

## DESAIN KOMUNIKASI QAM (*QUADRATURE AMPLITUDE MODULATION*) MENGGUNAKAN GNU RADIO

Nyaris Pambudiyatno, Bambang Bagus Harianto, Achmad Setiyo Prabowo  
Politeknik Penerbangan Surabaya  
E-mail: n.pambudi@gmail.com

---

### Abstrak

Dalam paper ini akan dipaparkan secara mendasar tentang perancangan Desain Komunikasi QAM menggunakan GNURadio. GNURadio merupakan *software* gratis dari sumber terbuka yang menyediakan teknik pemrosesan sinyal untuk diimplementasikan ke *software* radio. Perangkat lunak ini dapat digunakan dengan *hardware* RF (RTLSDR) atau dari hasil simulasi. Teknologi modern saat ini sudah banyak yang memakai konsep sistem komunikasi radio atau di definisikan sebagai *Software Defined Radio* (SDR). SDR adalah sistem komunikasi radio dimana komponen yang terdapat pada perangkat keras digantikan dengan pengimplementasian perangkat lunak pada perangkat komputer. Sistem SDR yang umum digunakan yakni GNU Radio. Hal ini yang mendasari peneliti untuk melakukan penelitian perancangan Desain Komunikasi QAM (*Quadrature Amplitude Modulation*) menggunakan GNURadio. Dalam penelitian ini akan ditunjukkan fungsi perblok *Transmitter* dan *Receiver*, serta *setting* perbloknya hingga hasil simulasi QAM pada GNURadio.

**Kata Kunci :** GNU Radio, *Software Defined Radio*, dan *Quadrature Amplitude Modulation*.

### Abstract

*In this paper, it will explain the basic design of the QAM Communication Design using GNURadio. GNURadio is free, open source software that provides signal processing techniques for implementation in radio software. This software can be used with hardware RF (RTLSDR) or simulation results. Many of today's modern technology have used the concept of a radio communication system or defined as Software Defined Radio (SDR). SDR is a radio communication system where the components contained in hardware are replaced by implementing software on computer devices. The SDR system that is commonly used is GNU Radio. This is what underlies the researcher to conduct research on the design of the QAM (Quadrature Amplitude Modulation) Communication Design using GNURadio. In this research, the function of the blockers Transmitter and Receiver will be shown, as well as the setting of the blockers to the QAM simulation results on GNURadio.*

**Keywords :** GNU Radio, *Software Defined Radio*, and *Quadrature Amplitude Modulation*.

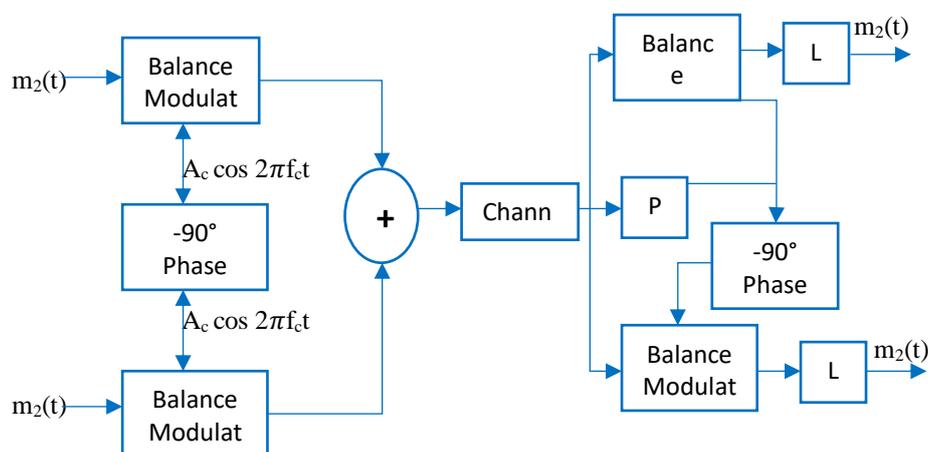
---

## PENDAHULUAN

Salah satu metode memodulasikan sinyal informasi dengan sinyal pembawa adalah Quadrature Amplitude Modulation (QAM). Sistem kerja QAM adalah mengubah sinyal informasi yang dikirimkan menjadi simbol-simbol. Simbol-simbol tersebut dikalikan dengan sinyal pembawa yang frekuensinya lebih tinggi daripada frekuensi dari sinyal informasi.

