

Desain dan Implementasi TCP/IP Menggunakan *Berkeley Socket* dan *Winsock API* serta GUI untuk Sistem Akuisisi Data Radar FMCW pada Kapal Feri

Ibrahim Amyas Aksar Tarigan¹, Arif Sasongko², Ratna Indra Wijaya³, Efa Maydhona⁴

Program Studi Teknik Elektro
Jurusan Sains, Institut Teknologi Sumatera
Bandar Lampung, Indonesia

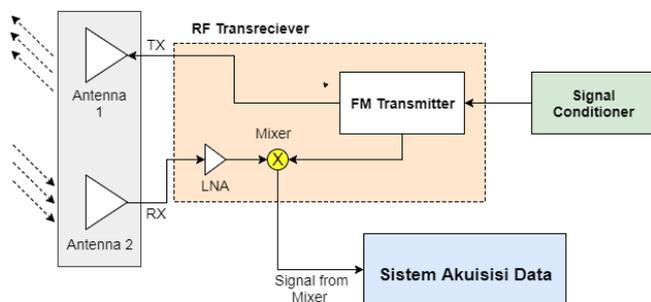
¹baimtarigan@gmail.com, ²asasongko@stei.itb.ac.id, ³ratna.indrawijaya@lipi.go.id
⁴maydhona@el.itera.ac.id

Abstrak—Pada makalah ini dijelaskan mengenai implementasi, dan pengujian dari TCP/IP menggunakan *Berkeley Socket* dan *Winsock API* serta GUI untuk sistem akuisisi data radar FMCW pada kapal feri. Masukan antarmuka berupa *file txt* yang berisi informasi proses hasil akuisisi data dari modul server (PC) yang dihubungkan dengan *client (board SoC)* menggunakan hubungan komunikasi TCP/IP *client/server*. *Berkeley Socket* dan *Winsock API* digunakan untuk mempermudah dalam pembuatan hubungan komunikasi pada TCP/IP. GUI diimplementasikan pada PC menggunakan *software MATLAB* dengan memanfaatkan fitur *Guide*. Output yang dihasilkan berupa grafik plot sinyal serta parameter amplitudo dan frekuensi dari *file* proses hasil sistem akuisisi data. Dari informasi ini maka dapat ditentukan jarak dari target yang terdeteksi.

Kata kunci—TCP/IP, *Berkeley Socket*, *Winsock API*, Radar FMCW, SoC, GUI.

I. PENGANTAR

Radar FMCW secara umum lebih sederhana dibandingkan jenis radar lainnya. Radar FMCW memiliki manufaktur yang lebih murah dari segi biaya, relatif sedikit mengalami kegagalan ketika beroperasi, dan menggunakan jauh lebih sedikit biaya dibandingkan *pulse-radar*. Namun, karena keluaran hasilnya masih berupa sinyal analog, informasi dan data yang diberikan oleh radar tidak begitu praktis sebab tidak dapat diolah lebih lanjut. Sinyal ini harus dikonversi dulu menjadi sinyal digital, kemudian diproses di PC agar dapat dimengerti oleh manusia. Diagram blok dari radar FMCW dapat dilihat pada **Gambar 1**.

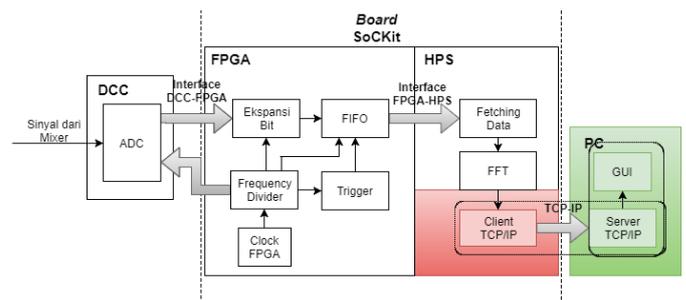


Gambar 1 – Diagram Blok Radar FMCW

Signal conditioner berfungsi untuk mengendalikan sinyal FM (*Frequency Modulated*) yang dipancarkan oleh FM transmitter. Sinyal pantulan yang diterima oleh antena selanjutnya diperkuat oleh LNA (*Low Noise Amplifier*) dan diteruskan ke dalam *mixer*. *Mixer* akan mencari selisih frekuensi

antara sinyal yang diterima dengan sinyal yang dipancarkan, kemudian memberikan sinyal output yang memiliki frekuensi tersebut. Sinyal ini yang akan diterima oleh sistem akuisisi data. Pada makalah ini, sinyal dari *mixer* akan disimulasikan oleh generator sinyal.

Proses konversi sinyal analog menjadi sinyal digital yang dilakukan membutuhkan sistem akuisisi data yang baik. Untuk itu digunakanlah *board SOC (FPGA dan HPS)* serta perangkat tambahan berupa DCC. Selain karena lebih mudah dalam implementasi desainnya, *board SoC* memiliki performa yang cukup tinggi, sehingga dapat melakukan akuisisi data dengan cepat. Untuk menghubungkan pengguna dengan *board SoC*, maka dibutuhkan antarmuka. Antarmuka yang dirancang berbentuk *Graphical User Interface (GUI)*, berfungsi untuk mengambil data dari radar FMCW yang sudah diproses pada *board SoC* kemudian menampilkannya pada layar PC. Hasilnya akan berbentuk plot grafik sinyal serta jarak dari target yang terdeteksi. Diagram blok dari sistem akuisisi data radar FMCW dapat dilihat pada **Gambar 2**.



