

## DESAIN LOW NOISE TRANSCEIVER 7 MHZ BERBASIS SOFTWARE DEFINED RADIO (SDR)

Sotyohadi <sup>1</sup>, Irrine Budi Sulistiawati <sup>2</sup>

<sup>1,2)</sup> Jurusan Teknik Elektro, Institut Teknologi Nasional Malang  
[sotyohadi@lecturer.itn.ac.id](mailto:sotyohadi@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 komersil seperti satelit komunikasi, 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. *Software defined radio* saat ini sangat dikenal dikalangan komunitas radio amatir di seluruh dunia. 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 digemari oleh komunitas radio amatir dikarenakan rangkaian elektroniknya yang sederhana. Namun dari kelebihan tersebut muncul permasalahan yang sering dijumpai adalah *noise* pada penerima yang tinggi dan sensitivitas rendah. Dengan latar belakang permasalahan tersebut, maka pada penelitian ini dirancanglah suatu *low noise transceiver 7 MHz* berbasis *software defined radio (SDR)*. Metode yang digunakan dalam mengurangi *noise* adalah dengan pemisahan *grounding* dari rangkaian digital, analog dan penguat RF serta pemilihan *chip Tayloe detector* yang memiliki nilai resistansi switch yang rendah yaitu 4 ohm. Berdasarkan hasil pengujian dengan menggunakan *integrated circuit (IC)* yang memiliki nilai resistansi *switching* yang rendah 4 ohm didapatkan kinerja *transceiver* yang lebih baik, dimana penerimaan lebih *sensitive* dan memiliki *noise* yang lebih rendah.

**Keyword :** *Software Defined Radio, Tayloe Detector, Transceiver, Low Noise*

### 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 di frekuensi radio. Salah satu pengembangan ini menggunakan teknologi SDR (*software defined radio*) yang saat ini digunakan secara luas dalam bidang komersil 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 *noise* yang tinggi dan rendahnya sensitivitas dari *receiver* berbasis SDR. Dari permasalahan tersebut, maka pada penelitian ini diusulkan sebuah metode perancangan yang menggunakan perangkat keras berbasis SDR yang mampu meningkatkan kinerja *transceiver*.

Metode yang diterapkan adalah dengan menggunakan *grounding* yang terpisah antara rangkaian analog dan digital serta pemilihan rangkaian IC yang memiliki nilai resistansi *switching* yang rendah, sehingga diharapkan mampu mengatasi kelemahan yang selama ini terjadi pada perangkat *transceiver* berbasis SDR.

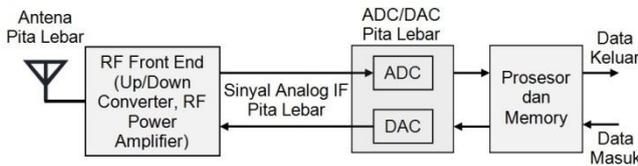
### 2. TINJAUAN PUSTAKA

#### 2.1. Teknologi Software Define Radio

Teknologi *Software defined radio (SDR)* diperkenalkan oleh Joseph Mitola pertama kali pada tahun 1991 [4]. Konsep SDR adalah sebuah radio yang dapat dikonfigurasi ulang atau diprogram ulang [2], sehingga menghasilkan sebuah perangkat komunikasi nirkabel dengan mode dan *band* frekuensi ditentukan oleh fungsi perangkat lunak. Teknologi SDR memiliki beberapa keuntungan karena memiliki sifat yang fleksibel, sehingga dapat dikonfigurasi ulang secara mudah sesuai dengan

perangkat komunikasi, selain hal itu teknologi SDR dapat diprogram ulang dan dapat diperluas [5].