*Quadrature Amplitude Modulation* atau QAM adalah suatu cara penransmisian pada laju bit-bit yang lebih tinggi pada saluran/kanal dengan lebar pita yang terbatas. Sebagai contoh penggunaan kumpulan sinyal QAM 16 titik memungkinkan 9600 bit/detik ditransmisikan pada saluran telepon dengan lebar pita 2700 Hz. Dalam kasus tersebut empat digit biner yang berurutan harus disimpan dan dikodekan kembali sebagai salah satu dari 16 bentuk sinyal yang ditransmisikan. Sinyal-sinyal yang dihasilkan dinamakan sinyal *Quadrature Amplitude Modulation* (QAM). Sinyal ini dapat ditafsirkan sebagai modulasi amplitudo multitingkat yang diterapkan secara bebas pada setiap dua pembawa kuadratur.

Sinyal *Quadratur Amplitude Modulation* (QAM) menggunakan dua pembawa kuadratur  $\cos 2\pi t$  dan  $\sin 2\pi t$ , masing-masing dimodulasikan oleh bit informasi. Metode dari transmisi sinyal memakai *Quadrature Carrier Multiplexing*. Inti dari QAM adalah gabungan antara modulasi PSK dan ASK hal ini digunakan ketika membutuhkan modulasi dengan kecepatan tinggi.



Gambar 1. Blok Diagram QAM

Pada QAM fase dan amplitudo dari sinyal *carrier* diubah-ubah untuk melambangkan data.

Sinyal QAM dapat dituliskan sebagai berikut:

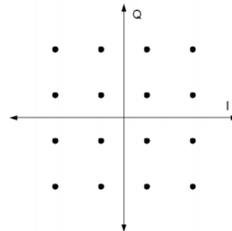
$$s(t) = I(t). \cos \omega_c t + Q(t). -\sin \omega_c t \quad (1)$$

dengan:

$$I(t) = A. \cos \theta \quad (2)$$

$$Q(t) = A. \sin \theta \quad (3)$$

Dari persamaan diatas, dapat dilihat bahwa sinyal QAM dapat dibentuk dengan menjumlahkan sebuah sinyal kosinus dengan amplitudo  $I(t)$  dan sebuah sinyal sinus dengan amplitudo  $Q(t)$ . Ini sama dengan menjumlahkan sebuah sinyal AM (*amplitude modulation*) yang menggunakan *carrier* kosinus dengan sebuah sinyal AM lain yang menggunakan *carrier* sinus. Kata quadrature pada QAM berasal dari kedua *carrier* yang berbeda fase  $90^\circ$ .



Gambar 2. Diagram Konstelasi 16-QAM

GNU Radio adalah sebuah perangkat lunak gratis dari sumber terbuka yang menyediakan teknik pemrosesan sinyal untuk diimplementasikan ke *software* radio. Perangkat lunak ini dapat digunakan dengan perangkat keras *Radio Frequency* (RF) atau dari hasil simulasi. Perangkat lunak ini umum digunakan untuk mempelajari, membuat, hingga menerapkan *Software Defined Radio* (SDR) dalam lingkup akademis maupun bisnis.

Aplikasi GNURadio sebagian besar ditulis menggunakan bahasa pemrograman Python, sedangkan bagian pemrosesan sinyal diimplementasikan dalam bahasa pemrograman C++ menggunakan prosesor ekstensi *floating-point*. Dengan demikian, pengembang dapat menerapkan sistem radio yang *real-time* dengan *throughput* tinggi namun dalam lingkungan aplikasi pengembangan yang bisa disederhanakan.

