

DETEKSI TARGET PADA RADAR MENGGUNAKAN METODE ORDER STATISTIK

TARGET DETECTION IN RADAR USING STATISTICAL ORDER METHOD

Muhammad Hegi Rinaldi¹, Fiky Y. Suratman², Dharu Arseno³

^{1,2,3} Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

¹hegirinaldi@students.telkomuniversity.ac.id, ²fysuratman@telkomuniversity.ac.id,

³darseno@telkomuniversity.ac.id

Abstrak

Radar mempunyai tiga subsistem utama yaitu *transmitter*, *receiver*, dan *signal processing*. Pada subsistem *processing signal* terdapat proses deteksi. Proses deteksi pada radar sangat penting, karena dapat menentukan ada atau tidaknya target. Namun dalam proses deteksi sering ditemukan deteksi *error*. Salah satunya adalah *false alarm*, *false alarm* merupakan kondisi dimana saat keadaan tidak ada target tetapi radar mendeteksi adanya target. Metode OS-CFAR digunakan untuk mengatasi *false alarm* pada radar. Metode OS-CFAR merupakan pengembangan dari CFAR yang didesain untuk kondisi lingkungan *noise* heterogen dan dapat menekan *masking* sasaran, dengan begitu OS-CFAR dapat menekan tepi *clutter* dari *false alarm*. Dari hasil pengujian pada kondisi lingkungan *noise* heterogen dengan parameter jumlah sampel = 2000, sel referensi = 16, sel guard = 2 dan nilai Probabilitas *false alarm* = 10^{-1} sampai 10^{-4} . Didapatkan hasil Pfa aktual sebesar 0,1458 pada Pfa nominal 10^{-1} , 0,0329 pada Pfa nominal 10^{-2} , 0,0062 pada Pfa nominal 10^{-3} dan 0,0026 pada Pfa nominal 10^{-4} . Pengujian ini juga menggunakan metode algoritma *Neyman-Pearson* dan *Cell Averaging CFAR* sebagai pembandingan dengan Order Statistik CFAR. Hasil perbandingan dengan algoritma *Neyman-Pearson* dan CA-CFAR adalah pada algoritma OS-CFAR dapat menekan target *masking* sehingga menghasilkan nilai *false alarm* yang konstan dan membuat parameter *threshold* lebih baik pada kondisi *noise heterogen*.

Kata Kunci : Radar, deteksi, *threshold*, CFAR, OS-CFAR, CA-CFAR, *Neyman-Pearson*, *false alarm*

Abstract

Radar have three main subsystems, it were transmitter, receiver, and processing signal. In the processing signal there is a detection process. The detection process in radar is very important, because it can determine exists or does not exist targets. But in the detection process error detection is frequently found. One of them is a false alarm, A false alarm is a condition where there is no target but the radar detects a target. To resolve false alarm on the radar, can be solved by using the detection method Order Statistics Constant False Alarm Rate (CFAR). The OS-CFAR method is the perfection of CFAR that is designed for heterogeneous noise conditions and can suppress target masking, so OS-CFAR can suppress the clutter of a false alarm. From the test result on heterogeneous noise conditions with the parameter number of samples = 2000, reference cells = 16, guard cells = 2 and Probability false alarm = 10^{-1} to 10^{-4} . Obtained the actual Pfa result of 0,1458 at nominal Pfa 10^{-1} , 0,0329 at nominal Pfa 10^{-2} , 0,0062 at nominal Pfa 10^{-3} and 0,0026 at nominal Pfa 10^{-4} . In this test also using *Neyman-Pearson* and *Cell Averaging CFAR* algorithm as a comparison for Order Statistic CFAR. The result of the comparison with *Neyman-Pearson* and CA-CFAR algorithm is OS-CFAR algorithm can suppress target masking that produce a constant false alarm value and make a better threshold parameter in heterogen condition.

