/Time Interleaver

- FECFRAME Size: Normal
- Constellation: 64QAM
- FEC blocks per frame: 9
- TI blocks per frame: 3

Input dari blok ini adalah 64QAM *cells* termodulasi. Output dari blok ini adalah *cell and time interleaved 64QAM cells* termodulasi.

#### 11. Frame Mapper

- FECFRAME Size: Normal
- Code Rate: 4/5
- Constellation: 64QAM
- Constellation Rotation: On
- FEC blocks per frame: 9
- TI blocks per frame: 3
- Extended Carrier Mode: Normal
- FFT Size: 8K
- Guard Interval: 1/8
- L1 Constellation: 64QAM
- Pilot Pattern: PP4
- T2 frames per super-frame: 2
- Number of Data Symbols: 200
- PAPR mode: Off

Blok ini berfungsi untuk memetakan simbol T2. Input dari blok ini adalah 64QAM cells termodulasi. Output dari blok ini adalah T2 *frame*.

#### 12. *Frequency Interleaver*

- *Extended Carrier Mode*: Normal
- *FFT Size*: 8K
- *Pilot Pattern*: PP4
- *Guard Interval*: 1/8
- *Number of Data Symbols*: 200
- *PAPR mode*: Off

Blok ini berfungsi untuk memasukkan frekuensi kedalam T2 *frame*. Input dari blok ini adalah T2 *frame*. Output dari blok ini adalah T2 *frame* yang sudah ditambahkan frekuensi.

#### 13. *Pilot Generator and IFFT*

- *Extended Carrier Mode*: Normal
- *FFT Size*: 8K
- *Pilot Pattern*: PP4
- *Guard Interval*: 1/8
- *Number of Data Symbols*: 200
- *PAPR mode*: Off
- *Bandwidth*: 8 MHz

Blok ini berfungsi menambahkan *pilot* kedalam T2 *frame*. Input dari blok ini adalah T2 *frame* yang sudah ditambahkan frekuensi. Output dari blok ini adalah T2 *frame* yang sudah ditambahkan *pilot* (dengan domain waktu).

#### 14. *OFDM Cyclic Prefixer*

- *FFT Length*: 8.192k
- *CP Length*: 1.024k
- *Rollof*: 0

Blok ini berfungsi untuk melakukan pembentukan pulsa pada simbol OFDM. Input dari blok ini adalah simbol OFDM (dalam domain waktu setelah IFFT). Output dari blok ini adalah aliran simbol skalar yang mencakup putaran dan pembentukan pulsa.

### 15. P1 Symbol Insertion

- *Extended Carrier Mode*: Normal
- *FFT Size*: 8K
- *Guard Interval*: 1/8
- *Number of Data Symbols*: 200

Blok ini berfungsi untuk memasukkan simbol P1 pada T2 *frame*. Input dari blok ini adalah OFDM T2 *Symbol*. Output dari blok ini adalah OFDM T2 *frame* dengan simbol P1.

### 16. Multiply Const

- *Constant*: 200m

Blok ini berfungsi sebagai pengali antara input dengan skalar atau konstanta vektor.

### 17. Rational Resampler

- *Interpolation*: 70
- *Decimation*: 64

Blok ini berfungsi untuk melakukan pengulangan pada pengambilan sampel.

### 18. WX GUI FFT Sink

- *Title*: FFT Plot
- *Sample Rate*: 10M
- *Baseband frequency*: 522M
- *Y per div*: 10 dB
- *Y divs*: 10
- *Ref level (dB)*: 0
- *Ref scale (p2p)*: 2
- *FFT Size*: 1.024k
- *Refresh rate*: 15

Blok ini berfungsi untuk menampilkan FFT Plot.

### 19. UHD: USRP Sink

- *Samp Rate (Sps)*: 10M
- *Ch0 Center Frequency (Hz)*: 522M
- *Ch0 Gain Value*: 20
- *Ch0 Antenna*: TX/RX



- Ch0 *Bandwith*: 8M

Blok ini berfungsi untuk mengekspor program kedalam USRP N210

Selanjutnya, perlu dilakukan verifikasi penggunaan USRP N210 menggunakan UHD dan GNU Radio untuk memastikan perangkat dapat beroperasi dengan baik dan benar. Beberapa tahapan dalam memverifikasi USRP N210 adalah sebagai berikut:

1. Melakukan cek koneksi UHD dengan USRP N210 dengan *command*  
`uhd_usrp_probe`
2. Melakukan cek perangkat yang terhubung dengan UHD dengan *command*  
`uhd_find_devices`
3. Melakukan benchmark dengan *command*  
`./benchmark_rate --rx_rate 10e6 --tx_rate 10e6`
4. Melakukan *transmit test signal* dengan *command*  
`./tx_waveforms --freq 915e6 --rate 5e6 --gain 0`

Setelah menguji dan memastikan bahwa perangkat dapat beroperasi dengan benar, selanjutnya kita bisa melakukan pengujian pada daya pancar. Sistem akan diuji pada tiga parameter yaitu daya pancar sistem serta kapasitas kanal dan frekuensi pada tiap kanal yang dapat dipancarkan. Hasil dari pengujian sistem tersebut akan ditampilkan pada *FFT Plot* untuk mengetahui berapa nilai frekuensi pada tiap kanal dan *power* yang dihasilkan untuk digunakan pada perhitungan daya pancar (EIRP).

*FFT Plot* adalah sebuah instrumen tampilan yang didalamnya terdapat metode untuk mengubah sinyal dari domain waktu kedalam domain frekuensi dengan cepat sehingga menghasilkan nilai variasi yang tinggi dan detail. Caranya adalah mengidentifikasi masing-masing nilai besaran pada frekuensi-frekuensi yang terdapat dalam sebuah gelombang.

Pengujian dilakukan berdasarkan standar dan regulasi yang telah ditentukan oleh lembaga yang berwenang dalam bidang telekomunikasi di Indonesia terkait dengan pemancar TV digital

standar T2 yang terdapat pada Peraturan Menteri Komunikasi dan Informatika No.6 Tahun 2019 Tentang “Rencana Induk Frekuensi Radio untuk Keperluan Penyelenggaraan TV Siaran Digital pada Pita UHF”. Hal-hal yang diatur dalam peraturan tersebut yang akan digunakan untuk pengujian adalah sebagai berikut:

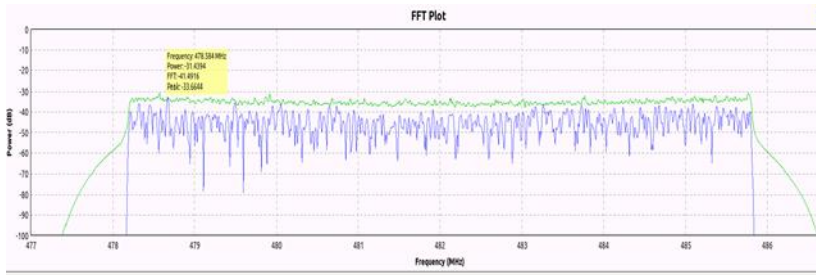
1. Frekuensi kerja yang dapat digunakan untuk televisi siaran digital terrestrial berbasis DVB-T2 adalah pada 478 MHz – 694 MHz.
2. Terdapat 27 kanal dengan nomor kanal pada Band IV adalah 22-37 dan pada Band V adalah 38-48 dengan masing-masing kanal memiliki *bandwidth* 8 MHz.

