

DESAIN SOFTWARE DEFINED RADIO TRANSCEIVER BERBASIS RED PITAYA**F. Yudi Limpraptono¹, Sotyohadi², Vivi Nur Cholidah³, Muhammad Rifky Arrohman⁴**^{1,2,3,4} Teknik Elektro, Institut Teknologi Nasional Malang

fyudil@lecturer.itn.ac.id

ABSTRAK

Software defined radio (SDR) adalah paradigma baru dalam desain perangkat komunikasi wireless. Teknologi SDR saat ini digunakan secara luas pada bidang telekomunikasi komersial, telepon bergerak dan banyak digunakan pada kalangan komunitas radio amatir. SDR adalah suatu sistem radio dimana komponen-komponennya yang biasanya di bangun oleh perangkat keras (mixer, filter, modulator, demodulator dll) digantikan fungsinya oleh perangkat lunak. Berbagai desain transceiver berbasis SDR yang bekerja pada band high frequency (HF) telah banyak diaplikasikan, dan berbagai perangkat lunak aplikasi SDR telah dikembangkan. Perangkat transceiver berbasis SDR cenderung digemari oleh komunitas radio amatir dikarenakan kesederhanaan dari rangkaian elektroniknya dan kualitas kinerjanya yang baik. Namun permasalahan yang sering dijumpai adalah bahwa pengguna harus menggunakan aplikasi dengan menggunakan personal komputer, bagi orang awam terkesan menggunakan Radio transceiver SDR terasa ribet dan sangat sulit untuk dioperasikan. Dari permasalahan tersebut, maka pada penelitian ini diusulkan sebuah purwarupa radio transceiver berbasis SDR Red pitaya dengan aplikasi yang tertanam pada sebuah Raspberry Pi. Metode penelitian yang digunakan dalam penelitian ini adalah metode penelitian dan pengembangan atau R&D. Setelah dilakukan pengujian pada radio SDR transceiver yang telah direalisasikan dapat disimpulkan bahwa sistem dapat bekerja dengan sempurna dan kinerja sistem adalah sangat baik dengan hasil pengujian frekuensi memberikan error sebesar 0.000068%, pengujian terhadap hasil pemodulasian memberikan hasil readability 92,5%, Strength 9, dan Quality 9. Diharapkan hasil penelitian ini dapat berkontribusi dan dapat dipakai sebagai panduan dalam desain radio transceiver SDR yang efisien dikalangan penggiat radio amatir.

Keyword : *Transceiver, Software Defined Radio, Red Pitaya***1. PENDAHULUAN**

Pada saat ini perkembangan bidang komputasi *digital* mengalami peningkatan secara berkelanjutan. Ditunjang dengan tersedianya komponen ADC (*analog to digital converter*) dan DAC (*digital to analog converter*) berkecepatan tinggi, memungkinkan untuk melakukan pemrosesan sinyal *digital* untuk sistem frekuensi radio. Salah satu pengembangan ini menggunakan teknologi SDR (*software defined radio*) yang saat ini digunakan secara luas dalam bidang komersial seperti satelit komunikasi sampai dengan telepon bergerak. Teknologi SDR juga sangat dikenal di lingkungan operator dan kalangan komunitas radio amatir [1]. *Software defined radio* adalah paradigma baru desain perangkat komunikasi *wireless*. Konsep *software radio* digagas pertama kali oleh Joseph Mitola pada awal tahun 1990 [2]. Joseph Mitola menggambarkan bahwa sebuah SDR ideal terdiri dari komponen fisik yaitu berupa antena dan sebuah ADC pada sisi penerima dan sebaliknya pada sisi pemancar ada DAC dan antena transmisi. Fungsi selebihnya akan ditangani oleh prosesor yang telah diprogram [3]. *Software defined radio* saat ini banyak dikembangkan dikalangan komunitas radio amatir. Berbagai desain *transceiver* berbasis SDR yang bekerja pada band *high frequency* (HF) telah banyak diaplikasikan, dan berbagai perangkat lunak dengan mengaplikasikan SDR telah banyak dikembangkan.