Gambar 2 – Diagram Blok Sistem Akuisisi Data Radar FMCW.

Makalah ini menjelaskan tentang implementasi komunikasi TCP/IP yang menggunakan *Berkeley Socket* dan *Winsock API* untuk menghubungkan *board SoC* ke PC, serta GUI yang berperan sebagai antarmuka sistem akuisisi data radar FMCW pada kapal feri. Dalam hubungan komunikasi TCP/IP, *Board SoC* akan berperan sebagai *client* yang mengirimkan data hasil akuisisi kepada PC yang berperan sebagai server menggunakan kabel ethernet. Data ini akan diterima PC dalam bentuk *file txt* yang kemudian diambil oleh GUI untuk kemudian diproses lebih lanjut sehingga menghasilkan bentuk grafis plot dan jarak target.

II. TINJAUAN PUSTAKA

A. TCP/IP

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah model jaringan komputer dan kumpulan protokol komunikasi yang dapat digunakan untuk menghubungkan komputer dengan internet, atau menghubungkan komputer dengan komputer lain. TCP/IP merupakan protokol yang populer digunakan. Alasannya karena TCP/IP cukup spesifik dalam mengatur data yang akan dikirimkan seperti format, alamat pengiriman, kecepatan transmisi, pengarahan atau routing, dan alamat destinasi tempat data diterima. Selain itu TCP/IP sendiri tidak hanya bisa dipakai pada internet saja, tetapi juga mampu pada suatu jaringan khusus/privat.

Dalam penelitian ini, digunakan komunikasi hubungan TCP/IP *client/server*. Server menyediakan layanan untuk masing-masing *client*. Dalam system ini, server adalah PC pengguna yang berbasis Windows, sedangkan *client* adalah *board* SoC yang berbasis Linux Yocto. *Client* dan server memiliki program masing-masing yang perlu dieksekusi agar bisa saling terhubung.

B. Berkeley Socket dan Winsock API

Berkeley Socket dan *Winsock* merupakan antarmuka pemrograman aplikasi (*application programming interface, API*) untuk *Internet socket* dan *Unix domain socket*. *Socket* sendiri merupakan representasi abstrak di dalam *local endpoint* pada jalur jaringan komunikasi. Dalam penelitian ini, *Berkeley socket* diterapkan pada *board* SoC yang berperan sebagai *client*. *Berkeley socket* didefinisikan dalam beberapa *header files* yang digunakan dalam penelitian ini:

- `sys/socket.h`, merupakan fungsi inti *socket* dan struktur data
- `netinet/in.h`, berisi standar *IP address* dan TCP dan UDP *port number*.
- `sys/un.h`, digunakan untuk komunikasi lokal antara program yang berjalannya dalam komputer yang sama
- `netdb.h`, fungsi untuk mentranslate nama protokol dan *host* menjadi *numeric addresses*.

Winsock, singkatan dari *Windows Socket*, merupakan versi upgrade dari *Berkeley Socket* yang dibuat oleh Windows. *Winsock* diterapkan pada PC yang juga menggunakan OS Windows untuk bertindak sebagai server.

C. GUI

Graphical User Interface (GUI) atau biasa disebut antarmuka pengguna adalah suatu metode interaksi pada alat elektronik secara grafis antara pengguna dengan komputer. GUI menjadi salah satu cara untuk memfasilitasi tampilan pada layar berupa simbol gambar, grafis sinyal, atau menu yang digunakan pengguna sebagai sarana untuk memberikan masukan melalui devais input. GUI pada modul antarmuka sistem akuisisi data radar FMCW ini dibuat melalui aplikasi MATLAB menggunakan menggunakan fitur *Guide*.

D. Board SoCkit

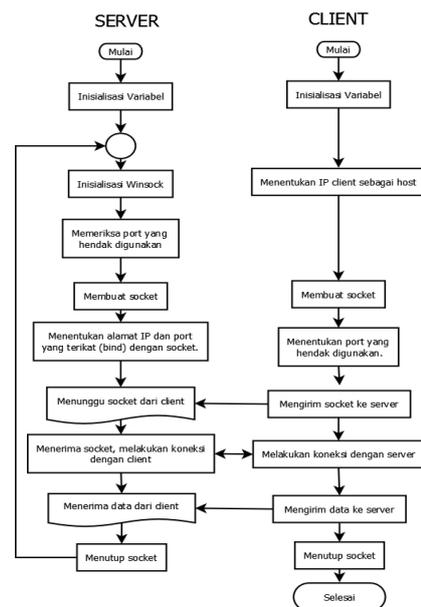
SoC merupakan sebuah *board* IC yang mengintegrasikan seluruh komponen komputer atau sistem elektronik lainnya ke

dalam sebuah *chip*. Perangkat ini sangat banyak digunakan dalam sistem *embedded* karena kemampuannya yang hanya membutuhkan daya rendah. SoC yang digunakan dalam proyek ini adalah SoCkit Cyclone V. *Board* ini dapat diprogram dengan menggunakan bahasa *hardware*. melalui *software* Quartus II 14.0 menggunakan bahasa Verilog HDL. Verilog HDL ini merupakan bahasa pemrograman yang digunakan untuk mendesain perangkat keras khususnya sistem digital. Verilog memiliki kecepatan *running* yang cepat karena dibangun dari kode-kode pemrograman sederhana dari bahasa C yang merupakan bahasa menengah setelah bahasa *assembly*.

