

PEMROSESAN RAW DATA SAR (SYNTHETIC APERTURE RADAR) MENJADI SAR IMAGE SPACE

SAR (SYNTHETIC APERTURE RADAR) RAW DATA TO SAR IMAGE SPACE PROCESSING

Syifa Haunan Nashuha¹, Erfansyah Yudhi Eka Ali, S.T., M.T.², Dr. Ir. Heroe Wijanto, M.T.³

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

¹haunannashuha@telkomuniversity.ac.id, ²erfansyahali@telkomuniversity.ac.id, ³heroe@telkomuniversity.ac.id

Abstrak

Synthetic Aperture Radar (SAR) merupakan teknologi penginderaan jauh yang potensial untuk diterapkan di Indonesia. Dengan menggunakan gelombang radio sebagai 'mata'-nya, penginderaan jauh dengan SAR memungkinkan pengamatan dilakukan pada kondisi iluminasi rendah dan cuaca apapun. Hal ini jelas lebih baik dibandingkan menggunakan sensor optis. SAR memanfaatkan informasi yang dihasilkan dari kombinasi pantulan sinyal dan pergerakan platform. Dengan informasi tersebut, platform seolah-olah memiliki antenna yang jauh lebih besar dari antenna sesungguhnya dan dapat dihasilkan citra beresolusi tinggi.

Penelitian ini membahas tentang bagaimana menerapkan proses pengolahan raw data SAR menjadi bentuk SAR image space. Adapun sinyal raw data SAR yang digunakan sebagai sumber dan data referensi dalam penelitian ini didapatkan dari file citra GIF berupa target titik tunggal dan jamak. Pemrosesan dilakukan dengan menggunakan perangkat lunak MATLAB dengan optimalisasi untuk menambah efisiensi dan mengurangi durasi proses. Didapatkan hasil pemrosesan pada single target dengan akurasi sebesar 98.22%.

Kata kunci : SAR, Imaging, Spotlight SAR, MATLAB

Abstract

Synthetic Aperture Radar (SAR) is a type of remote sensing technology that is very likely to be applied in Indonesia. With radio waves as the 'eyes', remote sensing by using SAR enables low-illuminated observation in any weather. This is much more preferable than using optical sensors. SAR utilize the information from combining reflected signal and platform movement. Using these information, the system synthesizes a bigger antenna compared to the physical antenna built on the platform. Thus producing a higher resolution image.

This research studies about how to process SAR raw data to SAR image space. The processed raw data is generated from GIF image file with single and multiple point targets in various sizes. The whole process is simulated in MATLAB with optimized source code in order to increase efficiency and decrease processing time. From this research, a processed image with 9.22% of accuracy is acquired.

Keywords: SAR, Imaging, Spotlight SAR, MATLAB

1. Pendahuluan

Teknologi SAR dapat memberikan resolusi yang lebih tinggi tanpa pemrosesan rumit. Hal ini disebabkan oleh sistem SAR yang memanfaatkan pergerakan antenna terhadap target. Umumnya, antenna dengan aperture kecil, yang menghasilkan beam lebar, membuat penentuan target semakin sulit dilakukan. Beam yang lebar menyulitkan penentuan dari target mana sebuah gelombang pantulan berasal. Untuk mengatasinya, dibutuhkan antenna dengan aperture

lebih besar, sehingga beam yang dihasilkan lebih kecil, dan memudahkan pembedaan ciri target sekaligus meningkatkan resolusi radar. Dengan menggunakan Synthetic Aperture Radar, antenna dengan aperture besar yang akan penambahan beban pada platform antenna (satelit, UAV, dll.) dapat digantikan fungsinya dengan antenna yang lebih kecil. Pergerakan antenna memungkinkan banyak beam yang dipancarkan dan banyak pantulan atas suatu target yang diterima. Dengan analisa lebih lanjut pada gelombang-gelombang pantulan, dapat dihasilkan citra SAR

dengan resolusi tinggi, seolah-olah dihasilkan oleh antenna beraperture besar [3].

