

Rancang Bangun Purwarupa Penerima Paket APRS Berbasis Raspberry Pi 2 untuk Stasiun Bumi

Agfianto Eko Putra^{*1}, Unggul Adhi Nugroho², Bakhtiar Alldino Ardi Sumbada³,
Catur Atmaji⁴

^{1,3,4}Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

²Program Studi Elektronika dan Instrumentasi, DIKE, FMIPA, UGM, Yogyakarta

e-mail : *¹agfi68@gmail.com, ²unggul.adn@gmail.com,

³b.alldino.as@ugm.ac.id, ⁴catur_atmaji@ugm.ac.id

Abstrak

Telah dirancang-bangun penerima paket APRS berbasis Raspberry Pi 2 untuk stasiun bumi. Tuner TV digunakan sebagai penerima sinyal, alat penerima diakses melalui laptop secara nirkabel. Antena Yagi dengan sebuah pengendali digunakan agar dapat secara otomatis mengarahkan ke satelit. Uji coba dilakukan dengan menerima paket APRS yang dipancarkan digipeater satelit International Space Station (ISS) dan satelit LAPAN-A2. Hasil penelitian menunjukkan bahwa alat ini mampu mendapatkan paket APRS satelit ISS dengan jumlah 6 paket dari 10 paket yang dipancarkan. Paket yang diterima memiliki rata-rata amplitudo pada frekuensi 1.200 Hz dan 2.200 Hz yang bernilai jauh lebih kecil dibandingkan amplitudo audio keseluruhan. Hal ini menunjukkan bahwa terdapat derau yang tinggi pada sinyal. Sedangkan paket APRS dari satelit LAPAN-A2 belum berhasil diperoleh.

Kata kunci— Radio Amatir, Dire Wolf, ISS, LAPAN-A2, rtl_fm, RTL-SDR

Abstract

The design of Raspberry Pi 2-based APRS packets for UGM earth stations has been designed. A TV tuner is used as a signal receiver, the receiver is accessed via a laptop wirelessly. A Yagi Antenna with a controller is used to automatically point to the satellite. The test was carried out by receiving the APRS package emitted by the International Space Station (ISS) satellite and the LAPAN-A2 satellite. The results showed that this tool was able to get the ISS satellite APRS package with a total of 6 packages out of 10 packets emitted. The package received has an average amplitude of 1,200 Hz and 2,200 Hz which is much smaller than the overall audio amplitude. This indicates that there is high noise in the signal. While the APRS package from the LAPAN-A2 satellite has not been successfully obtained.

Keywords— Amateur Radio, Dire Wolf, ISS, LAPAN-A2, rtl_fm, RTL-SDR

1. PENDAHULUAN

Satelit yang memiliki digipeater akan menunjang kinerja radio amatir dengan cara menerima sinyal (informasi) dari area *terrestrial* yang dapat berupa paket APRS (*Automatic Packet Reporting System*). Kemudian dipancarkan kembali ke area *terrestrial* sehingga radio amatir di area lain dapat mengetahui isi informasinya. Radio amatir menggunakan perangkat *transceiver* dengan dukungan APRS untuk saling bertukar informasi dalam bentuk teks, namun, perangkat dengan dukungan APRS memiliki harga yang relatif mahal.

Sruthi dkk. melakukan penelitian mengenai rancangan pemancar-penerima (*transceiver*) untuk SDR (*Software Define Radios*) menggunakan RTL-SDR. Pemancar-penerima berbasis *Personal Computer* (PC). Pemancar menggunakan *mixer* agar dapat mengakses seluruh pita

frekuensi DC (*Direct Current*) hingga 1.700 MHz. Sedangkan penerima, digunakan RTL-SDR yang memiliki jangkauan frekuensi 64 MHz hingga 1,7 GHz. Metode yang diusulkan tersebut menggunakan RF *mixer* dan RTL-SDR yang lebih cocok untuk transmisi sinyal dan penerimaan karena biayanya yang relatif rendah [1].

Sistem yang dibuat oleh Danymol dkk., yakni RTL-SDR dan Raspberry Pi sebagai penerima sinyal FM, ditransfer ke *remote* PC. RTL-SDR yang digunakan memiliki *tuner chip* Elonics E4000. Sedangkan proses pemantauan dilakukan pada lokasi yang berbeda secara jarak jauh menggunakan WiFi untuk mengirim data SDR, sehingga sistem komunikasi secara *realtime* dapat dilakukan. Sinyal FM pada frekuensi 98,3 MHz yang diterima oleh RTL-SDR dikirim menggunakan TCP server melalui LAN ke *remote* PC. Spektrum sinyal FM diwujudkan menggunakan perangkat lunak SDR Sharp yang dipasang di *remote* PC [2].

Hadjaveric dkk. melakukan implementasi *Client-Server* untuk APRS-IS (*APRS Internet Service*) menggunakan Raspberry Pi. Sistem yang dirancang memiliki komponen utama Raspberry Pi, RTL-SDR, *smartphone android* dan sebuah pemancar-penerima. Perangkat lunak yang digunakan adalah Raspbian Debian (Wheezy), *pymultimonaprs* dan *aprsdroid*. Raspberry Pi terhubung ke *router* atau *firewall* melalui kabel UTP atau WiFi untuk beralih ke perangkat komunikasi lainnya. Hasil dapat diamati pada laman <http://aprs.fi>, yaitu unit klien yang dapat mengirim perubahan lokasi GPS ke unit server melalui RF 144,800 MHz dengan radius 200 meter dengan hanya menggunakan antenna bawaan [3].

Berbeda dengan penelitian sebelumnya, penelitian ini bertujuan merancang-bangun sebuah penerima paket APRS berbasis Raspberry Pi 2 (dengan perangkat lunak **rtl_fm** dan **Dire Wolf**), RTL-SDR dengan memanfaatkan Antena Yagi yang dilengkapi dengan sebuah pengendali agar dapat secara otomatis mengarah ke satelit.