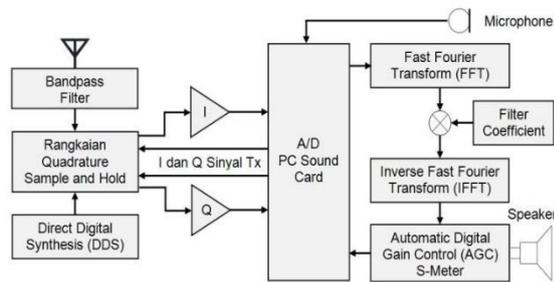
Perangkat *transceiver* berbasis SDR cenderung digemari oleh komunitas radio amatir dikarenakan kesederhanaan dari rangkaian elektroniknya dan kualitas kinerjanya yang baik. Namun permasalahan yang sering dijumpai adalah bahwa pengguna harus menggunakan aplikasi dengan menggunakan personal komputer, bagi orang awam terkesan menggunakan Radio transceiver SDR terasa ribet dan sangat sulit untuk dioperasikan. Dari permasalahan tersebut, maka pada penelitian ini diusulkan sebuah purwarupa radio transceiver berbasis SDR Red pitaya dengan aplikasi yang tertanam pada sebuah sistem embedded Raspberry Pi Versi 4, dengan harapan dapat berkontribusi untuk menyediakan perangkat radio SDR yang sederhana, kecil, murah dan mudah untuk digunakan.

2. TINJAUAN PUSTAKA

Teknologi SDR menawarkan beberapa keuntungan dalam memberikan solusi yang spesifik. Konsep SDR tidak hanya memiliki kemampuan memperbaiki *error* secara *real time* tetapi dari studi dapat diidentifikasi beberapa aplikasi yang telah dikembangkan secara pesat seperti *Dynamic Spectrum Positioning, Opportunity Driven Multiple Access (ODMA), Spectrum Regulation and Cost Reduction* (implementasi SDR lebih murah) [4]. Filosofi SDR mulai berdampak tinggi pada bidang telekomunikasi. Beberapa penelitian yang menggunakan teknologi SDR seperti *Driver*

Assistance [5], GPS signals Reception [6], HF Propagation Analysis [7], Interpretation of Cellular Technology Emission terutama modulasi OFDM [8] dan Identification of Radio Frequency Emissions [9], telah menunjukkan keandalan dan kemudahan teknologi ini dalam implementasinya untuk berbagai aplikasi telekomunikasi.

SDR adalah suatu sistem radio dimana komponen-komponennya yang biasanya di bangun oleh perangkat keras (*mixer, filter, modulator, demodulator* dll) digantikan fungsinya oleh perangkat lunak. Penerapan SDR bisa dilakukan dengan banyak cara dan berbagai kombinasi, yang blok diagramnya diperlihatkan pada Gambar 1., berikut:



Gambar 1. Desain Transceiver SDR

Dari Gambar 1, blok diagram desain transceiver SDR dapat dijelaskan sebagai berikut: pada proses penerimaan sinyal radio, frekuensi pembawa f_c yang berasal dari antenna diteruskan melalui *band pass filter* (BPF) dan kemudian diumpungkan ke *parallel mixer* (*quadrature mixer*). Pada *mixer* sebelah bawah sinyal *sinus* dari lokal osilator *direct digital synthesis* (DDS) dicampurkan dengan f_c . Sinyal hasil *mixer* dilewatkan *low pass filter* (LPF) sebelum dikonversi oleh *analog to digital converter* (ADC). Keluaran *mixer* sebelah atas menghasilkan *inphase* $I(t)$. Keluaran *mixer* sebelah bawah menghasilkan *quadrature* $Q(t)$. Sinyal I dan Q diteruskan ke *sound card personal computer* (PC) untuk diproses lebih lanjut secara *digital* menggunakan perangkat lunak SDR. Sinyal hasil olahan akan dikeluarkan melalui *audio out sound card* PC. Untuk proses pengiriman sinyal (*transmitter*) akan berlaku kebalikannya.

Saat ini transceiver berbasis SDR banyak dipergunakan dan dikembangkan di kalangan penggiat radio amatir. Salah satunya adalah transceiver SDR Genesis yang bekerja pada band frekuensi HF [10]. Sistem menggunakan software SDR POWERSDR berbasis sistem operasi Windows. Sistem masih menggunakan oscillator kristal sehingga lebar band masih terbatas.



