

## **PROTOTYPE LOW COST SOFTWARE DEFINED RADIO (SDR) USING RASPBERRY BOARD**

**Naufal Faris Saha Sika<sup>1)</sup>, Faridatun Nadziroh, S.ST., MT.<sup>2)</sup>, dan Dr. I Gede Puja Astawa, ST., MT.<sup>3)</sup>**

<sup>1,2,3</sup>Teknik Telekomunikasi, Politeknik Elektronika Negeri Surabaya  
E-mail: ljnaufal@te.student.pens.ac.id<sup>1)</sup>, faridatun@pens.ac.id<sup>2)</sup>, puja@pens.ac.id<sup>3)</sup>

### **Abstract**

*Software Defined Radio (SDR)* is a radio system in which components that are typically built by *Hardware* (mixers, filters, modulators, demodulators, etc.) are replaced by *Software*. *Software Defined Radio (SDR)* is currently well-known among the global amateur radio community. Various *Transceiver* designs based on *Software Defined Radio (SDR)* have been widely applied, and various *SDR application Software* has been developed. However, the use of *Software Defined Radio (SDR)* requires access to the Radio Frequency Spectrum, which is one of the most expensive and scarce resources. Therefore, in this final project, the use of Raspberry Pi as a signal generator and RTL-SDR as a signal receiver is proposed, enabling the transmission of data through radio frequencies. The data transmitted includes audio, text, and images. The research results show that Raspberry Pi can be used as a radio wave transmitter through the GPIO pins of the Raspberry Pi. The signal is then received by RTL-SDR, and the results are displayed using third-party *Software* that generates output according to its classification. The maximum testing distance for this project is up to 15 meters, using additional antennas on the transmitter side.

**Keywords:** *Software Defined Radio (SDR), RTL-SDR, Raspberry Pi, Radio Frequency Spectrum.*

### **Abstrak**

*Software Defined Radio (SDR)* merupakan suatu sistem radio di mana komponen yang biasanya dibangun oleh perangkat keras (mixer, filter, modulator, demodulator, dll.) Fungsinya digantikan oleh perangkat lunak. *Software Defined Radio (SDR)* saat ini sangat dikenal di kalangan komunitas radio amatir di seluruh dunia. Berbagai desain *Transceiver* berbasis *Software Defined Radio (SDR)* telah banyak diaplikasikan, dan berbagai perangkat lunak aplikasi SDR telah dikembangkan. Namun, penggunaan *Software Defined Radio (SDR)* memerlukan Spektrum Frekuensi Radio, dan Spektrum Frekuensi Radio merupakan salah satu sumber daya yang paling mahal dan langka. Untuk itu, dalam proyek akhir ini diusulkan penggunaan Raspberry Pi sebagai generator sinyal dan RTL-SDR sebagai penerima sinyal, yang dapat mentransmisikan data melalui frekuensi radio. Data yang dikirimkan berupa Audio, Teks dan Gambar. Hasil penelitian ini, raspberry pi dapat digunakan sebagai transmitter pemancarkan gelombang radio melalui pin GPIO Raspberry. Kemudian sinyal diterima oleh RTL-SDR dan hasilnya ditampilkan menggunakan *Software* pihak ketiga yang memunculkan output sesuai klasifikasinya. Jarak maksimal pengujian proyek ini sejauh 15 meter dengan menggunakan Antenna tambahan pada sisi transmitter.

**Kata Kunci:** *Software Defined Radio (SDR), RTL-SDR, Raspberry Pi, Radio Frequency Spectrum.*

## PENDAHULUAN

Standar konektivitas nirkabel telah dikembangkan oleh badan standarisasi untuk memenuhi persyaratan berbagai aplikasi. *Transceiver* nirkabel harus dilengkapi dengan *Transceiver* Frekuensi Radio (RF) untuk mendukung standar nirkabel. *Transceiver* RF tradisional dirancang dan diimplementasikan pada chip radio atau modul tertanam dalam *System-on-a-Chip* (SoC), memastikan ukuran kecil, kinerja tinggi, konsumsi daya rendah, dan biaya. Namun, desain implementasi tradisional ini membatasi kemampuan program dan fleksibilitas *Transceiver* RF secara langsung atau tidak langsung. Solusi alternatif untuk mengimplementasikan *Transceiver* RF adalah dengan menggunakan platform *Software Defined Radio* (Dereje, Hakim, Laurent, & Marion, 2022).

