

**RANCANG BANGUN SISTEM PENGOLAH SINYAL
REMOTE DISPLAY PADA MARINE RADAR
MENGUNAKAN MATLAB**

**SKRIPSI
KONSENTRASI TEKNIK TELEKOMUNIKASI**

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

SAHIRUL ALAM

NIM. 0610630096-63

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2013**

LEMBAR PERSETUJUAN

**RANCANG BANGUN SISTEM PENGOLAH SINYAL
REMOTE DISPLAY PADA MARINE RADAR
MENGUNAKAN MATLAB**

**SKRIPSI
KONSENTRASI TEKNIK TELEKOMUNIKASI**

*Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik*



Disusun oleh:

SAHIRUL ALAM

NIM. 0610630096-63

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dwi Fadila Kurniawan, ST., MT.
NIP. 19720630 200003 1 002

Rudy Yuwono, ST., MSc.
NIP. 19710615 199802 1 003



LEMBAR PENGESAHAN

**RANCANG BANGUN SISTEM PENGOLAH SINYAL
REMOTE DISPLAY PADA MARINE RADAR
MENGUNAKAN MATLAB**

Disusun oleh:

**SAHIRUL ALAM
NIM. 0610630096**

Skripsi ini telah diuji dan dinyatakan lulus pada
Tanggal 23 Juli 2013

Dosen Penguji

Ir. Erfan Achmad Dahlan, M.T.
NIP. 19530714 198203 1 003

Gaguk Asmungi, ST., MT.
NIP.

Ali Mustofa, S.T., M.T.
NIP. 19710601 200003 1 001

Mengetahui,
Ketua Jurusan Teknik Elektro

Dr. Ir. Sholeh Hadi Pramono, M.S.
NIP. 19580728 198701 1 001



PENGANTAR

Segala puji bagi Allah SWT. atas limpahan karunia, hidayah serta rahmat-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Rancang Bangun Sistem Pengolah Sinyal Remote Display Pada Marine Radar Menggunakan MATLAB” yang diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik. Shalawat dan salam penulis sampaikan kepada Rasulullah Muhammad SAW dan keluarganya serta sahabat-sahabatnya.

Dalam penulisan skripsi ini tentunya terdapat berbagai kesulitan dan hambatan. Namun akhirnya penulis dapat menyelesaikannya dengan rahmat-Nya dan bantuan serta dukungan dari berbagai pihak. Untuk itu penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Orang tua saya Bapak H. Mustofa dan Ibu Hj. Nurul Khotimah, kakak saya Mas Asif Awaludin, Mas Anton Abdurrahman, adik saya Imaduddin Ardiansyah serta seluruh keluarga saya yang selalu mendoakan saya serta memberi semangat, dukungan, dan kepercayaan kepada saya.
2. Bapak Dwi Fadila Kurniawan, S.T., M.T. selaku dosen pembimbing skripsi I dan Bapak Rudy Yuwono, S.T., M.Sc. selaku dosen pembimbing skripsi II yang telah memberikan saran, nasehat, ilmu, dan waktu dalam membimbing saya hingga skripsi ini dapat terselesaikan.
3. Bapak DR. Ir. Sholeh Hadi Pramono, M.S. selaku Ketua Jurusan Teknik Elektro dan Bapak M. Aziz Muslim, S.T., M.T., Ph.D. selaku Sekretaris Jurusan Teknik Elektro.
4. Bapak Ali Mustofa, S.T., M.T. selaku KKDK Telekomunikasi Jurusan Teknik Elektro.
5. Bapak Waru Djuriatno, S.T., M.T. selaku dosen penasihat akademik selama kuliah yang banyak memberikan pengarahan serta bimbingan akademik.
6. Bapak dan Ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro.

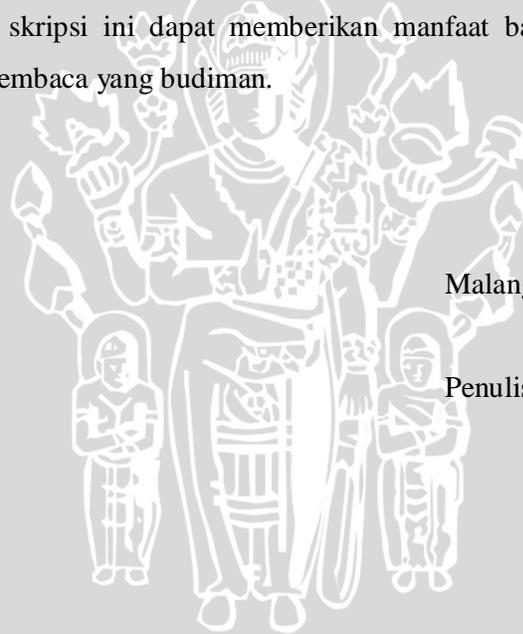
7. Bapak Ir. Halimurrahman, M.T. selaku Kepala Bidang Teknologi Atmosfer LAPAN Bandung yang telah berkenan memberikan saya kesempatan berharga untuk melakukan penelitian ini di LAPAN Bandung.
8. Keluarga besar Angkatan 2006 TEUB.
9. Sahabat-sahabat terbaik saya di kampus Universitas Brawijaya tercinta, Lucky Dedy, Yudhistira, Rifqie, Taufik, Rachmania, Rizal, Septian, mas Ari.
10. Dan untuk semua pihak yang tidak dapat penulis sebutkan satu persatu.

Dalam penulisan skripsi ini tentunya banyak terdapat kekurangan dan masih jauh dari kesempurnaan. Maka dari itu penulis mengharapkan kritik dan saran dari pembaca untuk memperbaiki segala kekurangan maupun kesalahan dari penulis.

Semoga laporan skripsi ini dapat memberikan manfaat bagi semua pihak khususnya penulis dan pembaca yang budiman.

Malang, Juli 2013

Penulis



DAFTAR ISI

PENGANTAR	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL.....	ix
ABSTRAK.....	x
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan.....	2
1.4. Batasan Masalah	3
1.5. Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1. Prinsip Radar	5
2.2. Sinyal Radar Pulsa	7
2.3. Konfigurasi Sistem Radar.....	8
2.3.1. Komponen <i>Scanner Unit</i>	9
2.3.2. Komponen <i>Display Unit</i>	11
2.4. <i>Clutter</i>	13
2.4.1. <i>Sea Clutter</i>	13
2.4.2. <i>Rain Clutter</i>	15
2.5. Kontrol <i>Display Radar</i>	16
2.5.1. <i>Gain Control</i>	16
2.5.2. <i>Sea Clutter Control</i>	17
2.5.3. <i>Rain Clutter Control</i>	18
2.6. Konversi Sistem Koordinat	20
2.7. Radar Pulsa X-Band Non Doppler Furuno 1932 Mark-2.....	21
2.8. A/D Converter Adlink PCI-9810.....	22



BAB III METODE PENELITIAN.....	23
3.1. Studi Literatur.....	23
3.2. <i>Sampling</i> Sinyal Radar Menggunakan ADC.....	24
3.3. Pembuatan Program Pengolah Sinyal Radar.....	25
3.4. Pembuatan GUI untuk <i>Display</i> Radar.....	28
3.5. Pengujian Sistem <i>Display</i> Radar dan Analisa.....	28
 BAB IV PEMBUATAN DISPLAY RADAR.....	 30
4.1. Pembuatan Program Pengolah Sinyal Radar.....	30
4.1.1 Pembuatan matriks yang berisi data posisi dan matriks yang berisi data target.....	30
4.1.2 <i>Data Mapping</i>	30
4.1.3 <i>Clutter Control</i>	32
4.1.4 Pengubahan Koordinat.....	33
4.1.5 Pembuatan <i>Plot</i>	33
4.1.6 <i>Handle Graphics</i>	33
4.2. Pembuatan GUI <i>Display</i> Radar.....	35
4.2.1 Pembuatan Jendela Utama <i>Display</i> Radar.....	35
4.2.2 Pembuatan Objek Grafik <i>Radar Map</i>	36
4.2.3 Pembuatan Tombol Pengatur Jangkauan dan Satuan Jarak Radar.....	36
4.2.4 Pembuatan <i>Gain Control</i>	43
4.2.5 Pembuatan <i>Clutter Control</i>	45
4.2.6 Pembuatan Tombol <i>Scan</i> dan Tombol Keluar.....	46
 BAB V PENGUJIAN DAN PEMBAHASAN.....	 48
5.1. Pengujian <i>Sampling</i> Sinyal Radar.....	48
5.2. Pengujian Program dan GUI Sistem <i>Display</i> Radar.....	50
5.2.1 Pengujian <i>Auto Gain</i> , Pemilih Satuan dan Pengatur Jangkauan Radar.....	50
5.2.2 Pengujian <i>Gain Control</i>	56
5.2.3 Pengujian <i>Sea Clutter Control</i>	65
5.2.4 Pengujian <i>Rain Clutter Control</i>	71
5.2.5 Waktu Simulasi.....	77

BAB VI PENUTUP	79
6.1. Kesimpulan.....	79
6.2. Saran.....	80
DAFTAR PUSTAKA.....	81
LAMPIRAN.....	82



DAFTAR GAMBAR

Gambar 2.1. Lokasi Gelombang Radar dalam Spektrum Elektromagnetik	6
Gambar 2.2. Pulsa yang dikirim dan diterima oleh radar	7
Gambar 2.3. Pantulan pulsa pertama berada pada periode pulsa kedua.....	8
Gambar 2.4. Rangkaian dan perangkat penyusun <i>scanner unit</i>	9
Gambar 2.5. <i>Heading, bearing</i> , dan <i>range</i> pada <i>display</i> radar	11
Gambar 2.6. Perangkat penyusun <i>Display Unit</i>	12
Gambar 2.7. Sinyal radar dipantulkan permukaan laut.	13
Gambar 2.8. Sinyal radar dipantulkan kembali oleh ombak di dekat posisi radar ..	14
Gambar 2.9. Ombak laut terlihat pada <i>display</i> radar jaman dulu	14
Gambar 2.10. Angin mempengaruhi bentuk ombak dan pemantulan sinyal radar..	14
Gambar 2.11. Bintik-bintik <i>clutter</i> menandakan adanya hujan lokal	15
Gambar 2.12. Nilai <i>Gain</i> yang tinggi (<i>threshold</i> semakin rendah) memungkinkan target serta <i>clutter</i> dengan sinyal lemah menjadi terlihat	17
Gambar 2.13. Nilai <i>sea control</i> yang tinggi memungkinkan target yang berada di antara <i>clutter</i> dapat terlihat lagi	18
Gambar 2.14. Dengan mengambil nilai absolut dari turunan sinyal, maka target di antara <i>clutter</i> dapat terlihat	19
Gambar 2.15. Koordinat Silinder	20
Gambar 2.16. Radar FURUNO 1932 MARK-2	21
Gambar 2.17. ADC ADLINK PCI-9810.....	22
Gambar 3.1. Diagram Alir Metode Penelitian.....	23
Gambar 3.2. Tampilan sinyal keluaran radar menggunakan osiloskop	24
Gambar 3.3. Contoh Tampilan <i>Radar Map</i> pada Radar Furuno.....	26
Gambar 3.4. Tahapan Proses dalam Program Pengolah Sinyal Radar.....	27
Gambar 3.5. Desain GUI Matlab Untuk Sistem <i>Display</i> Radar	28
Gambar 3.6. Diagram Blok Sistem Radar Saat Pengujian	29
Gambar 4.1. Diagram Alir Program Pengolah Sinyal Radar.....	34
Gambar 4.2. Diagram Alir Fungsi <i>Mapping Process</i>	35
Gambar 4.3. Diagram Alir Program “satuan_nm”	38
Gambar 4.4. Diagram Alir Program “satuan_km”	38

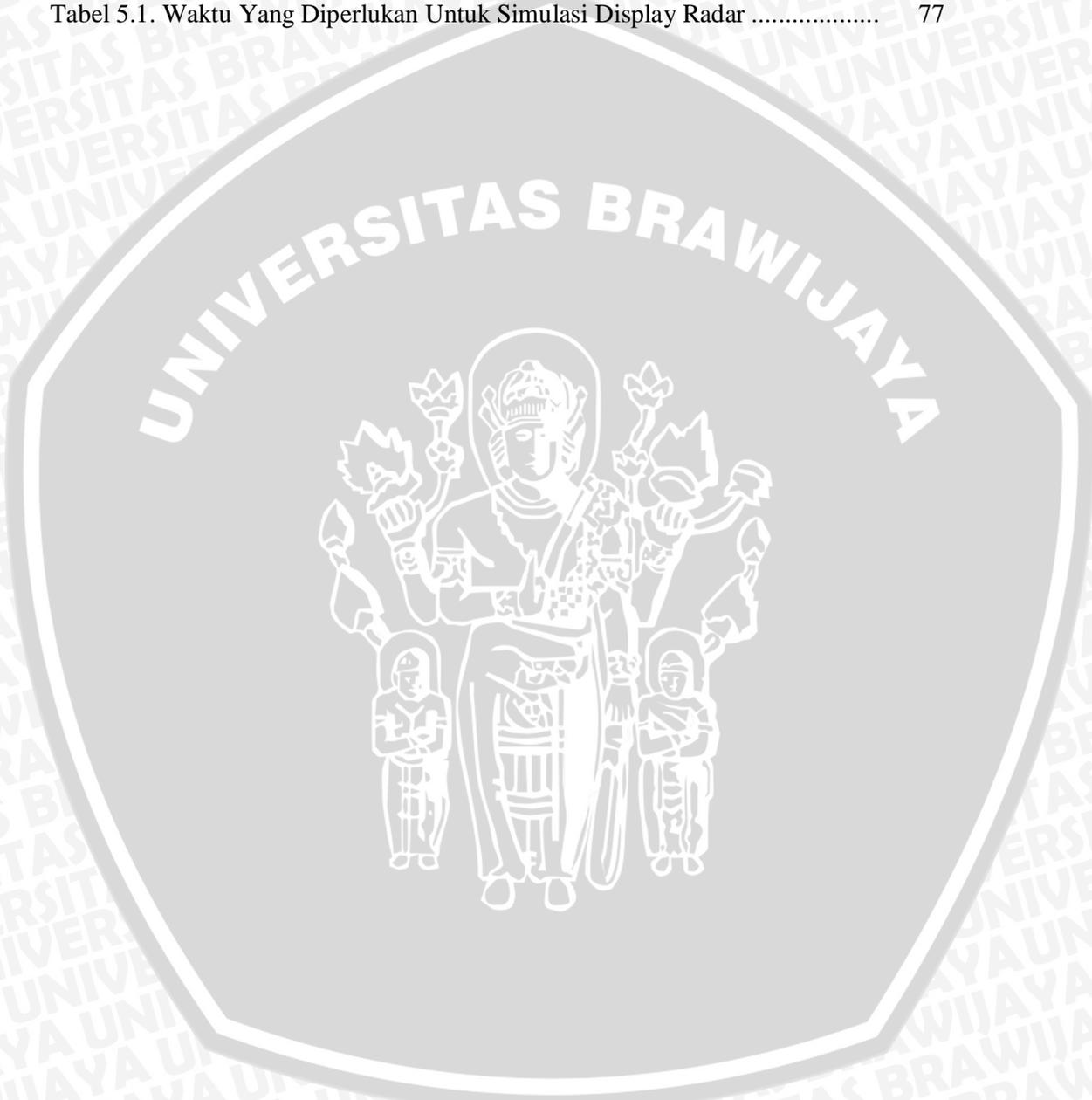
Gambar 4.5. Diagram Alir Fungsi “Jangkauan Maksimum Nautical Mile”	39
Gambar 4.6. Diagram Alir Fungsi “Jangkauan Maksimum Kilometer”	40
Gambar 4.7. Diagram Alir Program “kurangi_range”	42
Gambar 4.8. Diagram Alir Program “tambah_range”	43
Gambar 4.9. <i>Screenshot</i> GUI Sistem <i>Display</i> Radar	47
Gambar 5.1. (a) Posisi Radar; (b) Sistem Display dan Pengolah Sinyal Radar	48
Gambar 5.2. Variabel “data” Hasil <i>Sampling</i> Sinyal Radar	49
Gambar 5.3. Hasil <i>Sampling</i> Sinyal Radar	49
Gambar 5.4. Hasil Pengujian Satuan Nautical Mile.....	51
Gambar 5.5. Hasil Pengujian Satuan Kilometer	51
Gambar 5.6. <i>Radar Map</i> Pada <i>Display</i> Radar Furuno Dengan <i>Range</i> 12 NM.....	52
Gambar 5.7. Hasil Pengujian Jangkauan Radar 4 NM.....	53
Gambar 5.8. Hasil Pengujian Jangkauan Radar 6 NM.....	53
Gambar 5.9. Hasil Pengujian Jangkauan Radar 8 NM.....	53
Gambar 5.10. Hasil Pengujian Jangkauan Radar 10 NM.....	54
Gambar 5.11. Hasil Pengujian Jangkauan Radar 18 NM.....	54
Gambar 5.12. Hasil Pengujian Jangkauan Radar 24 NM.....	54
Gambar 5.13. Hasil Pengujian Jangkauan Radar 15 km	55
Gambar 5.14. Hasil Pengujian Jangkauan Radar 30 km	55
Gambar 5.15. <i>Overlay Radar Map</i> pada Foto Satelit di Wilayah Pengujian	56
Gambar 5.16. Hasil Pengujian <i>Gain Control</i> Dengan <i>Gain</i> Minimum.....	57
Gambar 5.17. Hasil Pengujian <i>Gain Control</i> Dengan <i>Gain</i> ¼ Skala.....	57
Gambar 5.18. Hasil Pengujian <i>Gain Control</i> Dengan <i>Gain</i> ½ Skala.....	58
Gambar 5.19. Hasil Pengujian <i>Gain Control</i> Dengan <i>Gain</i> ¾ Skala.....	58
Gambar 5.20. Hasil Pengujian <i>Gain Control</i> Dengan <i>Gain</i> 1 Skala Penuh	59
Gambar 5.21. Tampilan <i>Display</i> Radar Furuno Dengan <i>Gain</i> Minimum.....	60
Gambar 5.22. Tampilan <i>Display</i> Radar Furuno Dengan <i>Gain</i> ¼ Skala	60
Gambar 5.23. Tampilan <i>Display</i> Radar Furuno Dengan <i>Gain</i> ½ Skala	61
Gambar 5.24. Tampilan <i>Display</i> Radar Furuno Dengan <i>Gain</i> ¾ Skala	61
Gambar 5.25. Tampilan <i>Display</i> Radar Furuno Dengan <i>Gain</i> 1 Skala Penuh	62
Gambar 5.26. Sinyal Video pada <i>Bearing</i> 200 ⁰	63
Gambar 5.27. Sinyal Video Dan <i>Threshold</i> 0,1	64

Gambar 5.28. Sinyal Video Hasil Seleksi <i>Threshold</i> 0,1	64
Gambar 5.29. Hasil Pengujian <i>Sea Clutter Control</i> Dengan Nilai $\frac{1}{4}$ Skala	65
Gambar 5.30. Hasil Pengujian <i>Sea Clutter Control</i> Dengan Nilai $\frac{1}{2}$ Skala	66
Gambar 5.31. Hasil Pengujian <i>Sea Clutter Control</i> Dengan Nilai $\frac{3}{4}$ Skala	66
Gambar 5.32. Hasil Pengujian <i>Sea Clutter Control</i> Dengan Nilai 1 Skala	67
Gambar 5.33. Tampilan Display Radar Furuno Dengan A/C <i>Sea</i> Setengah Skala .	68
Gambar 5.34. Tampilan Display Radar Furuno Dengan A/C <i>Sea</i> Satu Skala Penuh.....	68
Gambar 5.35. Kurva <i>Sea Control</i>	71
Gambar 5.36. Hasil Pengujian <i>Rain Clutter Control</i> Dengan Nilai $\frac{1}{4}$ Skala	72
Gambar 5.37. Hasil Pengujian <i>Rain Clutter Control</i> Dengan Nilai $\frac{1}{2}$ Skala	72
Gambar 5.38. Hasil Pengujian <i>Rain Clutter Control</i> Dengan Nilai $\frac{3}{4}$ Skala	73
Gambar 5.39. Hasil Pengujian <i>Rain Clutter Control</i> Dengan Nilai $\frac{3}{4}$ Skala	73
Gambar 5.40. Tampilan <i>Display</i> Radar Furuno Dengan A/C <i>Rain</i> $\frac{1}{4}$ Skala	74
Gambar 5.41. Tampilan <i>Display</i> Radar Furuno Dengan A/C <i>Rain</i> $\frac{1}{2}$ Skala	75
Gambar 5.42. Tampilan <i>Display</i> Radar Furuno Dengan A/C <i>Rain</i> $\frac{3}{4}$ Skala	75
Gambar 5.43. Tampilan <i>Display</i> Radar Furuno Dengan A/C <i>Rain</i> 1 Skala.....	76
Gambar 5.44. Sinyal Video Hasil Penggunaan <i>Rain Clutter Control</i>	76
Gambar 5.45. Grafik Waktu Simulasi.....	78



DAFTAR TABEL

Tabel 2.1. Perbandingan Radar X-Band dan S-Band.....	9
Table 2.2. Spesifikasi Radar FURUNO 1932 MARK-2.....	21
Tabel 5.1. Waktu Yang Diperlukan Untuk Simulasi Display Radar	77



ABSTRAK

Sahirul Alam, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Juli 2013, *Rancang Bangun Sistem Pengolah Sinyal Remote Display Pada Marine Radar Menggunakan MATLAB*, Dosen Pembimbing: Dwi Fadila Kurniawan, S.T., M.T. dan Rudy Yuwono, S.T., M.Sc.