Pada kanal 22 akan dilakukan pengujian dengan nilai frekuensi tengah pada blok *UHD: USRP source* sebesar 482 MHz. Tujuan dari pengujian kanal 22 ini adalah untuk mengetahui nilai *peak frequency* dan *power* berdasarkan keluaran *WX GUI FFT Sink* di GNU Radio pada kanal 22.

Langkah - Langkah Pengujian

1. Rancang peralatan sesuai dengan blok diagram pada Gambar 4.34
2. Verifikasi penggunaan *USRP N210* menggunakan *UHD* dan *GNU Radio*
3. Buka aplikasi GNU Radio Companion
4. Buka file GRC yang telah dibuat
5. Atur nilai frekuensi tengah pada blok diagram *UHD: USRP Sink*
6. *Generate Flowgraph* untuk memastikan tidak ada kesalahan pada *flowgraph*
7. *Execute Flowgraph*

Dari hasil pengujian, diperoleh nilai spektrum frekuensi seperti ditunjukkan pada gambar 6.2.



Gambar 6.2. Spektrum frekuensi kanal 22

Berdasarkan gambar tersebut, pada nilai frekuensi tengah 482 MHz didapatkan *frequency peak* sebesar 478,58 MHz dan *power* sebesar -31,34 dB. Nilai EIRP pada kanal 22 ini adalah sebagai berikut:

$$\text{EIRP} = \text{PT}_x + \text{GT}_x - \text{LT}_x$$

$$\begin{aligned} \text{PT}_x &= 15 \text{ dBm} + (-31,34 \text{ dB}) \\ &= -16,34 \text{ dBm} \end{aligned}$$

$$\begin{aligned} \text{EIRP} &= -16,34 \text{ dBm} + 20 \text{ dB} - 0,2 \text{ dB} \\ &= 3,46 \text{ dBm} \end{aligned}$$

Sehingga dapat disimpulkan bahwa dengan frekuensi tengah sebesar 482 MHz nilai EIRP yang didapatkan sebesar 3,46 dBm. Hal ini menunjukkan bahwa pada kanal 22, USRP N210 dapat memancarkan sinyal dengan daya 3,46 dBm.

MNC Publishing

**Studi kasus:**  
**GNU Radio untuk PENERIMA TV DIGITAL STANDAR T2  
(DVB-T2) DENGAN MENGGUNAKAN USRP N210**

Tujuan yang ingin dicapai pada penelitian ini yaitu merancang USRP N-210 sebagai penerima TV digital standar T2 (DVB-T2) berupa Software Defined Radio yaitu GNU Radio. Selanjutnya perangkat penerima DVB-T2 berupa SDR akan dianalisis kapasitas kanalnya, serta frekuensi yang dapat digunakan pada setiap kanal. Metodologi yang digunakan

1. Perancangan dan implementasi penerima TV Digital DVB T2
2. Pengujian dan analisis penerima TV Digital DVB T2

## **1. Perancangan Penerima TV Digital Standar T2 (DVB-T2)**

### **1.1. Desain Umum Sistem**

Penerima TV Digital standar T2 (DVB-T2) dengan menggunakan *USRP N-210* dalam perancangannya melibatkan komponen-komponen yang terlibat dalam sebuah kesatuan sistem. Adapun secara garis besar terdapat sistem-sistem sebagai berikut:

- Sistem pada Daughterboard

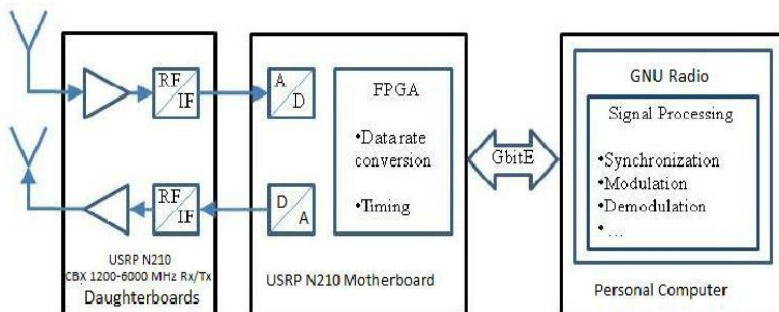
Sistem ini merupakan bagian penunjang dari sistem yang terdapat pada *USRP N-210* dimana didalam sistem *daughterboard* terdapat perangkat-perangkat yaitu:

- 1) *Antena*, yang berfungsi sebagai pengubah sinyal listrik menjadi sinyal elektromagnetik. Dan sebaliknya, antena juga dapat berfungsi untuk menerima sinyal elektromagnetik dari ruang bebas dan mengubahnya menjadi sinyal listrik.
- 2) *Transceiver*, yang berfungsi sebagai pemancar-penerima dapat digunakan untuk menghubungkan sebuah sistem ke sebuah jaringan sehingga sistem tersebut dapat memancarkan sekaligus menerima sinyal di dalam jaringan tersebut. Pada sistem ini, *transceiver* menjalankan fungsi sebagai penerima dengan ditentukan oleh program yang dibangun pada *GNU Radio*.