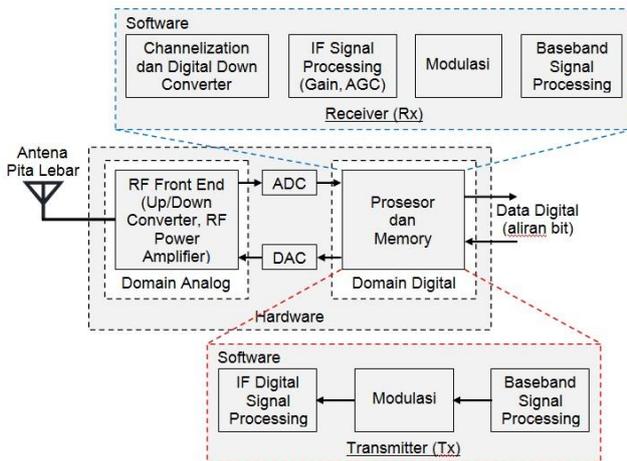
Arsitektur SDR yang ideal akan menempatkan ADC dan DAC sedekat mungkin dengan antena, sehingga membutuhkan ADC dan DAC *band* lebar. Fungsi radio akan dilakukan oleh perangkat lunak yang dijalankan oleh prosesor di komputer, sehingga menjadikan teknologi SDR lebih fleksibel [2].



Gambar 1. Arsitektur SDR yang Realistis.

Karena keterbatasan teknologi dan mahalanya ADC dan DAC *band* lebar, maka arsitektur SDR sedikit diubah dengan menempatkan ADC dan DAC setelah rangkaian RF *front end*, sehingga rangkaian yang realistis dapat diperlihatkan pada Gambar 1.

Arsitektur SDR yang realistis meletakkan posisi ADC dan DAC *band* lebar setelah *up/down converter*, sehingga proses konversi *analog* ke *digital* atau sebaliknya dilakukan terhadap sinyal *intermediate frequency* yang memiliki frekuensi lebih rendah dibandingkan dengan sinyal *radio frequency* (RF). Arsitektur SDR dilihat dari segi perangkat keras dan perangkat lunak yang diperlukan dengan mengacu pada arsitektur SDR realistis dapat ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur Transceiver SDR

Kesulitan dalam proses digitasi sinyal RF adalah keterbatasan kecepatan pencuplikan dari ADC. Sesuai dengan teori sampling Nyquist, bahwa sinyal *baseband* dengan frekuensi maksimum  $f_a$  harus dicuplik dengan frekuensi sampling  $f_s \geq 2 f_a$ . Proses digitasi sinyal RF atau IF dalam orde ratusan MHz sampai dengan GHz jika mengacu pada teori sampling tersebut, maka membutuhkan frekuensi

pencuplikan dua kali frekuensi yang dicuplik sehingga harga dari komponennya sangat mahal. Untuk menghemat biaya sistem yang diakibatkan mahalnya ADC frekuensi tinggi maka diterapkan *undersampling* dalam proses digitasi. Kaidah yang digunakan yaitu sinyal *bandpass* yang terletak antara frekuensi bawah  $f_L$ , frekuensi tengah  $f_C$  dan frekuensi atas  $f_U$  atau *bandwidth*  $B = f_U - f_L$  dapat disampling dengan frekuensi sampling  $f_s \geq 2B$  atau dengan frekuensi *sampling* yang besarnya dapat dipilih berdasarkan persamaan [6].

$$\frac{2f_U}{n} \leq f_s \leq \frac{2f_L}{n-1} \tag{1}$$

Dengan nilai  $n$  adalah bilangan integer yang memenuhi syarat :

$$1 \leq n \leq \left\lfloor \frac{f_U}{f_U - f_L} \right\rfloor \tag{2}$$

Pemilihan frekuensi dapat diuraikan sebagai berikut:

- Pilih frekuensi sampling sedemikian rupa sehingga kelipatan frekuensi samplingnya memenuhi :

$$\frac{(n-1)f_s}{2} \leq f_L \text{ dan } \frac{nf_s}{2} \geq f_U \tag{3}$$

Sehingga menjamin seluruh komponen spektrum *bandpass* sinyal jatuh pada daerah *baseband*  $[0, f_s/2]$  hasil *undersampling*.

- Persamaan tersebut dapat dinyatakan :

$$f_s \leq \frac{2f_L}{(n-1)} \text{ dan } f_s \geq \frac{2f_U}{n} \tag{4}$$

Dari persamaan (4) dapat ditulis dalam bentuk persamaan:

$$\frac{2f_U}{n} \leq f_s \leq \frac{2f_L}{n-1} \tag{5}$$

Dengan batasan

$$n \in \mathbb{N} \text{ sedemikian rupa sehingga } 1 \leq n \leq \left\lfloor \frac{f_U}{f_U - f_L} \right\rfloor$$

- Untuk  $n_{ganjil}$  maka spektrum hasil sampling tidak terbalik aksisnya dan  $n_{genap}$  maka spektrum hasil sampling akan terbalik aksisnya.