Marine radar secara umum digunakan sebagai alat navigasi juga piranti keselamatan pada transportasi laut untuk mencegah terjadinya tabrakan. Salah satu contoh *marine radar* adalah Furuno 1932 Mark-2, yang merupakan jenis radar pulsa *X-band non-doppler*. Kelemahan radar ini adalah tampilannya yang menggunakan monitor crt monokrom warna hijau untuk menampilkan target.

Pada skripsi ini akan dibahas mengenai pembuatan sistem *display* radar menggunakan GUI Matlab yang memiliki kelebihan dalam menampilkan target dengan gradasi warna berdasarkan kuat sinyal radar yang dipantulkannya. Untuk dapat menampilkan *radar map* pada sistem *display*, sinyal radar perlu diproses dalam beberapa tahapan. Pertama adalah proses *sampling* untuk mendapatkan data sinyal radar dalam bentuk diskrit. Selanjutnya data diproses dengan program pengolah sinyal radar yang meliputi *target mapping*, *gain control*, dan *clutter control* hingga menjadi citra *radar map*. Hasil akhir citra *radar map* kemudian ditampilkan pada GUI *display* radar.

Sistem *display* radar yang telah dibuat kemudian diuji dengan sinyal radar asli untuk menampilkan *radar map* dengan beberapa pengaturan, di antaranya pengaturan satuan dan jangkauan radar, *gain control*, dan *clutter control*. Radar map yang dihasilkan sistem *display* radar mirip dengan radar map pada *display* radar Furuno dan dengan gradasi warna. Waktu yang diperlukan untuk melakukan simulasi dengan sistem *display* yang telah dibuat lebih lama dibandingkan dengan *display* radar Furuno, sehingga simulasi hanya bisa dilakukan secara *non-real time*.

Kata kunci : *clutter control*, *display* radar, GUI Matlab, pemrosesan sinyal

BAB I PENDAHULUAN

1.1. LATAR BELAKANG

Radar telah menjadi instrumen yang sangat penting sejak masa Perang Dunia II. Kapal perang dan pesawat tempur dilengkapi dengan radar untuk mendeteksi keberadaan musuh. Pada zaman sekarang radar berperan penting dalam navigasi dan piranti keselamatan pada alat transportasi baik darat, laut, maupun udara. Bahkan dapat dikatakan bahwa untuk urusan keselamatan di laut, tidak ada perangkat elektronik lain di kapal yang sepenting radar. Dengan radar nahkoda dapat mengetahui kondisi di sekeliling kapalnya meliputi garis pantai, kapal, ataupun objek lainnya. Dengan demikian resiko terjadinya tabrakan dapat dihindari. Kelebihan radar adalah dapat menampilkan objek di sekeliling radar tanpa terpengaruh kondisi cuaca bahkan dalam kabut tebal sekalipun atau kondisi malam yang gelap (Wagener, 2011).

Salah satu jenis radar yang dapat digunakan oleh navigator pada kapal laut adalah radar pulsa *X-band non doppler* Furuno 1932 Mark-2. Kelemahan yang ada pada radar ini adalah sistem tampilannya yang kurang *user friendly* karena masih menggunakan monitor crt monokrom dengan 8 level kuantisasi warna hijau untuk menampilkan target membuat pengguna kesulitan membedakan target berdasarkan kuat sinyal pantulnya. Selain itu, kebutuhan pengolahan sinyal dan citra radar lebih lanjut untuk berbagai keperluan pemantauan target, misalnya pemantauan migrasi burung ataupun estimasi hujan, membutuhkan pengembangan sistem *display* baru radar berbasis komputer.

Dalam skripsi ini akan dibahas pembuatan sistem *display* radar berbasis radar pulsa *X-band* Furuno 1932 Mark-2 menggunakan MATLAB. Sistem *display* radar yang akan dibuat memerlukan sistem pengolah sinyal radar yang di dalamnya termasuk juga sistem pengurang *clutter* (*clutter control*). Sebelum memasuki tahapan pengolahan sinyal, perlu dilakukan proses *sampling* sinyal keluaran radar (dari *scanner*) untuk mendapatkan data hasil *scanning* radar. Data tersebut yang akan diolah dengan sistem pengolah sinyal yang dibuat

menggunakan Matlab. Data yang telah diolah kemudian ditampilkan pada sistem *display* radar yang dibuat dengan menggunakan fitur GUI pada Matlab.

GUI untuk sistem *display* radar dirancang tidak jauh berbeda dengan *display* radar sebenarnya. Akan tetapi tampilannya lebih disederhanakan agar mudah dioperasikan dan dengan fitur kontrol *display* yang memadai seperti pengatur jangkauan radar, *gain control*, *sea clutter control*, dan *rain clutter control*. Pada kontrol *display* juga dilengkapi dengan fitur pemilih satuan (antara Nautical Mile dan kilometer) dan *auto gain*.

Kelebihan utama pada sistem *display* radar yang akan dibuat dengan *display* radar sebenarnya adalah tampilan hasil *scanning* atau disebut juga *radar map*. Pada sistem *display* radar yang akan dibuat ini, *radar map* memiliki gradasi warna yang menunjukkan kuat sinyal pantul yang diterima radar. Warna yang digunakan adalah biru untuk sinyal yang lemah dan bergradasi sesuai kuat sinyal hingga menjadi warna merah untuk sinyal yang paling kuat. Dengan adanya gradasi warna ini diharapkan pengenalan target dapat dilakukan dengan lebih baik meskipun oleh pemakai radar yang masih awam.

1.2. RUMUSAN MASALAH

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka rumusan masalah yang akan dibahas dalam skripsi ini adalah sebagai berikut:

1. Bagaimana metode *sampling* yang digunakan untuk mendapatkan data dari sinyal keluaran radar?
2. Bagaimana cara pembuatan *display* atau visualisasi radar menggunakan MATLAB?
3. Bagaimana metode pengurangan *clutter* pada *marine radar*?
4. Bagaimana cara pengujian *remote display* radar yang dibuat dengan MATLAB menggunakan sinyal asli radar ?

1.3. TUJUAN PENELITIAN

Tujuan penelitian ini adalah merancang dan membuat sistem *display* radar menggunakan Matlab dengan kemampuan menampilkan *radar map* yang memiliki gradasi warna sesuai kuat sinyal radar yang dipantulkan oleh target.

1.4. BATASAN MASALAH

Pembahasan dalam skripsi ini dibatasi sebagai berikut:

1. Radar yang digunakan adalah radar pulsa *X-Band non doppler* FURUNO tipe 1932 MARK-2.
2. ADC yang digunakan untuk *sampling* sinyal radar adalah AD-LINK PCI-9810.
3. Perangkat keras tambahan dalam sistem berupa ADC, *filter*, dan pengkondisi sinyal tidak dibahas secara detail.
4. Tidak melakukan penghitungan atau analisis daya yang dipancarkan atau pun yang diterima oleh antena radar.
5. *Clutter* yang diperhitungkan dalam sistem radar adalah *sea clutter* dan *rain clutter*.
6. Metode pengurangan *clutter* yang digunakan adalah dengan pengaturan *threshold* sinyal yang dikendalikan dengan beberapa fitur kontrol pada sistem display radar, yaitu *Gain Control*, *Sea Clutter Control*, dan *Rain Clutter Control*.
7. Sistem *display* radar dibuat dengan memanfaatkan fitur GUI pada Matlab.
8. Lokasi radar berada di wilayah perkotaan (Bandung, Jawa Barat) dengan topografi pegunungan.
9. Pengujian dan simulasi sistem *display* radar dilakukan secara *non real time*.

1.5. SISTEMATIKA PENULISAN

Sistematika penulisan dalam skripsi ini adalah sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika penulisan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan sistem *display* radar.

BAB III Metode Penelitian

Membahas mengenai metode-metode yang digunakan dalam perancangan, pengujian, dan analisis data.

BAB IV Pembuatan Display Radar

Membahas pembuatan sistem *display* radar yang meliputi pembuatan program pengolah sinyal radar dan pembuatan GUI untuk *display* radar.

BAB V Pengujian dan Pembahasan

Membahas tata cara pengujian dan pembahasan berkaitan dengan hasil pengujian.

BAB IV Kesimpulan dan Saran

Memuat inti dari hasil pengujian dan menjawab rumusan masalah serta memberikan saran bagi penelitian selanjutnya agar lebih baik.



BAB II

TINJAUAN PUSTAKA

2.1 PRINSIP RADAR

Radar merupakan akronim dari *Radio Detection and Ranging* yaitu suatu peralatan yang digunakan untuk mengukur waktu yang diperlukan oleh suatu sinyal yang dipancarkan dan dipantulkan kembali oleh suatu objek serta sudut lokasi (*bearing*) relatif terhadap posisi radar.

Radar telah dikembangkan sejak masa Perang Dunia II. Hingga saat ini radar telah digunakan pada semua jenis kapal, mulai dari kapal pencari ikan hingga kapal pesiar. Radar berperan penting dengan kemampuannya mendeteksi objek (misal kapal) yang datang ke arah kita dalam cuaca kabut tebal, maupun menunjukkan lokasi pelabuhan dalam kondisi malam yang gelap. Dalam hal ini fungsi utama radar adalah untuk mencegah terjadinya tabrakan serta menyediakan informasi tentang keberadaan objek di sekeliling radar (misal kapal), garis pantai, dan lain-lain. Fungsi lain dari radar antara lain untuk navigasi di kegelapan maupun kabut, memantau pergerakan target, menentukan posisi kapal, navigasi untuk menuju ke suatu lokasi tertentu, navigasi untuk menentukan jalur lurus ke arah tujuan, juga untuk kegiatan memancing atau penangkapan ikan. (Furuno, 2008).

Prinsip kerja radar adalah dengan memancarkan pulsa atau gelombang radio dari *scanner* ke arah tertentu. Ketika pulsa menabrak suatu objek seperti kapal atau pulau maka sebagian energi akan kembali ke *scanner* radar. Arah di mana *scanner* menunjuk ketika pulsa pantulan diterima merupakan arah dari objek yang menyebabkan pantulan. Karena gelombang radio merambat dengan kecepatan yang mendekati konstan, maka waktu yang diperlukan pulsa pantulan untuk kembali ke *scanner* merupakan ukuran untuk menentukan jarak target.

Radar mendeteksi waktu yang diperlukan suatu pulsa dari saat dipancarkan hingga kembali ke radar karena pantulan. Namun hanya setengah dari waktu tersebut yang digunakan untuk menentukan jarak dari suatu target. Perhitungan jarak target ditunjukkan pada persamaan 2.1 berikut:

$$R = \frac{c\Delta t}{2} \quad (2.1)$$

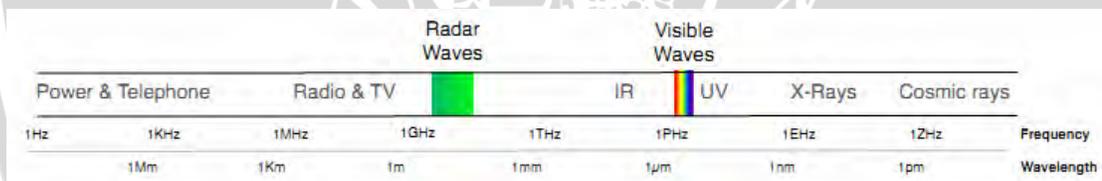
dimana R = jarak target dari lokasi radar (m)

c = kecepatan gelombang radar $3 \cdot 10^8$ m/s

Δt = selisih waktu antara saat pengiriman pulsa dan pulsa diterima (s)

Baik gelombang radio maupun cahaya merambat dengan kecepatan yang mendekati konstan yaitu 186.000 mil perdetik. Karena itu radar dapat memproses sejumlah besar informasi dalam waktu yang sangat singkat. Sebagai perbandingan yaitu *Sonar and Fish Finder* yang menggunakan gelombang ultrasonik yang kecepatan propagasinya hanya 1500 mil perdetik, maka pemrosesan sinyalnya menjadi lebih lambat jika dibandingkan dengan radar. (Furuno, 2008).

Menurut Rudolf F. Graf dalam *Modern Dictionary of Electronics*, radar menggunakan gelombang radio dengan frekuensi antara 1,55 sampai 5,2 GHz (gambar 2.1). Frekuensi ini cukup tinggi sehingga tidak terhalang oleh asap ataupun kabut, tetapi cukup rendah sehingga dapat dipantulkan oleh sebagian besar objek solid pada umumnya.



Gambar 2.1. Lokasi Gelombang Radar dalam Spektrum Elektromagnetik

Sumber: Wagener, 2011 (dari Rudolf F. Graf)

Lokasi target dilihat dari sudutnya relatif terhadap posisi radar atau disebut dengan *bearing* ditentukan dengan arah dari pantulan pulsa yang kembali ke radar.

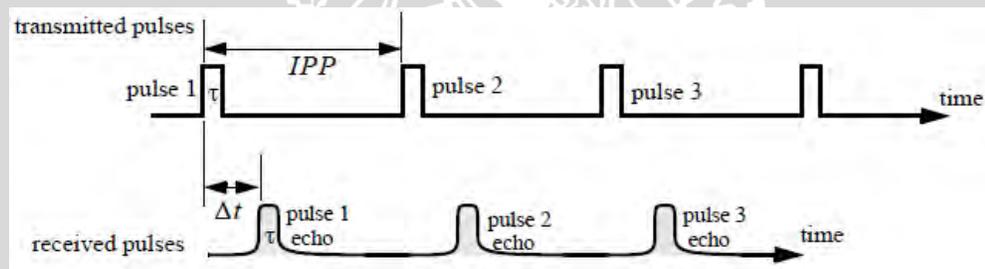
Scanner radar berotasi 360^0 . Pusat rotasi terletak di tengah atau sebagai pusat pada tampilan *plotter* radar. Untuk mendapatkan resolusi *bearing* dengan ketelitian yang bagus, antena memancarkan daya RF (*Radio Frequency*) dengan pola *directional beam*. “*Super*” *beam* mempunyai *horizontal beamwidth* dalam orde 1 derajat atau kurang, sehingga menghasilkan informasi

bearing dengan ketelitian yang cukup tinggi. Semakin tajam *beam*, maka *bearing* target dapat ditentukan dengan lebih akurat.

Target radar ditunjukkan dalam suatu *display* yang disebut *Plan Position Indicator* (PPI). Display ini menggunakan diagram polar dengan posisi radar berada di tengah atau pusat diagram. (Furuno, 2008).

2.2 SINYAL RADAR PULSA

Radar pulsa mengirimkan sinyal/pulsa dari antena radar dan menerima pantulan pulsa tersebut dari objek yang ingin dideteksi. Proses mengirim dan menerima pulsa ini berlangsung secara bergantian. Selisih waktu antara proses pengiriman pulsa dengan penerimaan pulsa ini yang digunakan untuk menghitung jarak objek/target dari radar. Gambar pulsa yang dikirim dan pantulan sinyal yang diterima antena radar diilustrasikan pada gambar 2.2.



Gambar 2.2. Pulsa yang dikirim dan diterima oleh radar.

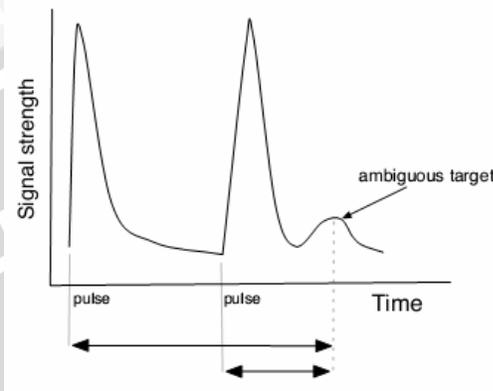
Sumber: Awaludin, 2013

Pada gambar 2.2, *IPP* atau *Inter Pulse Periode* adalah periode pulsa radar, τ adalah lebar pulsa, dan Δt adalah selisih waktu antara pengiriman pulsa dengan penerimaan pantulan pulsa.

IPP berkaitan dengan *PRF* (*Pulse Repetition Frequency*) dengan hubungan seperti pada persamaan 2.2:

$$PRF = \frac{1}{IPP} \quad (2.2)$$

Berdasarkan gambar 2.2, maka Δt tidak boleh melebihi IPP. Karena jika melebihi IPP maka akan ada pulsa pantulan pada periode pulsa berikutnya (gambar 2.3). Hal ini akan memunculkan target yang ambigu yang dapat menyebabkan kesalahan pendeteksian target oleh radar, dimana target yang jauh akan muncul pada *display* radar dengan jarak yang dekat.



Gambar 2.3. Pantulan pulsa pertama berada pada periode pulsa kedua.

Sumber: Wagener, 2011

Dengan demikian IPP dapat digunakan sebagai parameter untuk menentukan jarak maksimum radar (R_{max}). Dengan menggunakan persamaan 2.1 dan memasukkan nilai IPP ke dalam Δt (dalam hal ini Δt maksimum = IPP) maka akan didapat jarak maksimum radar.

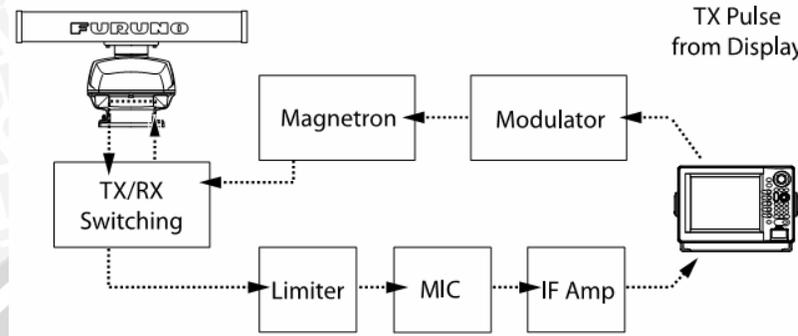
Sedangkan untuk mengetahui jarak minimum yang dideteksi radar dapat menggunakan parameter τ (lebar pulsa). Dengan cara seperti ketika mencari jarak maksimum radar, dengan memasukkan nilai τ ke dalam Δt (dalam hal ini Δt minimum = τ) maka akan didapat jarak minimum radar. Jarak minimum ini juga merupakan resolusi jarak radar.

2.3 KONFIGURASI SISTEM RADAR

Sistem radar dasar terdiri dari dua unit, yaitu *scanner unit* dan *display unit*. *Transceiver* (*transmitter/receiver unit*, atau t/r) biasanya ditempatkan dalam suatu wadah atau *gearbox* yang termasuk dalam *scanner unit*.

2.3.1 Komponen Scanner Unit

Sebagian besar *scanner unit* tersusun dari beberapa rangkaian dan perangkat seperti pada gambar 2.4.



Gambar 2.4. Rangkaian dan perangkat penyusun *scanner unit*.

Sumber: Furuno, 2008

Magnetron

Magnetron berfungsi membangkitkan pulsa radio. *Magnetron*, sebagaimana radar itu sendiri, dikelompokkan berdasarkan pita frekuensi transmisinya. Ada dua pita frekuensi utama untuk radar komersial, yaitu *X-Band* (9000 MHz band, panjang gelombang 3 cm) dan *S-Band* (3000 MHz band, panjang gelombang 10 cm). Daya output *magnetron* berkisar dari 1 kW untuk radar kecil hingga 60 kW untuk radar besar. Tabel 2.1 menunjukkan perbandingan frekuensi *S-Band* dan *X-Band*.