- 3) *Tuner* atau *Penala*, yang berfungsi sebagai pemilih kanal / stasiun dengan cara merubah gelombang radio yang diterima antena menjadi *signal IF (Intermediate Frequency)*. Didalam *Tuner* terdapat 3 rangkaian utama, yaitu: (1) Penguat frekuensi tinggi / Penguat RF (*RF Amplifier*), (2) Pencampur (*Mixer*) dan (3) Osilator lokal (*Local Oscillator*).
- Sistem pada *Motherboard*
    - 1) Sistem ini merupakan bagian utama dari sistem yang terdapat pada *USRP N-210* dimana didalam sistem *motherboard* terdapat perangkat-perangkat yaitu:
    - 2) *ADC (Analog to Digital Converter)*, berfungsi sebagai pengubah sinyal analog (sinyal kontinu) menjadi sinyal digital. Perangkat *ADC* dapat berbentuk suatu modul atau rangkaian elektronika maupun suatu *chip IC*. *ADC* juga berfungsi untuk menjembatani pemrosesan sinyal analog oleh sistem digital. Pada skripsi ini *ADC* berperan untuk mengubah sinyal analog yang diterima dari pemancar TV Digital untuk diteruskan ke *FPGA*.
    - 3) *DAC (Digital to Analog Converter)*, berfungsi sebagai pengubah sinyal digital (diskrit) menjadi sinyal analog (kontinu). Aplikasi *DAC* adalah sebagai antarmuka (*interface*) antara perangkat yang bekerja dengan sistem digital dan perangkat pemroses sinyal analog. Perangkat *DAC* dapat berupa rangkaian elektronika dan *chip IC DAC*.
    - 4) *Field-Programmable Gate Array (FPGA)*, berfungsi untuk pemrosesan komputasi dari algoritma integrasi numerik. Keuntungan implementasi *FPGA* digunakan untuk meningkatkan efisiensi rancangan dengan cara mengurangi pemakaian pemrograman perangkat lunak (*software*). *FPGA* mempunyai koreksi *error* yang kecil dan merupakan teknologi yang bebas (*technology-independent*) untuk diimplementasikan dalam berbagai algoritma. *FPGA* dapat berfungsi seperti kertas kosong yang bisa digunakan oleh para pemrogram untuk menulis instruksi-instruksi baru. Dalam sistem ini, *FPGA* juga

berperan dalam proses konversi kecepatan data serta mengontrol dan menyiapkan sinyal yang diinginkan untuk dikirim atau diterima dari dan ke *user*.

- Kabel Penghubung  
Pada sistem ini, dibutuhkan kabel *Gigabit Ethernet* berkecepatan tinggi yang berfungsi sebagai penghubung antara *computer (host)* dengan *USRP N-210*.
- Personal Computer  
Pada *personal computer* terdapat aplikasi *GNU Radio* yang berfungsi mengimplementasikan *system software defined radio* dengan melakukan perancangan, simulasi, serta pengukuran pada *USRP*. Secara khusus proses yang dilakukan adalah sinkronisasi, modulasi, dan demodulasi. Adapun blok diagram sistem tersebut ditunjukkan pada Gambar 1.



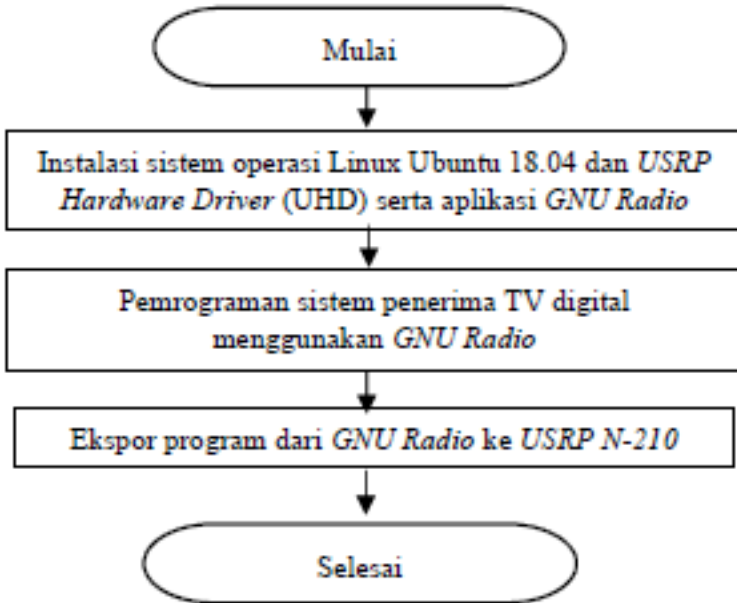
Gambar 1 Blok Diagram Sistem Penerima TV Digital standar T2 (DVB T2) dengan menggunakan *USRP N-210*

Sumber: Boppana, Lakshmi. 2013. "Concatenated Coding in OFDM for Wimax Using *USRP N210* and *GNU Radio*"

## 1.2. Instalasi Sistem

Instalasi sistem penerima TV Digital Standar T2 (DVB-T2) dengan menggunakan *USRP N-210* meliputi instalasi sistem operasi Linux Ubuntu 18.04 dan *USRP Hardware Driver (UHD)* serta aplikasi *GNU Radio*, pemrograman sistem penerima TV digital menggunakan *GNU Radio*, dan proses ekspor program yang telah dibuat pada *GNU Radio* ke *USRP N-210*. Diagram alir dari instalasi sistem penerima TV

Digital Standar T2 (*DVB-T2*) dengan menggunakan *USRP N-210* digambarkan dalam Gambar 2.



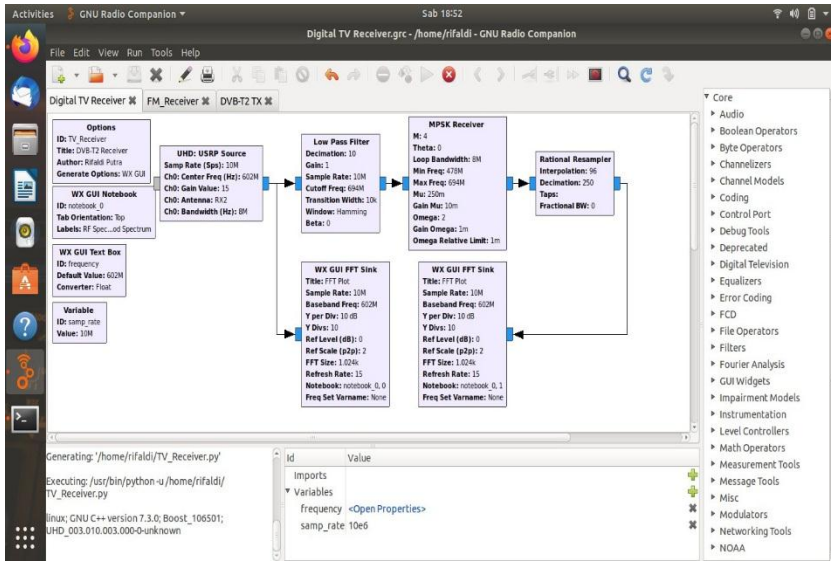
Gambar 2 Diagram Alir Perancangan Penerima TV Digital Standar T2 (*DVB T2*) Dengan Menggunakan *USRP N-210*  
Sumber : Perancangan (2019)

### 1.3. Implementasi Sistem

Sistem penerima TV digital ini akan menggunakan aplikasi *GNU Radio* untuk melihat apakah sistem yang dibangun dapat bekerja dan juga tentunya melihat parameter-parameter performansi yang sudah ditentukan yaitu kapasitas kanal yang dapat diterima serta frekuensi yang digunakan pada tiap kanalnya. Untuk dapat melakukan pengujian serta pembahasan, dibutuhkan implementasi sistem yang akan dilakukan pada perangkat keras dan perangkat lunak sistem.