Keywords : Radar, detection, *threshold*, CFAR, OS-CFAR, CA-CFAR, *Neyman-Pearson*, *false alarm*

1. Pendahuluan

Indonesia merupakan salah satu negara dengan kepulauan terbesar didunia, karena itu keamanan menjadi aspek penting bagi kedaulatan negara Indonesia. Dengan ribuan pulau yang terbentang dari sabang sampai merauke, Indonesia rawan terhadap pesawat dan kapal asing yang masuk ke wilayah Indonesia. Untuk itu diperlukan *Radio Detection and Ranging* (RADAR). Radar di Indonesia banyak digunakan untuk militer dan keperluan sipil seperti pengawas bandar udara. Militer di Indonesia memerlukan radar untuk dapat mendeteksi dan melacak keberadaan musuh yang masuk wilayah

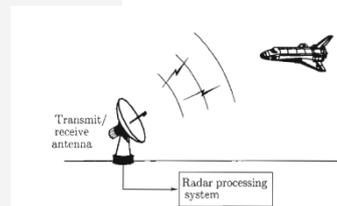
Indonesia tanpa izin sehingga dapat menjaga kedaulatan negara Indonesia. Pengawas bandar udara menggunakan radar untuk dapat mendeteksi dan mengetahui posisi pesawat supaya tidak terjadi *crash* antar pesawat. Dalam penggunaannya radar memancarkan gelombang elektromagnetik frekuensi radio ke wilayah tertentu kemudian gelombang elektromagnetik diterima dan dideteksi ketika ada pantulan terhadap objek di wilayah tersebut [1].

Fungsi utama radar adalah *detection*, *tracking* dan *imaging* [2]. Sinyal yang dipantulkan akan dianggap sebagai target oleh radar, tetapi terkadang radar menerima sinyal pantul yang sebenarnya disebabkan oleh *noise* yang ada di wilayah tersebut sebagai target disebut *false alarm*. Bila keadaan radar tidak mendeteksi adanya target tetapi sebenarnya ada target disebut *miss detection*. Terjadinya *false alarm* pada radar bergantung pada sinyal atau *noise* yang masuk tidak melebihi nilai dari parameter *threshold*. *Threshold* adalah nilai ambang batas yang dapat menyatakan bahwa ada tidaknya target dalam deteksi pada radar. Berdasarkan masalah diatas, deteksi target menjadi salah satu aspek yang penting. Metode deteksi pada radar mempunyai bermacam-macam algoritma, salah satunya menggunakan order statistik. Pada order statistik nilai parameter *threshold* dari target akan berubah-ubah sesuai dengan keadaan lingkungan deteksinya. Dengan bergitu nilai *false alarm* akan konstan karena mengikuti nilai dari parameter *threshold*.

2. Teori Dasar

2.1 RADAR

Radar merupakan akronim dari *Radio Detection and Ranging*. Fungsi utama dari radar adalah deteksi, *tracking* dan *imaging* [2]. Sejarah radar bermula ketika ditahun 1886 seorang ahli fisika dari Jerman bernama Heinrich Rudolf Hertz mendemonstrasikan refleksi dari gelombang radio dan ditahun 1990 *Tesla* mendeskripsikan sebuah konsep dari deteksi elektromagnetik dan pengukuran kecepatan. Pada tahun 1903 dan 1904 seorang insinyur dari Jerman bernama Hülsmeyer membuat percobaan dengan gelombang elektromagnetik dalam mendeteksi kapal [2]. Kerja suatu sistem radar adalah memancarkan gelombang elektromagnetik frekuensi radio ke suatu wilayah dan gelombang elektromagnetik diterima kemudian dideteksi ketika ada pantulan terhadap objek di wilayah tersebut. Subsistem utama pada radar adalah *transmitter*, *receiver*, antena dan *processing* sinyal [3].



Gambar 2.1 Cara kerja dasar radar [3].