Tabel 2.1. Perbandingan Radar *X-Band* dan *S-Band*

Sumber: Furuno, 2008

Frekuensi	Karakteristik
<i>X-Band</i>	<ul style="list-style-type: none"> - Panjang gelombang pendek untuk direktivitas yang lebih baik - Attenuasi/redaman dalam hujan lebih besar dibanding pada <i>S-Band</i> - Ukuran antena kecil dan ringan
<i>S-Band</i>	<ul style="list-style-type: none"> - Panjang gelombang lebih panjang untuk pendeteksian jarak jauh - Mampu menembus hujan dengan baik sehingga sesuai untuk komunikasi dalam cuaca buruk - Ukuran antena besar

Modulator

Modulator berperan untuk mengatur *magnetron*, memastikan *magnetron* bekerja sebagaimana mestinya dan mentransmisikan pulsa dengan frekuensi yang tepat sama pada keseluruhan durasi pulsa, serta waktu antar pulsa memiliki durasi yang tepat.

TX/TR Switching

Perangkat TX/RX *switching* memungkinkan radar untuk mentransmisikan pulsa radio dan menerima pulsa pantulannya melalui satu *scanner*. Perangkat *switching* yang digunakan oleh radar disebut dengan *circulator*, yang terdiri dari sebuah magnet permanen dan sebuah inti besi. Saat radar mengirim sinyal, ia menghubungkan pembangkit gelombang dengan antena *scanner* dan memutuskannya dengan rangkaian penerima. Saat radar menerima sinyal, ia menghubungkan antena *scanner* dengan rangkaian penerima dan memutuskannya dengan pembangkit gelombang.

Scanner

Scanner mentransmisikan pulsa radio dan menerima pulsa pantulannya. Sebagian besar *scanner* berotasi dengan kecepatan konstan sebesar 24 rpm. *Scanner* mempunyai direktivitas yang merupakan ukuran dari dua *beamwidth* (horizontal dan vertikal). Semakin sempit horizontal *beamwidth* maka *beam* semakin tajam. Vertikal *beamwidth* seharusnya lebar, biasanya 20 sampai 25 derajat. Alasan utama vertikal *beamwidth* berukuran lebar adalah untuk memastikan radar masih mampu menampilkan target saat kapalnya sendiri terombang-ambing oleh ombak.

Limiter

Limiter melindungi rangkaian penerima dari kerusakan ketika radar menerima pulsa radio dari radar lain. Ketika ini terjadi, *limiter* akan melemahkan pulsa tersebut untuk melindungi bagian rangkaian selanjutnya yaitu MIC.

MIC

MIC adalah singkatan dari *Microwave Integrated Circuit*. MIC terdiri dari *oscillator* lokal dan rangkaian *mixer*. Kolaborasi kedua piranti berfungsi meningkatkan kualitas, reliabilitas, sensitivitas, dan *noise figure* (nf) radar.

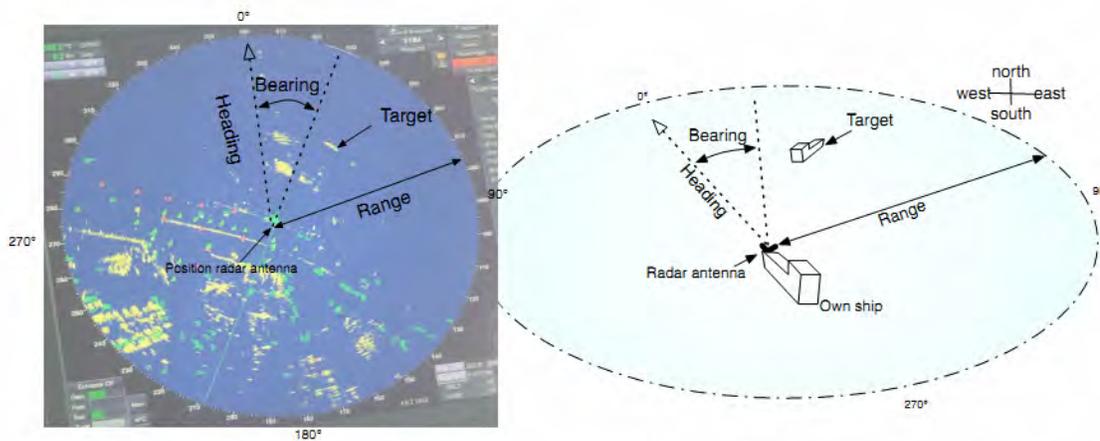
IF Amplifier

IF *amplifier* menguatkan sinyal output dari MIC pada frekuensi intermediet. (Furuno, 2008).

2.3.2 Komponen Display Unit

Display unit adalah bagian radar yang menampilkan peta kondisi di sekeliling radar, dengan pusat peta merupakan posisi radar. Jarak pada peta dari pusat hingga ke pinggir peta disebut dengan *range* (gambar 2.5). Semakin besar *range* maka skala peta pada radar semakin besar dan peta terlihat semakin *zoom out*. Besarnya *range* dapat diatur oleh operator radar tergantung pada situasi, meskipun *range* dibatasi oleh jangkauan maksimum antena radar.

Sebagian besar radar modern menggunakan *background* warna biru dan menampilkan objek dalam *range* radar dengan warna kuning. Semua objek yang penting/disorot pada display radar dan ada dalam kehidupan nyata serta ditampilkan pada posisi yang tepat, dikenal dengan sebutan *target*. Objek lain selain yang pokok/disorot disebut dengan *clutter*. *Clutter* dapat membingungkan operator dalam mengenali objek pada radar yang benar-benar ada dalam kehidupan nyata. Karena itu kemunculan *clutter* sebisa mungkin harus dikurangi.



(a) Display Radar

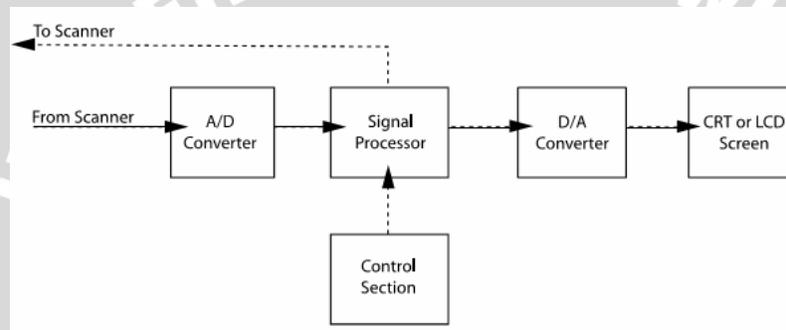
(b) Kondisi Nyata

Gambar 2.5. *Heading, bearing, dan range* pada *display radar*

Sumber: Wagener, 2011

Arah pada *display* radar dikenal dengan sebutan *bearing* dan diukur dalam derajat. *Bearing* adalah relatif terhadap haluan kapal. *Bearing* 0^0 berarti arah lurus ke depan, sedangkan *bearing* 90^0 berarti arah samping kanan dari kapal. *Bearing* digunakan oleh navigator untuk mengidentifikasi kapal dan memperhitungkan jika ada kapal lain yang kemungkinan memotong jalur atau akan menabrak. *Heading* dari kapal adalah arah haluan kapal relatif terhadap arah utara. Jika sebuah kapal mempunyai *heading* 0^0 , maka kapal tersebut berhaluan tepat ke utara, dan jika kapal mempunyai *heading* 90^0 maka haluan kapal ke arah timur.

Display unit umumnya tersusun dari perangkat seperti pada gambar 2.6.



Gambar 2.6. Perangkat penyusun *Display Unit*

Sumber: Furuno, 2008

A/D Converter

Sinyal IF yang diterima merupakan sinyal analog. Sinyal ini dikonversi menjadi sinyal digital agar dapat melalui beberapa proses pada *display unit*. A/D (*Analog to Digital*) Converter mengubah sinyal analog menjadi sinyal digital.

Signal Processor

Bagian ini merupakan inti dari radar yang terdiri dari komputer, memori, dan IC yang lain. Dengan teknik digital yang maju ini memungkinkan pengolahan sinyal yang cepat.

Control Unit

Control unit terdiri dari beberapa macam kunci dan kontrol untuk mengatur gambar yang ditampilkan radar.

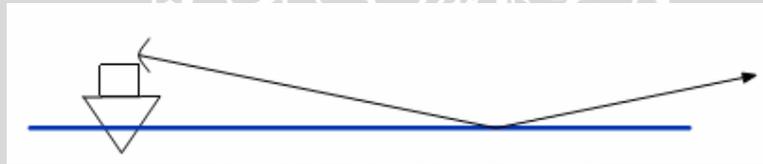
2.4 CLUTTER

Clutter adalah pantulan sinyal radar yang tidak diinginkan yang muncul dalam display radar. *Clutter* bisa disebabkan oleh pantulan sinyal dari lautan, daratan, hujan, hewan, atau objek lain yang tidak diinginkan. *Clutter* dapat mengurangi performansi radar, bahkan menimbulkan gangguan serius dalam sistem radar.

2.4.1 Sea Clutter

Sinyal radar dapat dipantulkan oleh sebagian besar objek solid yang ditabraknya bahkan oleh permukaan laut sekalipun. Pantulan sinyal dari permukaan laut ini tentu saja tidak diharapkan, karena jika sinyal pantul tersebut diterima oleh radar maka hampir seluruh *display* radar akan berwarna kuning sehingga sulit untuk melihat target pada display radar.

Untungnya karena bentuk permukaan laut rata dan tidak berhadapan langsung dengan *beam* antena, maka sebagian besar sinyal radar yang dipantulkan oleh permukaan laut tidak kembali ke antena (gambar 2.7).



Gambar 2.7. Sinyal radar dipantulkan permukaan laut.

Sumber: Wagener, 2011

Pada kondisi berangin, pada permukaan laut akan terdapat ombak. Jika permukaan ombak cukup curam, maka sinyal radar dapat dipantulkan oleh ombak kembali ke antena radar (gambar 2.8). Efek pemantulan ini yang dikenal dengan *sea clutter* dan muncul seperti ring *clutter* di sekitar pusat *display* radar (gambar 2.9). Efek *clutter* ini paling banyak terlihat di dekat pusat *display* radar dan semakin sedikit jika semakin jauh dari pusat display radar tergantung besarnya gelombang di permukaan laut. Namun umumnya semakin jauh dari radar maka semakin kecil/dangkal gelombang tersebut bagi sinyal radar untuk bisa memantulkan sinyal radar kembali ke antena.



Gambar 2.8. Sinyal radar dipantulkan kembali oleh ombak di dekat posisi radar

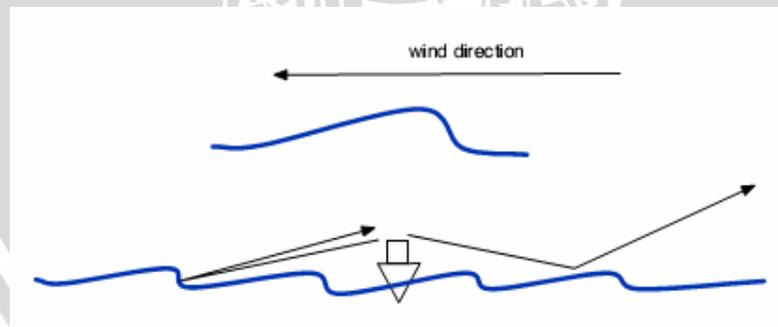
Sumber: Wagener, 2011



Gambar 2.9. Ombak laut terlihat pada display radar jaman dulu

Sumber: Wagener, 2011

Arah angin sangat mempengaruhi *sea clutter*. Sea clutter akan terlihat lebih mencolok pada arah kemana angin bertiup. Hal ini disebabkan karena angin mempengaruhi bentuk dari ombak/gelombang (gambar 2.10).



Gambar 2.10. Angin mempengaruhi bentuk ombak dan pemantulan sinyal radar

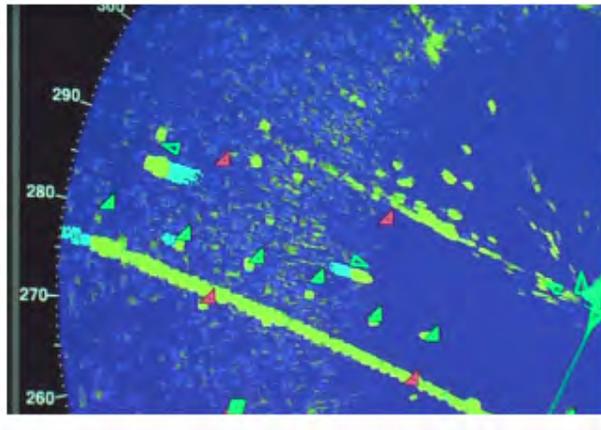
Sumber: Wagener, 2011

Angin mendorong ombak ke satu arah, yang menimbulkan permukaan curam pada satu sisi dan permukaan lebih dangkal pada sisi yang lain. Sisi yang

curam memantulkan sinyal radar kembali ke antena dengan lebih baik sehingga membuat *sea clutter* terlihat lebih banyak pada *display* radar.

2.4.2 Rain Clutter

Tak hanya air di lautan yang dapat memantulkan sinyal radar, tetapi juga air di udara. Air hujan mempunyai permukaan yang sangat kecil dan memantulkan hanya sebagian kecil dari sinyal radar. Tetapi jika air hujan muncul dalam jumlah besar (misalnya saat hujan lebat), maka jumlah sinyal radar yang bisa dipantulkan kembali ke antena akan signifikan sehingga bisa terlihat pada *display* radar (gambar 2.11).



Gambar 2.11. Bintik-bintik *clutter* menandakan adanya hujan lokal

Sumber: Wagener, 2011

Rain clutter pada *display* radar dapat dikenali sebagai *clutter* dengan pola berbintik samar yang senantiasa berubah yang membuat target real menjadi terlihat kabur. Meskipun kemampuan radar untuk mendeteksi butir-butir air hujan dapat digunakan pada aplikasi lain (misal radar cuaca) namun pada sistem *marine radar* hal ini tidak diinginkan. Efek *rain clutter* dapat dihilangkan sebagian dengan fitur *rain clutter reduction* pada sistem radar.

2.5 KONTROL *DISPLAY* RADAR

Ada beberapa kontrol *display* radar yang digunakan untuk mengatur pemrosesan *image* pada *display* radar. Di antaranya adalah *Gain*, *Rain*, dan *Sea Control*.

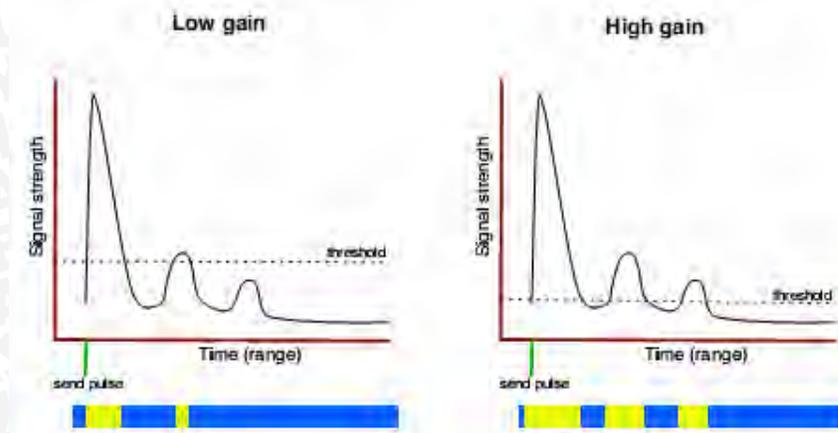
2.5.1 *Gain Control*

Pada awal kemunculan radar, ada *amplifier* tambahan yang diletakkan antara antena dan *display* radar yang dapat menguatkan sinyal agar lebih banyak target dan *clutter* yang terlihat. *Gain* (penguatan) yang dihasilkan oleh *amplifier* ini dapat diatur oleh operator radar untuk mendapatkan *image* yang memuaskan yang memperlihatkan semua target, namun dengan *clutter* sesedikit mungkin.

Pada sistem radar modern, *amplifier* telah digantikan dengan *digital signal processing*, tetapi masih menyisakan *gain control*. *Gain* di sini berbeda dengan gain antena (Wagener, 2011).

Pada saat ini *gain control* berfungsi untuk mengatur *threshold* yang digunakan untuk menentukan apakah sebuah *pixel* harus diwarnai biru (tidak ada apa-apa) atau kuning (target). Dari sinyal yang diterima antena, *display* harus menentukan sinyal mana yang cukup kuat untuk digambarkan pada *display* radar. Penentuan ini dilakukan dengan *threshold*. Semua nilai di atas *threshold* akan digambarkan sebagai target pada *display* sedangkan nilai di bawahnya tidak.

Dengan *gain control*, operator radar dapat mengatur *threshold* yang memungkinkan lebih sedikit ataupun lebih banyak target yang terlihat (gambar 2.12). Semakin tinggi gain, maka *threshold* menjadi semakin rendah. Nilai *gain* yang tinggi membuat *display* radar menjadi “semakin kuning”, karena semakin banyak target dan *clutter* yang berada di atas *threshold*. Operator radar berusaha sebaik mungkin untuk mengatur *gain* agar kapal-kapal di sekitar terlihat jelas, tetapi menyisakan *clutter-clutter* di bawah *threshold*.



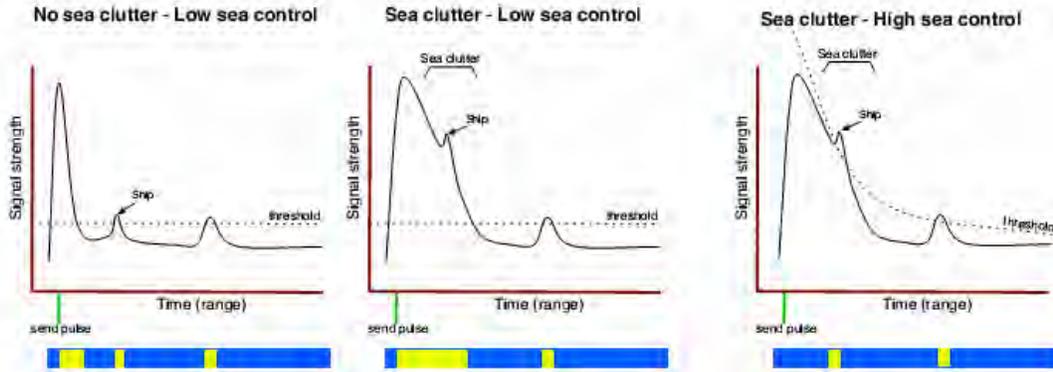
Gambar 2.12. Nilai *Gain* yang tinggi (*threshold* semakin rendah) memungkinkan target serta *clutter* dengan sinyal lemah menjadi terlihat

Sumber: Wagener, 2011

2.5.2 Sea Clutter Control

Sebagaimana telah dijelaskan sebelumnya bahwa *sea clutter* merepresentasikan pantulan dari ombak di sekitar kapal yang cukup besar sehingga muncul dalam *display* radar. Pantulan ini dapat membuat target menjadi terlihat samar karena bercampur dengan *clutter*. Dengan *sea clutter control*, operator radar dapat menghilangkan sebagian besar *clutter* dan membuat hanya target real yang ditampilkan.

Sea clutter dalam *display* radar dapat dikenali dengan kemunculannya yang paling banyak di area dekat radar dan semakin berkurang dengan bertambahnya jarak. Hal inilah yang dijadikan dasar *sea clutter control* untuk mengurangi hanya pantulan yang dekat dengan radar. Seperti halnya *gain control*, *sea control* mengatur *threshold line* yang menentukan jika sebuah sinyal harus digambarkan dalam *display* radar. Tetapi tidak dengan mengatur ketinggian dari seluruh *threshold line*, *sea control* mengatur ketinggian *threshold* bergantung pada jarak (gambar 2.13).



Gambar 2.13. Nilai *sea control* yang tinggi memungkinkan target yang berada di antara *clutter* dapat terlihat lagi.

Sumber: Wagener, 2011

Kurva yang digunakan pada *sea clutter control* adalah *inverse exponential* dimana kontrol akan mempengaruhi nilai eksponen. Hal ini dapat dirumuskan dengan persamaan 2.3:

$$T = C \frac{t}{R^s} \tag{2.3}$$

dimana T = nilai baru dari *threshold* t pada jarak R

t = nilai *threshold* asal

R = jarak terhadap radar atau waktu dalam grafik *threshold*

C = konstanta yang nilainya berbeda pada tiap sistem radar

s = *sea control*

Minimum nilai *sea control* biasanya 0, yang membuat nilai *threshold* tetap/sama. Sedangkan nilai maksimumnya berbeda pada tiap sistem radar. (Wagener, 2011).