Gambar 2. Sistem Transceiver Genesis SDR [10]

Selain Genesis SDR ada beberapa model front end SDR yang telah dikembangkan antara lain, LMR SDR, RSHFIQ dengan penambahan oscillator berbasis DDS sehingga band frekuensi lebih luas, sehingga bisa mengcover seluruh band HF dari 0Hz sampai 30Mhz. Selain PowerSDR aplikasi lain yang sering digunakan adalah aplikasi HDSDR. Sistem Transceiver SDR berbasis aplikasi Window ditunjukkan pada Gambar 2.

3. METODE PENELITIAN

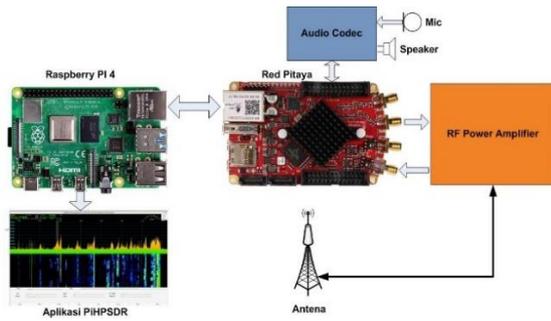
Metode penelitian yang digunakan dalam penelitian ini adalah metode penelitian dan pengembangan atau R&D. Diagram alir pelaksanaan penelitian ditunjukkan pada Gambar 3., langkah-langkah metode penelitian dan pengembangan diawali dengan studi literatur untuk mengumpulkan informasi yang dapat digunakan dalam memecahkan permasalahan, langkah kedua adalah tahap desain sistem SDR transceiver dan dilanjutkan dengan realisasi sistem dengan melakukan instalasi hardware dan software. Hasil dari penelitian akan menghasilkan sebuah purwarupa radio transceiver HF stand alone berbasis SDR Red Pitaya dengan pengendali utama adalah Raspberry Pi versi 4. Purwarupa sistem akan diuji di laboratorium secara eksperimental, untuk mendapatkan kesimpulan kinerja dari sistem yang telah direalisasikan.



Gambar 3. Diagram alir Pelaksanaan Penelitian

3.1. Desain Sistem

Desain Radio Transceiver HF SDR stand alone berbasis Red Pitaya ditunjukkan pada Gambar 4., sistem terdiri dari:



Gambar 4. Desain Sistem SDR Transceiver

1. Modul SDR Red Pitaya merupakan komponen utama dalam radio transceiver HF SDR stand alone. Modul SDR yang digunakan adalah Red Pitaya tipe STEM 125-14.
2. Unit pemroses sinyal dan pengendali sistem menggunakan Raspberry Pi 4.0 yang dilengkapi dengan LCD layar 4 inchi. Sistem operasi yang digunakan adalah Raspbian dan aplikasi SDR yang digunakan adalah aplikasi PiHPSDR.
3. Unit input output audio menggunakan Audio Codec WM8731SEDS
4. Purwarupa menggunakan RF linier power amplifier dengan daya 15 watt.
5. Unit Antena digunakan untuk mentransmisikan gelombang radio ke udara, dan jenis antena yang digunakan adalah open dipole.

3.2. Red Pitaya SDR

Red Pitaya sebenarnya adalah proyek yang ditujukan sebagai instrumen laboratorium pengukuran dan kontrol. Red Pitaya dikenal sebagai proyek open-source. Modul Red Pitaya memiliki dua input RF 125MS/s dan dua output RF 125MS/s, dengan bandwidth analog 50 MHz dan memiliki analog-ke-digital (ADC) 14 bit kecepatan tinggi. Perangkat lunak pada Red Pitaya mencakup aplikasi Red Pitaya sebagai osiloskop, penganalisis spektrum, generator sinyal, pengukur LCR dan sebagai radio SDR Transceiver pada band frekuensi HF (0-30 Mhz) [11][12]. Red Pitaya memiliki sebuah port USB 2.0, dan sebuah port Ethernet. Secara internal, Red Pitaya menggunakan Linux sebagai sistem operasi. Perangkat penyimpanan untuk sistem operasi adalah kartu micro-SD. Karena bandwidth ADC dan DAC yang lebar, Red Pitaya dapat digunakan sebagai penerima dan pemancar radio yang berbasis SDR. Modul Red Pitaya ditunjukkan pada Gambar 5.