Perangkat lunak GNURadio menyediakan kerangka kerja dan alat untuk membangun dan menjalankan radio perangkat lunak atau hanya aplikasi pemrosesan sinyal umum. *Software* GNURadio umumnya dikenal sebagai "*flowgraph*", yang merupakan serangkaian blok pemrosesan sinyal yang terhubung bersama, sehingga menggambarkan aliran data.

## PEMBAHASAN

### 1. Blok QAM Transmitter GNURadio

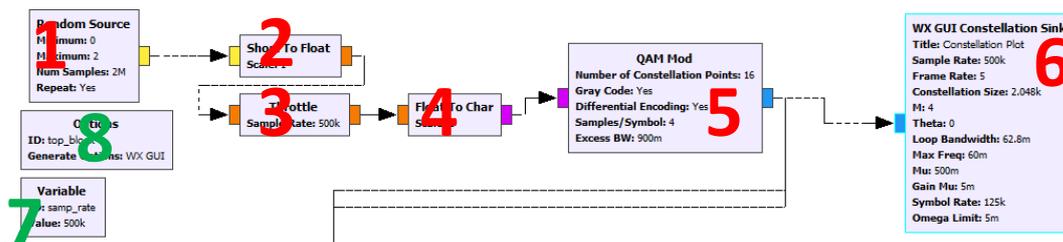
Simulasi dilakukan dengan spesifikasi *software* GNURadio tipe 3.7.13.4. Pada tipe ini, GNURadio telah memiliki blok yang beragam jenis seperti source, sinks, graphical sinks hingga misc. Sesuai dengan prinsip blok diagram QAM, pada GNURadio mentransmit sinyal QAM. Sehingga diperlukan beberapa blok *transmitter* untuk simulasi ini diantaranya *Random Source*, *Short to Float*, *Throttle*, *Float to Char*, *QAM Mod*, dan *WX GUI Constellation Sink*.

Perhatikan tabel 1 di bawah untuk meninjau pembagian blok transmit dan penunjang lebih jelasnya.

Tabel 1. Blok yang Digunakan pada *Transmitter* QAM

Blok Transmit	Blok Penunjang
<i>Random Source</i>	Options
Short to <i>Float</i>	Variable
<i>Float</i> to Char	
<i>Throttle</i>	
QAM Mod	
WX GUI Scope Sink	

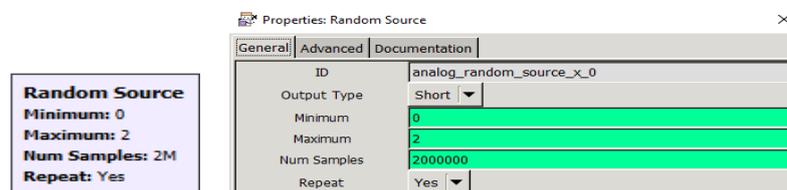
Pada pembahasan *transmitter* pada modulasi QAM di GNURadio ini akan dijelaskan secara bertahap mengenai proses *setting* tiap blok. Blok diagram *transmitter* dapat dilihat pada Gambar 3 di bawah. Warna merah menunjukkan blok diagram *transmitter* dan hijau menunjukkan blok penunjang. Pada simulasi ini blok *transmitter* dan *receiver* terdapat dalam satu grafik atau satu lembar kerja, sehingga blok penunjang merupakan bagian dari *receiver* juga.



Gambar 3. Blok Transmitter QAM pada GNURadio

#### Blok 1 : *Random Source*

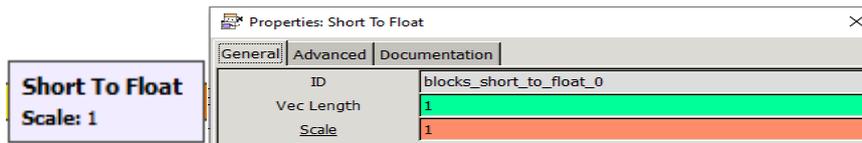
Blok ini berfungsi sebagai penghasil nomor *sample* secara acak tetapi kita mengatur *max* dan minimal nomor tersebut. *Setting* parameter lihat pada gambar 4 di bawah.