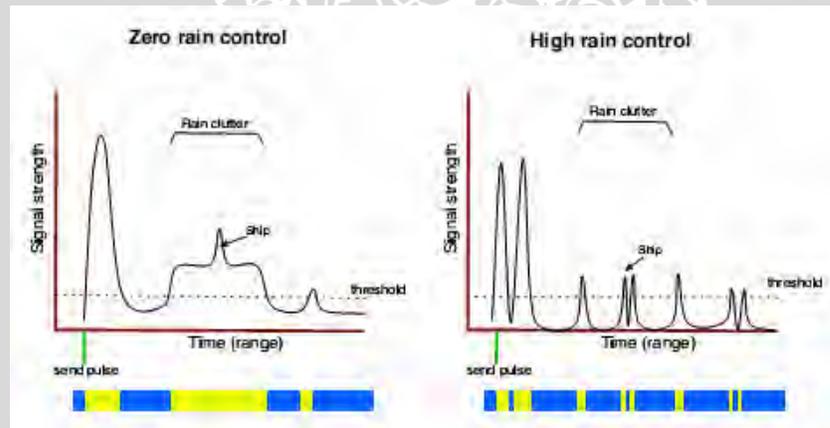
2.5.3 Rain Clutter Control

Rain clutter sebagaimana dijelaskan sebelumnya dikenali sebagai area dalam *display* radar dimana ada lebih banyak pantulan yang membuat target real menjadi samar. Hal ini biasanya terjadi ketika ada hujan lokal sehingga sinyal radar dipantulkan oleh hujan tersebut.

Gain control seringkali sulit untuk diterapkan pada kondisi seperti ini. Nilai gain yang rendah akan memunculkan target real dalam hujan, tetapi pada saat yang bersamaan target yang lemah yang berada di luar area hujan akan ikut terlihat. Hal yang diinginkan adalah nilai *gain* dibuat rendah hanya pada area hujan saja. *Rain control* mencoba mengatasi permasalahan ini dengan mengambil nilai absolut dari turunan sinyal. Misalkan *threshold* asli dinyatakan dengan fungsi $f(x)$, maka *threshold* dengan penerapan *rain control* yang dinyatakan dengan $f_r(x)$ didapatkan dengan persamaan 2.4.

$$f_r = |f'(x)| \quad (2.4)$$

Kurva turunan ini menunjukkan puncak ketika kurva asal mengalami kenaikan atau penurunan dan tetap nol ketika kurva asal tetap/konstan (seperti contoh pada gambar 2.14). Hasilnya adalah area yang luas dari *clutter* yang seragam hanya muncul di bagian awal dan akhir saja. Target yang berada di dalam area *clutter* dapat terlihat jelas lagi.



Gambar 2.14. Dengan mengambil nilai absolut dari turunan sinyal, maka target di antara clutter dapat terlihat.

Sumber: Wagener, 2011

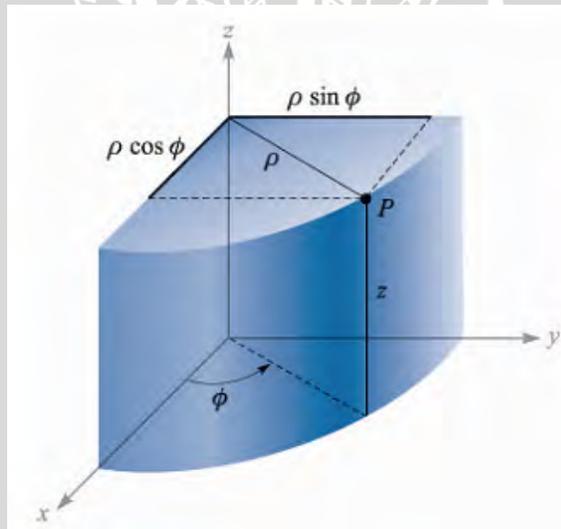
Rain control sendiri dapat diatur dari *fully on* hingga *fully off*. Meskipun *rain control* berguna, namun juga mempunyai kerugian. Dengan penerapan *rain control* dapat membuat pantulan dari target yang besar ataupun pantulan dengan intensitas yang meningkat bertahap akan terlihat lemah. Karena itu dalam praktek *rain control* tidak digunakan sesering *gain* dan *sea control*.

2.6 KONVERSI SISTEM KOORDINAT

Sistem koordinat yang paling umum digunakan adalah koordinat kartesian. Tetapi pada sistem radar, koordinat yang digunakan adalah koordinat polar (versi dua dimensi dari koordinat silinder). Untuk mengubah komponen koordinat polar (jarak dan sudut) menjadi komponen koordinat kartesian (sumbu-x dan sumbu-y) adalah dengan menggunakan persamaan 2.5.

$$\begin{aligned}x &= \rho \cos \phi \\y &= \rho \sin \phi \\z &= z\end{aligned}\quad (2.5)$$

Gambar 2.15 menunjukkan koordinat silinder dengan hubungan antara komponen koordinat silinder ρ , ϕ , z dengan komponen koordinat kartesian x , y , z .



Gambar 2.15. Koordinat Silinder

Sumber: Hayt, 2001

2.7 RADAR PULSA X-BAND NON DOPPLER FURUNO 1932 MARK-2

Radar FURUNO 1932 MARK-2 merupakan radar pulsa *non doppler* yang memiliki transmisi frekuensi pada rentang *X-band* dengan panjang gelombang 3,18 cm. Bentuk radar dan tampilan *plotter*-nya ditunjukkan dalam gambar 2.16. Spesifikasi radar tersebut ditunjukkan dalam tabel 2.2.



Gambar 2.16. Radar FURUNO 1932 MARK-2, (a) scanner unit; (b) plotter radar

Sumber: Furuno.com

Table 2.2. Spesifikasi Radar FURUNO 1932 MARK-2

Tipe radar	Radar X Band (9410±30 M)
Power Output	4 kW
Jangkauan	0,025 – 88,896 km (48 NM)
Kecepatan rotasi	24 rpm
Jenis antena	3.5 ft centre-fed waveguide slotted array
Beamwidth	Vertical: 27 ⁰ , Horizontal: 2,4 ⁰
Pulselength	0.08µs/2100Hz, 0.3µs/1200Hz, 0.8µs/600Hz
Polarisasi	Horizotal
IF	60 MHz. Bandwidth 25 MHz (0.08/0.3 µs) 3MHz (0.8µs)
Sistem Indikasi	PPI daylight display, monokrom dengan 8 level kuantiasasi

Sinyal keluaran radar pulsa Furuno terdiri empat sinyal, yaitu sinyal *trigger*, *video*, *bearing*, dan *heading*. Sinyal *trigger* merupakan sinyal pulsa dengan tegangan +12 V yang menandakan ditransmisikannya sinyal radar, sedangkan sinyal *video* merupakan sinyal pantul *baseband* dari target yang

diterima oleh radar dengan tegangan amplitudo antara 0 sampai -4 V. Sinyal *bearing* juga merupakan sinyal pulsa +12 V yang menandakan sudut arah berkas sinyal antenna radar, banyaknya sinyal ini adalah 360-450 pulsa persatu putaran antenna. Sinyal *heading* merupakan sinyal pulsa -12 V yang menentukan titik 0° arah putaran antenna yang terjadi pada tepi turun sinyal (Awaludin, 2013).

2.8 A/D CONVERTER ADLINK PCI-9810

ADLINK PCI-9810 merupakan salah satu jenis ADC yang digunakan untuk aplikasi pengolahan sinyal yang membutuhkan kecepatan *sampling* sangat tinggi. ADC ini memiliki resolusi 10 bit, dengan kecepatan *sampling* maksimal 20 MS/s, dan menggunakan *slot* PCI pada PC untuk aplikasinya. *Range* sinyal inputnya ± 5 V, ukuran *buffer* FIFOnya 32 k *sample*, dan mempunyai mode *trigger software*, analog (CH0 – CH3), dan digital (*trigger* eksternal). (Awaludin, 2013). Bentuk ADC ini seperti ditunjukkan dalam gambar 2.17.

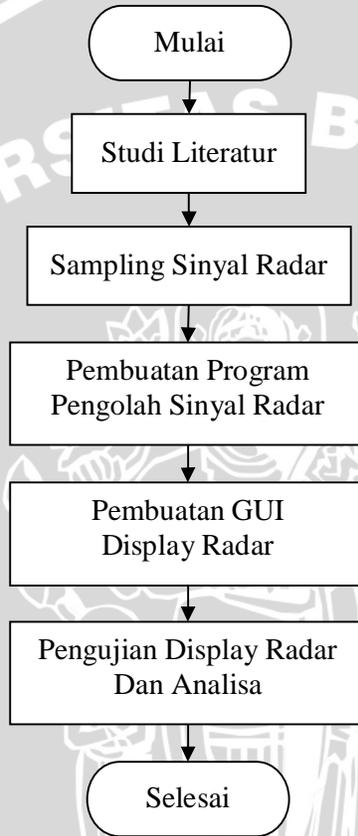


Gambar 2.17. ADC ADLINK PCI-9810.

Sumber: PCI 9812+9810 Datasheet

BAB III
METODE PENELITIAN

Kajian dalam skripsi ini bersifat perancangan dan simulasi sistem *display* pada *marine radar*. Diagram alir metode penelitian dalam skripsi ini ditunjukkan pada gambar 3.1.



Gambar 3.1. Diagram Alir Metode Penelitian

Rincian metode seperti yang ditunjukkan pada gambar 3.1 adalah sebagai berikut:

3.1. STUDI LITERATUR

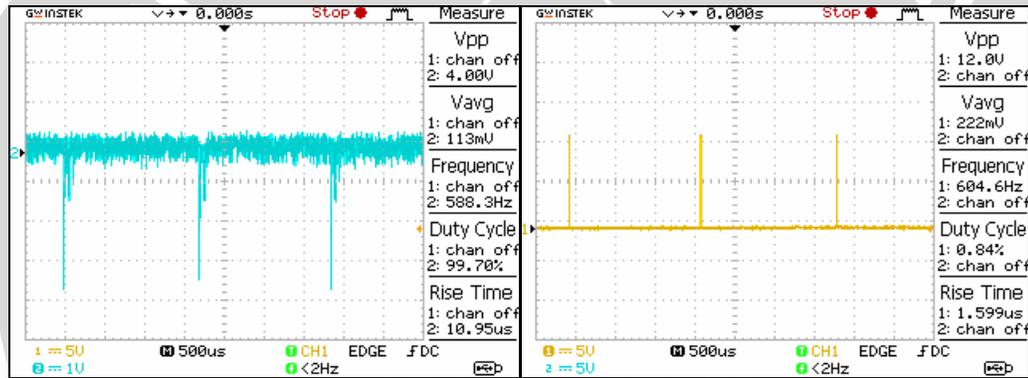
Segala konsep tentang *marine radar* yang diperlukan untuk menyusun skripsi ini diperoleh melalui studi pustaka. Konsep tersebut antara lain:

1. Prinsip radar dan konfigurasi sistemnya.

2. Spesifikasi *marine radar* (dalam hal ini FURUNO *Marine Radar* Tipe 1932 MARK-2) meliputi lebar pulsa, *pulse repetition rate*, kecepatan rotasi antena, *display unit*, dll.
3. Bentuk sinyal keluaran FURUNO *Marine Radar* Tipe 1932 MARK-2.
4. Metode *sampling* sinyal radar.
5. Pembuatan GUI dengan MATLAB.

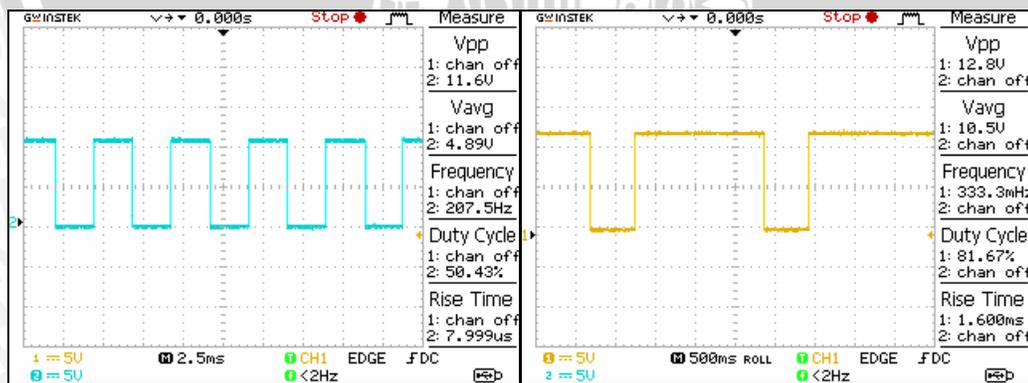
3.2. SAMPLING SINYAL RADAR MENGGUNAKAN ADC

Sinyal keluaran radar FURUNO terdiri dari empat buah yaitu sinyal video, *trigger*, *bearing*, dan *heading*. Tampilan keempat sinyal ini oleh osiloskop diperlihatkan pada gambar 3.1.



(a)

(b)



(c)

(d)

Gambar 3.2. Tampilan sinyal keluaran radar menggunakan osiloskop,

(a).Video, (b).Trigger, (c).Bearing, (d).Heading

Sumber: Awaludin, 2013

Seperti terlihat dalam gambar 3.2, sinyal *trigger*, *bearing*, dan *heading* berupa sinyal pulsa dengan amplitudo 12 Vpp. Ketiga sinyal ini perlu dikonversi amplitudonya menjadi 5 V agar dapat disampling menggunakan ADC ADLINK PCI-9810 yang mempunyai *range* input -5 V sampai +5V.

Sinyal video seperti pada gambar 3.2.a memiliki *noise* yang perlu dihilangkan terlebih dahulu sebelum dilakukan proses *sampling*. *Noise* pada sinyal video dapat dihilangkan dengan menggunakan *Low Pass Filter*.

Tujuan *sampling* adalah untuk mengubah sinyal radar menjadi data diskrit agar dapat diproses menggunakan komputer (program Matlab).

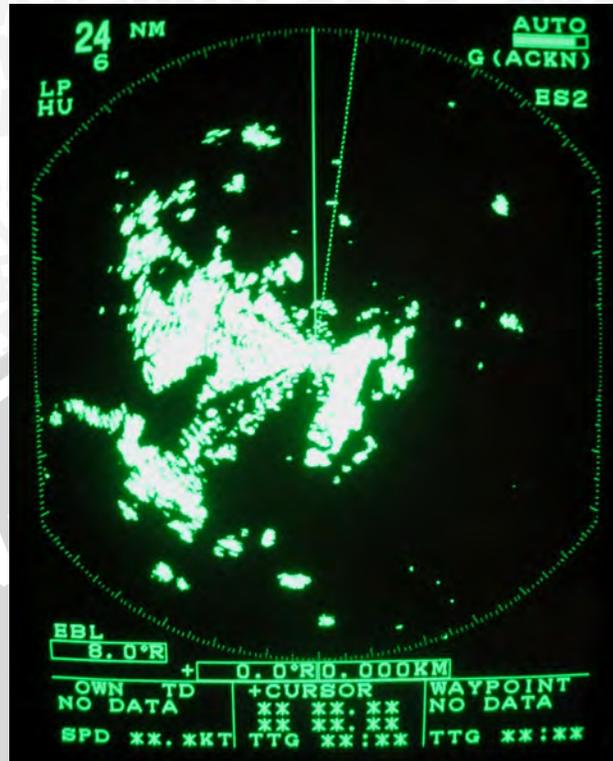
Proses *sampling* dilakukan dengan menghubungkan sinyal video, *trigger*, dan *bearing* pada CH0, CH1, dan CH2 ADLINK PCI-9810, sedangkan sinyal *heading* dihubungkan ke *external digital trigger*-nya. Proses *sampling* pada ADLINK PCI-9810 yang dipasang pada slot PCI komputer dikendalikan oleh perangkat lunak Matlab yang mempunyai *Data Acquisition Toolbox Adapter* DAQ-MTLB untuk ADLINK PCI-9810. Frekuensi *sampling* yang digunakan adalah 1,25 MHz dan jumlah sampel 3 juta.

3.3. PEMBUATAN PROGRAM PENGOLAH SINYAL RADAR

Hasil *sampling* sinyal radar berupa dua buah data, yaitu sinyal video dan sinyal *trigger* yang masing-masing dalam bentuk diskrit. Keduanya dihimpun dalam satu buah matrik yang diberi nama “data”.

Data hasil *sampling* tersebut kemudian akan diolah agar dapat ditampilkan menjadi grafik *radar map*, yaitu gambar hasil *scanning* radar seperti yang terlihat pada *display/plotter* radar sebenarnya (gambar 3.3).

Untuk mengolah data radar hingga menjadi grafik yang akan ditampilkan pada sistem *display*, maka perlu dibuat program khusus untuk mengolah data radar tersebut. Pada program yang akan dibuat itu juga akan ditambahkan proses untuk pengurangan *clutter*.



Gambar 3.3. Contoh Tampilan *Radar Map* pada Radar Furuno

Clutter yang diperhitungkan dalam skripsi ini adalah *sea clutter* dan *rain clutter*. Pengurangan *clutter* dilakukan dengan memanipulasi sinyal video dalam hal ini matriks “praz” sebelum dibandingkan dengan *threshold*. Teknik pengurangan *clutter* selanjutnya disebut *sea clutter control* dan *rain clutter control*.

Sea clutter control meredam *sea clutter* dengan mengurangi amplitudo sinyal video pada jarak dekat (dimana *clutter* paling banyak di lokasi ini), kemudian pengurangan amplitudo diperkecil secara bertahap seiring dengan peningkatan jarak, hingga tidak terjadi pengurangan amplitudo pada jarak dimana *sea clutter* sudah tidak ada lagi.

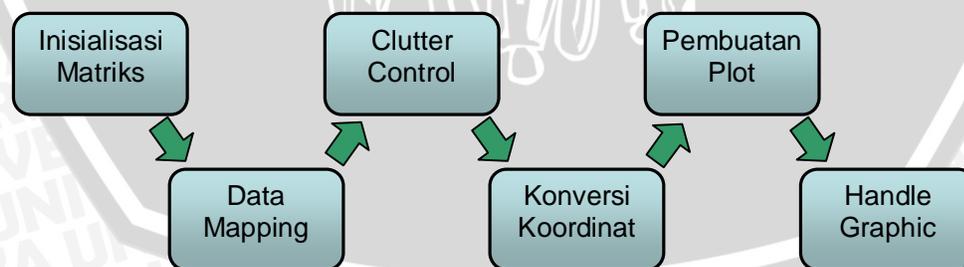
Vertical beamwidth dari antena didesain untuk mendeteksi target yang berada di permukaan, namun dengan desain ini *rain clutter* juga dapat terdeteksi sebagaimana target normal lainnya. *Rain clutter* tampak di *plotter* radar dengan pola berbintik. Dengan mengatur *rain clutter control* maka pola

berbintik tersebut akan tampak lebih samar sehingga target yang solid akan lebih mudah dikenali.

Rain clutter control menggunakan prinsip turunan sinyal. Hasil turunan sinyal akan menunjukkan amplitudo jika sinyal asli mengalami kenaikan atau penurunan amplitudo. Jika amplitudo sinyal asli bernilai sama pada rentang jarak tertentu, maka turunan sinyal akan bernilai nol. Area yang terdapat *rain clutter* akan menghasilkan sinyal pantul dengan amplitudo yang hampir sama kecuali jika terdapat target yang memiliki amplitudo lebih besar. Dengan demikian *rain clutter* dapat dihilangkan tapi target masih terlihat. Namun cara ini memiliki kelemahan, yaitu jika target cukup besar maka bagian tengah target akan hilang karena efek *rain clutter control*. Jadi *rain clutter control* juga perlu diatur dengan sebaik mungkin. Dalam kondisi nyata *rain clutter control* jarang digunakan karena *rain clutter* biasanya juga dapat dihilangkan dengan pengaturan *threshold* yang baik.

Karena data yang digunakan merupakan hasil *sampling* dari sinyal video sehingga merupakan data diskrit, maka *rain clutter control* tidak menggunakan turunan sinyal secara langsung. Namun menggunakan selisih data pada baris sebelum dan sesudahnya untuk memproses data pada tiap baris.