III. DESAIN DAN IMPLEMENTASI SISTEM

A. TCP/IP

Board SoC dihubungkan ke PC menggunakan komunikasi TCP/IP *client/server* melalui kabel ethernet. PC yang digunakan sebagai server TCP/IP akan terus menerima data dari *client* hingga *client* memutuskan koneksi.



Gambar 3 – Flowchart TCP/IP komunikasi *client/server*

Data hasil FFT dari submodul FFT dalam *board* SoC akan dikirimkan ke PC melalui hubungan TCP/IP. Untuk itu, pengondisian HPS sebagai *client* perlu dilakukan. *Client* diberikan informasi IP dan *port* server agar mampu melakukan hubungan TCP/IP. Setelah terhubung, barulah *client* mampu mengirim data ke PC. Masukannya adalah data-data 16-bit dengan keluaran 512 data tipe integer, dengan total $512 \times 4 = 2048$ byte dalam bentuk *file* txt.

Gambar 3 merupakan diagram alir dari proses TCP/IP komunikasi *client/server*. Prosesnya adalah sebagai berikut:

1. Inisialisasi variabel untuk *server* dan *client*.
2. *Server* dan *client* membuat *socket*, yaitu titik akhir dari komunikasi antara *board* dan komputer.
3. *Server* menentukan alamat IP dan *port* khusus untuk melakukan koneksi ke *client*.
4. *Server* menunggu *client* hingga *client* siap untuk melakukan hubungan TCP/IP.

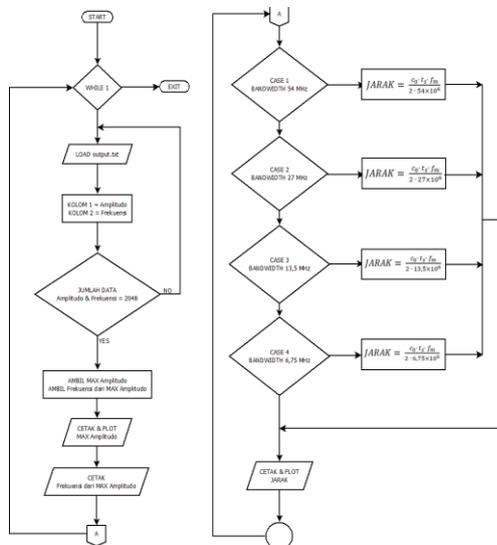
5. *Client* membuat hubungan dengan *server* melalui pengiriman *socket*, kemudian *server* menerimanya.
6. *Client* mengirimkan data ke *server*. *Server* menerima data dan menyimpannya dalam format txt.
7. *Server* dan *client* menutup *socket*.
8. *Server* mengerjakan kembali langkah 2-4, menunggu *client* diaktifkan kembali oleh pengguna.

PC yang digunakan sebagai server TCP/IP menggunakan sistem operasi Windows dari Microsoft. Microsoft telah menyediakan contoh *source code* untuk membuat server sederhana. *Source code* ini kemudian dimodifikasi untuk disesuaikan dengan besar data yang diterima, yaitu sejumlah 2048 data dengan tipe data integer. Server akan terus menerima data dari *client* hingga *client* memutuskan koneksi. Agar server bekerja secara berulang-ulang maka diberikan fungsi *superloop*. Setelah data diambil oleh submodul TCP/IP server, data diberikan label berupa informasi frekuensi, yang kemudian disimpan dalam *file* dengan format txt.

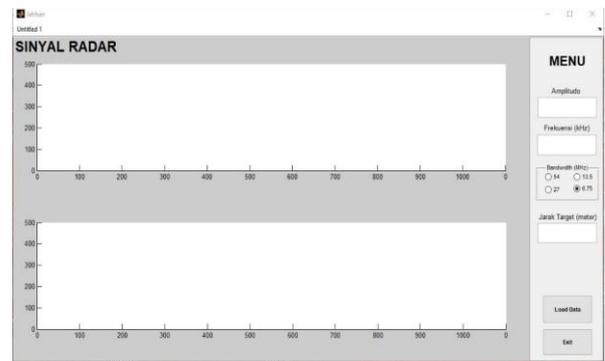
B. Graphical User Interface (GUI)

Graphical User Interface (GUI) adalah suatu metode interaksi pada alat elektronik secara grafis antara pengguna dengan komputer. GUI menjadi salah satu cara untuk memfasilitasi tampilan pada layar berupa simbol gambar, grafis sinyal, atau menu yang digunakan pengguna sebagai sarana untuk memberikan masukan melalui devais input. GUI pada modul antarmuka sistem akuisisi data radar FMCW ini dibuat melalui aplikasi MATLAB menggunakan menggunakan fitur *Guide*. Dengan GUI ini, pengguna akan lebih mudah dalam menganalisis hasil data dari radar yang telah diolah pada *board* SoCKit.

Diagram alir proses kerja GUI dalam penelitian ini diperlihatkan pada **Gambar 4**. Data diambil dalam bentuk *file* txt dari server. *File* tersebut berisi informasi nilai amplitudo dan frekuensi, dimana nilai-nilai ini kemudian akan di tampilkan pada layar GUI. Nilai frekuensi juga digunakan sebagai salah satu variabel untuk mendapatkan informasi besar jarak target. Selain itu juga pada GUI bisa diubah besar *bandwidth* untuk mengganti mode jarak yang akan dideteksi.