Adapun blok *receiver* yang akan dibangun pada *GNU Radio* adalah sebagai berikut ditunjukkan pada Gambar 3





Gambar 3. Blok Receiver.grc  
 Sumber: Perancangan (2020)

#### 1.4. Implementasi Perangkat Keras

Perangkat keras yang digunakan pada rancang bangun sistem penerima TV digital ini sebagai *host computer* adalah *Laptop*. Spesifikasi lengkapnya terdapat pada Tabel 1.

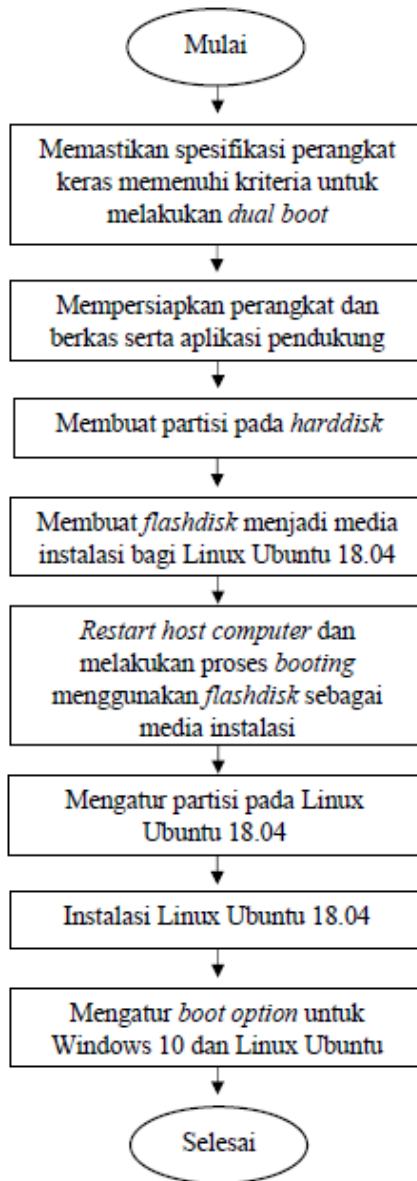
Tabel 1 Spesifikasi *Host Computer*

NO	ITEM	VALUE
1	OS Name	Microsoft Windows 10 Pro
2	Version	10.0.18362 Build 18362
3	Other OS Description	Linux Ubuntu 18.04
4	OS Manufacturer	Microsoft Corporation
5	System Name	DESKTOP-N118F80
6	System Manufacturer	Hewlett-Packard
7	System Model	HP Notebook
8	System Type	x64-based PC
9	System SKU	M7Q63PA#AR6

10	<i>Processor</i>	<i>Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz, 2201 Mhz, 2 Core(s), 4 Logical Processor(s)</i>
11	<i>BIOS Version/Date</i>	<i>Insyde F.02, 09/04/15</i>
12	<i>SMBIOS Version</i>	<i>2.8</i>
13	<i>Embedded Controller</i>	<i>94.13</i>
14	<i>BIOS Mode</i>	<i>Legacy</i>
15	<i>BaseBoard Manufacturer</i>	<i>Hewlett-Packard</i>
16	<i>BaseBoard Product</i>	<i>80BC</i>
17	<i>BaseBoard Version</i>	<i>KBC Version 94.0D</i>
18	<i>Platform Role</i>	<i>Mobile</i>
19	<i>Secure Boot State</i>	<i>Unsupported</i>
20	<i>PCR7 Configuration</i>	<i>Binding Not Possible</i>
21	<i>Windows Directory</i>	<i>C:\WINDOWS</i>
22	<i>System Directory</i>	<i>C:\WINDOWS\system32</i>
23	<i>Boot Device</i>	<i>\Device\HarddiskVolume1</i>
24	<i>Locale</i>	<i>United States</i>
25	<i>Hardware Abstraction Layer</i>	<i>Version = "10.0.18362.752"</i>
26	<i>User Name</i>	<i>DESKTOP-N118F80\Hewlett Packard</i>
27	<i>Time Zone</i>	<i>SE Asia Standard Time</i>
28	<i>Installed Physical Memory (RAM)</i>	<i>4,00 GB</i>
29	<i>Total Physical Memory</i>	<i>3,92 GB</i>
30	<i>Available Physical Memory</i>	<i>623 MB</i>
31	<i>Total Virtual Memory</i>	<i>5,43 GB</i>
32	<i>Available Virtual Memory</i>	<i>591 MB</i>
33	<i>Page File Space</i>	<i>1,50 GB</i>
34	<i>Page File</i>	<i>C:\pagefile.sys</i>
35	<i>Kernel DMA Protection</i>	<i>Off</i>
36	<i>Virtualization-based</i>	<i>Not enabled</i>

## 1.5. Implementasi Perangkat Lunak

### 1.5.1. Instalasi Sistem Operasi Linux Ubuntu 18.04 LTS



Instalasi sistem operasi Linux Ubuntu 18.04 LTS pada *host computer* ini dilakukan dengan teknik *dual boot* karena pada kondisi awal sudah terdapat sistem operasi Windows 10 Pro yang masih

dibutuhkan. Adapun penjelasan rinci langkah-langkah instalasi sistem operasi Linux Ubuntu 18.04 LTS adalah:

Tahapan Pertama yaitu pastikan bahwa spesifikasi dari *Host Computer* memenuhi persyaratan untuk melakukan *dual boot*. Spesifikasi minimum yang dibutuhkan adalah:

- *Dual Core Processor* : 2 GHz (64-bit) BIOS Legacy
- *RAM* : 4 GB
- *Space pada harddisk* : 25 GB
- *Skema partisi harddisk* : MBR (*Master Boot Record*)
- *Resolusi layar* : 1024×768

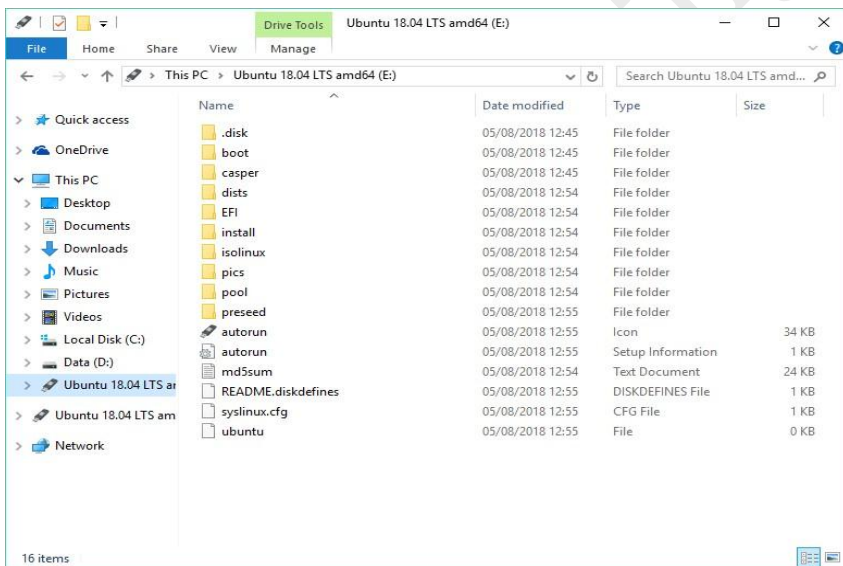
Tahapan kedua yaitu sediakan perangkat dan unduh berkas serta aplikasi pendukung yang akan digunakan. Perangkat dan berkas yang akan digunakan adalah sebagai berikut:

- Akses Internet dan USB *flashdisk* dengan kapasitas minimum 4 GB
- Berkas ISO Linux Ubuntu 18.04 LTS (Dapat diunduh pada situs resmi Ubuntu) Catatan: Pada skripsi ini saya menggunakan Ubuntu versi 18.04 karena pada saat saya melakukan penelitian versi tersebut adalah keluaran terakhir yang memiliki fasilitas *Long Term Support (LTS)* dimana keunggulan dari *LTS* ini adalah mendapat jaminan dukungan jangka panjang yaitu lima tahun sampai dengan tahun 2023 terkait pembaruan keamanan dan pemeliharaan.
- Aplikasi Rufus dan EasyBCD
- Catatan: Untuk pemilihan aplikasi dalam membuat *USB Bootable* saya memilih aplikasi Rufus dan untuk aplikasi dalam mengatur proses *booting* saya memilih aplikasi EasyBCD. Terdapat aplikasi-aplikasi lain seperti WinToFlash dan YUMI yang dapat menjadi pilihan.

Tahapan ketiga, lakukan partisi pada *harddisk* untuk memisahkan sistem operasi Windows 10 dan Linux Ubuntu 18.04 LTS. Caranya adalah buat partisi kosong pada *Disk Management* dengan cara klik kanan pada partisi (D:) lalu pilih *shrink volume*. Masukkan kapasitas yang diinginkan untuk partisi baru tersebut lalu klik *Shrink*. Di sini saya membuat kapasitas sebesar 55 GB. Setelah

selesai akan tampil sebuah partisi kosong (*unallocated*). Catatan: Salin data-data penting pada media eksternal sebagai pencegahan.

Tahapan keempat adalah membuat media *install* pada *flashdisk* dengan cara menyambungkan *flashdisk* ke komputer lalu buka aplikasi Rufus. Setelah terbuka pilih menu *Select* lalu pilih berkas ISO Ubuntu 18.04 LTS yang telah di unduh. Biarkan semua pengaturan dalam keadaan '*default*'. Klik *Start* untuk mulai proses. Setelah proses selesai akan terlihat *flashdisk* berisi data-data dari berkas ISO Ubuntu 18.04 LTS. Ini menandakan bahwa *flashdisk* telah berubah menjadi media instalasi dan siap digunakan. Untuk tahapan selanjutnya membutuhkan proses *restart*.



Gambar 4. Hasil Akhir Ketika *Flashdisk* Telah Dapat Berfungsi Sebagai Media Instalasi

Sumber: Perancangan (2020)

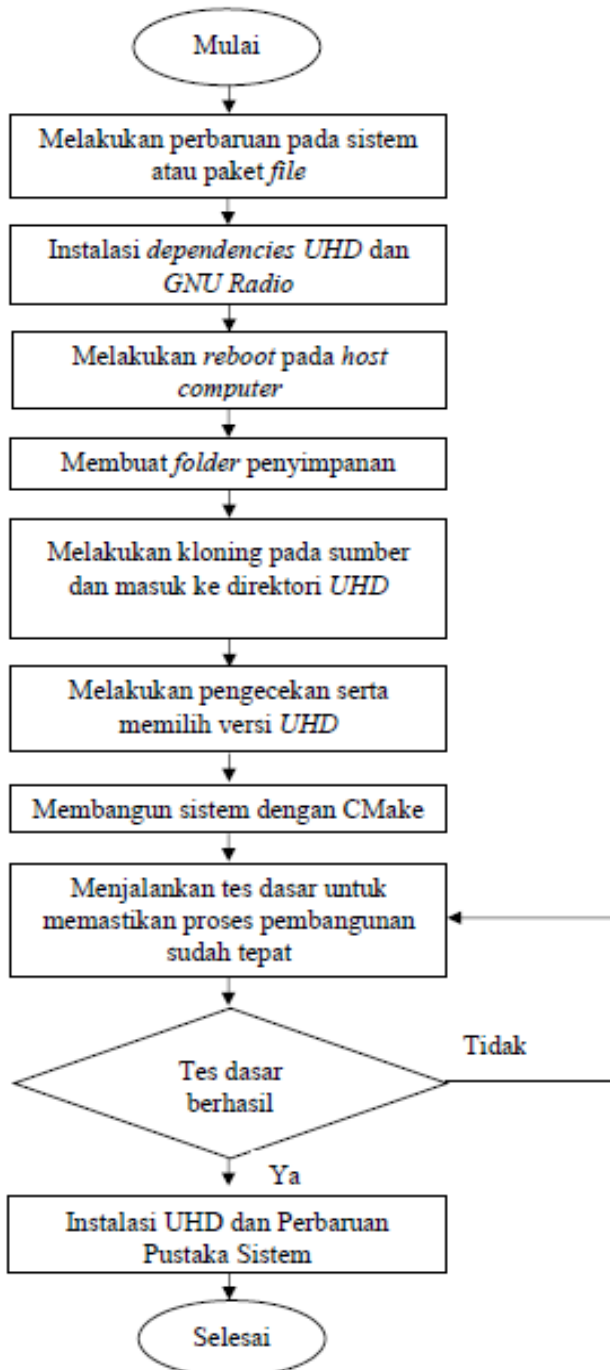
Setelah pada tahapan sebelumnya *host computer* menjalani proses *restart*, segera tekan tombol *Esc* untuk dapat memilih proses *Boot Device Option* dan menjalankan tahapan kelima. Lakukan proses *booting* dengan menggunakan *flashdisk* untuk masuk ke *operating system* Linux Ubuntu 18.04 LTS. Setelah berhasil *boot* dari *flashdisk*, akan tampil *wizard* dengan opsi '*Try Ubuntu*' dan '*Install Ubuntu*'. Karena partisi kosong yang tadi dibuat belum diatur pembagiannya,

maka di sini kita akan pilih opsi *Try Ubuntu*.. Catatan: Cara untuk menjalankan proses *booting* menggunakan *flashdisk* pada tiap komputer berbeda-beda, secara garis besar dibagi menjadi dua yaitu ada yang mengatur urutannya terlebih dahulu di BIOS dan ada juga yang memiliki menu khusus untuk memilih *boot*.