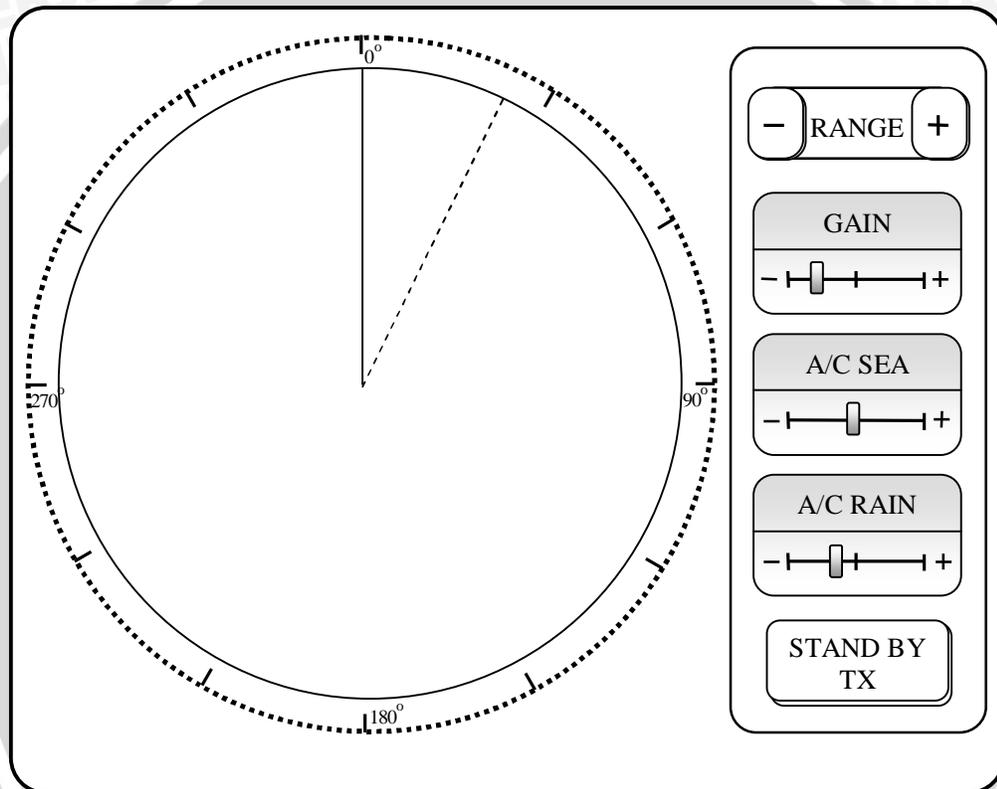
Secara keseluruhan terdapat beberapa tahapan dalam program untuk mengolah data radar hingga menjadi radar map. Tahapan tersebut diperlihatkan pada gambar 3.4.



Gambar 3.4. Tahapan Proses dalam Program Pengolah Sinyal Radar.

3.4. PEMBUATAN GUI UNTUK *DISPLAY* RADAR

GUI (*Graphical User Interface*) yang akan dibuat dirancang dengan konsep yang tidak jauh berbeda dengan *display* radar sebenarnya. GUI terdiri dari dua bagian, sebelah kanan adalah tampilan hasil *scan* radar atau disebut juga *radar map*, dan sebelah kiri adalah *display control* yang terdiri dari tombol-tombol pengatur *display* radar meliputi tombol *range*, *gain*, *clutter control*, dan lain-lain. Desain GUI untuk *display* radar adalah seperti gambar 3.3.

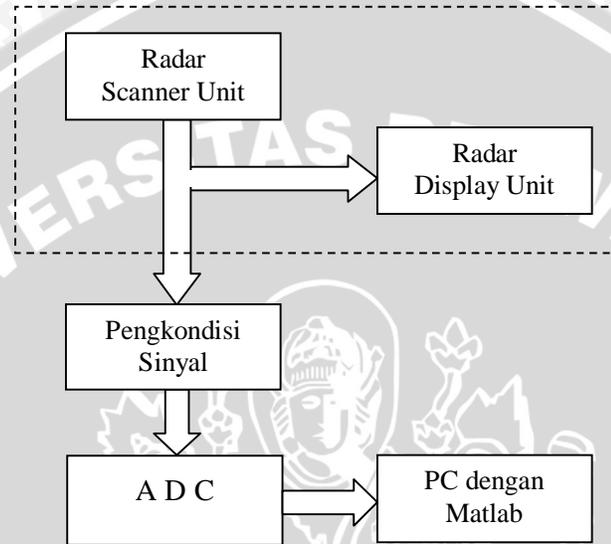


Gambar 3.5. Desain GUI Matlab Untuk Sistem *Display* Radar

3.5. PENGUJIAN SISTEM *DISPLAY* RADAR DAN ANALISA

Remote display radar yang telah dibuat akan diuji menggunakan data hasil *sampling* sinyal radar. Pengujian meliputi *scanning* untuk tiap *range* jangkauan, pengujian *gain control*, *sea clutter control*, dan *rain clutter control*. Gambar hasil *scanning* menggunakan sistem *display* radar yang telah dibuat kemudian dibandingkan dengan gambar yang ditampilkan oleh radar sebenarnya.

Diagram blok sistem radar yang digunakan saat pengujian ditunjukkan pada gambar 3.6. Bagian yang berada di dalam kotak dengan garis putus-putus adalah perangkat dari sistem radar Furuno. Berdasarkan gambar 3.6 sinyal radar yang diproses oleh sistem sistem *display* radar buatan diambil langsung dari *scanner unit* radar Furuno.



Gambar 3.6. Diagram Blok Sistem Radar Saat Pengujian

Hasil dari uji coba kemudian dianalisa untuk mengetahui kinerja sistem *display* radar yang telah dibuat. Jika kinerjanya bagus maka sistem *display* tersebut akan menunjukkan hasil yang sesuai dengan tampilan radar sesungguhnya. Analisa juga perlu dilakukan untuk mengetahui kelebihan dan kekurangan sistem *display* yang telah dibuat.

BAB IV

PEMBUATAN *DISPLAY* RADAR

Data hasil *sampling* sinyal radar akan diolah dengan program pengolah sinyal radar hingga menjadi grafik *radar map*. Selanjutnya grafik *radar map* akan ditampilkan pada sistem *display* radar. Pada bab ini akan dibahas pembuatan *display* radar yang terdiri dari program pengolah sinyal radar dan GUI untuk *display* radar.

4.1 PEMBUATAN PROGRAM PENGOLAH SINYAL RADAR

Data hasil *sampling* sinyal radar pada skripsi ini berupa matriks yang diberi nama “data” yang terdiri dari dua kolom. Kolom pertama berisi data hasil *sampling* sinyal video dan kolom kedua berisi data hasil *sampling* sinyal *trigger*. Program untuk mengolah data hasil *sampling* dibuat dengan menuliskan kode program pada M-File MATLAB. Isi program terdiri dari beberapa tahapan dalam mengolah data seperti pada gambar 3.4, yaitu:

4.1.1 Pembuatan matriks yang berisi data posisi dan matriks yang berisi data target

Matriks yang berisi data posisi diberi nama “radardat” yang terdiri dari 2 kolom dan jumlah baris sesuai dengan jumlah baris pada matriks “data”. Kolom pertama “radardat” digunakan untuk menyimpan data sudut target relatif terhadap *heading* radar, sedangkan kolom kedua digunakan untuk menyimpan data jarak target dari pusat peta pada *display* radar (posisi radar).

Matriks data target diberi nama “praz” yang terdiri dari satu kolom dan jumlah baris tergantung data. Matriks ini digunakan untuk menyimpan data hasil pengolahan sinyal video (dari kolom pertama matriks “data”) yang selanjutnya akan digunakan pada proses *clutter control* dan seleksi *threshold*.

4.1.2 Data Mapping

Tahapan ini merupakan inti dari program pengolah sinyal radar. Seleksi dimaksudkan untuk mengambil data yang masih dalam jangkauan radar. Karena dalam satu *scan line* radar memancarkan sinyal hingga jarak maksimum yaitu 250 km, sedangkan jangkauan radar yang digunakan memiliki jarak tertentu, misalkan 6 NM (11,112 km), 12 NM (22,224 km), dan pada

skripsi ini jarak maksimum yang digunakan adalah 24 NM (44,448 km). Dengan mengambil hanya data yang diperlukan, maka proses pengolahan data akan menjadi lebih cepat, karena data yang diolah lebih sedikit.

Data mapping diawali dengan mendeteksi tepi naik sinyal *trigger*. Yaitu dengan membandingkan nilai data baris ke- n dengan nilai data baris sebelumnya ($n-1$) pada kolom kedua matriks “data”. Jika nilainya lebih besar (pada skripsi ini digunakan pembanding sebesar 0,2 V), maka pada baris tersebut mulai terjadi tepi naik *trigger*. Tepi naik *trigger* menunjukkan bahwa radar memancarkan sinyal pada *scan line* baru. Secara fisik arah pancaran sinyal radar bergeser sebesar $0,26667^{\circ}$ searah jarum jam.

Setelah tepi naik *trigger* terdeteksi, selanjutnya matriks “data” diproses per baris. Pada tepi naik yang pertama sinyal *trigger*, yang bertepatan dengan *heading* radar, kolom kedua matriks “radardat” (menyatakan sudut) diberi nilai 0 dan kolom pertama (menyatakan jarak) juga diberi nilai 0. Data pada matriks “praz” pada baris yang sama kemudian dibandingkan nilainya dengan *threshold*, jika lebih besar maka nilai tersebut disimpan pada matrik “radz”.

Pada pemrosesan matriks “data” baris selanjutnya, kolom kedua matriks “radardat” diberi nilai sama dengan baris sebelumnya, sedangkan kolom pertama diberi nilai sama dengan baris sebelumnya ditambah resolusi jarak *sampling*. Pada frekuensi *sampling* 2,5 MHz resolusi jaraknya 60 m dan pada frekuensi *sampling* 1,25 MHz resolusi jaraknya 120 m. Dan seperti pemrosesan pada baris sebelumnya, data pada matriks “praz” pada baris yang sama dibandingkan nilainya dengan *threshold*, jika lebih besar maka nilai tersebut disimpan pada matrik “radz”.

Proses pada baris selanjutnya serupa dengan proses pada baris sebelumnya hingga kolom pertama matriks “radardat” nilainya mencapai jarak jangkauan radar. Jika ini sudah tercapai maka akan dilakukan pendeteksian tepi naik *trigger* selanjutnya.

Posisi baris pada matriks “data” di mana ditemukan tepi naik *trigger* yang baru merupakan awal dari urutan data pada *scan line* baru. Dengan demikian kolom kedua matriks “radardat” pada baris yang bersesuaian diberi nilai sama dengan baris sebelumnya ditambah 0,26667. Sedangkan kolom

pertama matriks “radardat” pada baris tersebut diberi nilai 0. Proses selanjutnya sama dengan proses pada *scan line* sebelumnya hingga mencapai jarak jangkauan radar. Proses ini terus berulang hingga didapat pemrosesan data satu *scan* penuh (360^0).

4.1.3 Clutter Control

Clutter control terdiri dari dua bagian, yaitu *sea clutter control* dan *rain clutter control*. *Clutter control* mengurangi *clutter* menggunakan persamaan khusus yang diterapkan pada sinyal video. Persamaan ini disisipkan pada program sebagai perintah yang ditempatkan pada bagian sebelum seleksi *threshold*.

Untuk *rain clutter control* perintah yang digunakan adalah sebagai berikut:

```
differ=abs(praz2(k+1,1)-praz2(k-2,1));
praz(k,1)=(praz(k,1)*(1-rc))+(differ*rc);
```

Variabel “differ” merupakan selisih amplitudo sinyal video pada urutan data sebelum (data ke-(k-2)) dan sesudah (data ke-(k+1)) data yang akan diterapkan *rain clutter control* (data ke-k). Variabel ini digunakan sebagai pengganti turunan sinyal. Variabel “differ” kemudian digunakan dalam perintah berikutnya untuk merubah amplitudo sinyal video pada urutan data ke-k. Agar sinyal video yang digunakan untuk mendapatkan variabel “differ” merupakan sinyal asal sebelum terkena efek *rain clutter control*, maka digunakan matriks “praz2” yang merupakan *backup* dari sinyal video. Nilai “rc” didapatkan dari *rain clutter control slider* yang telah diatur oleh pengguna radar.

Sea clutter control menggunakan perintah sebagai berikut:

```
scthd=(constsc/((radardat(nodat,2)/rangesc)^sc))-(0.25-0.5*sc);
if scthd<0
    scthd=0;
end
praz(k,1)=praz(k,1)-scthd;
```

Variabel “scthd” merupakan representasi dari kurva *sea control* pada tiap data ke-k. Nilai “scthd” akan besar pada jarak yang dekat dengan pusat

radar dan nilainya akan semakin kecil seiring meningkatnya jarak. Namun “scthd” perlu dibatasi agar tidak memiliki nilai di bawah nol. Konstanta “constsc” merupakan konstanta *sea control* yang digunakan pada program ini yang nilainya sebesar 0,2. Konstanta “rangesc” merupakan pembagi jarak, yaitu jarak yang diperkirakan *sea clutter* sudah tidak nampak lagi. Pada program ini “rangesc” bernilai 2500 (m). “sc” merupakan nilai *sea control* yang didapat dari *sea clutter control slider* yang telah diatur oleh pengguna radar. *Sea clutter control* mengurangi *sea clutter* dengan mengurangi sinyal video dengan kurva *sea control*.

4.1.4 Perubahan Koordinat

Matriks “radardat” yang telah dibuat pada tahapan sebelumnya yang menggunakan koordinat polar akan diubah menjadi koordinat kartesian. Hal ini perlu dilakukan karena data akan di-plot menggunakan fungsi *meshgrid*.

Pengubahan komponen dari koordinat polar ke koordinat kartesian menggunakan persamaan 2.5. Dan penerapannya pada program menggunakan perintah berikut.

```
radx=radardat(:,2).*cosd(radardat(:,1)-90);  
rady=radardat(:,2).*sind(radardat(:,1)-90);
```

4.1.5 Pembuatan Plot

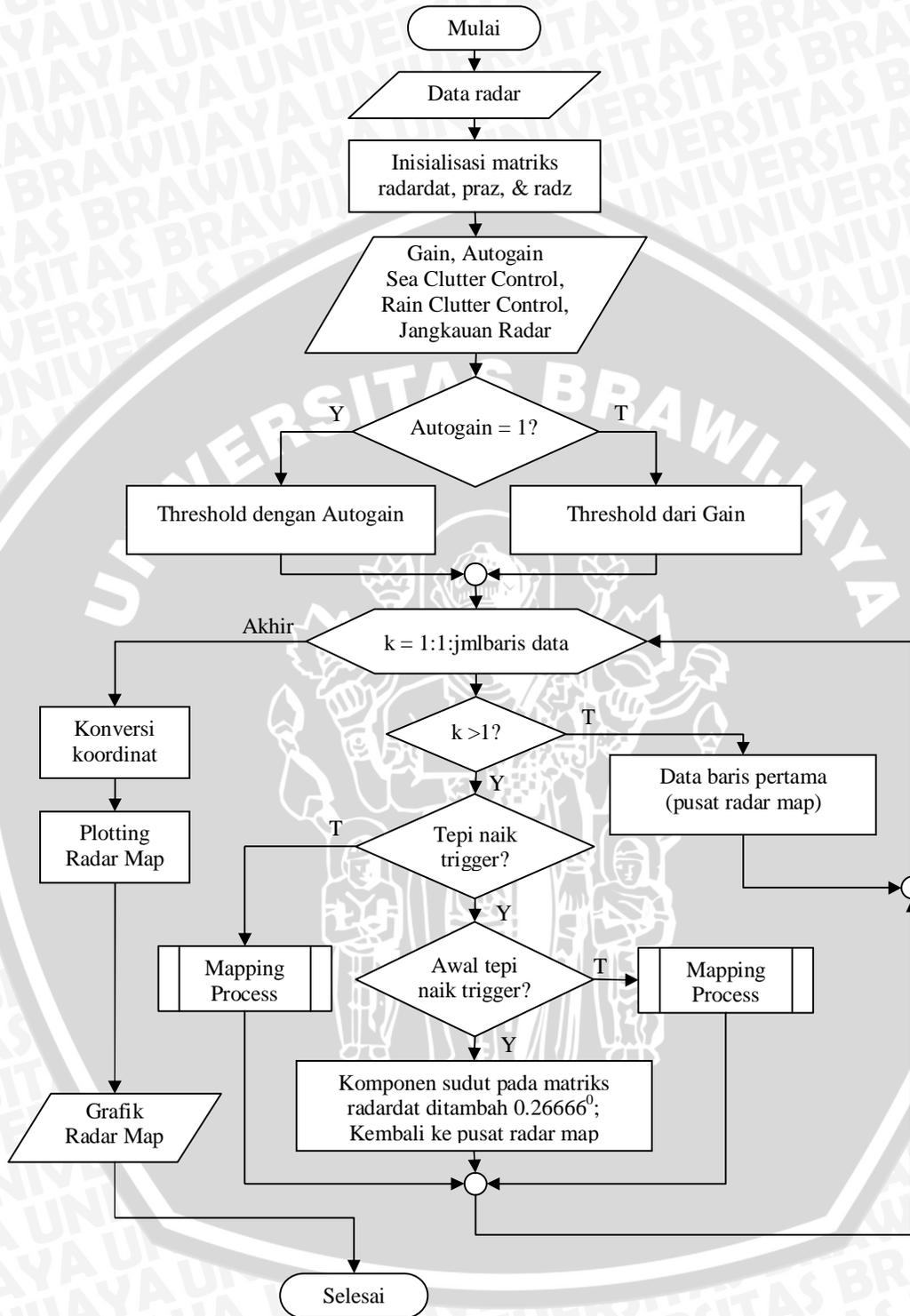
Hasil pengolahan data pada tahapan sebelumnya adalah matriks “radx” yang merupakan komponen sumbu-x pada grafik *display* radar, matriks “rady” sebagai komponen sumbu-y, dan matriks “radz” yang berisi data target yang direpresentasikan dalam nilai amplitudo.

Ketiga data tersebut akan dibuat *plot* yang akan ditampilkan sebagai grafik *display* radar. Pembuatan *plot* menggunakan fungsi *meshgrid* dan *griddata*.

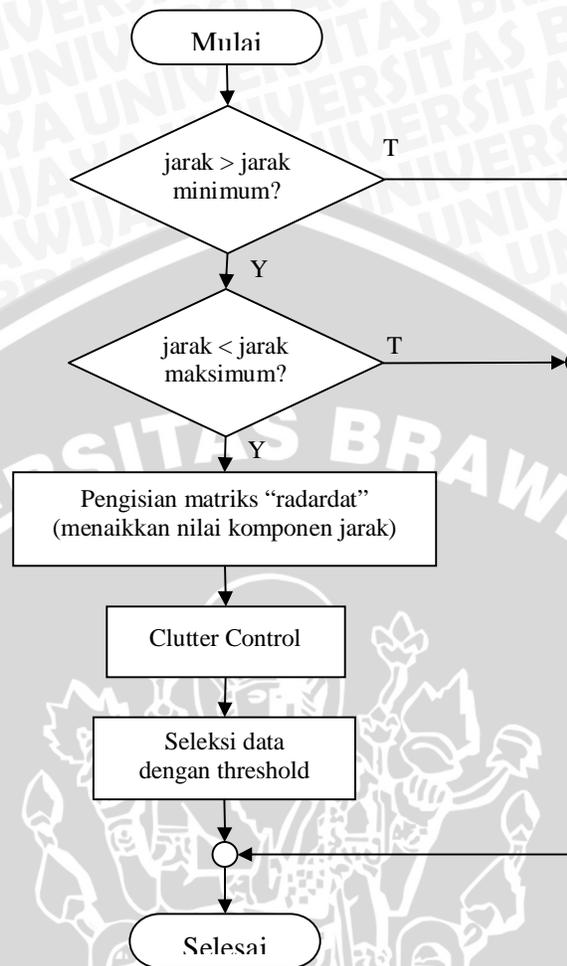
4.1.6 Handle Graphics

Hasil *plot* perlu diatur agar dapat ditampilkan sesuai dengan keinginan. Beberapa pengaturan yang dilakukan di antaranya penggunaan *colormap*, pengaturan label *axis* dan *colorbar*.

Diagram alir program pengolah sinyal radar ditunjukkan pada gambar 4.1 dan 4.2.



Gambar 4.1. Diagram Alir Program Pengolah Sinyal Radar



Gambar 4.2. Diagram Alir Fungsi *Mapping Process*

4.2. PEMBUATAN GUI *DISPLAY* RADAR

Display radar dibuat dengan menggunakan GUI pada Matlab. GUI untuk *display* radar dibangun berdasarkan desain yang telah dibuat seperti pada gambar 3.5. Sub bab ini membahas pembuatan GUI untuk *display* radar tahap demi tahap.

4.2.1 Pembuatan Jendela Utama *Display Radar*

Jendela utama untuk menyusun GUI *display* radar dibuat dengan fungsi `figure` dan `handle` yang diberi nama `win1`. Property value yang digunakan yaitu:

- units : points
- position : left 0, bottom 10, width 770, dan height 570
- menubar : none
- resize : off
- numbertitle : off
- name : Display Radar

Nilai position dirancang untuk monitor dengan resolusi 1024x768.