Gambar 4. Random Source Transmitter QAM

#### Blok 2 : *Short to Float*

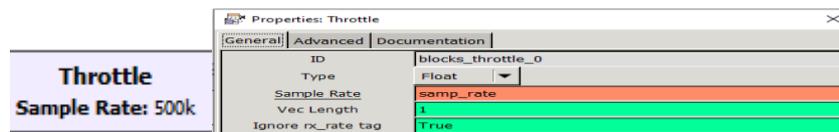
Blok ini berfungsi sebagai *converter* tipe *flow short* menjadi *float*. *Setting* parameter dapat dilihat pada gambar 5 di bawah.



Gambar 5 Short to Float Transmitter QAM

Blok 3 : *Throttle*

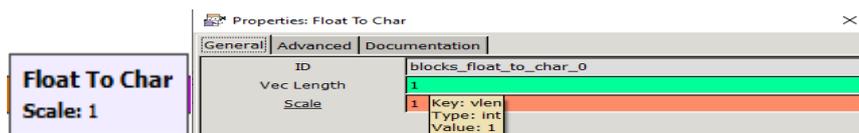
Blok ini digunakan untuk mengatur laju *sample*, *throttle* hanya digunakan untuk ketika tidak ada blok berupa *hardware* seperti (SDR, *speaker*, mikrofon). Untuk lebih jelasnya dapat dilihat pada gambar 6 di bawah.



Gambar 6. Throttle Transmitter QAM

Blok 4 : *Float to Char*

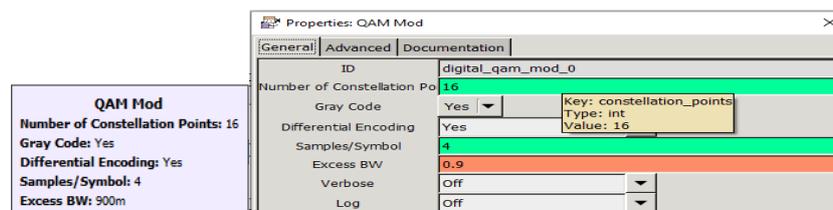
Blok ini berfungsi sebagai *converter* tipe flow short menjadi *float*. *Setting* parameter dapat dilihat pada gambar 7.



Gambar 7. Float to Char Transmitter QAM

Blok 5 : *QAM Mod*

Blok ini berfungsi sebagai pemodulasi sinyal informasi menjadi sinyal yang termodulasi secara QAM. *Setting* parameter dapat dilihat pada gambar 8 di bawah.



Gambar 8. QAM Mod Transmitter QAM

Blok 6 : *WX GUI Constellation Sink*

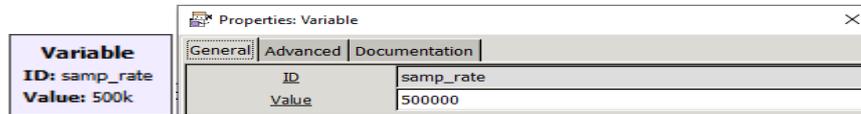
Blok ini berguna untuk menampilkan IQ constellation dari berbagai macam sinyal. *Setting* parameter yang digunakan lihat pada gambar 9 di bawah.



Gambar 9. WX GUI Constellation Sink Transmitter QAM

Blok 7 : *Variable*

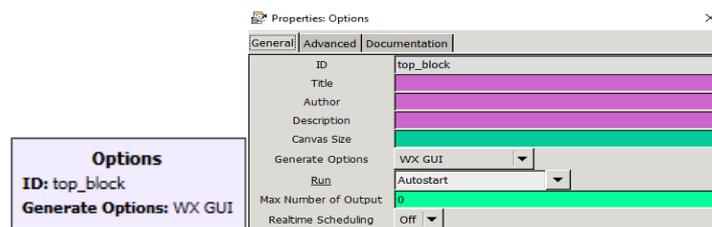
Blok ini berfungsi sebagai pengatur *flow rate* pada blok diagram yang dibuat. Untuk *setting* parameter seperti gambar 10 di bawah.