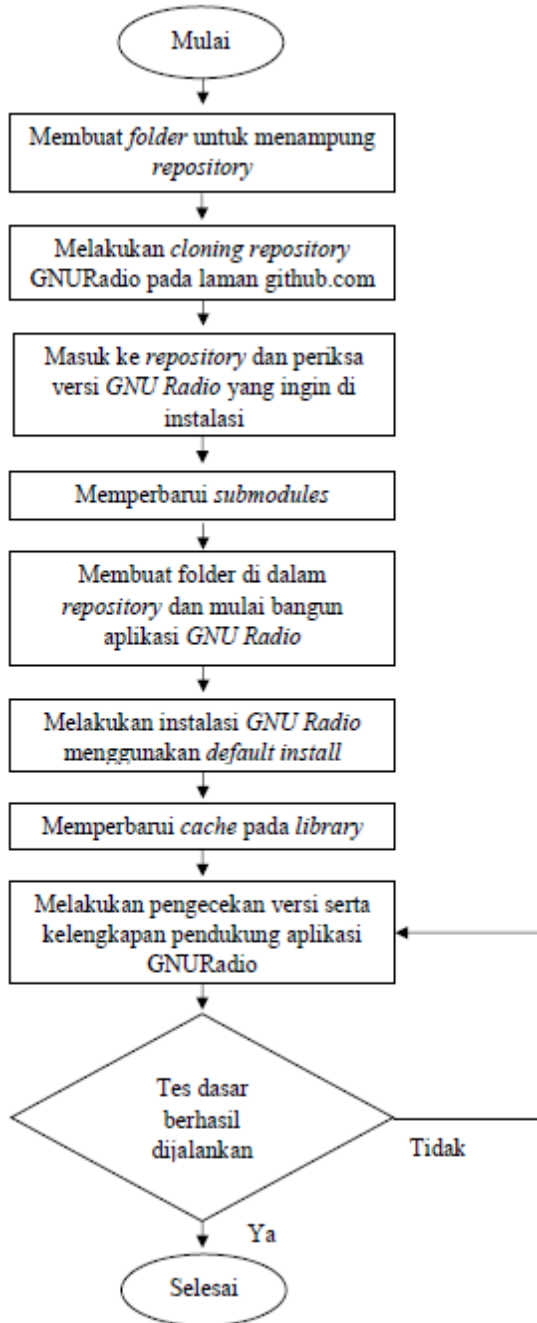
Tahapan keenam adalah mengatur partisi. Pada Linux Ubuntu 18.04 LTS mengatur partisi dapat dilakukan pada aplikasi GParted yang sudah tersedia. Mengatur partisi yang dimaksud pada tahap ini adalah membagi partisi kosong yang telah dibuat pada tahap ketiga. Cara untuk melakukan pengaturan partisi ini adalah:

Tahapan ketujuh adalah melakukan instalasi Linux Ubuntu 18.04 LTS. Sebelum instalasi disarankan untuk terhubung dengan internet terlebih dahulu agar proses instalasi ikut mengunduh pembaruan.

## Instalasi USRP Hardware Driver (UHD)



## Instalasi Aplikasi GNU Radio





### 1.5.2. Rancang Bangun Sistem Menggunakan GNU Radio

Rancang Bangun Sistem yang dimaksud disini adalah implementasi dari fungsi yang dibutuhkan oleh penerima TV digital standar T2 (DVB-T2) pada *diagram block* yang tersedia di aplikasi GNU Radio.

Adapun *block* yang dibutuhkan pada implementasi ini adalah sebagai berikut:

#### 1. Options

Blok ini berfungsi untuk:

- Menetapkan parameter khusus untuk *flow graph*. Hanya satu *options* yang diizinkan per *flow graph*.
- Judul, penulis, dan deskripsi dari parameter berguna untuk tujuan identifikasi.
- *The Generate Options* mengontrol jenis kode yang dihasilkan. *Flow graph* non-grafis harus menghindari penggunaan *sink* grafis atau kontrol variabel grafis.
- ID dari blok ini menentukan nama file yang dihasilkan dan nama kelas.

Seting Blok Options pada studi kasus ini:

- ID: *TV\_Receiver*
- *Generate Options*: WX GUI

#### 2. Variable 1

*Block* ini berfungsi untuk memetakan nilai ke variabel unik. Variabel pada parameter *block* lain, dapat menggunakan ID pada *Variable Block*. *Variable Block* adalah cara mudah untuk memiliki satu nilai yang digunakan di banyak tempat yang berbeda (yaitu parameter dari blok lain) sambil dapat dengan mudah mengubahnya.

Seting Blok Variabel 1: pada studi kasus ini:

- ID: *samp\_rate*
- Value: 10M

#### 3. WX GUI Notebook

Blok ini berfungsi untuk membuat dua grafik yang ingin ditampilkan dapat secara bersamaan ditabulasi untuk tampil.

Parameter:

- ID adalah parameter yang digunakan untuk memberi label nama variabel.
- Tab orientation adalah parameter untuk menentukan orientasi pada tab terletak disebelah atas.
- Labels adalah parameter untuk memberi judul pada tiap grafik yang diberi ID
- *"notebook\_0"*

Seting Blok WX GUI Notebook

- ID: notebook\_0
- Tab Orientation: Top
- Labels: "RF Spectrum", "Demod Spectrum"

#### 4. WX GUI Text Box

Seting Blok ini adalah:

- ID: frequency
- Default Value: 602M
- Converter: Float

Block ini berfungsi untuk memberikan nilai kepada setiap blok yang menggunakan ID "frequency"

Parameter:

- a. ID adalah parameter yang digunakan untuk memberi label nama variabel.
  - b. Default Value adalah nilai awal dengan ID "frequency".
  - c. Tipe Converter yang digunakan adalah float.
5. UHD: USRP Source

Seting blok ini adalah:

- Samp Rate (Sps): 10M
- Ch0: Center Freq (Hz): 602M
- Ch0: Gain Value: 15
- Ch0: Antenna: RX2
- Ch0: Bandwidth (Hz): 8M

Block ini berfungsi untuk mengalirkan sumber untuk sampel yang berasal dari perangkat keras USRP-N210.

Parameter:

- Center Freq (Hz) adalah frekuensi tengah yang akan menjadi frekuensi pusat dari aliran sampel yang diterima oleh USRP-

N210. Nilainya bervariasi dengan ID “frequency” yang akan diisi oleh WX GUI Text Box.

- Gain Value adalah parameter nilai penguatan yang akan diberikan.
- Antenna adalah parameter yang merujuk kepada antenna yang digunakan pada USRP N-210. Pada flowgraph ini antenna yang digunakan adalah RX2.
- Bandwidth (Hz) adalah parameter rentang frekuensi yang telah ditentukan oleh Kementerian Komunikasi dan Informatika untuk siaran televisi digital terrestrial yaitu 8 MHz.