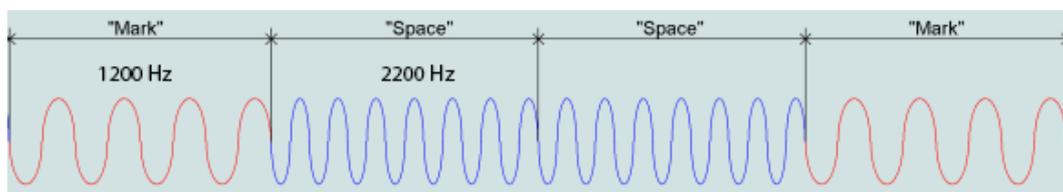
2. METODE PENELITIAN

2.1. APRS

APRS atau *Automatic Packet Reporting System*, yang dirancang oleh Bob Bruninga (WB4APR) dan diperkenalkan di TAPR/ARRL (*Tucson Amateur Packet Radio/American Radio Relay League*) *Digital Communications Conference* tahun 1992, adalah sebuah protokol paket komunikasi untuk menyebarkan data kepada semua orang di jaringan secara *real-time*. APRS merupakan kombinasi paket radio dengan jaringan satelit *Global Positioning System* (GPS), yang memungkinkan radio amatir untuk secara otomatis menampilkan posisi stasiun radio dan benda-benda lain pada peta di komputer atau laptop.

2.2 Paket Radio AFSK 1.200

APRS menggunakan teknik modulasi AFSK (*Audio Frequency Shift Keying*), dimana data digital diwakili oleh perubahan frekuensi (*pitch*) nada audio. Biasanya, audio yang ditransmisikan bergantian antara dua nada: "tanda", mewakili satu biner; yang lain, "ruang", mewakili nol biner. AFSK memiliki keuntungan yaitu mampu bekerja melalui jalur audio biasa, yang membuatnya cocok untuk digunakan pada radio yang dirancang untuk suara [5]. *Bit order* yang digunakan adalah LSB. *Data link layer* yang digunakan adalah AX.25.



Gambar 1. Urutan biner 1001 dengan modulasi FSK [6]

Gambar 1 menunjukkan urutan biner yang dimodulasi dengan FSK, frekuensi sinyal yang dipancarkan digeser dengan perubahan fasa minimal seperti tampak pada amplitudo modulasi sinyal tidak berubah. Karena hanya menggunakan dua frekuensi harmonik, spektrum dari sinyal yang ditransmisikan sempit (tidak lebih dari 3 kHz), yang memungkinkan sinyal untuk lewat dengan bebas melalui jalur *dial-up* komunikasi jarak jauh.

2.3 Perangkat Lunak Tracking Satelit

Perangkat lunak *tracking* satelit, Orbitron, adalah perangkat lunak yang digunakan untuk memprediksi posisi satelit. Data masukan yang digunakan untuk prediksi posisi satelit yaitu data *Two Line Element* (TLE) dan lokasi pengamat. Luarannya digunakan sebagai masukan ke pengarah atau *rotator* antena, berupa posisi satelit yang telah dikonversi ke dalam azimuth dan elevasi untuk pengamat (pengarah antena).

2.4 Perangkat Lunak APRSDroid

APRSDroid adalah sebuah aplikasi Android untuk operator radio amatir[8]. Perangkat lunak ini digunakan untuk pelaporan posisi pada APRS, perangkat lunak ini mampu menampilkan stasiun radio amatir di dekatnya serta untuk pertukaran data pesan APRS.

2.5 Perangkat Keras Rotator Antena

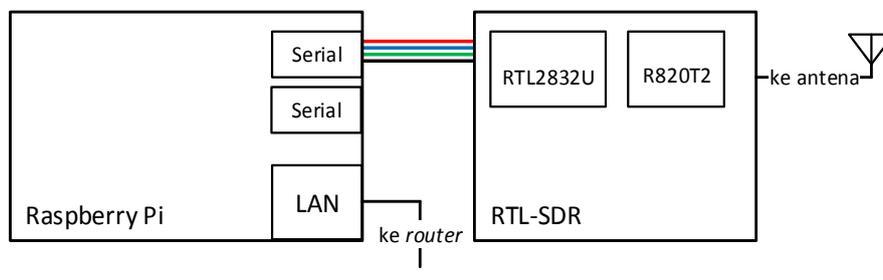
Rotator antena, sebagaimana ditunjukkan pada Gambar 2, digunakan untuk menggerakkan Antena Yagi dengan besar penguatan kurang lebih 8 dB. *Rotator* antena dikendalikan menggunakan mikrokontroler untuk mengarahkan antena ke posisi satelit secara *realtime* sesuai data masukan, yang berupa data TLE dari perangkat lunak *tracking* satelit.



Gambar 2. Perangkat keras *rotator* antena yang digunakan

2.6 Rancangan Perangkat Keras

Rancangan perangkat keras utama, sebagaimana ditunjukkan pada Gambar 3, terdiri atas Raspberry Pi 2 dan RTL-SDR. Raspberry Pi 2 dihubungkan dengan RTL-SDR melalui USB. *Tuner* yang digunakan adalah Rafael R820T2.