Pada penelitian sebelumnya, telah dilakukan simulasi pencitraan SAR tiga dimensi dengan bahasa pemrograman Java oleh Erfansyah Ali yang menggunakan Range Doppler Algorithm. Dengan menggunakan data raw yang di-generate pada MATLAB, penelitian ini menghasilkan suatu program pencitraan SAR tiga dimensi [3]. Matthew Schlutz pada tahun 2009 mengembangkan program berbasis MATLAB yang dapat men-generate raw data SAR dari citra format .GIF [4]. Penelitian tersebut menjadi dasar dari penelitian ini, dimana akan dilakukan optimalisasi pada source code. Range Doppler Algorithm dipilih untuk digunakan pada penelitian ini karena RDA merupakan salah satu algoritma yang dapat memproses sinyal SAR dengan kompleksitas yang rendah dan dalam waktu singkat, seperti dibuktikan pada penelitian oleh Bhanumurthy [5].

2. Dasar Teori/Metodologi/Perancangan

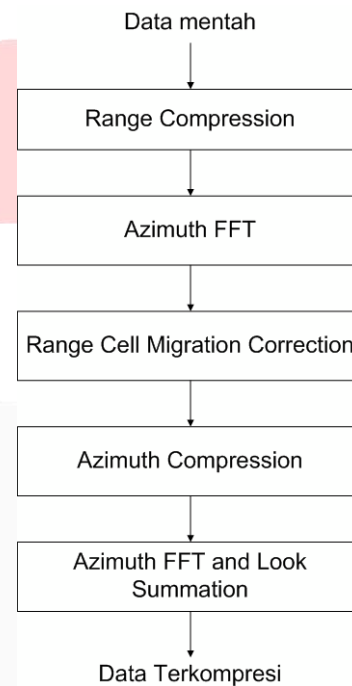
2.1 Synthetic Aperture Radar

Sebagai bentuk pengembangan dari radar konvensional, Synthetic Aperture Radar menjanjikan pencitraan baik di waktu siang maupun malam dan tidak terpengaruh cuaca. Citra beresolusi tinggi dapat dihasilkan dengan antenna beraperture kecil. Sistem ini berbasis pada geometri pencitraan tatap-samping berupa radar yang dipasang pada platform bergerak. Sistem radar memancarkan gelombang elektromagnetik berupa sinyal phase-encoded berdaya tinggi dan menerima pantulan dari sinyal backscatter secara sekuensial. [1]

Blok diagram sistem SAR pada umumnya terdiri dari dua bagian, ruas angkasa dan ruas Bumi. Antena berfungsi sebagai sensor yang memancarkan dan menerima pantulan sinyal. Sinyal yang

dipancarkan bersumber dari pembangkit sinyal pada satelit. Sinyal yang dipancarkan berupa pulse dengan frekuensi pengulangan (PRF, Pulse Repetition Factor) dan interval pengulangan (PRI, Pulse Repetition Interval) tertentu. PRF dan PRI menentukan durasi dimana antenna berfungsi pada mode transmit atau listen. [1]

2.2 Range Doppler Algorithm



Gambar 0.1 Basic RDA [2]

Range Compression

Tahap pertama dari RDA adalah Range Compression menggunakan metode Matched Filter yang dibahas pada sub BAB 2.3. Pada kasus ini, filter dibangkitkan dengan mengambil complex conjugate dari replica pulse yang diberi padding nol. Jika $S_0(f_r, \eta)$ adalah range transformasi Fourier dari raw data, dan $G(f_r)$ adalah matched filter domain frekuensi, maka keluaran dari matched filter dapat didefinisikan sebagai

$$\begin{aligned}
 & \left(\frac{1}{2} \right) \\
 & = \left\{ \left(\frac{1}{2} \right) \left(\frac{1}{2} \right) \right\} \\
 & = \\
 & 2 \left(\frac{1}{2} \right) \\
 & - \frac{1}{2} \left(\frac{1}{2} \right) \\
 & - \exp\{-4 \left(\frac{1}{2} \right)\} \\
 & / \} \quad (2.1)
 \end{aligned}$$

korelasi silang antara sinyal terima dengan replika konjugasi dari sinyal yang diinginkan dengan menggunakan matched filter. Pada konteks ini, “filter” korelasi adalah “matched” terhadap sifat fasa yang diharapkan dari sinyal. Secara umum apabila $s_r(t)$

adalah sinyal yang diterima dan $g(t)$ adalah replika sinyal, operasi matched filter dapat diimplementasikan dengan sebuah korelasi.