Di pasaran, platform *Software Defined Radio* hadir dengan konfigurasi, kinerja dan biaya yang beragam (Khan, 2011). Berdasarkan riset yang telah dilakukan di berbagai platform jual beli *online*. Harga *Software Defined Radio* sangat beragam, mulai dari ratusan ribu hingga ratusan juta tergantung spesifikasi. Sebagai perbandingan beberapa platform *Software Defined Radio* yaitu HackRF One di range harga 8 Juta, USRP N200 di range harga 46 Juta, BladeRF 2.0 *micro* xA9 di range harga 15 Juta dan RTL-SDR 2832U di range harga 200 Ribu. Dari berbagai jenis *Software Defined Radio* yang disebutkan RTL-SDR 2832U merupakan SDR yang paling terjangkau apabila ingin melakukan percobaan menggunakan platform *Software Defined Radio* (Laufer, 2014), namun RTL-SDR 2832U hanya bisa berperan sebagai *receiver*, hal tersebut disebabkan karena Desain Awal RTL-SDR 2832U dirancang untuk menerima sinyal TV digital dan Tuner yang digunakan memiliki cakupan frekuensi yang cocok untuk menerima berbagai sinyal radio, seperti FM, AM, ADS-B, dan berbagai sinyal frekuensi tinggi lainnya. Hal ini dapat diatasi dengan menambah Raspberry Pi yang berperan sebagai transmitter yang memancarkan gelombang radio melalui pin GPIO Raspberry pi. Dengan menambah Raspberry Pi yang berperan sebagai Transmitter sinyal Radio, maka eksperimen penggunaan platform SDR dapat dilakukan menggunakan RTL-SDR 2832U (Tomar & Bhatia, 2015). Penggunaan RTL-SDR 2832U dan penggunaan Raspberry pi pada pengembangan platform SDR dapat memangkas biaya hingga 50%.

Berdasarkan latar belakang yang ada, penelitian dilakukan tentang penerapan *Software Defined Radio* (SDR) dengan biaya rendah menggunakan Raspberry Pi yang dapat digunakan untuk mengirimkan data berupa audio, teks, dan gambar melalui

frekuensi radio. Raspberry Pi bertindak sebagai pemancar yang mengirimkan data melalui pin GPIO dan RTL-SDR yang terhubung ke komputer pribadi berfungsi sebagai penerima yang menerima frekuensi radio yang dipancarkan Raspberry, kemudian data yang dikirimkan akan diencode menggunakan beberapa *Software* untuk menampilkan output yang diinginkan.

Dalam penelitian *Prototype Low Cost Software Defined Radio using Raspberry Pi board* ditentukan batasan masalah sebagai berikut:

1. Raspberry Pi Berperan sebagai pemancar sinyal
2. RTL-SDR Berperan sebagai penerima sinyal
3. Pengiriman data berupa Audio, Teks, dan Gambar
4. Audio yang dikirimkan berformat .wav
5. Gambar yang dikirimkan berformat .png
6. Pengiriman data dilakukan pada jarak 3 Meter, 6 Meter dan 10 Meter
7. *Software* encoder yang digunakan adalah SDRSharp, PDW, MMSSTV.

Tujuan dari penelitian ini adalah terwujudnya *Prototype Low Cost Software Defined Radio* menggunakan raspberry pi yang dapat mengirim dan menerima data melalui frekuensi radio sebagai pengembangan dan pengaplikasian *Software Defined Radio* dengan biaya yang lebih murah dibandingkan dengan SDR yang ada dipasaran.

## **METODE PENELITIAN**

Metode penelitian yang digunakan dalam penelitian ini menggunakan pengembangan metode *waterfall*, dimana didalamnya terdapat 6 tahap yang dilakukan yaitu: Studi Literatur, Identifikasi Kebutuhan, Perancangan Sistem, Implementasi dan Pengujian Sistem, Pengambilan Data dan Analisa.