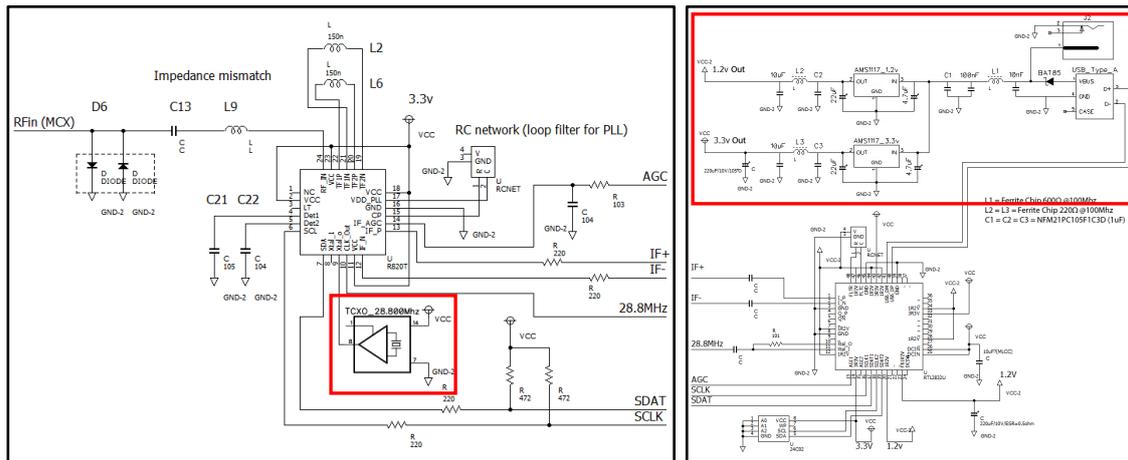


Gambar 3. Rancangan perangkat keras utama

2.6.1 Modifikasi RTL-SDR

RTL-SDR dapat secara langsung digunakan sebagai penerima frekuensi tinggi, akan tetapi penggunaan dalam jangka waktu lama dapat menyebabkan pergeseran *offset* pada frekuensi [9]. TCXO (*Temperature Compensated Crystal Oscillator*) digunakan agar *offset* frekuensi lebih stabil.

Gambar 4 (kiri) merupakan skematik RTL-SDR yang menggunakan TCXO (tanda kotak). Gambar 4 (kanan) adalah skematik ADC 7-bit Realtek RTL2832U, regulator daya 3,3V dan 1,2V. Hal yang umum menambahkan kapasitor *filter* pada VBUS USB [10], karena kabel jumper dapat berfungsi layaknya sebuah antena. Penapis pi ditambahkan setelah luaran regulator daya, digunakan untuk mengurangi riak pada luaran regulator daya.

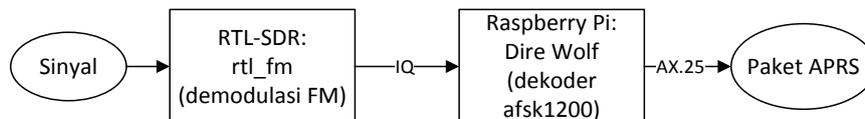


Gambar 4 Skematik pada rangkaian tuner dan TCXO (kiri), skematik pada rangkaian RTL2832U dan regulator daya (kanan)

2.7 Penggunaan Perangkat Lunak

2.7.1 Perangkat Lunak Pada Raspberry Pi 2

Perangkat lunak yang digunakan pada Raspberry Pi 2 adalah perangkat lunak pengoperasian RTL-SDR, yaitu `rtl_fm` dan **Dire Wolf**. Perintah yang diberikan untuk Raspberry Pi 2 adalah menjalankan `rtl_fm` kemudian di-*pipe* dengan perintah **Dire Wolf**. Diagram proses penerimaan sinyal hingga penampilan paket APRS ditunjukkan pada Gambar 5.



Gambar 5. Diagram proses penerimaan hingga penampilan paket APRS

2.7.2 Program Demodulator FM

Program demodulator FM yang digunakan adalah `rtl_fm`, yang merupakan demodulator FM untuk perangkat penerima berbasis RTL2832 [12]. *Source code* `rtl_fm` dapat diunduh di laman <https://github.com/keenerd/rtl-sdr/>. Pengujian dilakukan pada frekuensi 145,825 MHz dan laju pencuplikan 22.050 Hz. Pengaturan luaran pada program ini adalah *standard output* (stdout) agar dapat diteruskan oleh perintah pada program selanjutnya yaitu **Dire Wolf** sebagai dekoder APRS.

2.7.3 Program Dekoder APRS

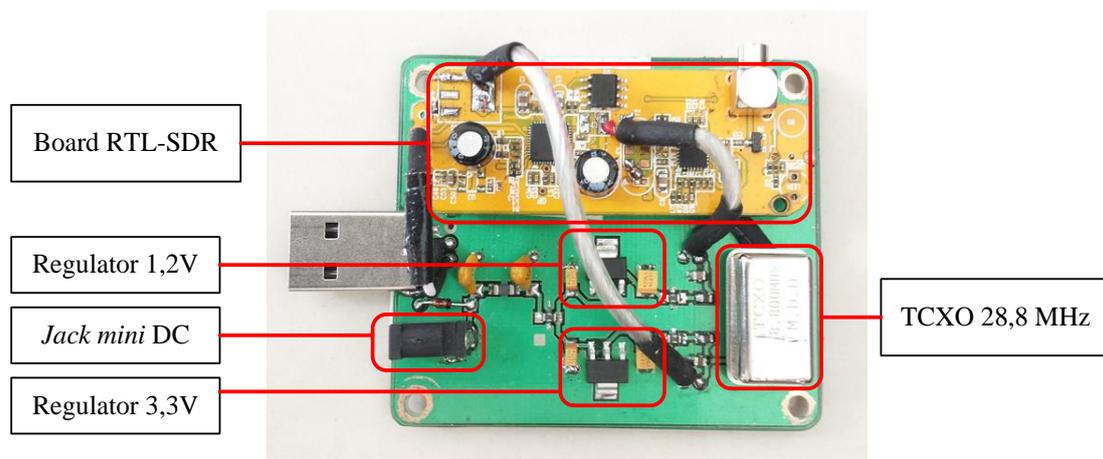
Program dekoder APRS yang digunakan adalah Dire Wolf (*Decoded Information from Radio Emissions for Windows Or Linux Fans*), yang merupakan perangkat lunak modem soundcard/TNC (*Terminal Node Controller*) dan untuk encode/dekode APRS. *Source code*-nya dapat diunduh pada laman <https://github.com/wb2osz/direwolf>.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Perangkat Keras