## 2.2. Perkembangan Aplikasi Software Defined Radio

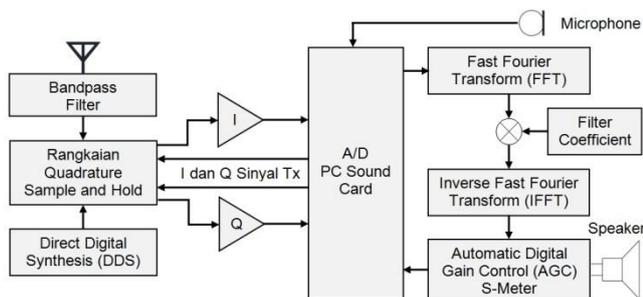
Teknologi SDR menawarkan beberapa keuntungan dalam memberikan solusi yang spesifik. Konsep SDR tidak hanya memiliki kemampuan memperbaiki *error* secara *real time* tetapi dari studi dapat diidentifikasi beberapa aplikasi yang telah dikembangkan secara pesat seperti *Dynamic Spectrum Positioning*, *Opportunity Driven Multiple Access (ODMA)*, *Spectrum Regulation and Cost Reduction* (implementasi SDR lebih murah) [6].

Filosofi SDR mulai berdampak tinggi pada bidang telekomunikasi. Beberapa penelitian yang

menggunakan teknologi SDR seperti *Driver Assistance* [7], *GPS signals Reception* [8], *HF Propagation Analysis* [9], *Interpretation of Cellular Technology Emission terutama modulasi OFDM* [10] dan *Identification of Radio Frequency Emissions* [11], telah menunjukkan keandalan dan kemudahan teknologi ini dalam implementasinya untuk berbagai aplikasi telekomunikasi.

### 2.3. Sistem Transceiver Berbasis SDR

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 3 berikut:



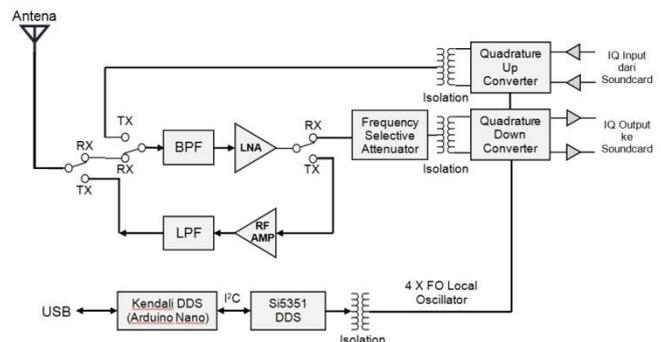
Gambar 3. Desain Transceiver SDR

Dari Gambar 3 blok diagram desain transceiver SDR dapat dijelaskan sebagai berikut:

- Pada proses penerimaan sinyal radio, frekuensi pembawa  $f_c$  yang berasal dari antena diteruskan melalui *band pass filter* (BPF) dan kemudian diumpankan 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 *in-phase*  $I(t)$ .
- 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.

### 3. DESAIN SISTEM

Desain dari *transceiver* SDR dapat dijelaskan seperti yang diperlihatkan pada blok diagram Gambar 4 berikut:



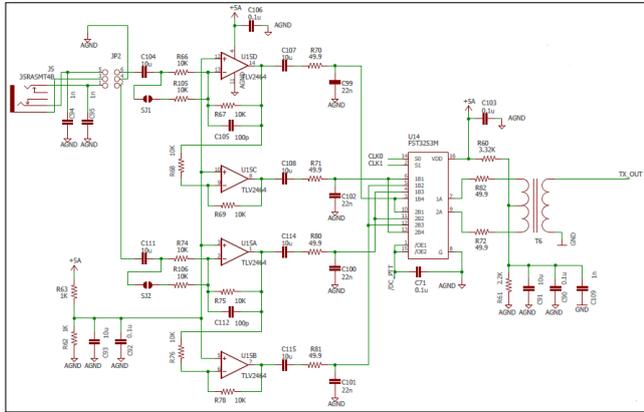
Gambar 4. Blok Diagram Sistem

Blok diagram sistem pada Gambar 4 pada saat posisi menerima (*receiver*) terdiri dari *band pass filter* (BPF) yang berfungsi untuk menekan sinyal yang tidak diinginkan dan meloloskan frekuensi tengah yang diinginkan, *low noise amplifier* (LNA) adalah penguat dengan *gain* yang menguatkan sinyal dari BPF, kemudian rangkaian *frequency selective attenuator* yang melemahkan kuat sinyal dari LNA untuk dilakukan pemilihan frekuensi yang diterima dan sebuah *quadrature down-converter* konvensional. Pada bagian pemancar SDR menggunakan sebuah *Quadrature up-converter* dan menggunakan *band pass filter* yang sama yaitu yang digunakan pada penerima, sebuah penguat daya dan sebuah *low pass filter*. Untuk menggeser frekuensi kerja pada sistem menggunakan *local oscillator* (LO) berbasis *chip* DDS Silicon Labs Si5351, dimana *chip* ini dapat menghasilkan 3 buah frekuensi keluaran, sinyal LO untuk *up* atau *down* converter, sebuah sinyal untuk kalibrasi  $I$  dan  $Q$  dan sebuah sinyal eksternal. *Chip* DDS Si5351 dikontrol oleh mikrokontroler Arduino Nano.

Desain rangkaian *low noise* SDR menggunakan tiga *ground* DC yang terpisah, yaitu *RF chassis ground, baseband* atau *audio ground* dan *digital ground*. Disain ini memungkinkan untuk menghubungkan *audio in* atau *out* dan *USB port* pada PC yang sama dengan *noise* yang rendah. Pemisahan *ground* pada desain *transceiver* SDR ini menggunakan *isolation* yaitu sebuah transformator sebagai kopling untuk menghubungkan antara rangkaian yang didedan dengan rangkaian berikutnya.

Proses dari modulasi dan demodulasi pada *transceiver* SDR menggunakan *taylor detector* yang merupakan komponen IC yang membagi sinyal yang diterima menjadi empat derajat sudut fasa yang berbeda. Pembagian ini dilakukan pada rangkaian *switching* yang terdapat di IC tersebut. Untuk dapat menekan *noise* pada proses *switching* maka dipilihlah IC untuk *Taylor detector* yang memiliki

nilai resistansi *switch* yang rendah. Resistansi *switch* merupakan tahanan penghubung antara *input* dan *output* *Taylor detector*. Dengan resistansi *switch* yang rendah maka *Taylor detector* dapat menghubungkan *input* dan *output* tanpa ada *delay* propagasi atau tanpa ada pembangkitan *ground bounce noise*. Rancangan *taylor detector* diperlihatkan pada Gambar 5 berikut ini.



Gambar 5. Rancangan *Taylor Detector*

**4. Hasil Dan Analisis**

**4.1. Perbandingan IC Taylor Detector**

Analisis dari perbandingan tiga jenis IC *taylor detector* dengan tipe FST3253, 4052 dan 4066 untuk nilai resistansi switching diperlihatkan seperti pada Tabel 1 berikut:

Tabel 1. Perbandingan Tipe Taylor Detector

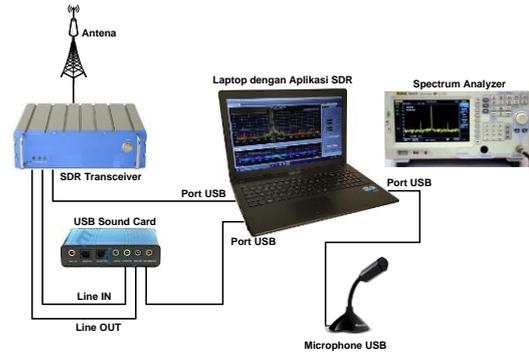
No.	Type IC	Resistansi Switching ( $\Omega$ )
1.	FST3253	4
2.	4052	25
3.	4066	15

Dari hasil perbandingan ketiga IC untuk *taylor detector* yang bersumber data sheet, didapatkan nilai resistansi *switching* yang terendah adalah 4 ohm dengan jenis chip atau IC FST3253.