Azimuth FFT

$$G(f) = S(f) * G(-f) \quad (2.3)$$

Selanjutnya keluaran dari tahap Range Compression masuk ke dalam blok Azimuth FFT yang akan memindahkan sinyal ke domain Doppler dengan operasi FFT pada setiap Azimuth bins.

Apabila sebuah filter $h(t)$ diaplikasikan dengan konvolusi daripada sebuah korelasi, inti dari filter adalah kompleks konjugasi waktu terbalik dari replika.

Range Cell Migration Correction (RCMC)

Fungsi dari Range Cell Migration Correction (RCMC) adalah untuk meluruskan trajektori sehingga parallel terhadap sumbu frekuensi azimuth. RCMC bekerja pada domain Doppler dan dependen pada range time dan frekuensi azimuth. RCMC didasarkan pada persamaan berikut.

$$\Delta(f) = \frac{1}{8} \quad (2.2)$$

Azimuth Compression

Setelah RCMC, sinyal keluaran kembali dimasukkan ke dalam matched filter untuk memfokuskan data pada arah azimuth. Matched filter dilakukan setelah RCMC untuk memanfaatkan domain Doppler yang merupakan karakteristik keluaran RCMC. Setelah itu pada hasil keluaran matched filter dilakukan IFFT, dan dihasilkan citra final dari Range Doppler Algorithm.

$$G(f) = S(f) * G(-f) \quad (2.4)$$

Pada kasus ini, “matched filter” berarti “filter konvolusi” dan integral konvolusi seperti pada persamaan berikut.

$$G(f) = S(f) * G(-f) \quad (2.5)$$

$$G(f) = S(f) G(-f) \quad (2.6)$$

Matched filter di domain waktu dapat dilakukan dengan konvolusi di domain waktu. Matched filter juga dapat diterapkan dengan konvolusi cepat pada domain frekuensi, menghasilkan hasil yang sama. Sebagai tambahan untuk diaplikasikan di domain frekuensi, matched filter juga dapat didesain di domain frekuensi dengan sedikit kesalahan dalam akurasi. Terkadang ini lebih mudah dilakukan. Persamaan matched filter pada domain frekuensi adalah seperti persamaan berikut.

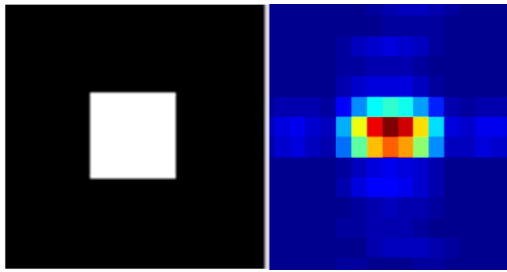
$$G(f) = S(f) G(-f) \quad (2.7)$$

2.3 Matched Filter [3]

Sinyal terima pada sistem radar dapat dipisahkan menjadi sinyal asli dan derau dengan cara melakukan

3. Pembahasan

Hasil simulasi pada Target A



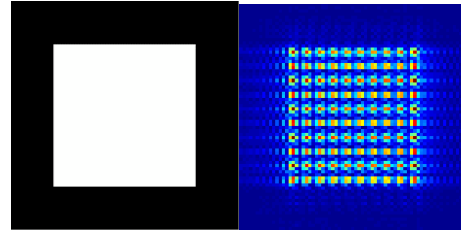
Gambar 0.1 Perbandingan citra raw(atas) dan hasil simulasi SAR (bawah) Target A berukuran 3x3 pixel

Simulasi pada target A dilakukan sebagai pengujian source code pada tahap awal penelitian. Ukuran yang relatif kecil mempersingkat waktu pemrograman sehingga dapat segera ditemukan error atau ketidaksesuaian pada program. Selain itu, target A juga disimulasikan sebagai perbandingan untuk Target B yang memiliki bentuk sama tapi dengan ukuran yang lebih besar.