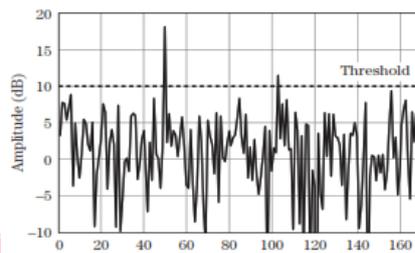
2.2 Deteksi Target

Dalam mendeteksi target radar mengirimkan gelombang elektromagnetik yang kemudian akan dipantulkan kembali sebagian menuju antena penerima radar apabila ada target. Pada kondisi tersebut, antena penerima radar tidak hanya menerima sinyal pantul tetapi ditambah dengan *noise* dan *interference* yang pasti ada pada setiap perangkat *receiver* [5]. Perangkat *receiver* dalam menerima gelombang pantul dari target terdapat dua hipotesis yaitu hasil deteksi hanya berupa *interference* (H_0) dan hasil deteksi berupa *interference* ditambah gelombang pantul dari target (H_1). Jika *sample* yang diuji dilambangkan dengan (y) maka:

$P_y(y|H_0)$ merupakan *probability density function* saat tidak ada target

$P_y(y|H_1)$ merupakan *probability density function* saat ada target

Probability density function adalah nilai probabilitas dari setiap kejadian [2]. Pada saat sinyal dideteksi, performansi detektor yang akan dievaluasi dengan model kanal baik yang tidak berfluktuasi (*swerling*) dan model kanal berfluktuasi dengan mengikuti distribusi eksponensial 1 dan 2 serta distribusi *chi-square* 3 dan 4 [1].



Gambar 2. 1 Deteksi target pada radar [3]

Pada Gambar diatas merupakan contoh deteksi target pada radar dengan menggunakan *threshold* sebagai nilai ambang batas dalam proses deteksi. Pada Gambar 2.3 terlihat bahwa *sample number* ke-50 dan ke-101, nilainya melebihi dari *threshold* maka hasil deteksinya adalah ada target (H1). Pada blok *Transmitter*, memancarkan satu atau lebih gelombang elektromagnetik ke arah target dan selanjutnya *receiver* menerima pantulan dari target untuk mendeteksi target dengan menggunakan nilai *threshold* (ambang batas). Data berupa *pulse* yang masuk ke *receiver* akan diintegrasikan untuk meningkatkan nilai *Signal to Noise Ratio* (SNR) [1]. Data yang masuk menentukan nilai *threshold* didefinisikan untuk membuat keputusan deteksi adanya target atau tidak [6]. Detektor pada *rectifier* yang biasa digunakan adalah linear dan *square-law* detektor.

2.2.1 Neyman-pearson detection

Neyman-pearson *detection* digunakan untuk memaksimalkan probabilitas dari deteksi (PD) dari kendala probabilitas *false alarm* supaya konstan [2]. Nilai dari probabilitas *false alarm* umumnya sangat rendah, nilainya berkisar 10^{-4} sampai 10^{-8} namun masih dapat menyebabkan *false alarm* setiap beberapa detik dan menit [2]. Detektor pada *neyman-pearson* memaksimalkan nilai *probability of detection* untuk meningkatkan *probability of false alarm*.

$$T = -\ln(PFA)\sigma_i^2 \tag{2.3}$$

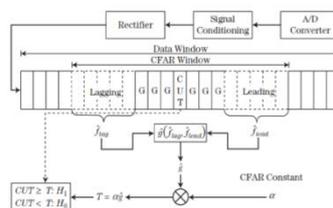
Pada persamaan *threshold* diatas terdapat dua persamaan. Persamaan pertama adalah nilai dari *probabilitas of false alarm* (PFA) yang diinginkan dan persamaan kedua adalah σ_i^2 adalah *output* dari *square law detector*.

2.3 Constant False Alarm Rate (CFAR)

CFAR adalah suatu sistem radar yang dirancang untuk mencapai dan mempertahankan nilai dari probabilitas *false alarm* yang diinginkan. Dengan nilai *false alarm* yang tinggi dapat menguras *resource* pada radar. Jika ada interferensi yang telah didistribusikan secara eksponensial, suatu probabilitas *false alarm* dapat dihitung dengan persamaan [1].

$$PFA = \int_T^\infty \frac{1}{\sigma_i^2} \exp\left(-\frac{z}{\sigma_i^2}\right) dz \tag{2.3}$$

Dimana *z* adalah *output* dari detektor dengan inputnya berupa sinyal interferensi yang kompleks. Untuk σ_i^2 adalah nilai rata-rata yang di harapkan dari *z*. Beberapa metode CFAR telah dikembangkan untuk meningkatkan kinerja yang lebih optimal, contohnya *Greatest of CA-CFAR*, *Smallest of CA-CFAR*, *Censored CFAR* atau *Trimmed Mean CFAR* dan *Order Statistic CFAR* [1].