**4.2. Pengujian Sistem SDR**

Pengujian sistem untuk *low noise transceiver* 7 MHz yang berbasiskan SDR membutuhkan perangkat pendukung antara lain satu unit PC atau *notebook* yang memiliki koneksi eksternal untuk *microphone*, *speaker* dan *port* USB.

Diagram dari susunan pengujian untuk sistem SDR ini diperlihatkan pada Gambar 6 berikut:



Gambar 6. Diagram Pengujian

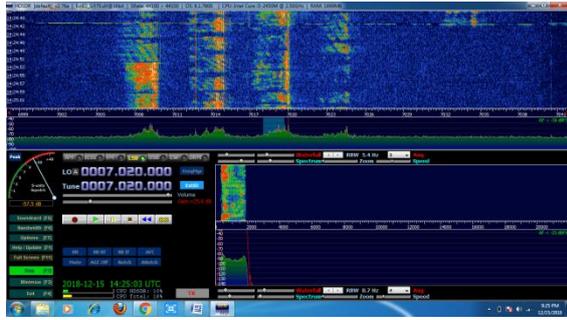
Pengujian sistem dilakukan dengan menghubungkan rangkaian *transceiver* SDR ke antena kemudian USB port dari microcontroller arduino *nano* terhubung dengan port USB di komputer. Selanjutnya *line out* dan *in* dari *transceiver* SDR terhubung dengan USB *sound card* eksternal pertama dan *microphone* terhubung dengan USB *sound card* kedua.

Setelah hubungan perkabelan dan perangkat telah terhubung secara lengkap dan benar, maka langkah selanjutnya ada mempersiapkan perangkat lunak SDR di komputer yang menggunakan aplikasi HDSDR. Pengujian sistem serta hubungan perkabelan dengan perangkat pendukung diperlihatkan pada Gambar 7 berikut ini.



Gambar 7. Pengujian Sistem SDR

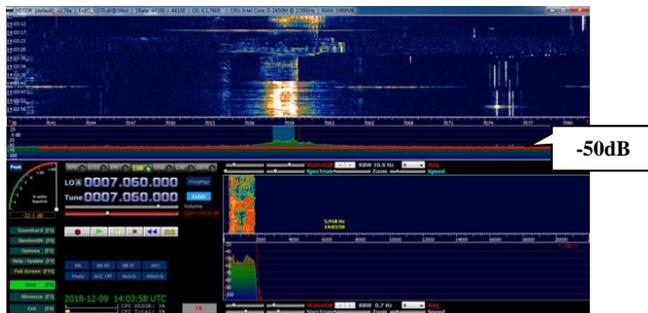
Perangkat lunak HDSDR yang telah terinstall di komputer dijalankan untuk melihat apakah perangkat keras SDR telah dikenali dan sudah terhubung dengan perangkat lunak yang digunakan, Gambar 8 memperlihatkan aplikasi HRSDR yang telah beroperasi dengan perangkat SDR.



Gambar 8. Aplikasi HSDR

### 4.3. Pengukuran Receiver SDR

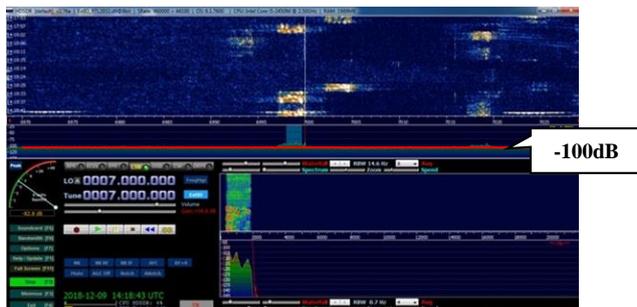
Pengujian transceiver *low noise* SDR dilakukan pada frekuensi 7MHz. langkah awal adalah melakukan pengujian untuk kualitas penerimaan (*receive*) terhadap *noise* yang muncul. Pengukuran dilakukan sebanyak dua kali, untuk membandingkan antara dua chip IC yang berbeda jenisnya. Pertama adalah mengukur chip IC 4052 yang hasilnya dapat diperlihatkan pada Gambar 9.