4.2.2 Pembuatan Objek Grafik *Radar Map*

Bagian utama dari *display* radar adalah peta radar (*radar map*). *Radar map* ditempatkan di bagian kanan sistem *display*. Objek grafik untuk menampilkan *radar map* dibuat dengan fungsi *axes* dan dengan *handle* yang diberi nama grafik 1. *Property value* yang digunakan yaitu:

- parent : win1
- units : pixel
- position : left 20, bottom 30, width 750, height 680

Karena fungsi *axes* selalu menampilkan garis sumbu, sedangkan untuk peta radar garis sumbu ini tidak diperlukan. Maka ditambahkan perintah *axis off* setelah pembuatan objek grafik.

Sebagai pelengkap *radar map* yang menggunakan gradasi warna, pada *display* radar ditambahkan *colorbar* untuk menjelaskan intensitas warna dalam hubungannya dengan kuat sinyal. *Colorbar* dipasang di sebelah kanan *radar map*. Sesuai dengan program pengolah sinyal radar yang telah dibuat sebelumnya, *colormap* yang digunakan adalah jenis “jet”. Maka warna biru menunjukkan sinyal yang lemah. Semakin kuat sinyal maka warna yang ditunjukkan pada peta radar mendekati merah.

4.2.3 Pembuatan Tombol Pengatur Jangkauan dan Satuan Jarak Radar

Sistem *display* radar ini menggunakan dua sistem satuan yaitu Nautical Mile (NM) dan kilometer (km) (1 Nautical Mile = 1,852 km). Setiap jenis satuan memiliki tujuh jangkauan (*range*) radar. Pada jenis satuan NM, jangkauannya adalah 4 NM, 6 NM, 8 NM, 10 NM, 12 NM, 18 NM, dan 24

NM. Pada jenis satuan km, jangkauannya adalah 8 km, 10 km, 12 km, 15 km, 20 km, 30 km, dan 42 km.

Untuk memilih sistem satuan, pada GUI dibuat *uicontrol* jenis *radio button*. Pada *uicontrol* untuk memilih satuan NM, *handle uicontrol* diberi nama “h_nm” dan *property value* yang digunakan adalah:

- units : points
- position : left 620, bottom 402, width 115, height 17
- string : Nautical Mile (NM)
- fontsize : 11
- style : radiobutton
- value : 1 (satuan NM digunakan sebagai default)
- callback : satuan_nm (M-File)

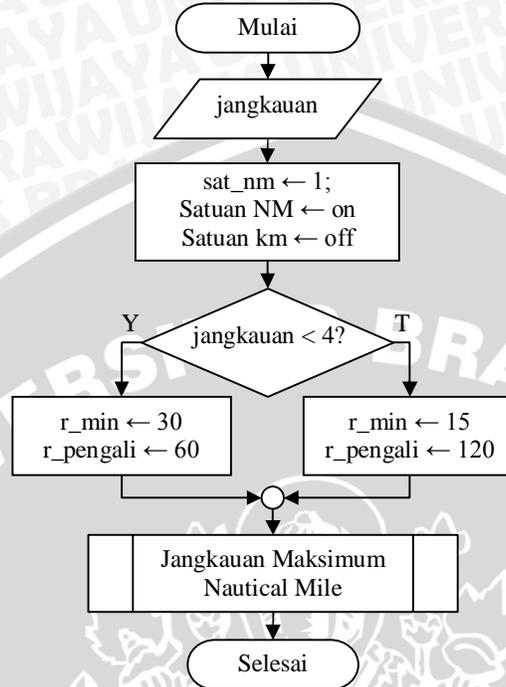
Pada *uicontrol* untuk satuan kilometer, *handle uicontrol* diberi nama “h_km” dan *property value* yang digunakan adalah:

- units : points
- position : left 620, bottom 385, width 115, height 17
- string : kilometer (km)
- fontsize : 11
- style : radiobutton
- value : 0
- callback : satuan_km (M-File)

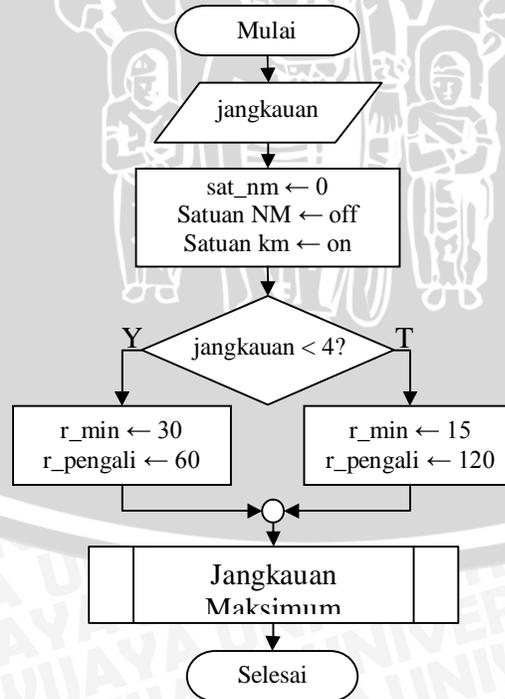
Program yang dijadikan *callback uicontrol* pemilih satuan NM (satuan_nm) berfungsi untuk memilih satuan NM dengan memberi nilai 1 pada variabel “sat_nm”. Pada program “satuan_nm” juga terdapat pengaturan jarak minimum dan maksimum radar dengan menentukan nomer data yang akan diproses. Selain itu pada program tersebut juga mengatur *string* yang akan ditampilkan pada GUI sebagai jangkauan dan satuan yang sekarang dipilih.

Program “satuan_km” yang menjadi *callback uicontrol* pemilih satuan km memiliki prinsip yang sama dengan program “satuan_nm”. Pada program “satuan_km”, variabel “sat_nm” akan diberi nilai nol. Jarak minimum dan maksimum radar akan disesuaikan dalam satuan kilometer, begitu juga dengan *string* jangkauan dan satuan yang akan ditampilkan pada GUI.

Diagram alir program “satuan_nm” dan “satuan_km” ditunjukkan pada gambar 4.3 dan 4.4.

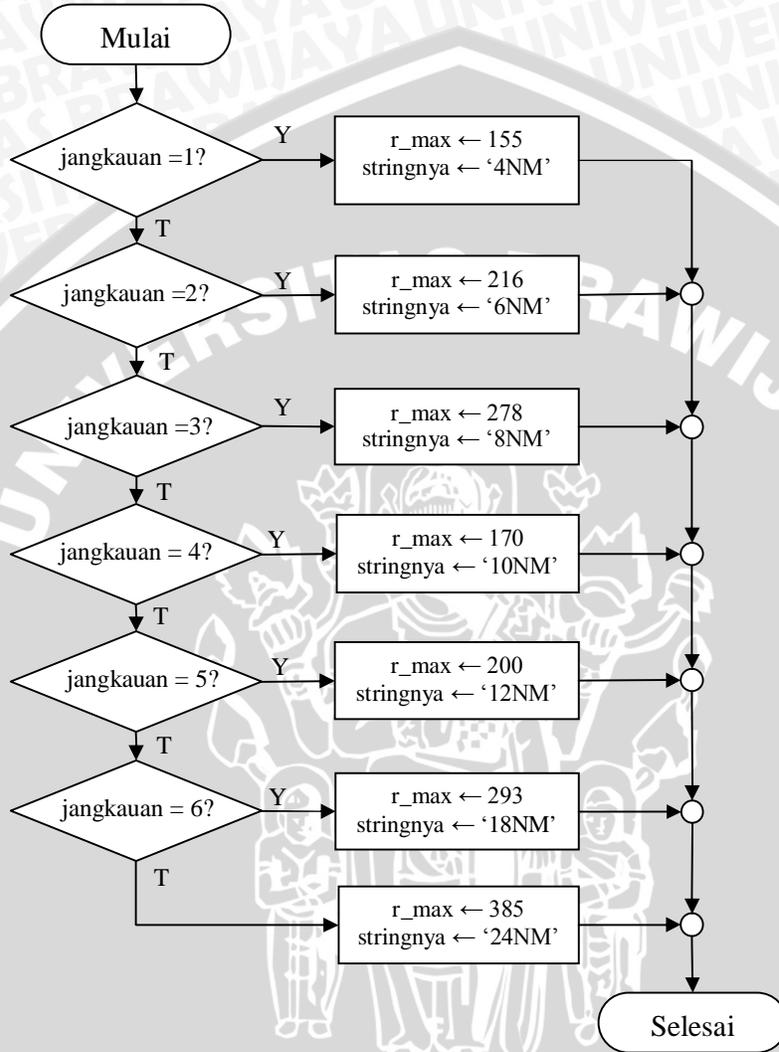


Gambar 4.3. Diagram Alir Program “satuan_nm”

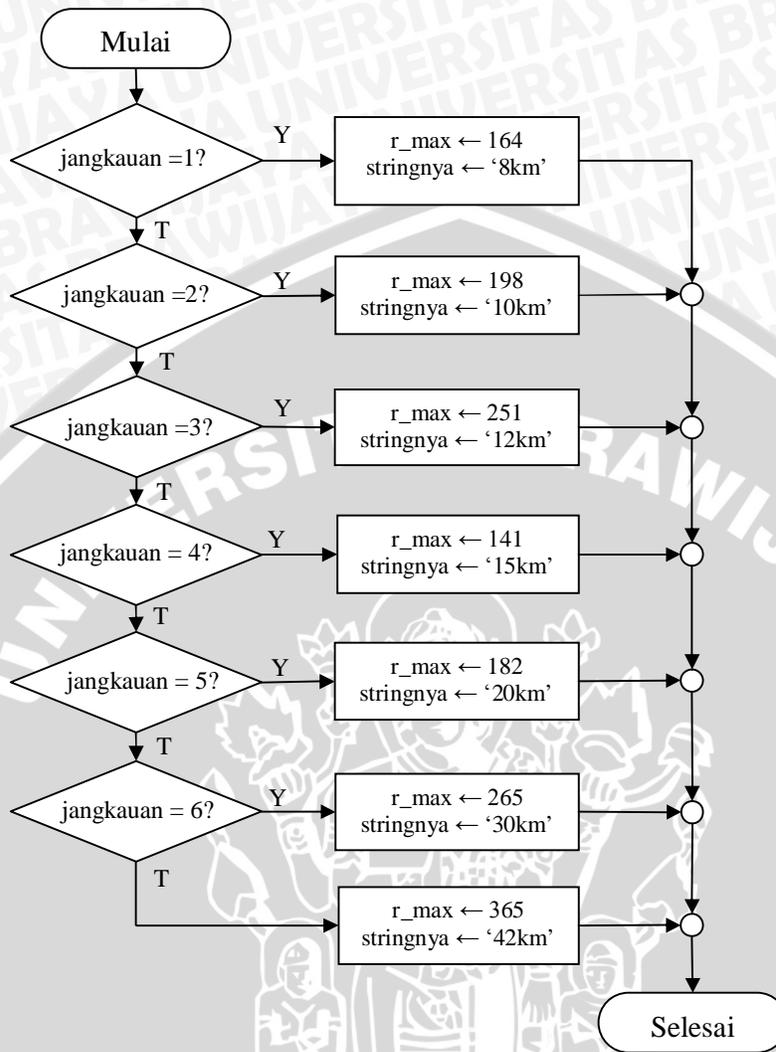


Gambar 4.4. Diagram Alir Program “satuan_km”

Diagram alir fungsi “Jangkauan Maksimum Nautical Mile” dan fungsi “Jangkauan Maksimum Kilometer” ditunjukkan pada gambar 4.5 dan gambar 4.6.



Gambar 4.5. Diagram Alir Fungsi “Jangkauan Maksimum Nautical Mile”



Gambar 4.6. Diagram Alir Fungsi “Jangkauan Maksimum Kilometer”

Untuk memilih jangkauan (*range*) radar, pada GUI dibuat tombol pengurang dan penambah jangkauan. Tiap jangkauan diberi nilai *integer*. Jangkauan yang paling pendek diberi nilai 1 dan jangkauan yang paling jauh diberi nilai 7. Jangkauan awal yang digunakan (*default*) adalah 10 NM atau 15 km yang sama-sama diberi nilai *integer* 4. Dengan menekan tombol pengurang atau penambah jangkauan maka nilai *integer* akan berganti, begitu juga dengan jangkauannya sesuai dengan nilai *integer* yang baru.

Property value pada *uicontrol* tombol pengurang jangkauan adalah:

- parent : win1
- units : points
- position : left 595, bottom 455, width 30, height 26
- style : pushbutton
- string : -
- fontsize : 15
- callback : kurangi_range (M-File)

Property value pada *uicontrol* tombol penambah jangkauan adalah:

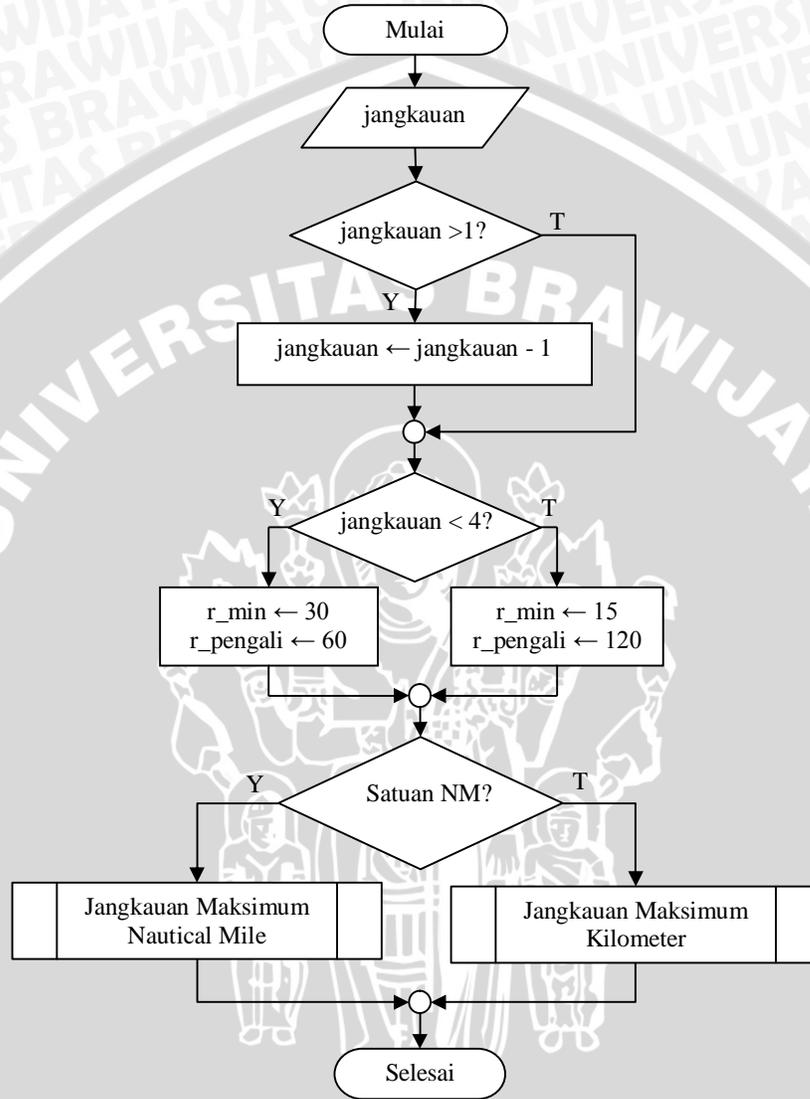
- parent : win1
- units : points
- position : left 715, bottom 455, width 30, height 26
- style : pushbutton
- string : +
- fontsize : 15
- callback : tambah_range (M-File)

Program yang dijadikan *callback uicontrol* pengurang jangkauan berfungsi untuk memilih jangkauan, memberi nilai *string* yang akan ditampilkan pada *display* radar, dan memberi nilai jarak minimum dan maksimum serta pengali jarak untuk pemrosesan data radar. Program untuk *callback uicontrol* penambah jangkauan juga memiliki fungsi yang sama dengan program untuk *callback uicontrol* pengurang jangkauan. Bedanya, program penambah jangkauan akan menaikkan nilai *integer* jangkauan sehingga jangkauan radar bertambah, sebaliknya program pengurang jangkauan akan mengurangi nilai *integer* jangkauan.

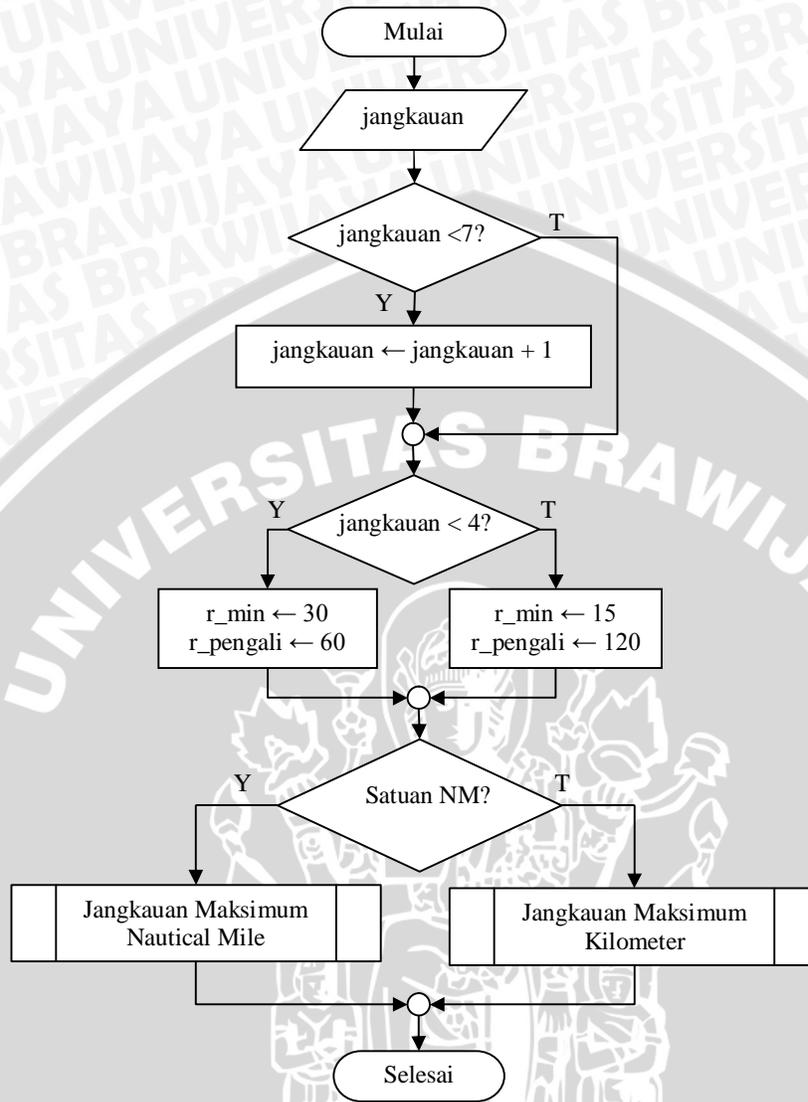
Beberapa variabel yang berhubungan dengan satuan dan jangkauan radar yang diberi nilai terlebih dahulu sebagai *default* adalah:

Variabel	Nilai
jangk (integer jangkauan radar)	4
r_min (jarak minimum radar)	15
r_max (jarak maksimum radar)	170
r_pengali (pengali jarak dalam meter)	120

Diagram alir untuk program “kurangi_range” dan “tambah_range” ditunjukkan pada gambar 4.7 dan 4.8.



Gambar 4.7. Diagram Alir Program “kurangi_range”



Gambar 4.8. Diagram Alir Program “tambah_range”

4.2.4 Pembuatan Gain Control

Gain control pada display radar sebenarnya berupa alat pemutar seperti pada potensiometer. Untuk menaikkan gain maka gain control diputar ke kanan. Pada sistem display radar yang akan dibuat, gain control dibuat menggunakan uicontrol tipe slider yang akan menaikkan gain jika slider digeser ke kanan. Pada uicontrol untuk gain control, handle uicontrol diberi nama “gain_sld” dan property value yang digunakan adalah sebagai berikut:

- parent : win1
- units : points
- position : left 600, bottom 290, width 140, height 20
- style : slider
- sliderstep : [.05 .2]
- max : 1
- min : 0

Program pengolah sinyal radar akan mengakses nilai *gain control* dengan perintah sebagai berikut:

```
prethd=get(gain_sld,'value');
```

Nilai *gain control* akan digunakan untuk mengatur nilai *threshold* dengan perintah berikut:

```
threshold=(1-prethd)*0.5;
```

Perintah di atas telah disesuaikan agar menghasilkan nilai *threshold* yang sesuai dengan rentang amplitudo sinyal radar yang diproses oleh program.