## **HASIL DAN PEMBAHASAN**

### **a. Studi Literatur**

Studi literatur mencari penelitian terkait untuk mendukung kebenaran dan keakuratan data pada proyek akhir. Referensi yang dikaji meliputi konsep dasar sdr, teori dan prinsip kerja raspberry pi sebagai generator sinyal, cara kerja raspberry mengirimkan audio dan data melalui frekuensi radio, cara kerja protokol pocsag dalam

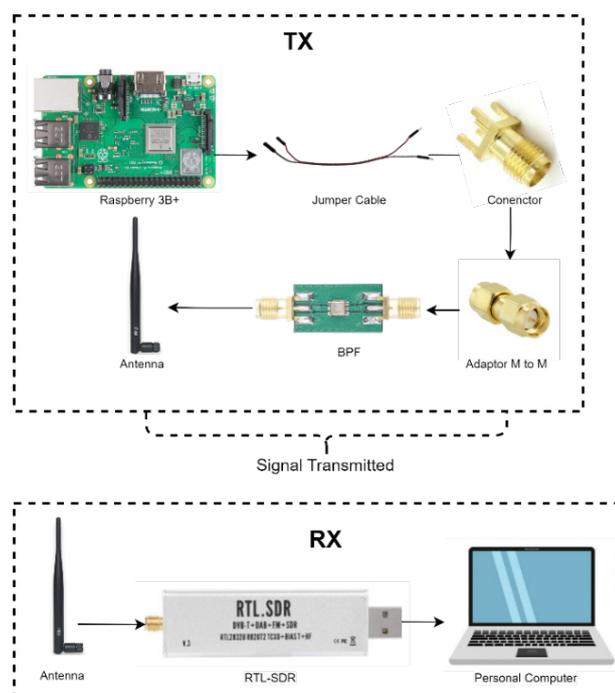
mengirim dan menerima teks, dan cara kerja SSTV dalam dalam mengirim dan menerima gambar.

#### b. Identifikasi Kebutuhan

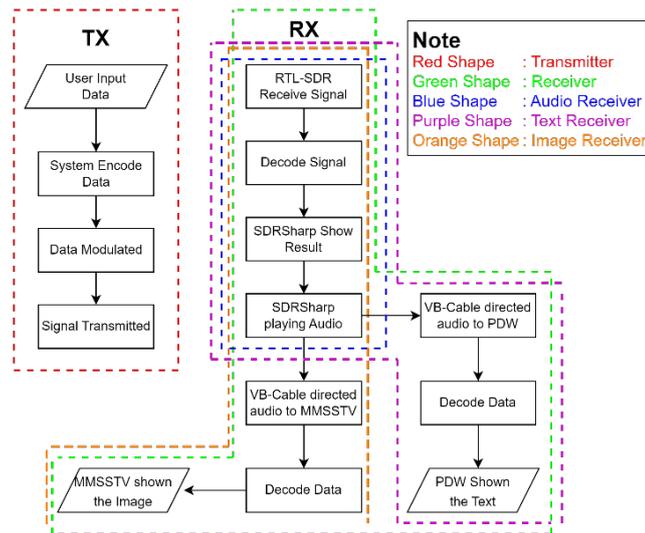
Identifikasi Kebutuhan yang dilakukan mencakup kebutuhan yang diperlukan pada sisi Transmitter dan *Receiver*, baik berupa *Software* maupun *Hardware*. Adapun yang diperlukan adalah : 1 set Raspberry pi model 3 B+, Kabel Jumper, Konektor SMA, Adaptor SMA Male to Male, *Band Pass Filter*, Antenna, PC/Laptop, RTL-SDR 2832U, Kabel LAN, Micro-SD Card.

#### c. Perancangan Sistem

Dalam penelitian ini, kami mengembangkan sistem untuk mengirim data melalui radio menggunakan Raspberry Pi dan RTL-SDR. Raspberry Pi dikonfigurasi dengan perangkat lunak rpitx untuk mengirim sinyal melalui pin GPIO, dengan perluasan pin menggunakan kabel jumper, konektor, adapter SMA, filter jalur lewat, dan antena SMA untuk meningkatkan jarak transmisi (Christiyono, Santoso, & Setiawan, 2009). Hasil pengiriman data mencakup kualitas puncak dan SNR audio selama pengiriman audio, kalimat yang ditampilkan selama pengiriman teks, dan kualitas gambar yang diterima selama pengiriman gambar. Ilustrasi implementasi sistem dapat dilihat pada Gambar 1.