Gambar 4 – Flowchart proses kerja GUI



Gambar 5 – Tampilan GUI pada MATLAB

Berikut ini penjelasan fitur GUI pada **Gambar 5**:

Tabel 1 – Fitur GUI

Nama Fitur	Tampilan	Deskripsi
Load & Plot Data		<ul style="list-style-type: none"> Dengan meng"klik" tombol ini maka data dari modul server akan diambil dan dibaca oleh GUI. Selain itu juga tombol ini berfungsi untuk melakukan plot sinyal serta plot jarak target pada Jendela Plot Sinyal dan Plot Jarak Target.
Parameter Bandwidth		<ul style="list-style-type: none"> Tempat untuk memilih bandwidth sinyal yang akan digunakan
Monitoring Amplitudo		<ul style="list-style-type: none"> Pada kotak ini akan ditambahkan besar amplitudo dari sinyal yang diplot
Monitoring Frekuensi		<ul style="list-style-type: none"> Pada kotak ini akan ditampilkan besar frekuensi sinyal yang diplot
Monitoring Jarak Target		<ul style="list-style-type: none"> Pada kotak ini akan ditampilkan jarak target benda yang didapat dari besar frekuensi sinyal yang diplot
Jendela Plot Sinyal		<ul style="list-style-type: none"> Jendela ini menampilkan hasil plot sinyal dari data yang diambil dari modul server.
Jendela Plot Jarak		<ul style="list-style-type: none"> Jendela ini menampilkan hasil plot jarak target dari data yang diambil dari modul server.
Exit Program		<ul style="list-style-type: none"> Dengan meng"klik" tombol ini maka program akan berhenti dan otomatis menutup GUI.

GUI ini didesain agar dapat melakukan monitoring hasil plot sinyal dari data yang diambil dari modul server. Pengguna dapat mengganti nilai *bandwidth* yang digunakan untuk mengganti *mode* ukuran jarak target. Dari GUI ini juga dapat dilakukan monitoring terhadap nilai jarak target yang informasi nilainya tidak didapat dari data yang diambil dari modul server. GUI ini sendiri bekerja secara berulang mengikuti informasi data yang berubah setiap waktunya (terbaru secara otomatis).

IV. PENGUJIAN DAN HASIL

A. Pengujian TCP/IP Komunikasi Client/Server

Dalam pengujian komunikasi *client/server*, input yang digunakan adalah 2048 data tipe integer yang dikirim dari *board* sebagai *client* dengan frekuensi 50 kHz. Output yang diamati adalah jumlah byte yang berhasil dikirim, dan *file* txt yang disimpan oleh modul *server* berisi label frekuensi dan 2048 data input. Pengujian dilakukan dengan mengamati jumlah byte data yang diterima oleh server harus sama dengan jumlah *byte* yang dikirim oleh *client*, yaitu $2048 \times 4 = 9192$ byte. Selain itu, *file* txt yang disimpan oleh modul *server* juga diamati. Setelah dieksekusi, modul *server* bekerja tanpa henti

```
Bytes sent : 2048
```

Gambar 6 – Informasi Pengiriman Data Client TCP/IP

```
Loop selesai
Listening...
Bytes received: 2048
Connection closing...
Loop selesai
Listening...
Bytes received: 2048
```

Gambar 7 – Informasi Pengiriman Data Server

Dari **Gambar 6**, terlihat bahwa data sudah berhasil dikirim dari *client*. Pengiriman data dilakukan untuk setiap 2048 byte data. Kemudian dari **Gambar 7**, terlihat bahwa data sudah berhasil diterima oleh *server*. Penerimaan data dilakukan untuk setiap 2048 data, dan dilakukan sebanyak empat kali. Hal ini terlihat pada jumlah byte yang diterima untuk setiap *loop*, bahwa ada empat kali penerimaan 2048 byte untuk *loop* server (di antara pernyataan “Loop selesai”). Hal ini menunjukkan bahwa modul server sudah berhasil menerima data sebanyak 2048 byte sebanyak 4 kali untuk setiap *loop*, dan dilakukan tanpa henti.