Selain *gain control* yang berupa *slider*, pada GUI juga dilengkapi dengan fitur *auto gain* untuk mendapatkan nilai *gain* secara instan. Nilai *gain* dari *auto gain* didapat dari rata-rata amplitudo sinyal radar (sinyal video). Fitur *auto gain* dibuat dengan menggunakan *uicontrol* tipe *checkbox*. *Auto gain* akan aktif jika *checkbox* bertanda centang. *Uicontrol* untuk *auto gain*, menggunakan *handle* yang diberi nama "aut_gain" dan *property value* sebagai berikut:

- units : points
- position : left 635, bottom 3345, width 115, height 17
- string : Auto Gain
- fontsize : 11
- style : checkbox
- value : 0 (default)

Program pengolah sinyal radar akan mengakses status *auto gain* apakah aktif atau tidak dengan perintah berikut:

```
autogain=get(aut_gain,'value');
```

Jika *auto gain* aktif, maka variable “autogain” akan bernilai 1, dan jika sebaliknya akan bernilai 0. *Auto gain* akan mengubah amplitudo sinyal video dengan perubahan yang besarnya tergantung dari rata-rata amplitudo sinyal video. Berikut ini adalah perintah pada program pengolah sinyal radar jika fitur *auto gain* aktif.

```
rataavid=mean(data(1:jmlbaris,1));
praz=rataavid-data(1:jmlbaris,1);
threshold=0.1;
```

Variabel “rataavid” merupakan rata-rata dari amplitudo sinyal video (data). Matriks “praz” kemudian menyimpan data (sinyal video) yang telah diubah amplitudonya. Sinyal video yang baru ini diberikan nilai *threshold* sendiri yang hanya digunakan jika fitur *auto gain* aktif.

4.2.5 Pembuatan *Clutter Control*

Clutter control pada *display* radar sebenarnya seperti pada *gain control*, yaitu berupa alat pemutar. Pada sistem *display* yang akan dibuat ini, *clutter control* juga dibuat dengan *uicontrol* tipe *slider*, seperti pada pembuatan *gain control*. *Clutter control* terdiri dari dua bagian, yaitu *sea clutter control* dan *rain clutter control*.

Uicontrol untuk *sea clutter control* menggunakan *handle* yang diberi nama “sea_sld” dan *properti value* sebagai berikut:

- parent : win1
- units : points
- position : left 600, bottom 215, width 140, height 20
- style : slider
- sliderstep : [.1 .25]
- max : 1
- min : 0

Program pengolah sinyal radar akan mengakses nilai *sea clutter control* dengan perintah sebagai berikut:

```
sc=(get(sea_sld,'value'))*0.5;
```

Nilai minimum *slider* adalah 0 dan nilai minimumnya 1. Tetapi pada program pengolah sinyal radar, variabel “sc” (*sea control*) menggunakan nilai setengah nilai *slider*.

Rain clutter control menggunakan *handle uicontrol* yang diberi nama “rain_sld” dengan *property value* sebagai berikut:

- parent : win1
- units : points
- position : left 600, bottom 140, width 140, height 20
- style : slider
- sliderstep : [.1 .25]
- max : 1
- min : 0

Program pengolah sinyal radar akan mengakses nilai *rain clutter control* dengan perintah sebagai berikut:

```
inputrc=get(rain_sld,'value');
```

4.2.6 Pembuatan Tombol *Scan* dan Tombol *Keluar*

Setelah *user* mengatur satuan dan jangkauan radar, kemudian mengatur *gain control* dan *clutter control*, maka selanjutnya untuk memulai proses *scanning* radar dibuatlah tombol *scan*. Dengan menekan tombol *scan* maka program pengolah sinyal radar akan dijalankan. Kemudian keluaran program yaitu grafik *radar map* hasil *scanning* radar akan ditampilkan pada sistem *display* radar.

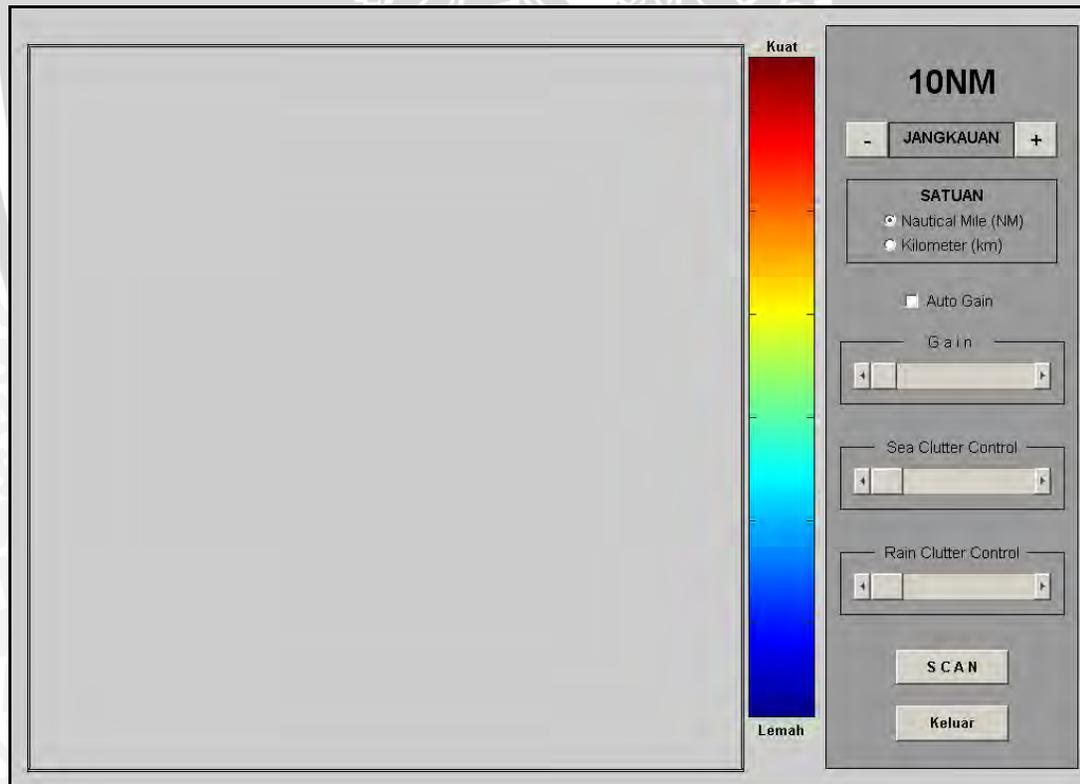
Tombol *scan* dibuat dengan *uicontrol* tipe *pushbutton* menggunakan *handle* bernama “tmblscan” dan *property value* sebagai berikut:

- parent : win1
- units : points
- position : left 630, bottom 80, width 80, height 25
- style : pushbutton
- string : S C A N
- fontweight : bold
- fontsize : 10
- callback : imageproc_total (M-File)

Sebagai pelengkap GUI *display* radar, untuk menutup jendela GUI maka dibuatlah tombol keluar. Untuk membuat tombol keluar digunakan *uicontrol* tipe *pushbutton* dengan *handle* “tmbkeluar” dan *property value* sebagai berikut:

- parent : win1
- units : points
- position : left 630, bottom 40, width 80, height 25
- style : pushbutton
- string : Keluar
- fontweight : bold
- fontsize : 10
- callback : close

Demikian proses pembuatan GUI untuk *display* radar. Hasil *screenshot* dari sistem *display* radar yang telah dibuat ditunjukkan pada gambar 4.9.



Gambar 4.9. Screenshot GUI Sistem Display Radar

BAB V

PENGUJIAN DAN PEMBAHASAN

5.1 PENGUJIAN *SAMPLING* SINYAL RADAR

Tahap pertama pengujian sistem *display* radar adalah *sampling* sinyal. Tempat pengujian adalah di kantor LAPAN Jl. Dr. Junjuna 133 Bandung. Posisi radar yang digunakan berada di atas gedung (lantai lima) seperti terlihat pada gambar 5.1.(a). Proses *sampling* sinyal radar dilakukan di ruangan pada lantai empat seperti terlihat pada gambar 5.1.(b). Pengujian dilakukan ketika cuaca di sekitar tempat pengujian sedang terjadi hujan. Dengan demikian diharapkan radar akan menangkap adanya *rain clutter*.



(a)

(b)

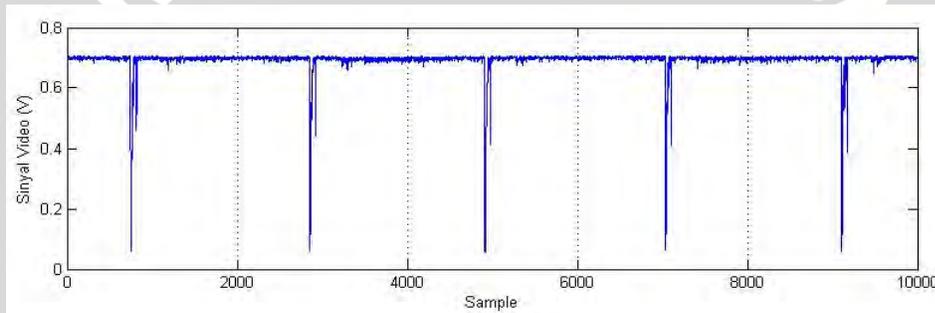
Gambar 5.1. (a) Posisi Radar; (b) Sistem *Display* dan Pengolah Sinyal Radar

Sinyal keluaran *scanner* radar dihubungkan dengan rangkaian pengkondisi sinyal yang kemudian dihubungkan dengan ADC Adlink PCI-9810 yang terpasang pada PC. Proses *sampling* oleh ADC terhadap sinyal radar dijalankan dengan kode Matlab. Hasil *sampling* adalah variabel “data” yang berupa matriks dua kolom seperti ditunjukkan pada gambar 5.2. Kolom pertama adalah hasil *sampling* sinyal video dan kolom kedua adalah hasil

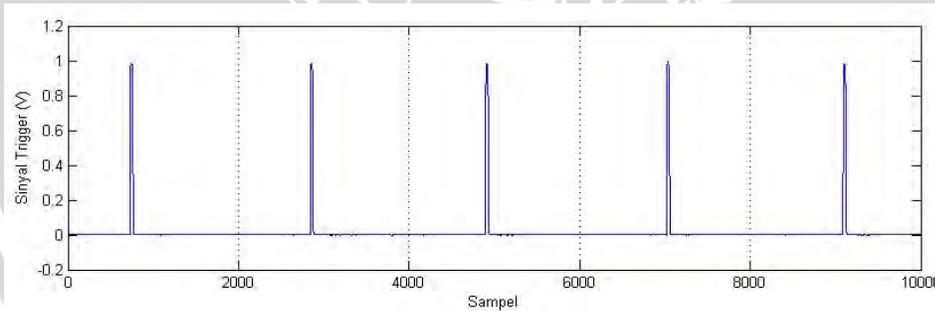
sampling sinyal *trigger*. Gambar 5.3 menampilkan hasil *sampling* kedua sinyal dari variabel “data” dalam bentuk grafik.

Name	Size	Bytes	Class	1	2
data	3000000x2	48000000	double	0.699244678...	1.525902189...
				0.701197833...	1.525902189...
				0.699244678...	1.525902189...
				0.701197833...	1.525902189...
				0.703150988...	1.525902189...
				0.701197833...	1.525902189...
				0.699244678...	1.525902189...
				0.699244678...	1.525902189...

Gambar 5.2. Variabel “data” Hasil *Sampling* Sinyal Radar



(a) Sinyal Video



(b) Sinyal *Trigger*

Gambar 5.3. Hasil *Sampling* Sinyal Radar

Pada gambar 5.3 dapat dilihat bahwa dalam setiap satu periode sinyal *trigger*, sinyal video memiliki pola yang hampir sama. Saat terjadi kenaikan amplitudo pada sinyal *trigger* berarti radar memancarkan sinyal pada *scan line* baru. Pola yang dibentuk oleh sinyal video menunjukkan bahwa terjadi pantulan sinyal radar yang kuat pada area dekat radar. Hal ini akan dapat terlihat jelas

setelah sinyal tersebut diolah dan ditampilkan menjadi *radar map*. Berdasarkan bentuk sinyal video, maka pada radar map akan muncul target-target yang kuat di area dekat radar.

5.2 PENGUJIAN PROGRAM DAN GUI SISTEM *DISPLAY* RADAR

Pengujian sistem *display* radar dilakukan secara *non-real time*. Data hasil sampling sinyal radar disimpan untuk digunakan sebagai bahan pengujian sistem *display* radar.

Pada pengujian program pengolah sinyal radar, hal yang ingin diuji adalah kemampuan program dalam menampilkan *radar map* serta kemampuan dalam *clutter control*. Sedangkan pada GUI *display* radar, pengujian ditekankan pada fungsi fitur-fitur GUI, yaitu pemilih satuan, pengatur jangkauan, *auto gain*, dan *clutter control*.

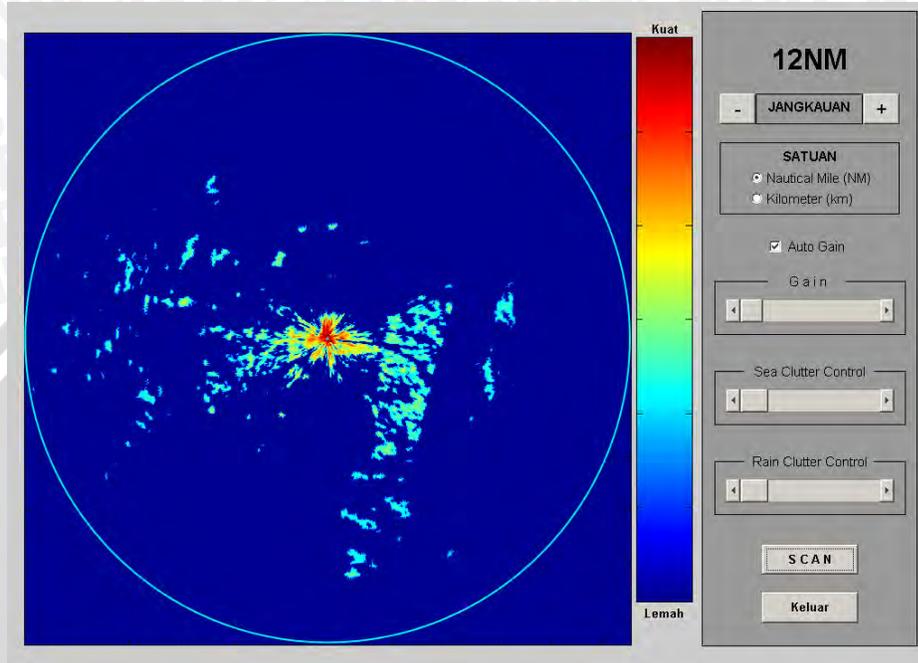
5.2.1 Pengujian *Auto Gain*, Pemilih Satuan dan Pengatur Jangkauan Radar

Pengujian fitur pemilih satuan dilakukan dengan dua kali simulasi. Untuk satuan Nautical Mile digunakan jangkauan 12 NM (22,224 km) dan untuk satuan kilometer digunakan jangkauan 20 km. Pada simulasi ini sekaligus dilakukan pengujian fitur *auto gain* namun tanpa *clutter control*. Karena itu *slider sea clutter control* dan *rain control* diatur pada posisi nol (minimum). Hasil kedua simulasi ini diperlihatkan pada gambar 5.4 dan 5.5.

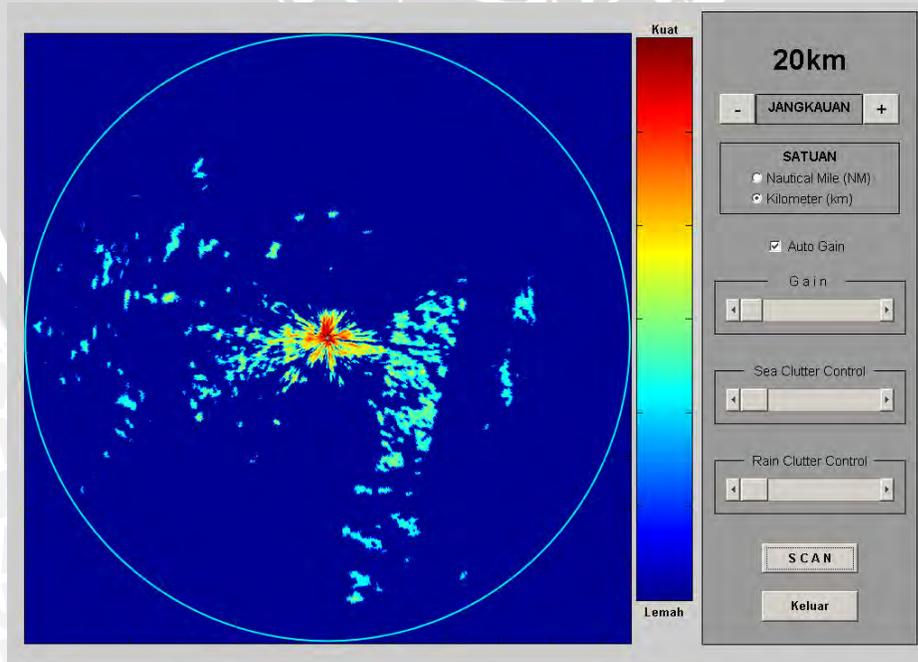
Karena nilai jangkauan pada dua simulasi tidak sama persis (22,224 km dan 20 km), maka citra *radar map* yang dihasilkan juga memiliki area yang sedikit berbeda. Area *radar map* disesuaikan dengan nilai jangkauan pada masing-masing simulasi. Sebagai perbandingan, gambar 5.6 menunjukkan *radar map* yang ditampilkan oleh *display* radar sebenarnya pada jangkauan 12 NM dan dengan mengaktifkan fitur A/C Auto (*auto clutter control*).

Radar map pada gambar 5.6 sedikit berbeda dengan gambar 5.4 dan 5.5. Hal ini karena fitur A/C Auto memiliki *setting* yang berbeda dengan fitur *auto gain* pada sistem *display* radar yang telah dibuat. A/C Auto merupakan fitur pengurang *clutter* secara otomatis yang menggabungkan A/C Sea dan A/C Rain pada *control display* radar Furuno. Sedangkan fitur *auto gain* digunakan

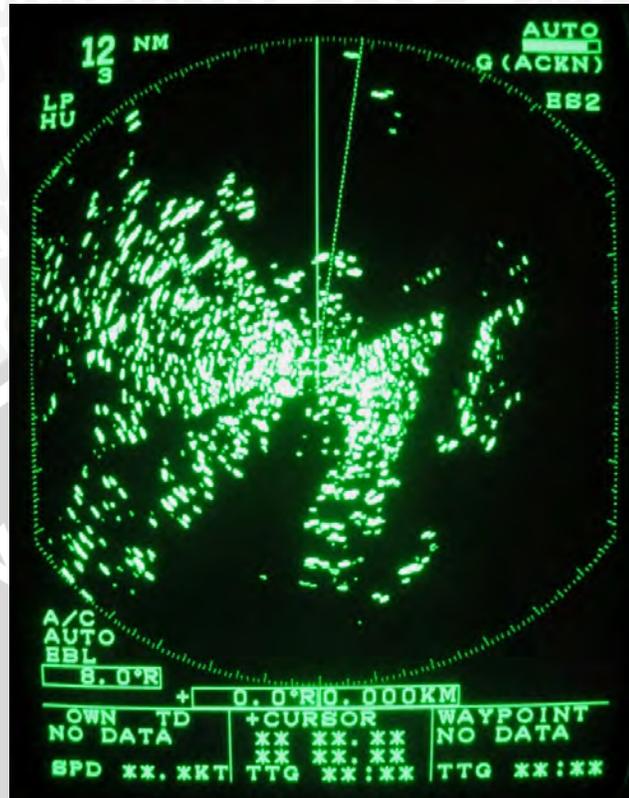
untuk mengatur *threshold* sinyal secara otomatis berdasarkan rata-rata amplitudo sinyal video.