Gambar 1. Ilustrasi Perancangan sistem

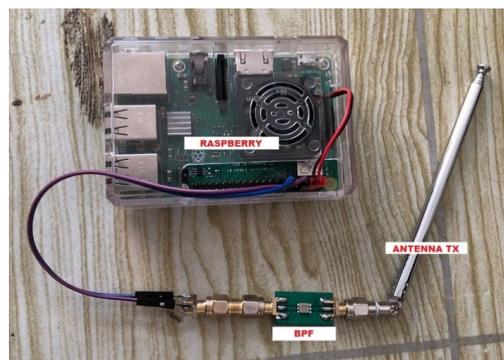


Gambar 2. Flowchart Sistem

Gambar 2 menjelaskan tentang implementasi flowchart sistem. Bagian pertama melibatkan Raspberry Pi sebagai pengirim sinyal radio dengan konfigurasi dari repository rpitx (Pella & Lami, 2022). Pengguna dapat mengirimkan data audio dalam format .wav yang diubah menjadi modulasi FM atau teks dalam bentuk ASCII dengan sinyal Tinggi untuk kode 1 dan Rendah untuk kode 0. Gambar .png atau .jpg juga dapat dikirimkan setelah diubah ke format SSTV. Bagian kedua melibatkan RTL-SDR yang berfungsi sebagai penerima sinyal pada frekuensi tertentu. Sinyal tersebut kemudian ditampilkan di perangkat lunak SDRSharp untuk audio, dan perangkat lunak VB-Cable dan dekoder seperti PDW Paging Decoder untuk teks serta MMSSTV untuk gambar digunakan untuk mendekode sinyal tersebut.

#### d. Implementasi dan Pengujian Sistem

Implementasi dari perancangan yang sudah dibuat dapat dilihat pada gambar 3 dan gambar 4.



Gambar 3. Implementasi Transmitter

Gambar 4. Implementasi *Receiver*

Sistem diuji dengan mengirim data pada jarak 3, 6, dan 10 meter dengan menyamakan frekuensi pada sisi pemancar dan penerima untuk mengevaluasi kualitas transmisi. Masing-masing pengujian memiliki parameter tersendiri, untuk pengiriman audio parameter utama yang digunakan adalah SNR karena ini adalah perbandingan antara daya sinyal dan daya noise. Setelah mendapatkan SNR melalui SDRSharp, dilakukan perhitungan manual untuk memverifikasi nilai SNR menggunakan rumus (1) (Kadence WP, 2023).

$$\text{SNR} = \text{Peak} - \text{Floor} \quad (1)$$

Nilai puncak (*Peak*) dan lantai (*Floor*) yang diperoleh dari perangkat lunak SDRSharp menggunakan unit dBFS. Jika Anda ingin melakukan perhitungan manual, Anda harus mengonversinya ke unit dB menggunakan rumus(2).

$$\text{dB} = \text{dBFS} - 20 \times \log(2) \quad (2)$$

Setelah mendapatkan SNR dari program dan SNR yang dihitung, Anda dapat menghitung kesalahan (error) menggunakan rumus (3) untuk menentukan akurasi SNR program.

$$\text{Error} = \frac{(\text{SNR Program} - \text{SNR Manual})}{\text{SNR Manual}} \times 100\% \quad (3)$$

*Peak* = Kekuatan Sinyal

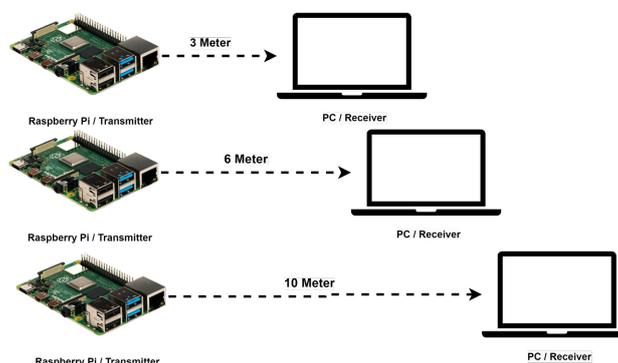
*Floor* = Interferensi Frekuensi

SNR = Rasio antara Puncak (*Peak*) dan Interferensi Frekuensi (*Floor*)

Untuk pengiriman gambar, terdapat dua parameter, yaitu "Jelas" yang mengindikasikan sejauh mana gambar terlihat baik, dan "gambar rusak" yang mengacu pada sejauh mana gambar mengalami kerusakan atau distorsi. Sedangkan untuk pengiriman teks, parameter yang diperhatikan adalah apakah teks yang dikirim dan diterima sama persis, tanpa perubahan atau modifikasi.