3.1.1 TCXO dan regulator daya pada RTL-SDR

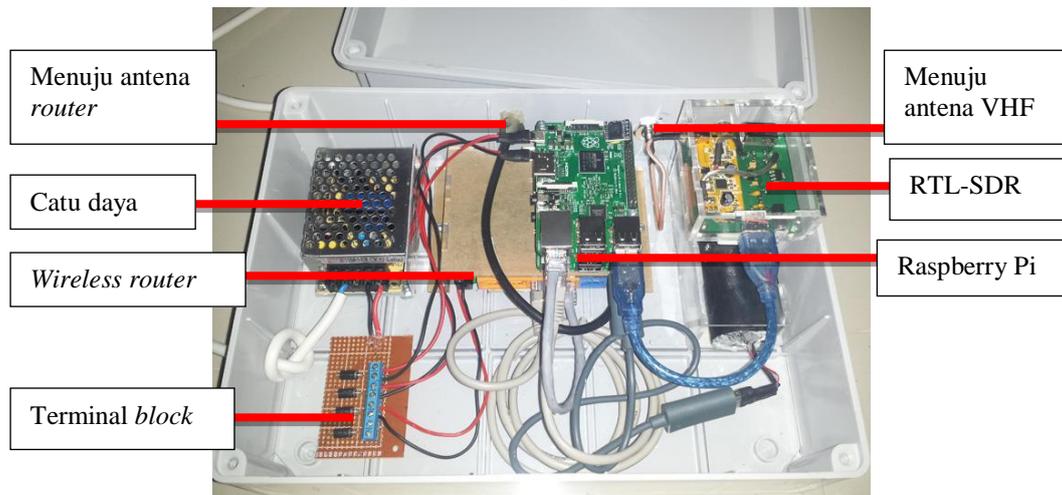
Gambar 6 menunjukkan papan rangkaian RTL-SDR, regulator daya 1,2 V dan 3,3V, *filter* untuk regulator daya, TCXO 28,8Mhz dan *jack mini* DC untuk daya ekstra apabila daya yang dikeluarkan Raspberry Pi 2 tidak mencukupi. Dioda (Schottky BAT85) ditambahkan pada bagian antara VBUS USB dengan Vout pada *jack mini* DC. Dioda ini berfungsi untuk mencegah arus mengalir dari Vout *jack* ke VBUS USB sehingga Raspberry Pi 2 tetap aman.



Gambar 6 TCXO dan regulator daya

3.1.2 Perangkat Keras Utama

Gambar 7 menunjukkan rangkaian perangkat keras utama. Raspberry Pi 2 dihubungkan dengan PC/laptop melalui *wireless router*, juga dihubungkan dengan RTL-SDR melalui USB. Antena Yagi dihubungkan ke RTL-SDR menggantikan antena bawaan. Rangkaian dengan dioda dan *terminal block* digunakan untuk mendistribusikan daya ke Raspberry Pi 2 dan *wireless router*. *Wireless router* pada penerima dikonfigurasi sebagai *server*. Sedangkan *wireless router* sebagai *client* digunakan untuk koneksi ke PC/laptop.



Gambar 7. Perangkat keras utama

3.2 Implementasi Perangkat Lunak

3.2.1 Sesi SSH pada Raspberry Pi

Sesi SSH ditunjukkan pada Gambar 8, pengaturan alamat ip untuk Raspberry Pi 2 adalah 192.168.0.107, sedangkan laptop beralamatkan 192.168.0.7. Laptop dikonfigurasi sedemikian rupa sehingga koneksi internet laptop dapat dibagi ke Raspberry Pi agar dapat dilakukan pembaruan dan pemasangan perangkat lunak.

```

      • MobaXterm 8.5 •
      (SSH client, X-server and networking tools)

> SSH session to pi@192.168.1.101
• SSH compression : ✓
• SSH-browser      : ✓
• X11-forwarding  : ✓ (remote display is forwarded through SSH)
• DISPLAY         : ✓ (automatically set on remote server)

> For more info, ctrl+click on help or visit our website

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 12 18:32:17 2016 from takamine-pc.mshome.net
pi@raspberrypi ~ $

```

Gambar 8 Sesi SSH pada Raspberry Pi dengan MobaXterm

3.2.2 Program rtl_fm

Gambar 9 merupakan tampilan pilihan perintah pada rtl_fm yang menunjukkan pilihan beberapa perintah yang tersedia. Beberapa pilihan yang tersedia yaitu pengaturan frekuensi, jenis modulasi, pengaturan *sample rate* (nilai *default* 24kHz,) indeks perangkat (*default* 0), dan *tuner gain*.

```

pi@raspberrypi ~ $ rtl_fm -h
rtl_fm, a simple narrow band FM demodulator for RTL2832 based DVB-T receivers

Use:    rtl_fm -f freq [-options] [filename]
        -f frequency_to_tune_to [Hz]
           use multiple -f for scanning (requires squelch)
           ranges supported, -f 118M:137M:25k
        [-M modulation {default: fm}]
           fm, wbfm, raw, am, usb, lsb
           wbfm == -M fm -s 170k -o 4 -A fast -r 32k -l 0 -E deemp
           raw mode outputs 2x16 bit IQ pairs
        [-s sample_rate {default: 24k}]
        [-d device_index {default: 0}]
        [-g tuner_gain {default: automatic}]

```

Gambar 9 Pilihan perintah pada program rtl_fm

3.2.3 Program Dire Wolf

Gambar 10 menunjukkan tampilan menu bantuan pada Dire Wolf dengan pilihan beberapa perintah yang tersedia. Beberapa pilihan yang tersedia adalah pengaturan nama berkas konfigurasi, lokasi berkas *log*, *sample rate* audio, jumlah *channel*, *bit-depth* audio, dan *data rate* (nilai *default* adalah 1.200).

```

Dire Wolf version 1.3

Usage: direwolf [options] [ - | stdin | UDP:nnnn ]
Options:
  -c fname      Configuration file name.
  -l logdir     Directory name for log files. Use . for current.
  -r n          Audio sample rate, per sec.
  -n n          Number of audio channels, 1 or 2.
  -b n          Bits per audio sample, 8 or 16.
  -B n          Data rate in bits/sec for channel 0. Standard values are 300, 1200, 9600.
                If < 600, AFSK tones are set to 1600 & 1800.
                If > 2400, K9NG/G3RUH style encoding is used.
                Otherwise, AFSK tones are set to 1200 & 2200.