Tabel 2 – 128 Data Output yang Disimpan dalam File txt Beserta Label

n	Label	Data	n	Label	Data	n	Label	Data	n	Label	Data
1	0.509	65.12207	33	16.785	2.63261	65	35.061	53.91657	97	49.337	1337.553
2	1.017	6.471702	34	17.293	13.71608	66	33.569	46.0459	98	49.845	767.4285
3	1.526	22.17031	35	17.802	20.56409	67	34.078	28.15626	99	50.354	1094.424
4	2.035	5.594893	36	18.311	25.36736	68	34.587	27.84552	100	50.863	1297.551
5	2.543	16.06026	37	18.819	20.63969	69	35.095	77.13072	101	51.371	115.4725
6	3.052	4.885213	38	19.328	14.52818	70	35.604	26.05467	102	51.88	879.0861
7	3.56	7.419042	39	19.836	26.03378	71	36.112	41.07801	103	52.389	149.0606
8	4.069	29.99116	40	20.345	32.59969	72	36.621	25.88762	104	52.897	133.3724
9	4.578	14.8401	41	20.854	17.0713	73	37.13	27.81894	105	53.406	204.5858
10	5.086	14.9582	42	21.362	24.66398	74	37.638	76.43175	106	53.914	256.5612
11	5.595	12.1826	43	21.871	7.456307	75	38.147	30.94412	107	54.423	205.445
12	6.104	3.093308	44	22.38	9.063447	76	38.656	29.24547	108	54.932	78.97475
13	6.612	29.69985	45	22.888	10.62524	77	39.164	56.805	109	55.44	30.3808
14	7.121	1.685248	46	23.397	42.5804	78	39.673	59.65435	110	55.949	75.92371
15	7.629	9.486586	47	23.905	5.821801	79	40.181	65.23415	111	56.458	162.7009
16	8.138	4.237361	48	24.414	5.853377	80	40.69	75.45919	112	56.966	252.3564
17	8.647	14.56191	49	24.923	21.45699	81	41.199	103.3214	113	57.475	124.9827
18	9.155	21.90226	50	25.431	5.460666	82	41.707	150.2648	114	57.983	69.16344
19	9.664	11.46848	51	25.94	56.48133	83	42.216	137.8311	115	58.492	57.77562
20	10.173	16.37446	52	26.449	16.99194	84	42.725	50.94288	116	59.001	93.00188
21	10.681	7.031756	53	26.957	16.46443	85	43.233	41.5662	117	59.509	65.82044
22	11.19	18.35042	54	27.466	30.71026	86	43.742	23.35478	118	60.018	25.36494
23	11.698	26.32926	55	27.974	20.20326	87	44.25	100.0111	119	60.527	51.0287
24	12.207	17.20272	56	28.483	23.68152	88	44.759	131.4981	120	61.035	112.9608
25	12.716	12.10892	57	28.992	14.4602	89	45.268	167.1525	121	61.544	58.00867
26	13.224	10.13482	58	29.5	18.50973	90	45.776	115.4247	122	62.052	40.14799
27	13.733	12.11564	59	30.009	50.22687	91	46.285	13.21171	123	62.561	74.95953
28	14.242	29.49534	60	30.518	17.2198	92	46.794	337.5958	124	63.07	41.54705
29	14.75	17.05218	61	31.026	44.28664	93	47.302	175.6031	125	63.578	70.45213
30	15.259	15.85937	62	31.535	35.38541	94	47.811	195.2354	126	64.087	71.52312
31	15.767	16.36969	63	32.043	32.09884	95	48.319	131.7532	127	64.596	37.32788
32	16.276	29.37783	64	32.552	33.24944	96	48.828	367.2379	128	65.104	60.30167

Pemberian label frekuensi diberikan kepada masing-masing data agar data dapat ditampilkan pada MATLAB. Selain itu, label frekuensi akan memudahkan MATLAB untuk mencari frekuensi dari data ke-*n* yang memiliki nilai terbesar. Perhitungan label frekuensi dilakukan dengan rumus berikut:

$$\text{Label Frekuensi} = \frac{n}{4096} \cdot \text{Frekuensi Sampling}$$

Frekuensi sampling yang digunakan adalah 2.083 MHz. Label frekuensi disimpan oleh *modul* server dalam kolom pertama *file* txt, sedangkan data disimpan dalam kolom kedua. Label frekuensi disimpan dengan satuan kHz.

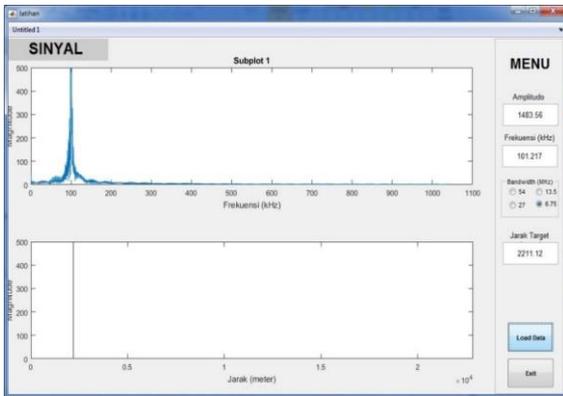
Tabel 2 berisi data ke-1 sampai data ke-128 dari 2048 data pada *file* txt. Dari **Tabel 2**, terlihat bahwa label frekuensi yang memiliki nilai data terbesar adalah label frekuensi 49.337. Proses ini yang nantinya akan digunakan oleh modul GUI untuk mencari nilai frekuensi data terbesar.

Input yang digunakan memiliki frekuensi sebesar 50 kHz, dan label frekuensi data terbesar pada **Tabel 2** adalah 49.337 kHz. Hal ini menunjukkan bahwa modul *server* sudah berhasil memberikan label frekuensi berdasarkan nilai *n* dan frekuensi

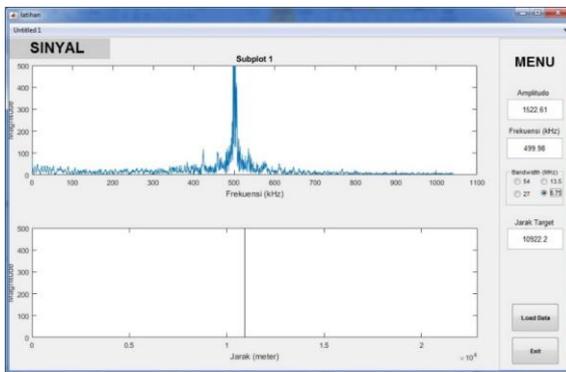
sampling, dengan galat sebesar $\frac{50-49.337}{50} \cdot 100\% = \frac{0.663}{50} \cdot 100\% = 1.326\%$

B. Pengujian Grafik Plot Data

Untuk melakukan pengujian grafik plot sinyal, GUI harus mengambil data *file* txt dari modul server terlebih dahulu. *File* ini berisi data besar amplitudo dan frekuensinya yang berjumlah 2048 data. Karena data dalam *file* ini terus berubah sepanjang waktu, maka proses plot data pada antarmuka GUI dilakukan secara berulang. Dalam melakukan pengujian, diambil 2 nilai frekuensi yang berbeda, 100 kHz dan 500 kHz. Hasilnya seperti yang terlihat pada **Gambar 6-7** berikut:



Gambar 8 – Plot Data pada Frekuensi 100 kHz

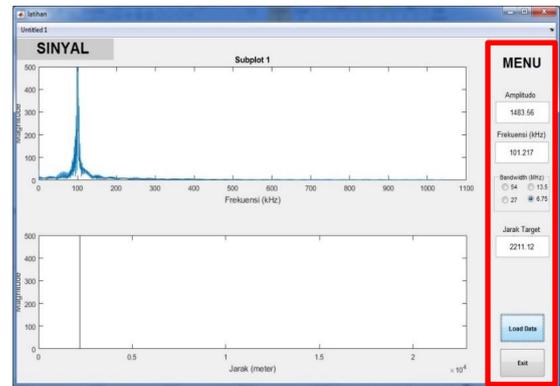


Gambar 9 – Plot Data pada Frekuensi 500 kHz

Dari **Gambar 8-9** bisa dilihat bahwa titik puncak amplitudo bergeser ke arah kanan sesuai dengan bertambahnya nilai frekuensi. Perhatikan juga bahwa dari gambar-gambar tersebut, nilai frekuensi pada GUI tidak sama persis dengan frekuensi yang diberikan. Penjelasan mengenai hal tersebut akan dijelaskan lebih dalam pada pengujian hasil perhitungan frekuensi. Hasil ini membuktikan bahwa grafik plot data telah berhasil bekerja sesuai dengan yang diharapkan.

C. Pengujian Parameter Amplitudo dan Frekuensi pada GUI

Bagian kanan antarmuka (GUI) merupakan posisi *menu* yang berisi jendela monitoring amplitudo, frekuensi, parameter *bandwidth*, dan jarak tempuh. Dalam pengujian hasil perhitungan ini, yang diperhatikan adalah besaran nilai amplitudo, frekuensi, serta jarak target yang ditampilkan pada antarmuka (GUI).



Gambar 10 – Posisi Menu pada GUI

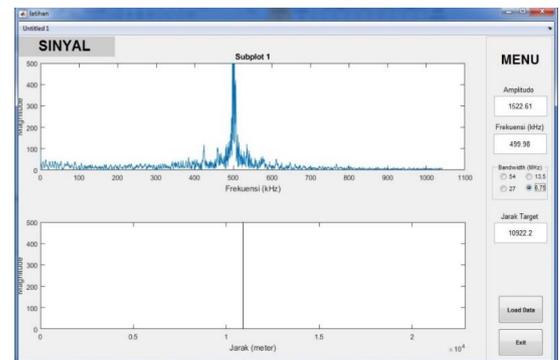
Nilai amplitudo dan frekuensi yang ditampilkan pada GUI adalah data yang didapat pada *output.txt*. Isinya terdiri dari dua buah kolom, kolom pertama merupakan nilai frekuensi dan kolom kedua merupakan amplitudo dari frekuensi tersebut. Untuk menampilkan nilai amplitudo pada GUI, maka diambil nilai terbesar dari 2048 data yang berada pada kolom kedua *output.txt*. Nilai kolom pertama yang tepat berada di samping nilai amplitudo tadi, akan ditampilkan sebagai frekuensinya pada GUI.

Pengujian dilakukan dengan mencocokkan nilai frekuensi yang ditampilkan kemudian membandingkannya dengan nilai frekuensi yang diberikan melalui generator sinyal. **Gambar 10** diambil ketika data *output.txt* diplot saat frekuensi yang diberikan berada pada 100 kHz. Hasil yang ditampilkan pada GUI menunjukkan nilai frekuensi 101,217 kHz seperti pada **Gambar 11**.



Gambar 11 – Amplitudo dan Frekuensi pada GUI saat Frekuensi 100 kHz

Terlihat bahwa nilai yang ditampilkan memiliki perbedaan dengan frekuensi yang diberikan pada generator sinyal. Kemudian, perhatikan hasil plot grafik ketika frekuensi 500 kHz seperti pada **Gambar 12** berikut:



Gambar 12 – Plot Grafik Frekuensi 500 kHz

Lihat bahwa besar frekuensi yang ditampilkan adalah 499.98 kHz. Besar frekuensi yang ditampilkan pada GUI tidak tepat persis sama seperti dengan frekuensi yang diberikan dari generator sinyal. Hal ini diakibatkan karena proses *sampling* sinyal itu sendiri dilakukan pada modul server. Frekuensi *sampling* yang dilakukan adalah 4096 data, maka resolusi dari data yang diambil adalah $2083 \text{ kHz}/4096 = 0,509 \text{ kHz}$. Resolusi ini digunakan pada modul server sebagai label untuk data *input* sehingga dapat langsung dapat ditampilkan pada GUI (label = kolom satu, data = kolom dua pada *output.txt*). Dengan resolusi ini maka, frekuensi yang ditampilkan tidak mungkin bernilai persis sama dengan yang diberikan generator sinyal, dimana nilainya akan terus berada pada kelipatan 0,509 kHz.

Maka, dari dua hasil pengujian tersebut bisa disimpulkan bahwa hasil pengujian ini benar, karena hasil frekuensi yang ditampilkan pada GUI hanya memiliki galat yang kecil sehingga dapat ditoleransi.