Gambar 2.3 Arsitektur CFAR [1]

Pada arsitektur CFAR terdapat data-data statistik dari hasil deteksi yang berupa sinyal target dan juga sinyal *interference* atau *noise*. Data yang didapatkan akan diproses oleh *CFAR window* yang berada dalam *data window*. *CFAR window* didalamnya terdapat beberapa bagian yaitu *lagging*, *leading*, *guard cell* (G) dan *Cell Under Test* (CUT). Pada *lagging* dan *leading* merupakan bagian dari jendela referensi yang digunakan untuk menghitung perkiraan statistik dari *noise* yang ada pada *data window*. Perhitungan

antara *lagging* dan *leading* dilakukan secara terpisah, kemudian hasil dari kedua perhitungan tersebut akan digabungkan.

2.3.1 Cell Averaging CFAR

Algoritma relatif sederhana karena menghitung *threshold* berdasarkan perkiraan kekuatan interferensi rata-rata di jendela referensi [1]. CFAR dirancang untuk beroperasi pada lingkungan *noise* homogen, dimana pada kondisi lingkungan *noise* homogen ini bersifat identik. Cara kerja pada *cell averaging* CFAR adalah dengan mengestimasi daya *noise* CA-CFAR window [1][7].

$$Z_{ca} = \frac{1}{N} \sum_{i=1}^N X_i \quad (2.7)$$

Dimana N adalah jumlah sel referensi dan nilai X_i merupakan sel ke-i, $X = \{X_1, X_2, \dots, X_i\}$. Sedangkan untuk nilai *threshold* pada CA-CFAR dapat ditentukan dengan persamaan [1].

$$T_{ca} = \alpha_{CA} Z_{CA} \quad (2.8)$$

Nilai dari α_{CA} merupakan konstanta CA-CFAR dan dapat ditentukan dengan persamaan [1].

$$\alpha_{CA} = N \left[P_{fa} \frac{1}{N} - 1 \right] \quad (2.9)$$

P_{fa} ditentukan berdasarkan estimasi daya *noise*. Pada CA-CFAR cara kerjanya adalah menghitung rata-rata nilai dari data yang berada di sel *lagging* dan sel *leading* pada CA-CFAR window, selanjutnya dikalikan dengan α_{CA} untuk menghasilkan *threshold* pada CA-CFAR.

2.3.2 Greatest of Cell Averaging CFAR

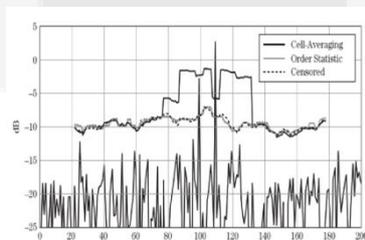
Greatest of Cell Averaging CFAR adalah pengembangan dari metode Cell Averaging CFAR untuk mengurangi gangguan *false alarm* pada daerah yang bersifat heterogen. Gangguan dari *false alarm* dapat dikurangi dengan cara menghitung data statistik dari *interference* atau *noise* rata-rata pada *lagging* dan *leading cell* secara terpisah dan memilih yang paling besar dari data perhitungan [1][10]. Hasil dari data tersebut akan digunakan untuk menghitung nilai *threshold*. Data statistik dari *interferensi* atau *noise* pada jendela GOCA-CFAR dapat dihitung dengan persamaan [1].