Gambar 9. Pengukuran *Noise* dengan IC 4052

Garis merah pada Gambar 9 memperlihatkan level *noise* pada -50 dB.

Selanjutnya melakukan pengukuran pada chip FST3253 dan *ground* terpisah yang hasilnya dapat diperlihatkan pada Gambar 10 berikut ini.



Gambar 10. Pengukuran *Noise* dengan IC FST3253

Garis merah pada Gambar 10 diatas merupakan hasil pengukuran level *noise* pada -100 dB.

### 4.4. Pengukuran Transmit SDR

Pengukuran *transmit* untuk SDR menggunakan alat ukur *spectrum analyzer*. Gambar 11 memperlihatkan pengukuran yang dilakukan untuk pemancar SDR.



Gambar 11. Pengukuran *Transmit* pada Frekuensi 7,035 MHz

Pengukuran dilakukan untuk menguji kestabilan dari perangkat penguat RF, karena perangkat pemancar yang baik akan menghasilkan penguatan di frekuensi kerja dengan nilai harmonisa yang kecil. Perangkat SDR dioperasikan untuk memancar pada frekuensi 7,035 MHz, yang kemudian diukur dengan *spectrum analyzer* dengan hasil -2dB, serta harmonisa yang rendah. Pengujian pancaran dari transceiver SDR dapat diperlihatkan pada aplikasi HSDR pada Gambar 12.



Gambar 12. Pengukuran *Transmit* HSDR pada Frekuensi 7,035 MHz

## 5. KESIMPULAN

Dalam penelitian ini telah berhasil melakukan desain *low noise transceiver* 7 MHz dengan menggunakan SDR. Berdasarkan desain yang telah dibuat serta telah dilakukan pengujian serta pengukuran maka dapat disimpulkan, bahwa pemilihan jenis *chip tayloe detector* sangat berpengaruh terhadap kualitas *noise* yang dihasilkan oleh *transceiver* SDR. Dari hasil pengukuran

didapatkan bahwa *chip* yang memiliki nilai resistansi *switching* yang terendah menghasilkan nilai *noise* yang lebih kecil. Jenis *chip tayloe detector* dengan nilai *noise* yang rendah adalah FST3253 dengan resistansi *switching* 4 ohm. Dengan resistansi *switch* yang rendah maka *Tayloe detector* dapat menghubungkan *input* dan *output* tanpa ada *delay* propagasi atau tanpa ada pembangkitan *ground bounce noise*. Selain itu implementasi dari *ground* terpisah antara rangkaian *analog* dan *digital* juga menyumbang hasil *noise* yang rendah pada batas -100 dB.

Penguatan untuk *transmit* dari *transceiver* SDR juga menunjukkan hasil dengan nilai harmonisa yang sangat kecil, hal ini dapat dilihat dari hasil pengukuran dimana nilai sinyal harmonisa dibawah -70 dB jika dibandingkan dengan nilai sinyal pembawa yang mencapai puncaknya di -2 dB.

#### DAFTAR PUSTAKA

- [1]. A. 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]. J. H. Reed, "Software Radio: A Modern Approach to Radio Engineering", New Jersey, Prentice Hall, 2002.
- [5]. F. Christensen, "A Scalable Software Defined Radio Development System", [On-line] [http://www.xilinx.com/publications/xcellonline/xcell\\_51/xc\\_es-sundance51.pdf](http://www.xilinx.com/publications/xcellonline/xcell_51/xc_es-sundance51.pdf), 2004.
- [6]. S. Gultchev, K. Moessner, and D. Thilakawardana, Evaluation of Software Defined Radio Technology, Center for Communication System Research, University of Surrey, 2005.
- [7]. 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.
- [8]. 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.
- [9]. P.B. Nagaraju, E. Koski, And T. Melodia, "A Software Defined Radio Ionospheric Chirpsounder for HF Propagation Analysis", 2009.
- [10]. 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.
- [11]. 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.