Setelah melalui proses simulasi RDA, dihasilkan citra seperti ditampilkan pada Gambar 4.1. Gambar sebelah kiri adalah citra target, sedangkan gambar sebelah kanan adalah citra hasil simulasi RDA. Terlihat bahwa citra hasil dapat dikenali sebagai target A, dimana blok bernilai tinggi di tengah gambar mewakili point target di tengah Target A.

Untuk evaluasi dengan MSE, didapatkan nilai MSE 1.78 atau ketepatan sebesar 98.22%. Hal ini menunjukkan bahwa citra hasil pemrosesan hampir identik dengan citra target.

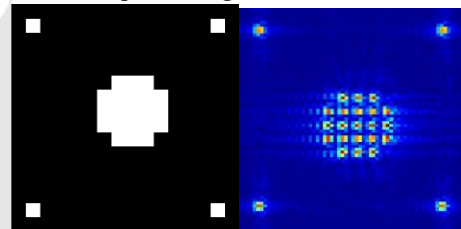
Hasil Simulasi pada Target B



Gambar 0.2 Perbandingan citra raw (kiri) dan hasil simulasi SA (kanan) Target B berukuran 16x16 pixel

Pada Target B, disimulasikan target dengan point berukuran 16x16 pixel. Pada Gambar 4.2 dapat dilihat perbandingan dari citra target dan citra hasil simulasi. Sama seperti pada target A sebelumnya, hasil simulasi dapat dengan mudah dikenali sebagai citra target dengan melihat bagian bernilai tinggi yang merupakan representasi point target di tengah citra. Garis-garis pada citra sumbu azimuth dikarenakan resolusi azimuth yang terbatas, sehingga citra hasil tidak mem-blok penuh pada kedua arah azimuth dan range. Didapatkan nilai MSE sebesar 87.99, yang berarti ketepatan program dalam memproses citra hanya sebesar 12,01%.

Hasil Simulasi pada Target C



Gambar 0.3 Perbandingan citra raw (kiri) dan hasil simulasi SA (kanan) Target C berukuran 16x16 pixel

Citra yang selanjutnya disimulasikan adalah Target C yang berupa citra 16x16 pixel dengan multiple target. Pada Gambar 4.3 dapat dilihat bahwa kedua citra, target dan hasil simulasi) dapat dikenali sebagai citra yang sama. Tetapi sama seperti yang terlihat jelas pada hasil simulasi Target B, tampak gap

antar bins pada sumbu azimuth yang disebabkan oleh resolusi azimuth yang kurang baik. Nilai MSE pada citra C ini menghasilkan nilai 99.61 atau ketepatan sebesar 0.39%. Berdasarkan analisis MSE, hal ini berarti citra hasil sangat berbeda dengan citra target.

Analisa pada Target 32x32 dan 64x64 pixel

Untuk menganalisa kinerja lebih jauh dari program, maka dilakukan pengujian pada citra satelit yang masing-masing berukuran 32x32 pixel (Gambar 4.16) dan 64x64 Pixel (Gambar 4.17). Kedua gambar ini diperoleh secara acak dari Google Earth dan merupakan gambar satelit dari Makuhari Bay di Chiba, Jepang.

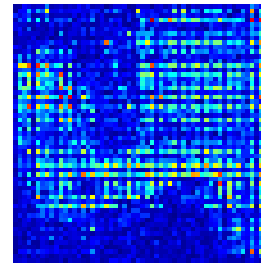


Gambar 0.4 Citra satelit 32x32 pixel

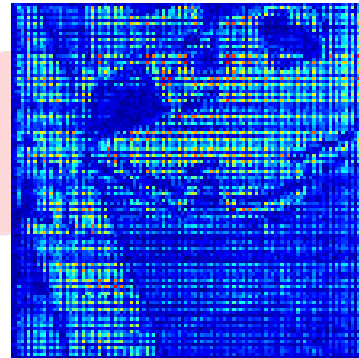


Gambar 0.4 Citra satelit 64x64 pixel

Dari kedua citra tersebut didapatkan citra hasil sebagaimana pada Gambar 4.6 dan 4.7.



Gambar 0.6 Hasil Simulasi Citra 4.16



Gambar 0.7 Hasil Simulasi Citra 4.17

Analisis MSE menghasilkan nilai yang sangat besar, yaitu 234.05 untuk citra 64x64. Tetapi citra 32x32 pixel justru menghasilkan nilai MSE yang cenderung baik, yaitu 4,36.