#### 6. Low Pass Filter

Seting Blok ini adalah:

- *Decimation*: 10
- *Gain*: 1
- *Sample Rate*: 10M
- *Cutoff freq*: 694M
- *Transition width*: 10k
- *Window*: Hamming
- *Beta*: 0

*Block* ini berfungsi untuk menyaring atau membatasi frekuensi yang akan dialiri pada *flowgraph*.

Parameter:

- *Decimation* adalah parameter untuk memberikan nilai kompresi pada aliran sampel.
- *Gain* adalah parameter untuk memberikan penguatan pada aliran sampel setelah melalui penyaringan.
- *Sample Rate* adalah parameter nilai sampel yang akan diproses pada *flowgraph*.
- *Cutoff freq* adalah parameter yang menentukan batas nilai frekuensi pada aliran sampel.
- *Transition width* adalah parameter yang memberikan nilai jeda waktu antara *stop-band* dan *pass-band*.
- *Window* adalah parameter yang menentukan kerangka apa yang akan digunakan pada aliran sampel.

- Beta adalah parameter yang hanya digunakan pada kerangka jenis Kaiser.

#### 7. MPSK Receiver

Seting Blok ini adalah:

- M: 4
- Theta: 0
- *Loop Bandwidth*: 8M
- *Min Freq*: 478M
- *Max Freq*: 694M
- Mu: 250m
- *Gain Mu*: 10m
- *Omega*: 2
- *Gain Omega*: 1m
- *Omega Relative Limit*: 1m

*Block* ini berfungsi sebagai penerima aliran sinyal digital dan melakukan demodulasi terhadap sinyal tersebut.

Parameter:

- a. M adalah parameter yang menentukan nilai atau jumlah level digital (M-ary).
  - b. *Loop Bandwidth* merupakan nilai yang menentukan kapan *bandwidth* akan berulang.
  - c. *Min* dan *Max Freq* menentukan frekuensi minimum dan maksimum yang dapat dimodulasi.
- #### 8. Rational Resampler

- *Interpolation*: 96
- *Decimation*: 250
- *Taps*:
- *Fractional BW*: 0

*Block* ini berfungsi untuk melakukan pengulangan pada pengambilan sampel. Parameter:

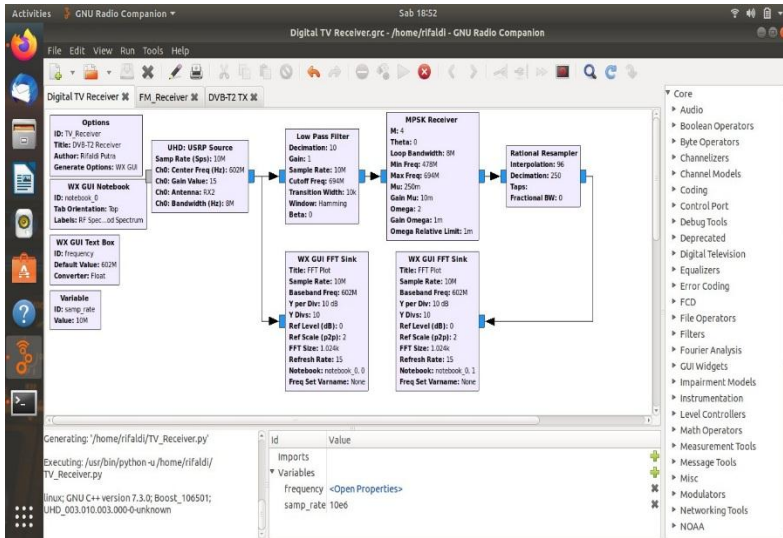
- a. *Interpolation* adalah nilai interpolasi yang diberikan pada aliran sampel.
- b. *Decimation* adalah parameter untuk memberikan nilai kompresi pada aliran sampel.

#### 9. WX GUI FFT Sink

*Block* ini berfungsi untuk menampilkan FFT Plot.

Parameter:

- Sample Rate merupakan nilai aliran sampel yang telah diambil ulang pada block rational resampler.
- FFT Size adalah parameter ukuran FFT yang akan ditampilkan.
- Notebook adalah parameter untuk menggabungkan FFT Plot kedalam Tab yang sudah diatur oleh WX GUI Notebook dengan mengatur ID.



Gambar 5 Flowgraph pada GNU Radio Companion

Sumber: Penulis (2020)

## 2. Pengujian dan Analisis Penerima TV Digital Standar T2 (DVB T2)

Pengujian akan dilakukan untuk mengetahui apakah rancang bangun yang dikerjakan dapat berhasil dan melihat parameter-parameter yang telah ditentukan yaitu kapasitas kanal dan penggunaan frekuensi pada tiap kanal. Sistem ini dibuat bertujuan untuk menyisipkan informasi berupa video yang berformat MPEG2 atau MPEG-TS pada sinyal yang ditangkap pada frekuensi tertentu. Adapun langkah-langkah pengujian yang dapat dilakukan sebagai berikut:

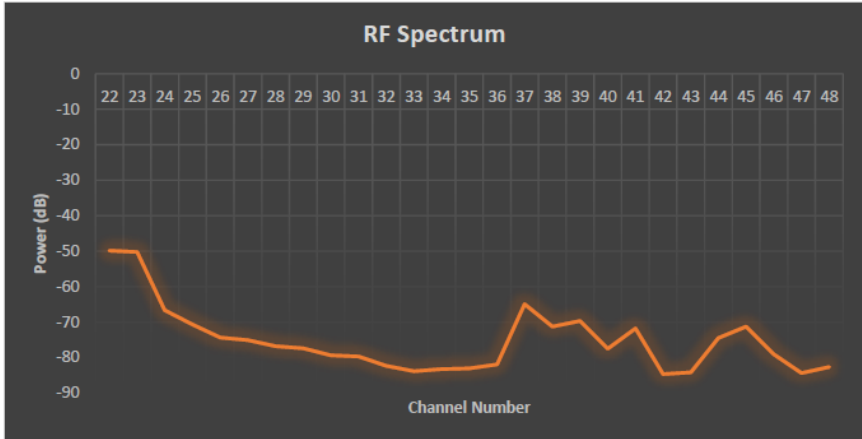
- Pengujian pertama yang akan dilakukan adalah pengujian *GNU Radio*. Dimana listing program untuk penerima TV

Digital Standar T2 (*DVB-T2*) diperiksa apakah sudah benar atau belum, kemudian dipastikan tidak ada *error* dan program dapat berjalan.