2.3.3 Order Statistic CFAR

Order Statistic CFAR dirancang untuk menekan *masking* sasaran, sehingga pada kondisi *noise* heterogen dapat meminimalkan terjadinya *false alarm* dan *miss detection*. Pada OS-CFAR nilai N adalah jumlah sampel dan memilih sampel k sebagai CFAR statistik, dimana nilai $k > N/2$ [1][7]. Sampel k merupakan sampel pada sel referensi yang telah diurutkan dari yang terkecil ke terbesar. Dengan begitu OS-CFAR mampu mengolah dan menekan tepi *clutter* dari *false alarm* [1]. Untuk mencari nilai dari P_{fa} pada OS-CFAR adalah sebagai berikut [1][9].

$$P_{FAOS} = k \binom{N}{k} \frac{(k-1)! (\alpha_{OS} + N - k)!}{(\alpha_{OS} + N)!} \quad (2.11)$$

Dimana nilai α_{OS} berasal dari *constant* CFAR dan Fungsi dari nilai k adalah untuk memeriksa kerugian CFAR yang disebabkan oleh OS-CFAR. Dengan kemampuannya untuk menekan target *masking*, jendela referensi CFAR mungkin mempunyai lebih dari satu target tetapi algoritma OS-CFAR dirancang untuk menekan *masking sasaran* tersebut.



Gambar 2. 4 Perbandingan *Cell Averaging* dan *Order Statistic* [1]

Pada gambar diatas dapat dibuktikan bahwa dengan menggunakan OS-CFAR dapat menekan *masking* sasaran sehingga dapat menekan *threshold* yang terbentuk saat kondisi lingkungan heterogen.

Dengan menggunakan metode Order Statistik, data statistik yang masuk dari detektor akan masuk ke CFAR window. Data yang telah masuk akan dibagi-bagi ke dalam *Cell Under Test*

(CUT), *sell guard* dan sel referensi. CUT merupakan sel yang sedang diuji, sel yang akan uji meliputi seluruh sampel. Sel *guard* merupakan sel penjaga, sel *guard* tidak masuk dalam perhitungan karena bersebelahan langsung dengan CUT sehingga khawatir mengandung *noise*. Sel referensi terdiri dari dua bagian yaitu sel *lagging* dan sel *leading*. Setelah data statistik masuk pada CFAR *window*, data yang berada pada sel *lagging* dan *leading* akan digabungkan menjadi satu. Setelah digabungkan data selanjutnya *diurutkan* dari yang terkecil ke terbesar seperti persamaan berikut [7][8].

$$X(1) \leq X(2) \leq X(3) \leq \dots \leq X(N) \quad (2.12)$$

Selanjutnya menentukan sampel yang akan diuji (k), dengan cara $k = 3N/4$. Setelah menentukan sampel k , harus mengambil salah satu nilai yang lebih besar dari sampel (k). Selanjutnya keluaran nilai tersebut akan dihitung dengan *constant* CFAR untuk menentukan nilai parameter *threshold* [7][9].

$$T_{ca} = (\alpha CA)(k) \quad (2.13)$$

Setelah itu hasil dari parameter *threshold* akan dibandingkan dengan CUT pada saat itu, proses ini diulang hingga semua sampel pada OS-CFAR *window* diuji. Algoritma OS-CFAR menghasilkan kerugian CFAR yang lebih besar dari CA-CFAR. Kelebihan dari algoritma OS-CFAR yaitu menekan *masking* sasaran secara bersamaan, panjang jendela referensi OS-CFAR dapat ditambah untuk mengurangi kerugian CFAR [1][6].

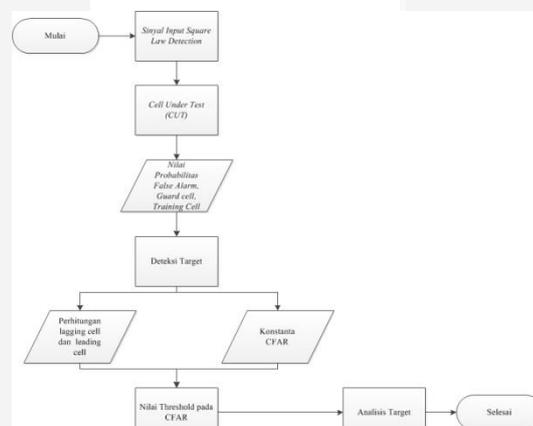
3. Pembahasan

3.1 Perancangan Sistem

Pada tugas akhir ini akan dilakukan simulasi menggunakan *software*. Sistem algoritma yang digunakan adalah *Constant False Alarm Rate* (CFAR), bertujuan untuk mendapatkan nilai *false alarm* yang konstan terhadap *noise*. Dalam perancangan ini akan membandingkan sistem algoritma *Neyman-Pearson*, *Cell averaging* CFAR dan *Order Statistic* CFAR.