Gambar 10. Variable Transmitter QAM

Blok 9 : *Options*

Blok ini berguna untuk memilih mode kerja dari diagram yang dibuat, dan menamai blok diagram yang telah dibuat. Untuk *setting* yang diatur lihat gambar 11 di bawah.



Gambar 11. Options Transmitter QAM

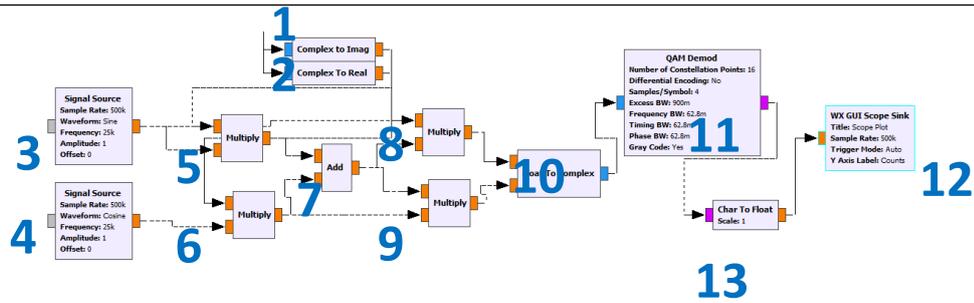
## 2. Blok QAM Receiver GNURadio

Pada *receiver* QAM GNURadio terdapat beberapa blok yang digunakan, untuk lebih jelasnya perhatikan tabel 2 di bawah.

Tabel 2. Blok yang Digunakan pada *Receiver* QAM

Blok Receive	Blok Penunjang
Complex to Imag	Options
Complex to <i>Real</i>	Variable
Signal Source	
Multiply	
Add	
<i>Float</i> to Complex	
QAM Demod	
Char to <i>Float</i>	
WX GUI Scope Sink	

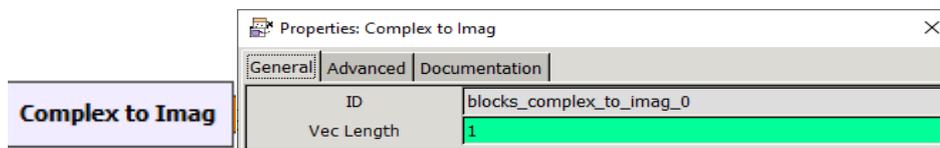
Pada pembahasan ini akan dijelaskan proses *receive* pada modulasi QAM di GNURadio serta proses *setting* tiap bloknnya. Blok diagram *receiver* dapat dilihat pada Gambar 12 di bawah. Warna biru menunjukkan blok diagram *receiver*. Blok penunjang *receiver* ada pada gambar blok *transmitter* di atas, namun bertugas untuk blok *transmitter* maupun *receiver*.



Gambar 12. Blok Receiver QAM

Blok 1 : *Complex to Imag*

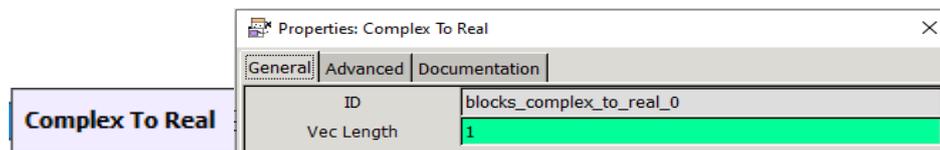
Blok ini digunakan untuk *converter* dari *setting flow complex* yang berasal dari *transmitter* QAM menjadi bilangan imajiner dikarenakan akan dibandingkan dengan *signal source* yang menghasilkan gelombang *sinus*. *Setting* parameter dapat lihat pada gambar 13 di bawah.