### e. Skenario Pengujian

Pengujian ini bertujuan untuk mengetahui berapa jauh daya pancar transmitter. Ilustrasi pengujian dapat dilihat pada gambar 5.



Gambar 5. Skenario Pengujian

Pengujian sistem dilakukan untuk mengevaluasi daya pancar transmitter dan kinerja pengiriman data pada jarak 3, 6, dan 10 meter. Hasilnya penting untuk menentukan keandalan dan kualitas transmisi data serta mengidentifikasi potensi perbaikan. Data tersebut akan digunakan sebagai dasar untuk mengoptimalkan daya pancar transmitter dan menjaga kualitas pengiriman data pada berbagai jarak.

### f. Pengujian dan Analisa Data

Dari pengujian yang telah dilakukan pada jarak 3,6 dan 10 Meter untuk masing-masing pengiriman data, dapat dilihat pada Tabel 1, 2 dan 3.

Tabel 1  
Data Transmisi Audio

Data	Jarak 3 meter			Jarak 6 meter			Jarak 10 meter		
	SNR-P (dB)	SNR-M (dB)	Error (%)	SNR-P (dB)	SNR-M (dB)	Error (%)	SNR-P (dB)	SNR-M (dB)	Error (%)
Audio 1	28,2	28,2	0	26,4	26,4	0	19,4	19,4	0
Audio 2	30,3	30,4	0,3	27,3	27,3	0	20,6	20,6	0
Audio 3	30,3	30,4	0,3	25,4	25,4	0	19,6	20,0	0,5
Audio 4	29,0	29,0	0	29,2	29,2	0	19,6	19,6	0
Audio 5	32,9	35,0	6	30,0	30,0	0	19,3	19,3	0

Pengiriman audio dilakukan sebanyak 5 kali dengan menggunakan audio yang berbeda. Data SNR-P (Program) diperoleh secara acak selama pengujian, sementara SNR-M (Manual) dihitung dengan rumus sebelumnya. Hasil menunjukkan bahwa SNR-P dan SNR-M tidak berbeda jauh, dengan kesalahan kurang dari 10%. SNR-P masuk dalam klasifikasi Sinyal sangat baik (25dB–40dB) dan Sinyal Lemah (15dB-25dB), hal

ini menunjukkan bahwa Raspberry Pi mampu berperan sebagai pemancar sinyal radio lokal, meskipun daya input Raspberry Pi 3 Model B+ hanya 5V/2.5A.

Tabel 2  
Data Transmisi Teks

Teks dikirim	Teks diterima		
	Jarak 3 meter	Jarak 6 meter	Jarak 10 meter
Pasca.Sarjana	Pasca.Sarjana	Pasca.Sarjana	Pasca.Sarjaeaw
bismillahh	bismillahh	bismillah	bismillahh
12345	12345	12345	12223
Sukolilo	Sukolilo	Sukolilo	Suko1eq2
Alhamdulillah	Alhamdulillah	Alhamdulillah	Alhamdulillah

Pengiriman teks dilakukan sebanyak 200 kali pada masing-masing jarak dan teks. Tabel 2 menunjukkan bahwa pada jarak 3 dan 6 meter, teks diterima tanpa perubahan. Namun, pada jarak 10 meter, beberapa teks mengalami perubahan akibat gangguan dan penurunan kualitas sinyal di pihak penerima. Proses pengodean kode ASCII menjadi teks terpengaruh oleh gangguan ini, sehingga menghasilkan perbedaan dalam teks yang diterima.