D. Pengujian Hasil Perhitungan Jarak Target pada GUI

Untuk mendapatkan jarak target harus dilakukan perhitungan yang berbeda. Besar jarak target benda didapatkan melalui perhitungan sebagai berikut:

$$R = \frac{c_0 \cdot t_s \cdot f_m}{2 \cdot BW}$$

- R merupakan jarak dari target yang terdeteksi radar (meter).
- c_0 merupakan kecepatan rambat cahaya ($3 \times 10^8 \text{ m/s}$).
- t_s merupakan *sweep time*, yakni waktu yang diperlukan sinyal dari mulai dipancarkan hingga kembali lagi setelah dipantulkan ($983,04 \times 10^{-6} \text{ s}$).
- f_m merupakan frekuensi yang terukur. Dalam hal ini frekuensi yang diberikan dari generator sinyal. (Hz)
- BW merupakan *bandwidth* yang dipilih sendiri pengguna. (MHz)

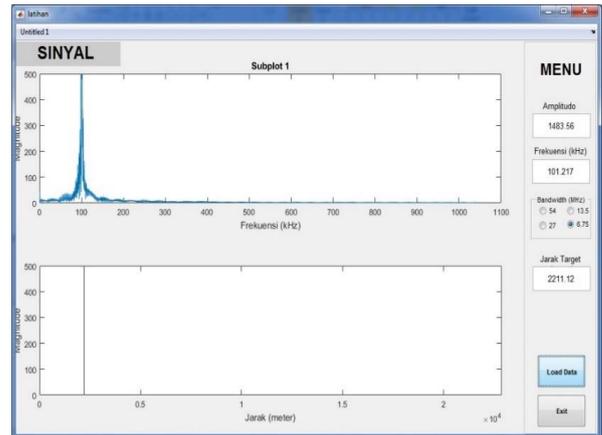
Dari persamaan ini, maka hanya dibutuhkan nilai input frekuensi untuk bisa menentukan jarak target pada GUI. Selanjutnya, tinggal ditentukan pada nilai *bandwidth* yang mana untuk mengetahui nilai jarak targetnya. GUI ini memiliki 4 pilihan nilai *bandwidth*, dimana tiap nilainya akan menentukan mode jarak target yang ingin dideteksi radar. Misalnya, *bandwidth* 54 MHz digunakan untuk mendeteksi target yang sangat dekat, sedangkan *bandwidth* 6,75 MHz digunakan untuk mendeteksi target yang memiliki jarak sangat jauh.



Gambar 13 – Parameter Bandwidth dan Monitoring Jarak Target pada Menu GUI

Pengujian jarak target akan dilakukan dengan menggunakan dua nilai frekuensi yang berbeda, 100 kHz dan 500 kHz. Untuk setiap nilai frekuensi, akan dipilih 4 nilai *bandwidth* secara

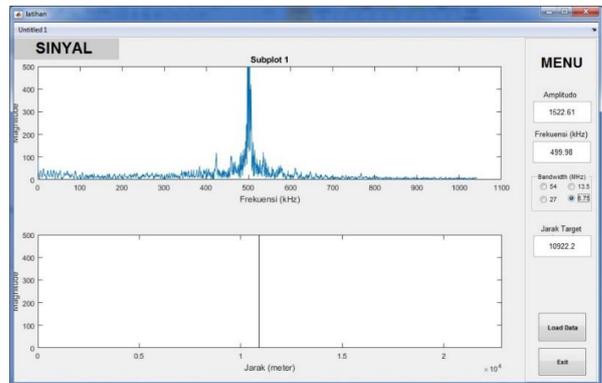
bergantian. Hasil yang ditampilkan pada GUI kemudian akan dibandingkan dengan hasil perhitungan teori yang dilakukan secara manual. Berikut adalah gambar hasil pengujian pada frekuensi 100 dan 500 kHz.



Gambar 14 – Grafik Plot pada Frekuensi 100 kHz



Gambar 15 – Pengujian pada Frekuensi 100 kHz pada Bandwidth 6,75, 13,5, 27, dan 54 MHz



Gambar 16 – Grafik Plot pada Frekuensi 500 kHz



Gambar 17 – Pengujian pada Frekuensi 500 kHz pada Bandwidth 6,75, 13,5, 27, dan 54 MHz

Selanjutnya pengujian dilakukan dengan menggunakan persamaan teori. Pertama-tama dilakukan perhitungan untuk setiap nilai *bandwidth* pada frekuensi 100 kHz, dilanjutkan kemudian dengan frekuensi 500 kHz. Berikut hasil perhitungannya:

- Frekuensi 100 kHz, *bandwidth* 6,75 MHz

$$R = \frac{c_0 \cdot t_s \cdot f_m}{2 \cdot BW} = \frac{3 \times 10^8 \cdot 983,04 \times 10^{-6} \cdot 10^5}{2 \cdot 6,75 \times 10^6} = 2184,533 \text{ m}$$
- Frekuensi 100 kHz, *bandwidth* 13,5 MHz

$$R = \frac{c_0 \cdot t_s \cdot f_m}{2 \cdot BW} = \frac{3 \times 10^8 \cdot 983,04 \times 10^{-6} \cdot 10^5}{2 \cdot 13,5 \times 10^6} = 1092,267 \text{ m}$$
- Frekuensi 100 kHz, *bandwidth* 27 MHz

$$R = \frac{c_0 \cdot t_s \cdot f_m}{2 \cdot BW} = \frac{3 \times 10^8 \cdot 983,04 \times 10^{-6} \cdot 10^5}{2 \cdot 27 \times 10^6} = 546,133 \text{ m}$$
- Frekuensi 100 kHz, *bandwidth* 54 MHz