Gambar 13. Complex to Imag Receiver QAM

Blok 2 : *Complex to Real*

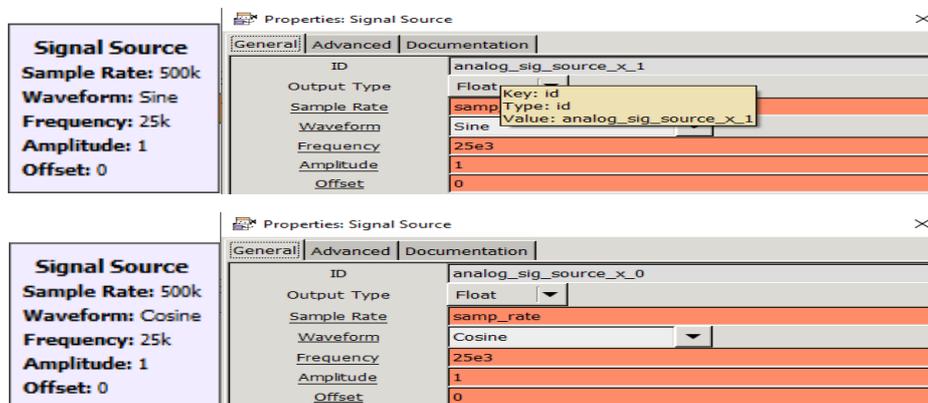
Blok ini digunakan untuk *converter* dari *setting flow complex* menjadi bilangan *real* dikarenakan akan dibandingkan dengan *signal source* yang menghasilkan gelombang *cosinus*. *Setting* parameter dapat lihat pada gambar 14 di bawah.



Gambar 14. Complex to Real Receiver QAM

Blok 3 & 4 : *Signal Source*

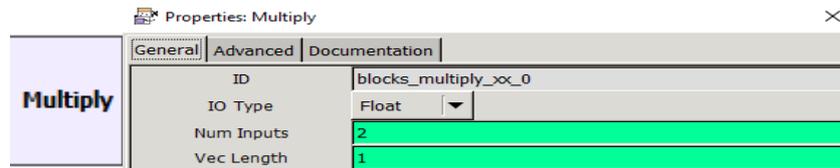
*Signal Source* pada blok 3 digunakan untuk pembuat sinyal *carrier* dan pada blok 4 digunakan untuk pembuat sinyal modulating, untuk lebih jelasnya lihat gambar 15 di bawah beserta parameternya.



Gambar 15. Signal Source Receiver QAM

Blok 5, 6, 8, 9 : *Multiply*

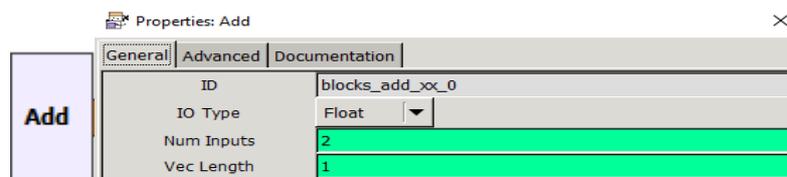
Blok ini berguna untuk mengalikan 2 *input* menjadi 1 *output*, pada blok 5 menggabungkan *output* dari blok 1 dengan blok 3, blok 6 menggabungkan *output* blok 2 dan 4, blok 8 menggabungkan *output* blok 3 dan blok 7, blok 9 menggabungkan *output* dari blok 4 dan blok 7, untuk lebih jelasnya Anda dapat melihat gambar 16 di bawah. Pada blok *multiply* ini semua blok mempunyai nilai parameter yang sama.



Gambar 16. Multiply Receiver QAM

Blok 7 : *Add*

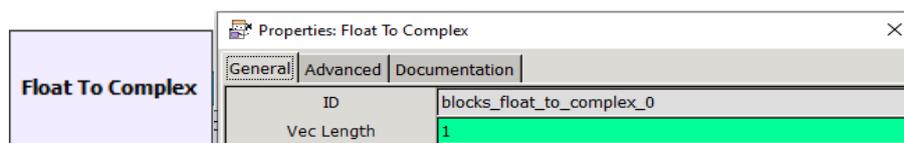
Blok ini berguna untuk menambahkan 2 *input* menjadi 1 *output*, *input* pada blok 7 merupakan *output* dari blok 5 & 6, untuk lebih jelasnya lihat pada gambar 17 di bawah.