Analisa Durasi Pemrosesan

Setelah dilakukan evaluasi visual, dilakukan analisis performa source code dengan menganalisa durasi pemrograman. Dalam melakukan evaluasi ini, digunakan perangkat keras dengan spesifikasi seperti ditampilkan pada Tabel 4.1

Item	Deskripsi
Prosesor	Intel® Core™ i5-4210Y CPU @1.50GHz
Memori (RAM)	4.00 GB

Tipe Sistem	64-bit System, processor	Operating x64-based
-------------	--------------------------------	------------------------

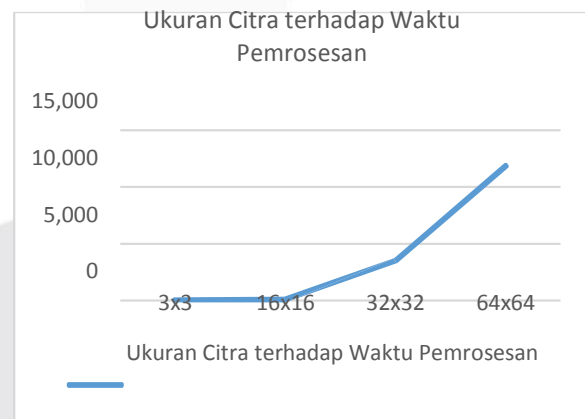
Tabel 0.1 Sistem hardware yang digunakan pada simulasi

Range	2.092571	2.176673
Compression		
Azimuth	0.437378	0.475496
FFT		
RCMC	0.915787	1.054292
Azimuth	0.500722	0.606586
Compression	1010.731	897.7908
+IFFT		
Durasi		
Total		

Tabel 0.2 Perbandingan durasi pemrograman sebelum dan sesudah optimasi

Target	Proses	Durasi Sebelum Optimalisasi (detik)	Durasi Setelah Optimalisasi (detik)
Target A	Generate	34.375132	32.059345
	Echoes		
	Range	2.190786	2.116749
	Compression		
	Azimuth	0.464751	0.494845
	FFT RCMC		
	Azimuth	0.893524	1.068784
	Compression	0.493174	0.612966
	+IFFT	39.56346	39,50046
	Durasi Total		
Target B	Generate		
	Echoes	989.392208	835.792275
	Range		
	Compression	2.210366	2.158668
	Azimuth		
	FFT	0.446272	0.452575
	RCMC		
	Azimuth	0.928712	1.051929
	Compression	0.503520	0.609929
	+IFFT	997.3695	840.0654
Target C	Durasi Total		
	Generate		
Target C	Echoes	1003.353878	890.356994

Dari Tabel 4.2 terlihat bahwa sesudah dilakukan optimasi pada source code [4], waktu pemrosesan dapat diselesaikan dengan lebih cepat. Selisih pada durasi pemrograman Target A adalah 0.063 detik, tidak terlalu signifikan dibandingkan selisih waktu pemrograman pada Target B (12,996884 detik) an Target C (12,9402 detik).



Pada Tabel 4.3 dapat dilihat hubungan antara ukuran citra target dengan waktu pemrosesan. Kurva pixel terhadap waktu naik dengan cukup drastis pada setiap sampelnya. Citra ukuran 3x3 pixel dan 16x16 pixel masing-masing memakan waktu pemrosesan seperti tercantum pada Tabel 4.2 sedangkan pemrosesan pada target berukuran 32x32 pixel memakan waktu 3494.5

detik dan citra target 64x64 pixel memakan waktu kurang lebih 11.837.5 detik. Nilai ini sangatlah bergantung pada tingkat kerumitan dari citra target sendiri. Dapat dilihat pada Tabel 4.2 bahwa citra B dan citra C yang sama-sama memiliki ukuran 16x16 pixel menghabiskan waktu pemrosesan yang berselisih. Hal ini dikarenakan perbedaan kerumitan citra B yang hanya berupa titik-titik target di tengah citra, dan citra C yang lebih rumit. Simulasi dilakukan menggunakan perangkat keras dengan spesifikasi yang sama, dan hanya dilakukan pada empat ukuran (3x3, 16x16, 32x32, dan 64x64) dikarenakan perangkat keras yang tidak memiliki spesifikasi yang cukup untuk memproses citra yang lebih besar.