Tabel 3  
Data Transmisi Gambar

Percobaan Pengiriman	Klasifikasi Gambar		
	Jarak 3 meter	Jarak 6 meter	Jarak 10 meter
Percobaan ke-1	Jelas	Jelas	Jelas
Percobaan ke-2	Jelas	Jelas	Jelas
Percobaan ke-3	Jelas	Jelas	Jelas

Pengiriman gambar dilakukan 3 kali dalam setiap percobaan, dengan tujuan mengevaluasi kualitas pengirim dan penerima menggunakan mode SSTV yang sama, yaitu ScottieS1 yang telah terbukti memberikan hasil terbaik selama pengujian sistem. Hasil pengujian, seperti yang tercantum dalam Tabel 3, menunjukkan bahwa transmisi gambar pada jarak hingga 10 meter diklasifikasikan sebagai "jelas," dengan gambar yang dapat dilihat dengan sedikit distorsi. Namun, saat pengujian sistem dilakukan dalam waktu yang terlalu lama dan transmitter menjadi panas, terjadi klasifikasi "gambar rusak," yang mengakibatkan kerusakan gambar yang dikirim dan membuatnya tidak dapat dikenali.

## SIMPULAN

*Software Defined Radio* (SDR) memisahkan perangkat keras dan perangkat lunak dalam sistem komunikasi nirkabel, memberikan fleksibilitas, rekonfigurabilitas, dan efisiensi dalam pengembangan serta pengujian sistem komunikasi. Pemanfaatan *Band*

*Pass Filter* (BPF) dengan koneksi pin GPIO Raspberry Pi ke antena meningkatkan kualitas dan jangkauan transmisi, sedangkan hanya mengandalkan pin GPIO menghasilkan transmisi berkualitas rendah dengan banyak noise dan jangkauan terbatas.

Secara umum, kualitas pengiriman audio, akurasi pengiriman teks, dan kejelasan gambar dipengaruhi oleh jarak transmisi. Pada jarak 3 meter, kualitasnya baik dengan SNR tinggi, pengiriman teks yang error kecil, dan gambar yang jelas tanpa frame yang hilang. Pada jarak 6 meter, terjadi penurunan SNR audio tetapi masih cukup tinggi, dengan pengiriman teks yang bebas dari kesalahan dan gambar yang tetap jelas tanpa frame yang hilang. Namun, pada jarak 10 meter, terjadi penurunan signifikan dalam kekuatan sinyal audio, beberapa kesalahan dalam pengiriman teks, dan sedikit noise atau distorsi pada sinyal gambar. Meskipun demikian, sistem masih mampu memulihkan sebagian besar data dalam transmisi gambar.

Sebagai saran untuk penelitian masa depan, pertimbangkanlah untuk menambahkan mode pengiriman seperti video streaming dan kamera live streaming, mendalami penelitian tentang koreksi kesalahan pada jarak > 6 meter untuk memastikan keakuratan teks yang diterima, serta menciptakan TX-RX pada Raspberry Pi untuk menghasilkan *Software Defined Radio* yang optimal dengan biaya yang terjangkau.

## DAFTAR PUSTAKA

- Christiyono, Y., Santoso, I., & Setiawan, B. (2009). Perancangan Antena  $5/8\lambda$  Berpolarisasi Circular Pada Band VHF (30-300Mhz). *Jurnal Teknik Elektro*, 2(1), 53-59.
- Dereje, M. M., Hakim, B., Laurent, G., & Marion, B. (2022). Software Defined Radio Platforms for Wireless Technologies. *IEEE Access*, 10, 26203-26229. <https://doi.org/10.1109/ACCESS.2022.3154364>
- Kadence WP. (2023). What is dBFS in Audio? How it's Calculated and Used. <https://audiointerfacing.com/dbfs-in-audio/>
- Khan, A. (2011). *Software Defined Radio(SDR) Prototype: Basic Concepts and Methods*. LAP LAMBERT Academic Publishing.
- Laufer, C. (2014). *The Hobbyist's Guide to the RTL-SDR: Really Cheap Software Defined Radio*.
- Pella, S. I., & Lami, H. F. J. (2022). Pengembangan Radio Digital Pembelajaran Jarak Jauh Menggunakan Modul Rpitx Raspberry Pi. *Jurnal Media Elektro*, XI(1), 45-51.
- Tomar, V. S., & Bhatia, V. (2015). Low Cost and Power Software Defined Radio using Raspberry Pi for Disaster Effectuated Regions. *Procedia Computer Science*, 58, 401-407.