Gambar 17. Add Receiver QAM

Blok 10 : *Float to Complex*

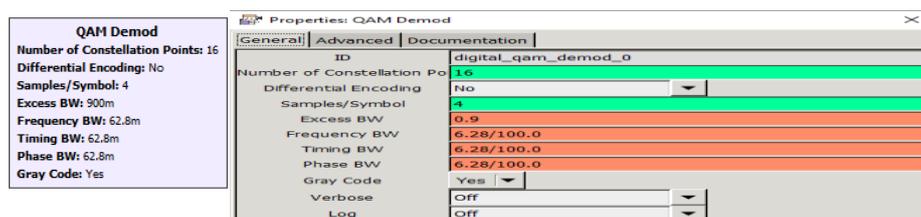
Blok ini berguna untuk menkonversi *setting flow float* menjadi *setting flow complex* hal ini dilakukan karena *input* blok QAM Demod menggunakan *setting flow complex*. Untuk *setting* parameter seperti gambar 18 di bawah.



Gambar 18. Float to Complex Receiver QAM

Blok 11 : *QAM Demod*

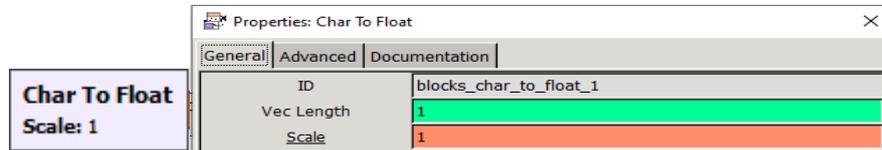
Blok ini berguna untuk mendemodulasi sinyal yang telah termodulasi oleh QAM sehingga akan didapat hasil sinyal informasi. Untuk *setting* parameter dapat dilihat pada gambar 19 di bawah.



Gambar 19. QAM Demod Receiver QAM

### Blok 12 : Char to Float

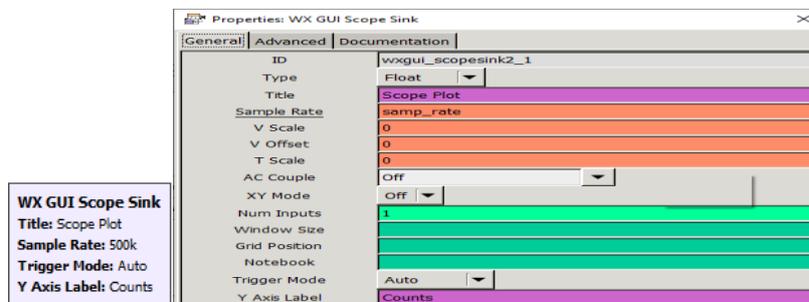
Blok ini berguna sebagai converter *setting* flow char menjadi *setting* flow Float supaya dapat terbaca pada blok WX GUI Scoper Sink. *Setting* parameter dapat dilihat pada gambar 20 di bawah.



Gambar 20. Char to Float Receiver QAM

### Blok 13 : WX GUI Scope Sink

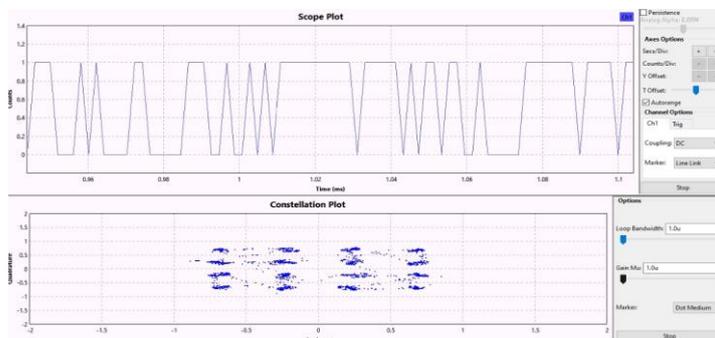
Blok ini berguna untuk memonitor sinyal yang masuk dan memantau sinyal seperti layaknya *oscilloscope*, pada blok ini *input* sinyal yang dimonitor yaitu sinyal *char to float*, *Throttle*, untuk lebih jelasnya lihat gambar 21 di bawah.