4. Kesimpulan

Setelah diselesaikannya penelitian ini, dapat disimpulkan bahwa:

1. Program MATLAB yang digunakan bekerja dengan cukup baik dalam memproses data mentah SAR menjadi SAR image space. Citra 3x3 dengan target tunggal dapat diproses dengan akurasi sebesar 98.22%. Secara visual semua citra hasil dapat dikenali sebagai gambaran SAR dari citra target, tetapi dilihat dari nilai MSE hanya citra 3x3 pixel dengan target tunggal dan citra 32x32 pixel dengan citra target hasil pencitraan satelit yang memenuhi target akurasi diatas 70%.
2. Program dioptimalkan untuk sensor SAR dengan frekuensi 4,5 GHz dan dapat digunakan dengan baik pada target single point maupun multiple point. Untuk target single point masih terlihat skewing yang kemungkinan besar disebabkan oleh kurang optimalnya proses Azimuth FFT atau RCMC,

tetapi juga simulasi dilakukan pada citra yang berukuran lebih besar skewing tidak begitu kentara sehingga citra hasil masih dapat dikenali dengan baik.

3. Perlu dilakukan evaluasi lebih lanjut dengan metode evaluasi selain MSE (Mean Square Error) dikarenakan citra hasil pemrosesan berupa colormap yang berbeda format pewarnaan dengan citra asal yang berformat warna greyscale. Hal ini menyebabkan citra hasil dideteksi sebagai citra dengan nilai kesalahan yang tinggi (diwakilkan oleh nilai MSE yang besar) karena perubahan format colormap ke grayscale tidak dapat mewakili hasil yang terlihat secara visual.
4. Masih ada kekurangan dalam waktu pemrosesan, yang dapat dikurangi dengan mengoptimasi source code lebih jauh lagi terutama pada bagian generasi gelombang pantul (echo) yang paling menghabiskan banyak waktu.

DAFTAR REFERENSI

- Ali, E. Y. (2012). "Simulasi Pencitraan Synthetic Aperture Radar (SAR) 3D dengan Menggunakan JAVA". Bandung, Indonesia: Institut Teknologi Bandung.
- Bhanumhurty, M. Y., & Reddy, Y. M. (2011). SAR Data Processing with Range Cell Migration. *International Journal of Engineering Science and Technology*, 3(1), 388-398.
- Chan, Y. K., & Koo, V. C. (2008). An Introduction to Synthetic Aperture Radar (SAR). *Progress in Electromagnetics Research B*, 2, 27-60.
- Cumming, I. G., & Wong, F. H. (2005). *Digital Processing of Synthetic Aperture Radar Data : Algorithms and Implementation*. Massachusetts, United States of America: Artech House.
- Dastgir, N. (2007). "Processing SAR Data using Range Doppler and Chirp Scaling Algorithms". Stockholm, Sweden: Royal Institute of Technology.

Heckbert, P. (1998, January 27). Carnegie Mellon University School of Computer Science. Dipetik July 31, 2016, dari <https://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/2001/pub/www/notes/fourier/fourier.pdf>

Maitre, H. (Penyunt.). (2008). Processing of Synthetic Aperture Radar Images. London: ISTE Ltd.

Mon, S. S., & Jiancheng, F. (2011). SAR Image Formation Process and Non-Stationary Errors in Aircraft Motion. International Journal on Machine Learning and Computing, 1(5), 498-503.

Moreira, A., Prats-Iraola, P., Younis, M., Krieger, G., Hajnsek, I., & Papathanassiou, K. .. (2013). A Tutorial on Synthetic Aperture Radar. IEEE Geoscience and Remote Sensing Magazine(April 2014), 6-43.

Prihartoni, D. (2010). Kompleksitas Algoritma Transformasi Fourier Cepat. Bandung: Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.

Schlutz, M. (2009). "Synthetic Aperture Rar Imaging Simulated in MATLAB". California, United States of America: California Polytechnic State University.

Skolnik, M. L. (2001). Introduction to Radar Systems 3rd Ed. Singapore: McGrawHill Book Co.