3.2 Skema Simulasi

Algoritma pada *Cell Averaging* CFAR dan *Order Statistic* CFAR menggunakan skema pengujian yang sama yaitu algoritma CFAR. Pada algoritma CFAR data statistik yang masuk akan dibagikan kedalam sel-sel pada CFAR *window*. CFAR *window* terdiri dari *Cell Under Test* (CUT), *guard cell*, *lagging cell* dan *leading cell*. Berikut ini adalah Skema *flowchart* pengujian pada CFAR.



Gambar 3. 1 Blok Diagram Alir Perancangan Sistem

Pada *flowchart* diatas merupakan cara kerja dari algoritma CFAR. Pada CA-CFAR sinyal input masuk dari detektor *square law*, kemudian sinyal input yang berisikan data statistik akan dibagi kedalam sel-sel CFAR *window*. Pada CFAR *window*, sel-sel akan diperiksa satu persatu kedalam sel referensi. Pada CA-CFAR data statistik akan dirata-ratakan untuk memperoleh nilai parameter *threshold*. Sedangkan pada *order statistic* CFAR data statistik yang masuk akan diurutkan dari yang terkecil ke terbesar, supaya mendapatkan nilai *threshold* untuk menekan target *masking*.

3.3 Parameter Simulasi

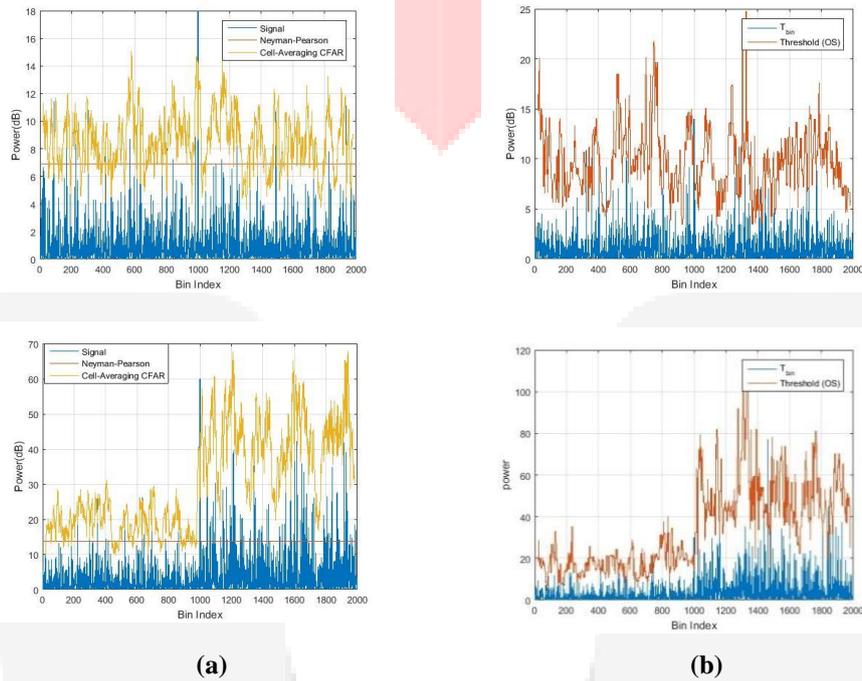
Pada perancangan simulasi ini akan digunakan estimasi nilai-nilai parameter yang dipakai pada Algoritma *Neyman-Pearson*, CA-CFAR dan OS-CFAR. Nilai-nilai parameter yang digunakan sebagai berikut.