```

Gambar 10 Pilihan perintah pada program Dire Wolf

3.2.4 Demodulasi dan dekode

Proses demodulasi dan dekode pada Raspberry Pi dilakukan menggunakan perintah yang ditunjukkan pada Gambar 11. Perintah “rtl_fm” digunakan untuk melakukan demodulasi sinyal frekuensi tinggi. “-f 145.825M” untuk melakukan demodulasi pada frekuensi 145,825 Mhz. “-s 22050” *sample rate* diatur ke 22.050 Hz. Tanda “-” digunakan untuk memasukkan data ke *stdin*. “|” untuk melakukan *pipe* perintah selanjutnya. Pada perintah Dire Wolf terdapat “-n 1” yaitu *channel* diatur ke 1, “-r 22050” *sample rate* diatur ke 22.050 Hz. “-D 1” membagi *sample* audio menjadi 1. “-a 300” menampilkan statistik audio setiap 300 detik. “-l /home/pi/Documents/” menyimpan log data ke lokasi /home/pi/Documents/.

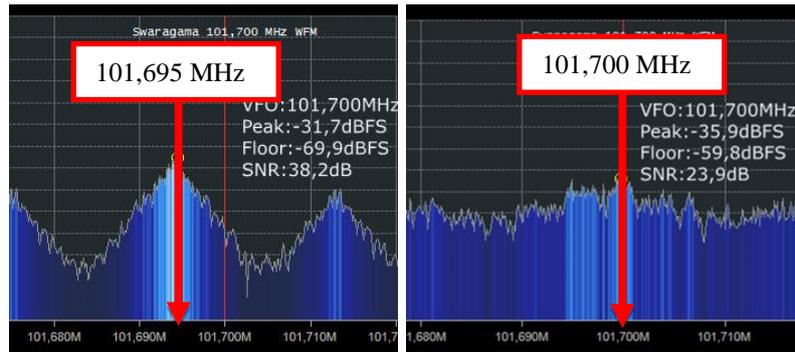
```
rtl_fm -f 145.825M -s 22050 - | direwolf -n 1 -r 22050 -D 1 -a 300 -l /home/pi/Documents/ -
```

Gambar 11 Perintah untuk melakukan demodulasi dan dekode

3.3 Pengujian Sistem

3.3.1 Pengujian Kestabilan Frekuensi

Perangkat lunak yang digunakan untuk pengujian adalah SDRSharp, Gambar 12 (kiri) menunjukkan pergeseran *offset* frekuensi pada 101,700 Mhz ke 101,695 Mhz atau sekitar 5 kHz setelah 10 menit penggunaan pada RTL-SDR bawaan. Kemudian setelah digunakan TCXO, seperti ditunjukkan pada Gambar 12 (kanan), performa RTL-SDR pada penerimaan radio FM, setelah lebih dari 10 menit dinyalakan, tampak *offset* frekuensinya lebih stabil pada 101,700 MHz.



Gambar 12 Pergeseran *offset* frekuensi (kiri) dan *offset* frekuensi lebih stabil (kanan)

Kestabilan *offset* frekuensi akan sangat berpengaruh terhadap penerimaan sinyal dari satelit. Ketika terjadi pergeseran frekuensi akibat efek *doppler*, total pergeseran yang lebih besar dapat terjadi bila *offset* frekuensi pada RTL-SDR juga bergeser. Misalnya, dari satelit, frekuensi bergeser -4 KHz akibat efek *doppler* dan dari RTL-SDR bergeser -5 KHz. Maka total pergeseran adalah -9 KHz, dari frekuensi 145,825 MHz menjadi 145,816 MHz. Hal ini dapat menyebabkan paket APRS tidak dapat didekode oleh dekoder.

3.3.2 Pengujian pada Satelit ISS

Pemancaran pada *digipeater* APRS satelit dilakukan menggunakan radio UHF/VHF dengan daya 10 Watt dan antena yagi. Pemancaran dilakukan menggunakan aplikasi APRSDroid pada *smartphone* yang dihubungkan ke radio. Posisi pemancaran paket adalah tetap. Volume suara untuk APRSDroid diatur ke 100%. Antena yagi digunakan pada radio. Radio diatur ke frekuensi 145,825 Mhz dan penerima diatur ke frekuensi yang sama pula. Pemancaran *beacon* akan dilakukan 1 menit sekali, karena pada APRSDroid dibatasi minimal 1 menit setiap transmisi *beacon*. Potongan paket yang diterima dari *digipeater* ISS ditunjukkan pada Gambar 13.

```
digipeater RS0ISS audio level = 71(10/11) [NONE] _:|_|_
[0.3] YC2WKC-7>APDR13,RS0ISS*,YBSAT,WIDE2-2:=0746.03S/11022.57Es/A=000633 AFSK Test - Gadjah Mada University
Position, original primary Phone, APRSDroid Android App http://aprsdroid.org/
S 07 46.0300, E 110 22.5700, alt 633 ft
AFSK Test - Gadjah Mada University
```

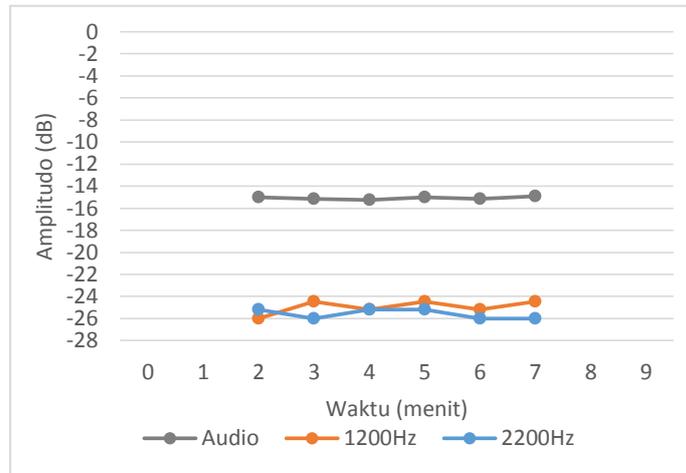
Gambar 13 Potongan paket yang diterima dari *digipeater* ISS