- Pengujian Kedua yang akan dilakukan adalah menampilkan spektrum. Dengan menampilkan spektrum dapat dipastikan bahwa penerima TV Digital Standar T2 (*DVB-T2*) telah berfungsi dengan baik. Kemudian akan dilakukan pencarian frekuensi siaran televisi digital. Prinsip Kerja pada sistem ini perangkat *USRP N-210* menjadi penerima dan akan *decoding* sinyal.
- Pengujian ketiga yang akan dilakukan adalah perhitungan berapa banyak kanal yang dapat diterima serta frekuensi yang digunakan pada tiap-tiap kanal oleh penerima TV digital dengan standar T2 (*DVB-T2*) dengan menggunakan *USRP N-210*.

### **Hasil Data *RF Spectrum* penerima TV Digital DVB T2**

*RF Spectrum* adalah visualisasi konseptual untuk mengatur dan memetakan fenomena fisik dari gelombang elektromagnetik yang ditangkap oleh *USRP-N210*. *RF Spectrum* ini merupakan keluaran dari blok *WX GUI FFT Sink* pada *GNU Radio* yang menampilkan daya pada tiap frekuensi yang belum melalui proses demodulasi. *FFT Plot* adalah sebuah instrumen tampilan yang didalamnya terdapat metode untuk mengubah sinyal dari domain waktu kedalam domain frekuensi dengan cepat sehingga menghasilkan nilai variasi yang tinggi dan detail. Grafik *RF Spectrum* dapat dilihat pada Gambar 6.



Gambar 6 *RF Spectrum*  
 Sumber: Penulis (2020)

Berdasarkan Gambar 4.21 dapat dianalisa bahwa mengacu pada rumusan masalah yang ada pertama, semua kanal yang dialokasikan untuk siaran televisi digital dapat diterima oleh *USRP N210*. Kedua, daya terbesar yang dapat diterima oleh *USRP-N210* adalah sebesar  $-49,85$  dB pada kanal 22 tepatnya pada frekuensi  $482,055$  MHz serta daya terkecil yang diterima adalah sebesar  $-84,66$  dB pada kanal 42 tepatnya pada frekuensi  $642,005$ . Ketiga, nilai daya rata-rata pada *RF Spectrum* adalah  $-75,196$  dB. Dan yang terakhir, nilai tengah untuk daya *RF Spectrum* yang didapatkan adalah  $-77,36$  dB.

### **Pengujian Daya Terima dan Frekuensi pada *Demod Spectrum***

Tujuan pengujian ini adalah untuk mengetahui daya dari frekuensi yang diterima pada *Demod Spectrum* berdasarkan keluaran dari blok *WX GUI FFT Sink* yang disusun diakhir *flowgraph* setelah blok *rational resampler*. Pada *Demod Spectrum* ditampilkan spektrum sinyal setelah melalui proses demodulasi.

#### **Langkah-langkah Pengujian:**

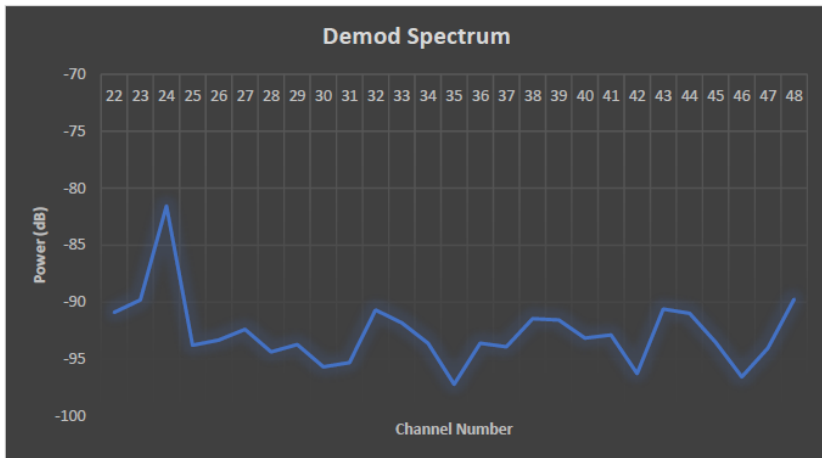
1. Rangkaian peralatan sesuai dengan Gambar 4.20.
2. Verifikasi penggunaan *USRP N210* menggunakan *UHD* dan *GNU*

## Radio

3. Buka aplikasi *GNU Radio Companion*
4. Buka file *GRC* yang telah dibuat
5. Atur frekuensi tengah yang ingin diproses pada *WX GUI Text Box*
6. *Generate flowgraph*
7. *Execute Flowgraph*
8. Catat hasil *Demod Spectrum*

### Hasil Data *Demod Spectrum*

*Demod Spectrum* adalah grafik yang menunjukkan daya tiap kanal setelah melalui proses demodulasi pada rangkaian *flowgraph* dalam *GNURadio*. Grafik *Demod Spectrum* dapat dilihat pada Gambar 4.22.



Gambar 7 *Demod Spectrum*

Sumber: Penulis (2020)

Berdasarkan Gambar 7 dapat dianalisa bahwa pertama, secara keseluruhan daya yang diterima oleh *USRP-N210* setelah melalui proses demodulasi relatif konstan, dapat dilihat bahwa perubahan yang signifikan hanya terjadi antara kanal 24 dan 25 dengan perbedaan nilai daya hanya sebesar 4,01 dB. Kedua, nilai daya terbesar yang didapatkan adalah sebesar -81,553 dB pada kanal 24 tepatnya pada frekuensi 494,3 MHz dan nilai daya terkecil adalah sebesar -97,2 dB pada kanal 35 tepatnya pada frekuensi



586,005 MHz. Ketiga, nilai daya rata-rata pada *demod spectrum* adalah -92,67 dB. Dan yang terakhir, nilai tengah untuk daya *Demod Spectrum* yang didapatkan adalah -93,33 dB.

Berdasarkan pengujian terhadap 27 kanal frekuensi dengan *bandwidth* pada masing-masing kanal sebesar 8 MHz dan *Sample Rate* yang digunakan adalah 10MHz. Hasil pengujian menunjukkan bahwa *USRP-N210* dapat menerima seluruh kanal dengan rentang frekuensi 478 MHz - 694 MHz dan didapatkan *Peak Frequency* serta besar *Power* pada tiap kanal.

MNC Publishing

MNC Publishing

## DAFTAR PUSTAKA

- [1] Di Pu, Alexander M. Wyglinski, "Digital Communication Systems Engineering with Software-Defined Radio", Artech House, London, 2013
- [2] C. Richard Johnson, Jr., William A. Sethares, "Telecommunication Breakdown", Prentice Hall, NJ, 2003
- [3] Rodger H' Hosking, "Software Defined Radio Handbook", 12<sup>th</sup> edition, Pentek Inc. New Jersey, 2016
- [4] Qinetiq, "An Evaluation of Software Defined Radio - Main Document", QinetiQ Ltd, 2006
- [5] Arthur A. Giordano, Allen HH. Levesque, "Modelling of Digital Communication System Using Simulink", Wiley, USA, 2015

MNC Publishing