Tabel 3. 1 Parameter Simulasi

Parameter	Nilai
Jumlah sampel	2000
Jumlah sel referensi	16
Jumlah sel <i>Guard</i>	2
Nilai Probabilitas <i>false alarm</i>	$10^{-1}, 10^{-2}, 10^{-3}$ dan 10^{-4}

4. Hasil dan analisis

Dari sekian banyak simulasi akan ditampilkan hasil simulasi dengan $P_{fa} = 10^{-3}$ pada kondisi *noise* homogen dan heterogen



Gambar 3.2 kiri menggunakan CA-CFAR dan kanan menggunakan OS-CFAR, homogen

Pada pengujian ini $P_{fa} = 10^{-3}$, Hasil pengujian untuk *noise* homogen dengan CA-CFAR, banyaknya sinyal *noise* ataupun target yang melebihi batas *threshold* dari *Neyman-Pearson* adalah 22 sampel dengan nilai P_{fa} aktual 0,0112. Pada CA-CFAR adalah 22 sampel dengan nilai P_{fa} aktual 0,0112 menggunakan nilai $P_{fa} = 10^{-3}$ lebih baik dibanding dengan menggunakan $P_{fa} = 10^{-2}$ karena nilai *false alarm* akan mengecil, tetapi bila menggunakan nilai P_{fa} nominal yang sangat kecil maka nilai probabilitas deteksi akan semakin kecil. Pada *noise* heterogen banyaknya sinyal *noise* ataupun target yang melebihi batas *threshold* dari *Neyman-Pearson* adalah 118 sampel dengan nilai P_{fa} aktual 0,0601. Pada CA-CFAR adalah 20 sampel dengan nilai P_{fa} aktual 0,0102.

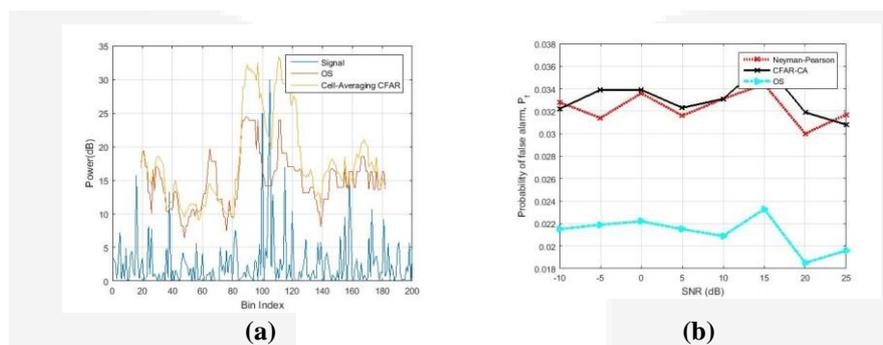
Sedangkan pada OS-CFAR, Pengujian kedua dengan $P_{fa} = 10^{-3}$ seperti pada Gambar 3.1 (b) terlihat jelas bahwa parameter *threshold* sangat baik pada saat kondisi lingkungan *noise* homogen. Sinyal *noise* yang melewati *threshold* sangat berkurang, sinyal yang melewati *threshold* berjumlah 23 dengan nilai P_{fa} 0,0117. Pada *noise* heterogen banyaknya sinyal *noise* ataupun target yang melebihi batas *threshold* dari OS-CFAR adalah 12 sampel dengan nilai P_{fa} aktual 0,0062.

Tabel 3. 2 Hasil Simulasi Kondisi *Noise* Homogen

Pfa	Neyman-Pearson	CA-CFAR	OS-CFAR
10^{-1}	0,1186	0,1258	0,1512
10^{-2}	0,0285	0,0326	0,0280
10^{-3}	0,0112	0,0112	0,0117
10^{-4}	0,0041	0,0056	0,0020

Tabel 3. 3 Hasil Simulasi Kondisi *Noise* Heterogen

Pfa	Neyman-Pearson	CA-CFAR	OS-CFAR
10^{-1}	0,2495	0,1314	0,1458
10^{-2}	0,1191	0,0372	0,0329
10^{-3}	0,0601	0,0102	0,0062
10^{-4}	0,0224	0,0031	0,0026