Gambar 21. WX GUI Scope Sink Receiver QAM

## 3. Hasil Simulasi QAM GNURadio

Pada gambar 22 di bawah merupakan hasil sinyal transmisi pada modulasi QAM yang hanya dihasilkan oleh 1 blok utama. Komponen I dan Q merupakan hasil dari pembuatan dari blok *random source*, pada titik plot tidak terbentuk konstelasi yang sempurna karena disebabkan oleh filter RRC yang kurang sempurna.



Gambar 22. Random Source binary data dan QAM Constellation

## PENUTUP

### Kesimpulan

Dari hasil simulasi yang telah dilakukan diatas, dapat disimpulkan bahwa *software* GNURadio dapat menunjukkan adanya modulasi QAM dan titik plot konstelasi. Pemahaman mengenai fungsi per blok *transmitter* dan *receiver* sangat penting, karena terwujudnya hasil simulasi bergantung pada hal tersebut dan besar usaha yang dilakukan untuk mencoba terus menerus.

Paper ini dapat dijadikan dasar bagi para pemula untuk belajar dan mengembangkan diri baik dalam bidang akademis ataupun bisnis. Modulasi digital lain seperti *Amplitude Shift Keying* (ASK), *Frequency Shift Keying* (FSK), *Binary-Phase Shift Key modulation* (BPSK), dan *Orthogonal Frequency Division Multiplexing* (OFDM) juga dapat dikembangkan dengan GNURadio. GNURadio juga dapat didownload pada website GNURadio secara gratis, sehingga sangat mudah untuk dipelajari tanpa menghabiskan banyak biaya. Perkembangan teknologi saat ini sedang pesatnya, oleh karena itu memanfaatkan waktu tanpa menghabiskan biaya dengan belajar hal-hal baru tentang teknologi komunikasi, merupakan pilihan paling baik untuk saat ini.

### DAFTAR PUSTAKA

- Blossom, E. (2004). GNU radio: tools for exploring the radio frequency spectrum. *Linux Journal*.
- Buchali, F., Steiner, F., Böcherer, G., Schmalen, L., Schulte, P., & Idler, W. (2016). Rate Adaptation and Reach Increase by Probabilistically Shaped 64-QAM: An Experimental Demonstration. *Journal of Lightwave Technology*.  
<https://doi.org/10.1109/JLT.2015.2510034>
- Bulow, H. (2009). Polarization QAM modulation (POL-QAM) for coherent detection schemes. *Conference on Optical Fiber Communication, Technical Digest Series*.  
<https://doi.org/10.1364/ofc.2009.owg2>
- Buracchini, E. (2000). Software radio concept. *IEEE Communications Magazine*.  
<https://doi.org/10.1109/35.868153>
- Gandhiraj, R., Ram, R., & Soman, K. P. (2012). Analog and digital modulation toolkit for software defined radio. *Procedia Engineering*.  
<https://doi.org/10.1016/j.proeng.2012.01.975>
- Hanzo, L., Webb, W. T., & Keller, T. (1999). Single- and Multi-carrier Quadrature Amplitude Modulation: *Methods*.
- Li, X. (2008). Simulink-based Simulation of Quadrature Amplitude Modulation (QAM) System. *Proceedings of IAJC-IJME International Conference*
- National Instrument. (2007). Quadrature Amplitude Modulation (QAM), 1–27.
- Rakesh. (2015). QAM Modulation.
- Rtl-sdr, A. G. T. O. T. H. E., Defined, S., & By, R. (n.d.). THE HOBBYIST ' S GUIDE TO THE RTL-SDR : REALLY CHEAP SOFTWARE.