$$R = \frac{c_0 \cdot t_s \cdot f_m}{2 \cdot BW} = \frac{3 \times 10^8 \cdot 983,04 \times 10^{-6} \cdot 10^5}{2 \cdot 54 \times 10^6} = 273,067 \text{ m}$$
- Frekuensi 500 kHz, *bandwidth* 6,75 MHz

$$R = \frac{c_0 \cdot t_s \cdot f_m}{2 \cdot BW} = \frac{3 \times 10^8 \cdot 983,04 \times 10^{-6} \cdot 5 \times 10^5}{2 \cdot 6,75 \times 10^6} = 10922,667 \text{ m}$$
- Frekuensi 500 kHz, *bandwidth* 13,5 MHz

$$R = \frac{c_0 \cdot t_s \cdot f_m}{2 \cdot BW} = \frac{3 \times 10^8 \cdot 983,04 \times 10^{-6} \cdot 5 \times 10^5}{2 \cdot 13,5 \times 10^6} = 5461,333 \text{ m}$$
- Frekuensi 500 kHz, *bandwidth* 27 MHz

$$R = \frac{c_0 \cdot t_s \cdot f_m}{2 \cdot BW} = \frac{3 \times 10^8 \cdot 983,04 \times 10^{-6} \cdot 5 \times 10^5}{2 \cdot 27 \times 10^6} = 2730,667 \text{ m}$$
- Frekuensi 500 kHz, *bandwidth* 54 Hz

$$R = \frac{c_0 \cdot t_s \cdot f_m}{2 \cdot BW} = \frac{3 \times 10^8 \cdot 983,04 \times 10^{-6} \cdot 5 \times 10^5}{2 \cdot 54 \times 10^6} = 1365,333 \text{ m}$$

Tabel 3 – Tabel Perbandingan Hasil Jarak Target pada GUI dan Perhitungan Teori

Frekuensi (kHz)	Bandwidth (MHz)	Hasil GUI (m)	Hasil Teori (m)
100	6,75	2211,12	2184,533
	13,5	1105,56	1092,267
	27	552,78	546,133
	54	276,39	273,067
500	6,75	10922,2	10922,667
	13,5	5461,11	5451,333
	27	2730,56	2730,667
	54	1365,28	1365,333

Dari **Tabel 3** bisa dilihat adanya perbedaan nilai hasil pada GUI dan dari hasil teori perhitungan. Perbedaan ini nilainya cenderung kecil sekali, terutama pada saat pengujian frekuensi 500 kHz yang hasilnya hanya berbeda sekitar 0,1 meter. Adanya perbedaan ini terjadi karena frekuensi yang diterima oleh GUI merupakan frekuensi yang telah diberi label oleh modul server seperti yang telah dijelaskan pada pengujian perhitungan amplitudo dan frekuensi. Nilai frekuensi ini tidak

akan pernah sama persis dengan nilai frekuensi pada generator sinyal, namun memiliki nilai yang sangat dekat. Sebagai contoh, pada **Gambar 14** dan **Gambar 17** frekuensi yang ditampilkan pada GUI adalah 101,217 kHz dan 499,98 kHz sementara frekuensi yang diberikan adalah 100 kHz dan 500 kHz.

Maka, dari hasil pengujian ini dapat disimpulkan bahwa hasil perhitungan jarak target untuk setiap *bandwidth* telah berhasil dilakukan.

V. KESIMPULAN

Dalam sistem ini, *board* SoC berperan sebagai *client* yang mengirimkan data kepada server (PC) secara terus menurut dalam hubungan komunikasi TCP/IP yang dibangun menggunakan *Berkeley Socket* dan *Winsock* API. GUI yang diimplementasi pada PC, dibangun menggunakan MATLAB dengan bantuan *Guide*, berfungsi untuk membuat grafik plot dari data yang diambil dari server, kemudian menampilkannya pada layar GUI beserta hasil perhitungannya jarak target.

Sejauh ini implementasi dan pengujian hubungan TCP/IP dan GUI yang dilakukan telah berhasil, dengan berhasilnya dikirim data dari *client* menuju server secara berurut tanpa kehilangan data. GUI yang dibuat juga mampu menampilkan hasil plot grafik sinyal serta jarak target yang terdeteksi di layar, serta dapat di-*update* secara otomatis mengikuti informasi data yang berubah-ubah.

REFERENSI

- [1] Austerlitz, Howard. *Data Acquisition Techniques Using PCs*. 2nd Edition. Academic Press: 2002
- [2] Fatourou, Panagiota, "Presentation: Introduction to Sockets Programming in C using TCP/IP", May 2012, University of Crete.
- [3] Hwang, Yin-Tsung, Yi-Chih Chen, Chen-Ru Hong. *Design and FPGA Implementation of a FMCW Radar Baseband Processor*. IEEE: 2012 (diakses tanggal 1 Agustus 2016, 10.46)
- [4] Hyun, Eugin, Sang-Dong Kim, Yeong-Hwan Ju. *FPGA Based Signal Processing Module Design and Implementation for FMCW Vehicle Radar System*. IEEE: 2011
- [5] Suto, Kyohei, J. T. Sri Sumantyo, M. Y. Chua, W. G. Cheaw. *Development of SAR Base-Band Signal Processor Using FPGA and Onboard PC*. IEEE: 2014 (diakses tanggal 1 Agustus 2016, pukul 10.30)