**Gambar 3.3** Hasil Perbandingan CA-CFAR dan OS-CFAR dan Perbandingan dengan Kurve ROC

Pada pengujian ini menggunakan sampel yang lebih sedikit dengan nilai $Pfa = 10^{-2}$, karena untuk melihat lebih jelas perbedaan antara kedua algoritma tersebut pada kondisi lingkungan *noise* homogen. Pengujian ini menggunakan dua sinyal target yaitu pada bin ke-100 dan ke-105. Algoritma pada CA-CFAR tidak mampu menekan tepi *clutter threshold* pada saat terdapat sinyal target yang berdekatan, selanjutnya membandingkan dengan kurva ROC Terlihat pada Gambar 3.3 (b), nilai Pfa estimasi = 10^{-2} algoritma yang paling mendekati dari nilai Pfa simulasi adalah OS-CFAR. Hal ini membuktikan bahwa metode OS-CFAR menghasilkan Pfa yang baik dibandingkan dengan CA-CFAR dan *Neyman-Pearson*.

5. Kesimpulan

Setelah melakukan pengujian dengan simulasi dan hasil analisis data yang didapatkan dalam tugas akhir ini, maka dapat diambil kesimpulan sebagai berikut:

1. Dari hasil simulasi diketahui bahwa algoritma Order statistik CFAR lebih baik untuk menekan target *masking* dibandingkan dengan *Cell Averaging* CFAR.
2. Nilai dari *Probability of false alarm* nominal berbanding lurus dengan probabilitas deteksi. Apabila nilai Pfa nominal mengecil maka nilai dari probabilitas deteksi juga ikut mengecil.
3. Berdasarkan pengujian OS-CFAR pada kondisi *noise heterogen* dengan menggunakan Pfa nominal dari 10^{-1} sampai 10^{-4} diperoleh nilai Pfa aktual $10^{-1}= 0,1458$, $10^{-2}= 0,0329$, $10^{-3}= 0,0062$ dan $10^{-4}= 0,0026$. Hasil Pfa aktual yang dihasilkan OS-CFAR lebih baik dibandingkan dengan *Neyman-Pearson* dan CA-CFAR.

DAFTAR PUSTAKA

- [1] J. . Scheer, W. A. Holm, and M. A. Richards, "*Principles of Modern Radar. Basic principles*". SciTech Publishing, 2010.
- [2] M.A. Richards, "*Fundamentals of Radar Signal Processing*". The McGraw-Hill Companies, 2005.
- [3] S. M. Kay, "*Fundamentals of Statistical Signal Processing*". Upper Saddle River, 1993.
- [4] F. Gustomo, Suwadi, Titiék Suryani "Analisa Penggunaan Sinyal Radar Bentuk Pulsa dan Gelombang Kontinyu untuk Target Bergerak dengan Model Clutter Terdistribusi Rayleigh," Institut Teknologi Sepuluh Nopember, 2013.
- [5] F. Y. Suratman, A. Adya Pramudita, and D. Arseno, "Deteksi Sinyal: Overview Model Parametrik menggunakan Kriteria Neyman-Pearson" *Elkomika*, vol. 7, no. 1, pp. 14–28, Jan. 2019.
- [6] A. Youssef, P. F. Driessen, F. Gebali, and M. Belaid, "*Performance Evaluation of Time Compression Overlap-Add Radar Systems Based on Order-Statistics CFAR under Convolution Noise Jamming*." University of Victoria, 2017.
- [7] H. Rohling, "*Radar CFAR Thresholding in Clutter and Multiple Target Situations*", IEEE Transactions On Aerospace and Electronic System, vol. 19, no. 4, Jul.1983.
- [8] P. Swerling, "*Probability of detection for fluctuating targets*", Mar.1954.
- [9] M. Shor, N. Levanon, "*Performances Of Order Statistics CFAR*", IEEE Transactions On Aerospace and Electronic System, vol. 27, no. 2, Mar.1991.
- [10] P. P. Gandhi, S. A. Kassam, "*Analysis Of CFAR Processor In Nonhomogenous Background*", IEEE Transactions On Aerospace and Electronic System, vol. 24, no. 4, Jul.1988.