Paket yang dipancarkan berupa *beacon* lokasi pemancaran dan dengan isi komentar “AFSK Test - Gadjah Mada University”. *Callsign* pengguna adalah YC2WKC-7. *Path* yang digunakan adalah RS0ISS, YBSAT dan WIDE2-2. “0746.03S/11022.57Es/A=000633” adalah posisi radio pemancar. Paket yang diterima memiliki data *payload* yang sama dengan data pemancar. Perbedaannya adalah pada *path*, yaitu tanda *asterisk* (*) pada RS0ISS* yang menandakan bahwa paket APRS telah berhasil dilakukan *digipeating* oleh ISS. Dire Wolf menampilkan amplitudo audio secara presentase, pada paket pertama, 71 adalah *volume* audio, yaitu nilai amplitudo puncak ke puncak pada audio. Sedangkan angka 10/11 adalah nilai amplitudo setengah puncak ke puncak. 10 adalah amplitudo untuk frekuensi 1.200 Hz (untuk *mark* atau 1), sedangkan 11 adalah amplitudo untuk frekuensi 2.200 Hz (untuk *space* atau 0). Rasio amplitudo dalam desibel didapatkan menggunakan persamaan (1).

$$G_{dB} = 20 \log_{10}(A_2/A_1) \quad (1)$$

Satelit ISS melintas dalam jangkauan penerima selama kurang lebih 10 menit. Waktu ini disebut dengan waktu *Acquisition of Signal* (AOS), pada 10 kali pemancaran paket *beacon*, hanya didapatkan 6 paket dari *digipeater* satelit. Dua paket pada menit awal dan menit akhir tidak diterima oleh penerima. Dua menit awal dan akhir, sudut elevasi antara satelit dan penerima bernilai kecil. Sudut elevasi yang kecil menyebabkan sinyal yang dipancarkan dapat

terhambat oleh permukaan bumi. Gambar 14 menunjukkan grafik amplitudo sinyal yang diterima dari *digipeater* ISS.



Gambar 14 Grafik amplitudo pada penerimaan paket dari *digipeater* ISS

Amplitudo audio memiliki nilai rata-rata -15,08 dB, rata-rata amplitudo pada 1.200 Hz adalah -24,95 dB dan 2.200 Hz adalah -25,61 dB. Amplitudo audio keseluruhan memiliki nilai yang lebih besar, hal ini dapat disebabkan oleh interferensi dari lingkungan luar sehingga menyebabkan adanya derau yang tinggi. Selain itu dapat pula disebabkan oleh efek *doppler shift*, yang menyebabkan pergeseran frekuensi yang diterima dan dipancarkan satelit. Sehingga pada frekuensi 145,825 MHz tidak diperoleh sinyal yang kuat. Nilai rata-rata amplitudo yang diperoleh ditunjukkan pada Tabel 1.

Tabel 1. Nilai rata-rata amplitudo hasil pengujian pada satelit ISS

	Audio	1.200 Hz	2.200 Hz
Amplitudo	-15,08 dB	-24,95 dB	-25,61 dB

3.3.3 Pengujian pada satelit LAPAN-A2

Pengujian penerimaan paket APRS dari *digipeater* satelit LAPAN-A2 tidak berhasil. *Digipeater* LAPAN-A2 memiliki daya 5 Watt. Sedangkan daya *digipeater* APRS pada satelit ISS adalah 20 Watt. Penguatan (*gain*) pada daya dalam desibel dihitung menggunakan persamaan (2).

$$Gain_{dB} = 10 \log_{10}(P_2/P_1) \quad (2)$$

Sehingga diketahui penguatan daya pada LAPAN-A2 adalah 6,99 dB, sedangkan pada ISS adalah 13,01 dB. Jadi, penguatan daya pada *digipeater* satelit LAPAN-A2 lebih kecil 6,01 dB dibandingkan *gain* daya pada *digipeater* ISS.

Satelit LAPAN-A2 mengorbit dengan ketinggian kurang lebih 636 km, ketinggian ini lebih jauh dibandingkan dengan satelit ISS yang sejauh kurang lebih 406 km. Jarak yang di tempuh oleh sinyal menuju stasiun bumi menyebabkan sinyal yang diterima menjadi jauh lebih kecil. Hubungan antara jarak dan kerugian sinyal atau *free-space path loss* (*FSPL*) ditunjukkan oleh persamaan (3).

$$FSPL_{dB} = 20 \log_{10}(d) + 20 \log_{10}(f) + 20 \log_{10}\left(\frac{4\pi}{c}\right) \quad (3)$$