Gambar 5. Modul Red Pitaya SDR

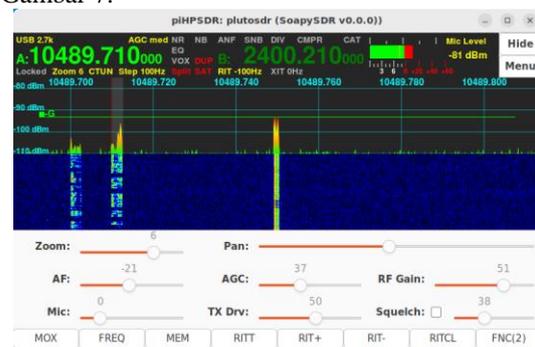
3.3. Raspberry Pi dan Aplikasi piHPSDR

Raspberry Pi adalah unit pemroses sinyal dan pengendali sistem dalam sistem radio SDR transceiver yang direncanakan, Raspberry Pi yang digunakan adalah Raspberry Pi versi 4.0 model B dengan sistem operasi linux Bullseye 64 bit. Raspberry Pi memiliki RAM 8GB, dan memori sistem menggunakan SD Card kapasitas 64 GB, memiliki 4 buah port USB dan sebuah port ethernet dan 2 buah port HDMI, selain itu Raspberry Pi 4.0 memiliki jalur komunikasi WiFi dan Bluetooth. Gambar modul Raspberry Pi ditunjukkan pada Gambar 6.



Gambar 6. Modul Raspberry Pi

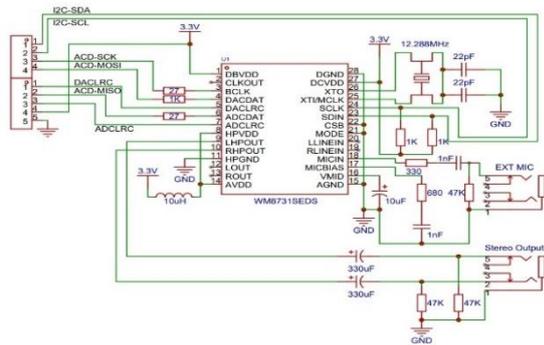
Raspberry Pi dalam menjalankan fungsinya sebagai pemroses sinyal, menggunakan aplikasi SDR piHPSDR yang dikembangkan oleh John Melton G0ORX/N6LYT dibawah lisensi GNU (free software foundation) *general public license*. piHPSDR adalah program SDR yang dapat mendukung modul SDR Red Pitaya. piHPSDR akan memproses sinyal I dan Q yang dikirim oleh Red Pitaya saat proses receive atau menghasilkan dan mengirimkan sinyal I dan Q ke modul Red Pitaya pada saat proses transmit. Selain itu aplikasi piHPSDR juga sebagai antarmuka grafis untuk pengguna, piHPSDR menggunakan toolkit GTK-3, sedangkan pemrosesan sinyal menggunakan library WSP. piHPSDR mengatur transfer data frekuensi radio digital (RF) antara perangkat keras radio dan WSP, selain itu berfungsi mentransfer data audio (baik dari mikrofon atau ke headphone), serta memproses sinyal input dari pengguna (baik dengan mouse/layar sentuh, keyboard, atau tombol eksternal). Aplikasi piHPSDR mampu menampilkan sinyal RF dalam bentuk grafis pada LCD monitor dalam bentuk spektrum maupun waterfall. Tampilan layar aplikasi piHPSDR ditunjukkan pada Gambar 7.