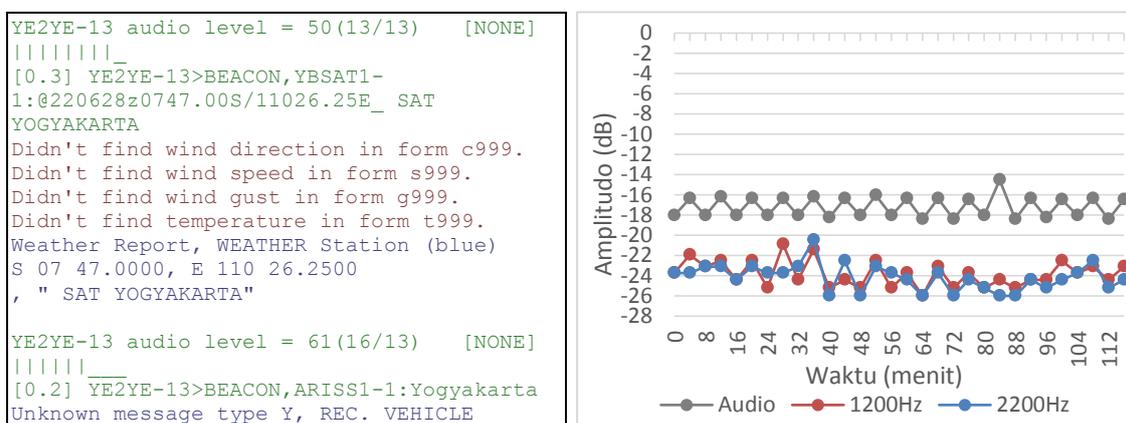
Frekuensi APRS untuk satelit LAPAN-A2 sama dengan satelit ISS yaitu 145,825 MHz. Sehingga diketahui nilai *FSPL* (*Free Space Path Loss*) untuk satelit LAPAN-A2 adalah 131,79 dB sedangkan untuk satelit ISS adalah 127,89 dB. Jadi nilai *FSPL* untuk satelit LAPAN-A2 lebih besar 3,9 dB dari pada nilai *FSPL* untuk satelit ISS. Hasil ini menunjukkan bahwa besar jarak berbanding lurus dengan besar kerugian sinyal selama transmisi, semakin besar jarak maka semakin besar kerugian sinyal, menyebabkan semakin kecil sinyal yang diterima oleh

stasiun bumi. Rata-rata amplitudo frekuensi 1.200 Hz dan 2.200 Hz (nada AFSK) pada paket yang didapatkan dari satelit ISS secara berurutan bernilai -24,95 dB dan -25,61 dB, maka pada satelit LAPAN-A2 bernilai lebih kecil. Sehingga nada AFSK tidak dapat dideteksi oleh dekoder dan tidak ada paket yang bisa ditampilkan.

Dengan demikian, pengujian paket data APRS pada satelit LAPAN-A2 tidak berhasil dilakukan karena satelit memiliki penguatan daya yang lebih kecil dan kerugian sinyal yang lebih besar dibanding satelit ISS. Hal ini juga berlaku untuk satelit lainnya, karena sebagian besar satelit memiliki *digipeater* APRS dengan daya yang lebih kecil dan memiliki orbit dengan ketinggian yang lebih dari satelit ISS, sehingga paket tidak berhasil diperoleh. Selain itu, tidak semua satelit memiliki *digipeater* APRS yang aktif, sehingga hanya diperoleh paket APRS dari satelit ISS.

3.3.4 Pengujian pada stasiun cuaca

Pengujian dilakukan pada stasiun cuaca (*weather station*) dengan *callsign* YE2YE-13, yang memiliki jarak sekitar 7 km dari posisi penerima. Penerima diatur ke frekuensi yang sama. Gambar 15 (kiri) menunjukkan potongan paket yang diterima dari YE2YE-13 dan Gambar 15 (kanan) adalah grafik amplitudo-nya.



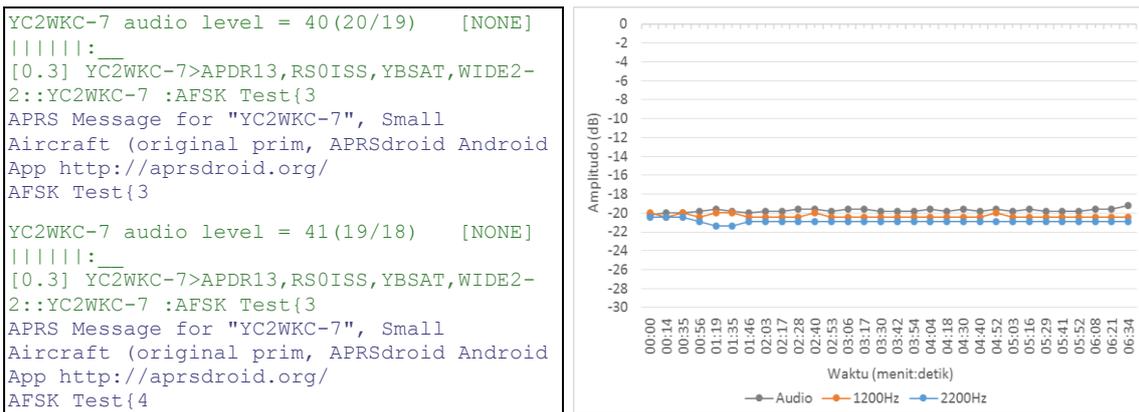
Gambar 15 Potongan paket yang diterima dari YE2YE-13 (kiri) dan grafik amplitudo (kanan)

Terdapat paket *error* yang ditunjukkan oleh lonjakan pada grafik dengan nilai amplitudo audio -14,54 dB, amplitudo frekuensi 1.200 Hz dan 2.200 Hz masing-masing adalah -24,44 dB dan -26,02 dB. Amplitudo pada frekuensi 1.200 Hz dan 2.200 Hz terlihat tidak stabil yang dapat disebabkan karena pengaturan pemancar atau dapat pula karena gangguan di luar lingkungan. Hasil rata-rata dari amplitudo audio adalah -17,19 dB, amplitudo pada frekuensi 1.200 Hz sebesar -23,83 dB dan amplitudo pada frekuensi 2.200 Hz sebesar -24,18 dB. Amplitudo audio menunjukkan nilai yang lebih besar dari pada amplitudo frekuensi 1.200 Hz dan 2.200 Hz, meskipun tidak sekecil pada penerimaan dari *digipeater* ISS. Hal ini menunjukkan bahwa YE2YE-13 memiliki sinyal yang lebih kuat dari pada *digipeater* ISS dari lokasi stasiun bumi.

3.3.5 Pengujian Paket dari Radio UHF/VHF

Radio dan penerima diatur ke frekuensi 144,390 Mhz, paket yang dipancarkan berupa pesan. Pemancaran dilakukan menggunakan aplikasi APRSDroid pada *smartphone* yang dihubungkan ke radio UHF/VHF. Volume suara untuk APRSDroid diatur ke 100%. Pemancaran dilakukan dengan jeda kurang lebih 10 detik, pada tempat yang sama dengan jarak kurang lebih 20 meter dari posisi penerima.