Gambar 7. Tampilan Layar Aplikasi piHPSDR

3.4. Audio Codec

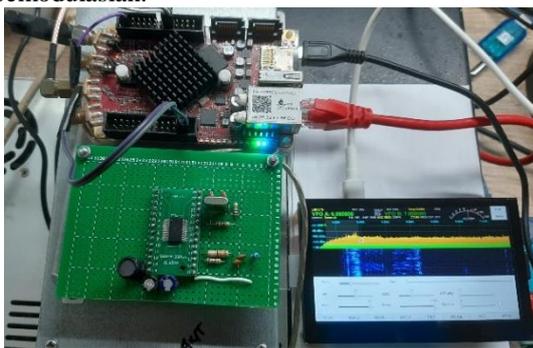
Rangkaian audio codec berfungsi untuk mengkonversi digital audio menjadi sinyal audio analog, selain itu berfungsi juga untuk mengkonversi sinyal audio analog yang berasal dari mikrofon menjadi audio digital. Transfer data antara Raspberry Pi dengan modul audio codec dilakukan melalui port Serial Peripheral Interface (SPI), sedangkan kontrol untuk operasi board audio codec dilakukan melalui port komunikasi I2C. Modul audio codec menggunakan chip WM8731SEDS, dengan spesifikasi, membutuhkan daya rendah, dengan tegangan supply 3.3V, Stereo codec dan memiliki driver untuk earphones. Gambar Rangkaian audio codec ditunjukkan pada Gambar 8.



Gambar 8. Rangkaian Audio Codec

4. HASIL DAN PEMBAHASAN

Sistem Radio transceiver SDR yang telah direalisasikan ditunjukkan pada Gambar 9. Untuk mengetahui kinerja dari sistem yang telah direalisasikan, dilakukan beberapa pengujian pada Radio SDR transceiver, yang meliputi: keakuratan frekuensi sinyal pembawa, pengujian pemodulasian berbagai mode pemodulasian (AM, SSB, FM), dan pengujian penerimaan sinyal RF berbagai mode pemodulasian.

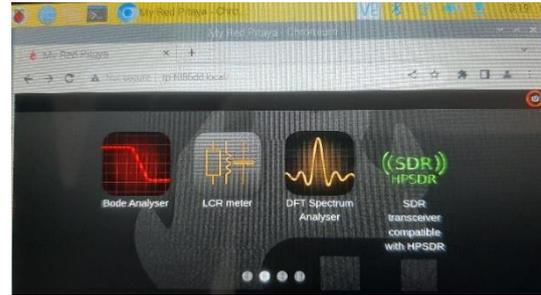


Gambar 9. Perangkat Transceiver SDR

Untuk memulai menjalankan radio transceiver SDR yang telah direalisasikan dengan Langkah-langkah sebagai berikut:

1. Membuka web browser yang ada pada menu Raspberry Pi, kemudian mengakses Modul Red Pitaya dengan cara mengetikkan IPnya atau ketik Mac Address dari Red Pitaya dengan format rp-xxxxxx.local/.

2. Kemudian akan tampil menu web dari Red Pitaya seperti ditunjukkan pada Gambar 10.



Gambar 10. Menu Web Modul Red Pitaya

3. Langkah selanjutnya adalah memulai menjalankan Aplikasi piHPSDR dengan mengklik icon yang ada di layar, seperti ditunjukkan pada Gambar 11.



Gambar 11. Tampilan Icon Aplikasi piHPSDR

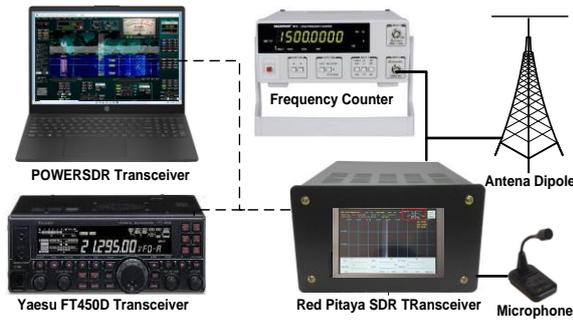
4. Tampilan aplikasi transceiver piHPSDR yang telah berjalan ditunjukkan pada Gambar 12.



Gambar 12. Tampilan Aplikasi piHPSDR

4.1. Pengujian keakuratan frekuensi sinyal pembawa.

Salah satu parameter penting dari sebuah transceiver adalah keakuratan frekuensi yang dipancarkan atau diterimanya. Untuk itu perlu diuji keakuratan dan kestabilan frekuensi yang dibangkitkan. Metode pengujian dilakukan dengan jalan menguji dan mengukur frekuensi yang dibangkitkan disetiap Band frekuensi kerjanya. Pengukuran dengan menggunakan sebuah frekuensi counter. Diagram pengujian frekuensi ditunjukkan pada Gambar 13. Hasil pengukuran ditunjukkan pada Tabel 1., menunjukkan error rata-rata sekitar 0.000068%.