Amplitudo audio yang diperoleh memiliki nilai rata-rata -19,72 dB, rata amplitudo pada frekuensi 1.200 Hz adalah -20,36 dB dan pada frekuensi 2.200 Hz adalah -20,90 dB. Nilai amplitudo pada pengujian ini lebih besar dari pada pengujian penerimaan dari satelit ISS dan stasiun cuaca YE2YE-13, terutama perbedaannya yang signifikan terhadap perbandingan antara amplitudo audio dengan amplitudo frekuensi. Hal ini menunjukkan bahwa diperoleh sinyal yang lebih kuat. Nilai amplitudo audio yang hampir sama dengan amplitudo frekuensi menunjukkan tingkat derau yang kecil. Gambar 16 (kiri) menunjukkan potongan paket yang diterima dari radio UHF/VHF dengan *callsign* YC2WKC-7, sedangkan Gambar 16 (kanan) adalah grafik amplitudo.



Gambar 16 Potongan paket yang diterima dari radio UHF/VHF (kiri) dan grafik amplitudo (kanan)

4. KESIMPULAN

Purwarupa penerima paket APRS berbasis Raspberry Pi 2 untuk stasiun bumi telah berhasil dibuat dan mampu menerima paket APRS dari *digipeater* ISS, stasiun cuaca YE2YE-13, dan radio UHF/VHF. Penerima paket APRS dapat menerima paket dari *digipeater* ISS dengan jumlah 6 paket dari 10 paket yang dipancarkan. Paket APRS yang diterima dari *digipeater* ISS memiliki rata-rata amplitudo audio sebesar -15,08 dB, amplitudo pada frekuensi 1.200 Hz sebesar -24,95 dB dan pada 2.200 Hz sebesar -25,61 dB, nilai ini menunjukkan tingkat derau yang tinggi. Paket dari *digipeater* satelit LAPAN-A2 tidak dapat diperoleh, hal ini bisa disebabkan karena daya *digipeater* satelit yang kecil dan jarak yang lebih jauh dibanding satelit ISS. Sudut elevasi satelit, besar daya pada *digipeater* satelit dan jarak satelit atau pemancar berpengaruh pada transmisi sinyal.

5. SARAN

Beberapa saran terkait dengan apa yang telah dilakukan pada penelitian ini dan bisa menjadi pengembangan topik penelitian ini antara lain, antena dengan faktor penguatan yang lebih besar perlu digunakan agar kerugian sinyal akibat jarak dapat dikurangi, demikian juga antena pada radio untuk pemancaran paket ke satelit. Posisi penerima perlu ditempatkan di lokasi yang memiliki gangguan lingkungan lebih sedikit, sehingga sistem dapat menerima sinyal dengan lebih baik. Perangkat SDR dengan kemampuan pancar-terima perlu digunakan agar dapat melakukan pengiriman dan penerimaan paket APRS.

DAFTAR PUSTAKA

- [1] M.B. Sruthi, M. Abirami, A. Manikkoth, R. Gandhiraj, and K.P. Soman, "Low cost digital transceiver design for Software Defined Radio using RTL-SDR". in 2013 International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing Proceeding, 2013 [Online], pp.852–855. Available: <https://ieeexplore.ieee.org/document/6526525>
- [2] R. Danyamol, T. Ajitha, and R. Gandiraj, "Real-Time Communication System Design using RTL-SDR and Raspberry". in 2013 International Conference on Advanced Computing and Communication Systems (ICACCS -2013) Proceeding, 2013 [Online], pp.1–5. Available: <https://ieeexplore.ieee.org/document/6938691>
- [3] K. Hajdarevic and S. Konjicija, 2015, "A Low Energy Computer Infrastructure for Radio VOIP Supported Communication and SDR APRS in Education and Disaster Relief Situations", in 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 [Online], pp.556–561. Available: <https://ieeexplore.ieee.org/document/7160334>
- [4] S. Miller, "1200 Baud Packet Radio Details" N1VG's Projects and Stuff, <http://n1vg.net/packet/index.php> [Accessed March 14, 2019].
- [5] Electron18, "Interface specification for transmission of binary data by frequency manipulation FSK, 2010 [Online]. Available: http://www.softelectro.ru/bell202_en.html [Accessed March 14, 2019].
- [6] APRSdroid, "APRSdroid - APRS for Android", 2017 [Online]. Available: <https://aprsdroid.org/> [Accessed March 14, 2019].
- [7] Laidukas, "Mods and performance of R820T2 based RTL SDR receiver" RTL-SDR.com, <https://www.rtl-sdr.com/mods-performance-r820t2-rtl-sdr/> [Accessed March 14, 2019].
- [8] FTDI, "Application Note AN_146 USB Hardware Design Guidelines for FTDI ICs. Technology", 2014 [Online]. Available: https://www.ftdichip.com/Documents/AppNotes/AN_146_USB_Hardware_Design_Guidelines_for_FTDI_ICs.pdf
- [9] Mobatek, "MobaXterm - Enhanced terminal for Windows with X11 server, tabbed SSH client, network tools and much more", 2018 [Online]. Available: <http://mobaxterm.mobatek.net> [Accessed March 14, 2019].
- [10] K. Keen, "RTL-SDR experimental branch", 2015 [Online]. Available: <https://github.com/keenerd/rtl-sdr> [Accessed March 14, 2019].
- [11] J. Langner, "Dire Wolf User Guide", 2016 [Online]. Available: <https://github.com/wb2osz/direwolf/blob/master/doc/User-Guide.pdf>
- [12] M. Wakita, "All Satellites Frequency List Update", 2019 [Online]. Available: <http://www.ne.jp/asahi/hamradio/je9pel/satslist.doc> [Accessed March 13, 2019].