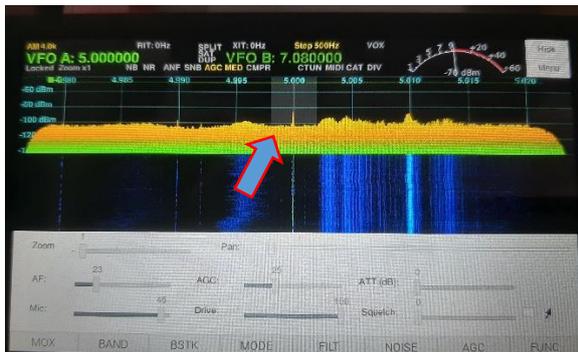


Gambar 13. Blok Diagram Pengujian

Tabel 1. Hasil pengujian keakuratan frekuensi

No	Band (meter)	Frekuensi Kerja (Mhz)	Frekuensi Pengukuran (Mhz)	Error (%)
1.	160	2.000000	2.000001	0.00005
2.	80	3.500000	3.500003	0.000085
3.	40	7.000000	7.000005	0.000071
4.	20	14.000000	14.000011	0.000079
5.	10	28.000000	28.000015	0.000054
Error rata-rata				0.000068

Selain itu pengujian frekuensi dilakukan juga dengan menggunakan frequency calibration 5 Mhz dengan menggunakan mode WWV. Hasil ditunjukkan pada Gambar 14., dengan hasil yang sangat presisi.



Gambar 14. Tampilan Pengujian Frekuensi dengan wwv frequency calibration 5Mhz

4.2. Pengujian pemodulasian

Pengujian pemodulasian SDR transceiver dilakukan dengan melakukan proses transmit pada berbagai macam mode pemodulasian seperti AM, SSB baik USB dan LSB, dan mode FM. Proses pengujian dilakukan dengan menggunakan radio receiver berbasis SDR dan Radio transceiver analog All Band Yaesu FT450D, gambar diagram blok pengujian ditunjukkan pada Gambar 13. Pengamatan dilakukan dengan mengamati spektrum maupun kualitas modulasinya. Hasil pengujian dapat ditampilkan pada Tabel 2., hasil pengamatan diperoleh kualitas pemodulasian diberbagai band dan berbagai mode menghasilkan kualitas yang baik sekali, dengan mengacu pada parameter RSQ reporting radio amatir [14]. Dengan penjelasan sebagai berikut:

Tabel 2. Hasil pengujian pemodulasian

No	Mode Pemodulasian	Hasil Pengujian		
		Readability (%)	Strength	Quality
1.	AM	95	9	9
2.	LSB	90	9	9
3.	USB	90	9	9
4.	FM	95	9	9
Rata-rata		92,5	9	9

Penjelasan hasil pengujian:

- Readability 92,5% (R5) berarti hasil *perfectly readable*.
- Strength 9 (S9) berarti hasil *very strong trace*
- Quality 9 (Q9) berarti *clean signal*

4.3. Pengujian proses penerimaan (demodulasi)

Pengujian demodulasi atau penerimaan radio SDR transceiver dilakukan dengan melakukan proses receive pada berbagai macam mode modulasi seperti AM, SSB baik USB dan LSB, dan mode FM. Proses pengujian dilakukan dengan menerima sinyal yang dihasilkan dari Radio transceiver analog All Band Yaesu FT450D, diagram pengujian ditunjukkan pada Gambar 13. Pengamatan dilakukan dengan mengamati spektrum maupun kualitas modulasinya. Hasil pengujian dapat ditampilkan pada Tabel 3., hasil pengamatan diperoleh kualitas penerimaan diberbagai band dan berbagai mode menghasilkan kualitas yang baik sekali, dengan mengacu pada parameter RSQ reporting radio amatir [14]. Dengan penjelasan sebagai berikut:

Tabel 3. Hasil pengujian penerimaan (demodulasi)

No	Mode Pemodulasian	Hasil Pengujian	
		Readability(%)	Quality
1.	AM	95	9
2.	LSB	95	9
3.	USB	95	9
4.	FM	95	9
Rata-rata		95	9

Penjelasan hasil pengujian:

- Readability 95% (R5) berarti hasil *perfectly readable*.
- Quality 9 (Q9) berarti *clean signal*

4.4. Perbandingan SDR berbasis Red Pitaya dengan SDR konvensional berbasis PC.

Tabel 4. Perbandingan Efisiensi

No	Parameter Efisiensi	SDR Berbasis PC	SDR Red Pitaya
1.	Konsumsi daya	>80 Watt	3,5 Watt
2.	Ukuran fisik	11.342 cm ³	252 cm ³
3.	Biaya sistem	> Rp. 9 Juta	< RP.7 Juta

Penjelasan:

Pemakaian SDR berbasis Red Pitaya lebih efisien dibandingkan SDR konvensional Berbasis PC, ditinjau dari konsumsi daya, bentuk fisik dan biaya.

5. KESIMPULAN

Setelah dilakukan pengujian pada radio SDR transceiver yang telah direalisasikan dapat disimpulkan bahwa sistem dapat bekerja dengan sempurna dan kinerja sistem adalah sangat baik, dengan prosentase error dari pengujian frekuensi hanya sekitar 0.000068%, demikian juga pengujian terhadap hasil pemodulasian (transmit) memberikan hasil readability 92,5%, Strength 9, dan Quality 9, sedangkan proses demodulasi (*receive*) diberbagai mode pemodulasian diperoleh hasil readability 95%, dan Quality 9 sehingga dapat disimpulkan bahwa kinerja sistem transceiver adalah sangat baik. Radio SDR transceiver yang telah direalisasikan merupakan Radio transceiver yang efisien dengan biaya rendah karena unit pemroses sinyal telah digantikan oleh sistem embedded berbasis Raspberry Pi. Diharapkan hasil penelitian ini dapat berkontribusi dan dapat dipakai sebagai panduan dalam desain radio transceiver SDR di kalangan penggiat radio amatir.

DAFTAR PUSTAKA

- [1] Retzler, "OpenWebRX:SDR Web Application for the Masses", *34th ARRL and TAPR Digital Communications Conference*, 2015.
- [2] J. Mitola, "Software Radio Survey, Critical Evaluation And Future Directions", *IEEE National Telesystems Conference*, Pp. 13-15, 1992.
- [3] J. R. Machado, "Software Define Radio: Basic Principles and Applications", *Revista Facultad de Ingenieria*, Vol. 24 No.38 pp.79-96, 2015.
- [4] S. Gultchev, K. Moessner, and D. Thilakawardana, Evaluation of Software Defined Radio Technology, Center for Communication System Research, University of Surrey, 2005.
- [5] N. Haziza, M. Kassab, and R. Knoopp, "Multi technology vehicular cooperative system based on Software defined Radio (SDR)", *Communication Technologies for Vehicles*, pp 84-95, Springer, 2012.
- [6] J. Seo, Y.H. Chen, and D.S. De Lorenzo, "A Real-Time Capable Software Defined Receiver Using GPU for Adaptive Anti-Jam GPS Sensors", *Sensor Journal*, 2011.
- [7] P.B. Nagaraju, E. Koski, And T. Melodia, "A Software Defined Radio Ionospheric Chirpsounder for HF Propagation Analysis", 2009.
- [8] H. S. Yoo, B. Park, and S. H. Kim, "Seamless Vertical Handover in Software Defined Radio Terminal", *International Journal of Control and Automation*, vol. 2, 2009.
- [9] M. Islam, M.A. Hannan, S. A. Samad, and A. Hussain, "Software Defined Radio for RFID Application", *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, 2009.
- [10] DE Robby, WB5RVZ Genesis Radio G40 Build Experience, Diakses pada 1 Desember 2023, https://www.wb5rvz.com/sdr/genesis_g40/index.htm
- [11] Ibrahim, Dogan (2016). *Explore, experiment, program Red Pitaya for test & measurement*. London, UK: Elektor International Media BV. ISBN 978-1-907920-53-0.
- [12] Richards, Mike (July 2016). "Pi Updates and Red Pitaya". *Radio User*. Bournemouth, UK: PW Publishing Ltd. 11 (7): 17. ISSN 1748-8117
- [13] YC3BCS, *Buku Amatir Radio ORARI dan Kegiatannya*, ORARI lokal Malang Raya, 2009