Gambar 5.4. Hasil Pengujian Satuan Nautical Mile



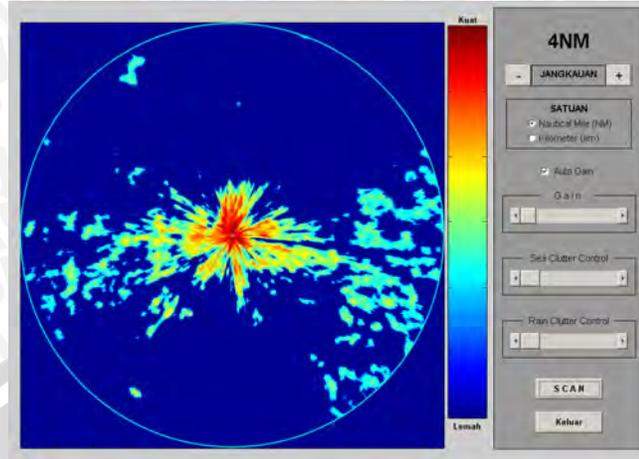
Gambar 5.5. Hasil Pengujian Satuan Kilometer



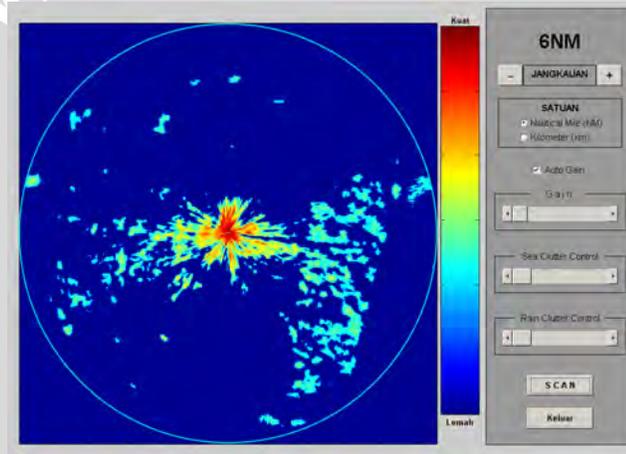
Gambar 5.6. Radar Map Pada Display Radar Furuno Dengan Range 12 NM

Pengujian fitur pengatur jangkauan radar dilakukan dengan simulasi pada jangkauan radar 4 NM, 6 NM, 8 NM, 10 NM, 12 NM, 18 NM, dan 24 NM untuk satuan Nautical Mile. Sedangkan untuk satuan kilometer, hasil simulasi yang ditampilkan hanya pada jangkauan radar 15 km, 20 km, dan 30 km. Hasil pengujian fitur pengatur jangkauan radar ditunjukkan pada gambar 5.7 sampai 5.14. Untuk jangkauan radar 12 NM dan 20 km sama seperti pada gambar 5.4 dan 5.5.

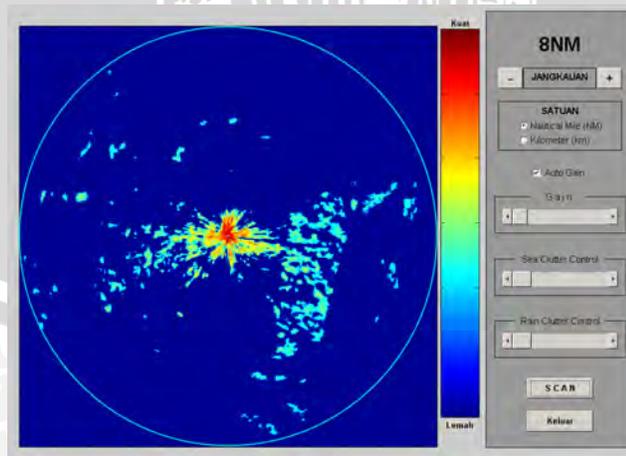
Radar map dengan jangkauan yang lebih kecil tampak seperti *zoom in* dari *radar map* dengan jangkauan yang lebih besar. Hal ini karena sinyal radar yang diproses memang hanya yang berada dalam jarak jangkauan yang dipilih, sedangkan jarak maksimum sinyal radar adalah 250 km.



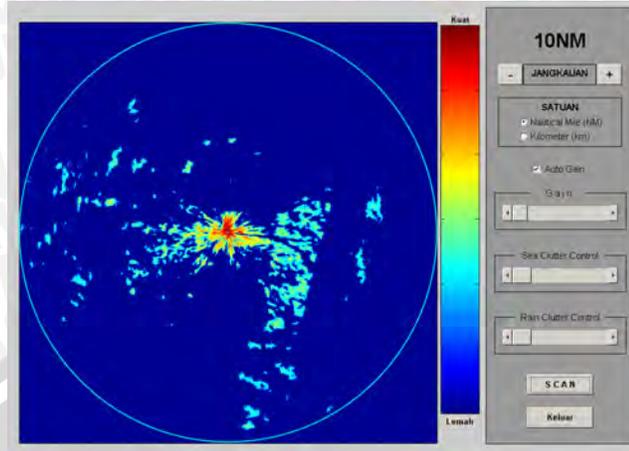
Gambar 5.7. Hasil Pengujian Jangkauan Radar 4 NM



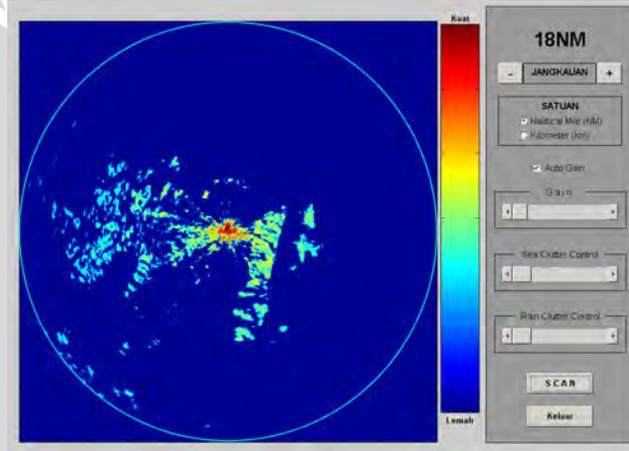
Gambar 5.8. Hasil Pengujian Jangkauan Radar 6 NM



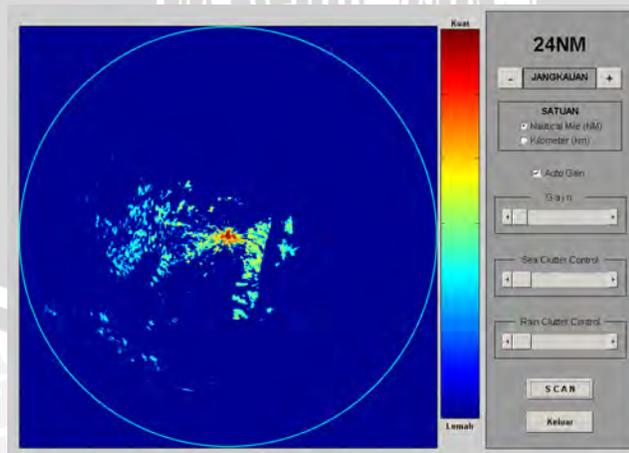
Gambar 5.9. Hasil Pengujian Jangkauan Radar 8 NM



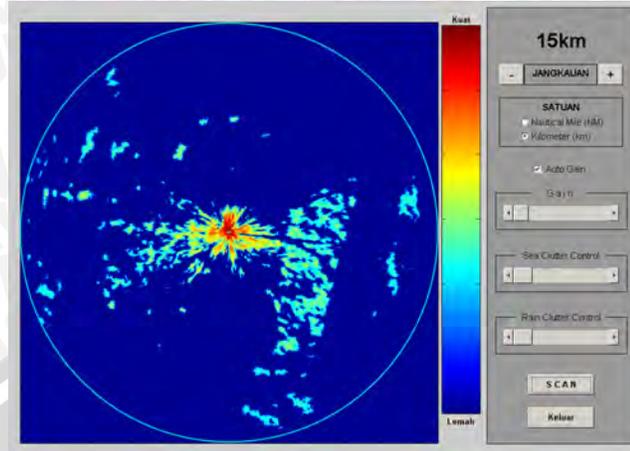
Gambar 5.10. Hasil Pengujian Jangkauan Radar 10 NM



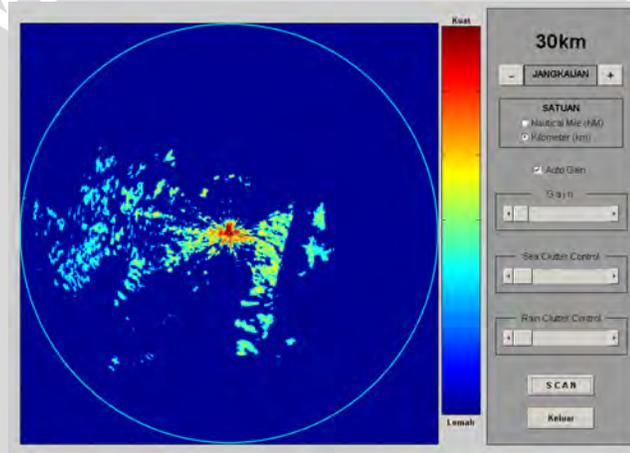
Gambar 5.11. Hasil Pengujian Jangkauan Radar 18 NM



Gambar 5.12. Hasil Pengujian Jangkauan Radar 24 NM



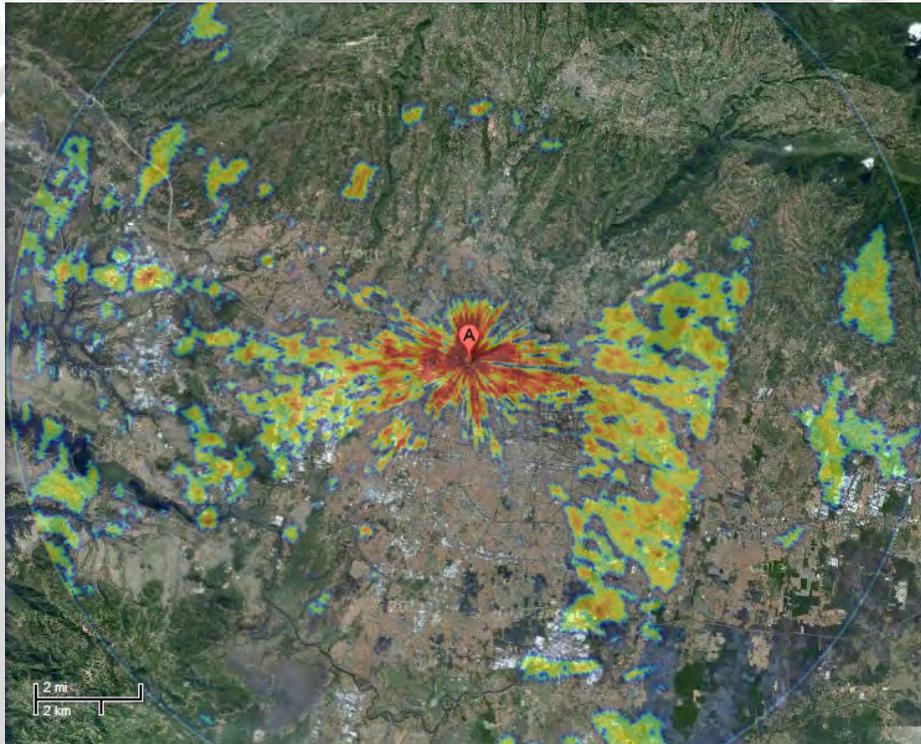
Gambar 5.13. Hasil Pengujian Jangkauan Radar 15 km



Gambar 5.14. Hasil Pengujian Jangkauan Radar 30 km

Hasil *overlay radar map* pada foto satelit kota Bandung yang di-*download* dari google map ditunjukkan pada gambar 5.15. *Radar map* dihasilkan dari pengujian *display* radar dengan jangkauan 15 km dan gain 0,65 skala. Label A menunjukkan posisi radar yaitu di gedung LAPAN. Nampak bahwa target yang muncul pada *radar map* adalah gedung-gedung dan beberapa lereng gunung yang curam. Topografi wilayah tempat pengujian juga memengaruhi target yang terlihat pada *radar map*. Gedung yang berada pada ketinggian tanah yang hampir sama dengan ketinggian radar akan muncul pada *radar map*. Gedung-gedung di dekat posisi radar tentu saja banyak yang memantulkan sinyal radar dengan amplitudo yang cukup kuat sehingga

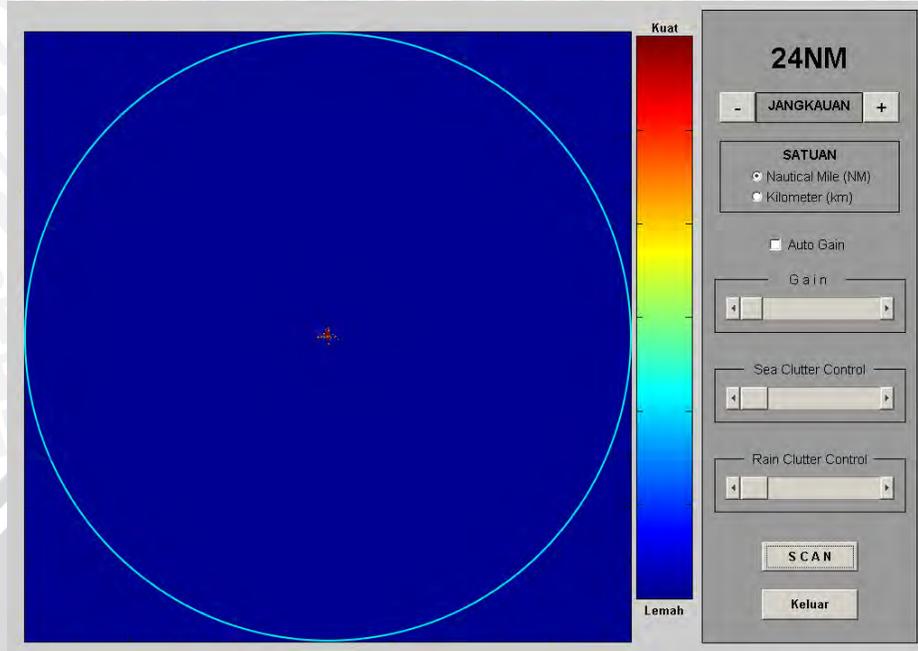
terdeteksi oleh radar. Karena gedung tersebut masih berada pada ketinggian tanah yang hampir sama dan jaraknya yang cukup dekat dengan posisi radar. Karena itu pada *radar map* di area pusat (dekat posisi radar) muncul banyak target yang kuat yang ditunjukkan dengan warna merah. Sedangkan target lainnya yang agak jauh dari posisi radar cukup lemah memantulkan sinyal radar sehingga muncul warna kuning hingga yang paling lemah yaitu warna biru muda.



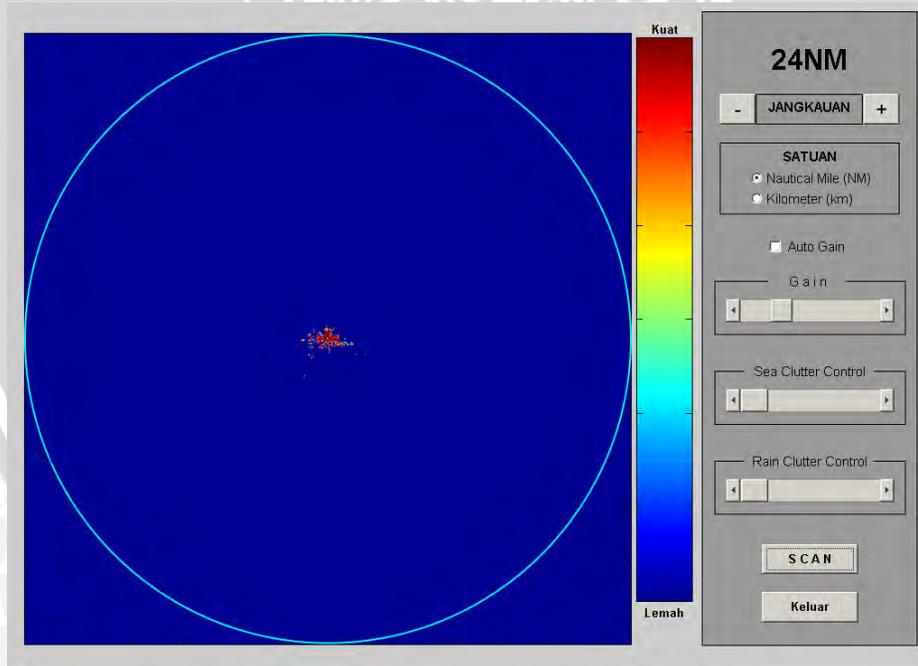
Gambar 5.15. *Overlay Radar Map* pada Foto Satelit di Wilayah Pengujian

5.2.2 Pengujian *Gain Control*

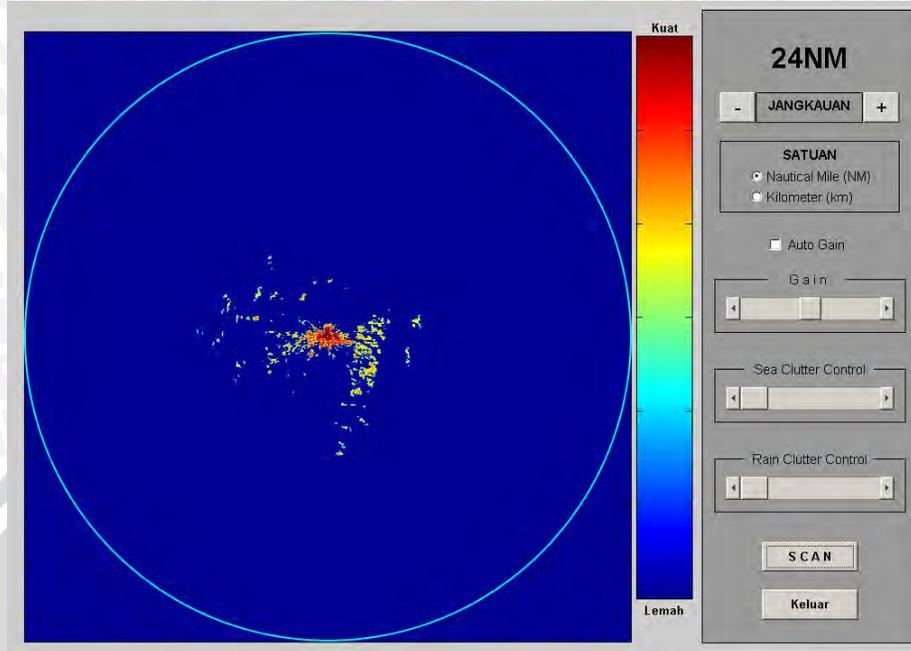
Pengujian fitur *gain control* dilakukan dengan simulasi sistem *display* radar menggunakan *gain control* pada lima kondisi *gain control*. Yaitu pada kondisi *gain* minimum, seperempat skala, setengah skala, tiga perempat skala, dan satu skala penuh. Jangkauan radar yang digunakan dalam pengujian *gain control* ini adalah 24 Nautical Mile. Hasil simulasi untuk pengujian *gain control* pada tiap kondisi *gain* ditunjukkan pada gambar 5.16 hingga 5.20.



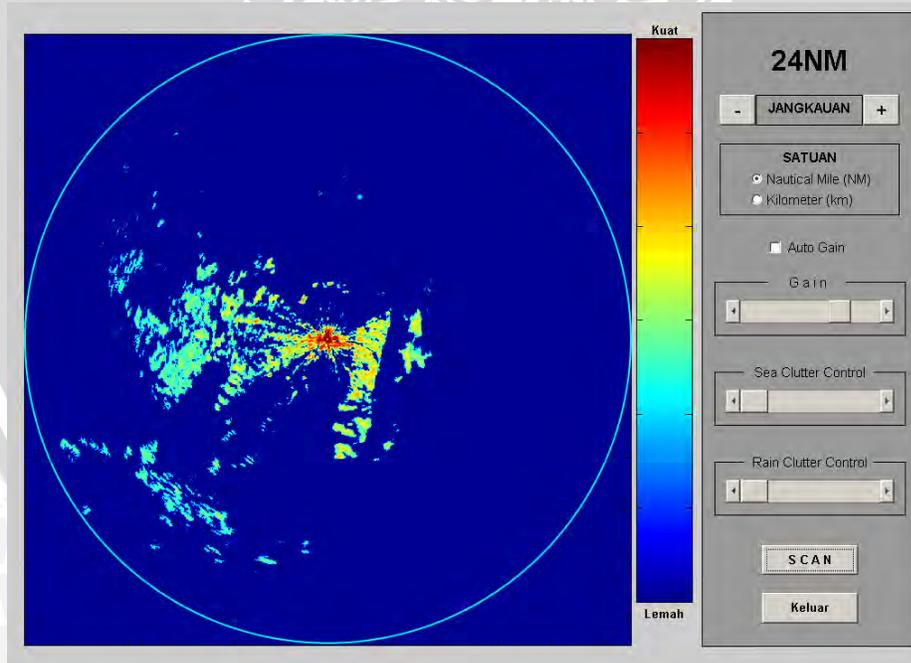
Gambar 5.16. Hasil Pengujian Gain Control Dengan Gain Minimum



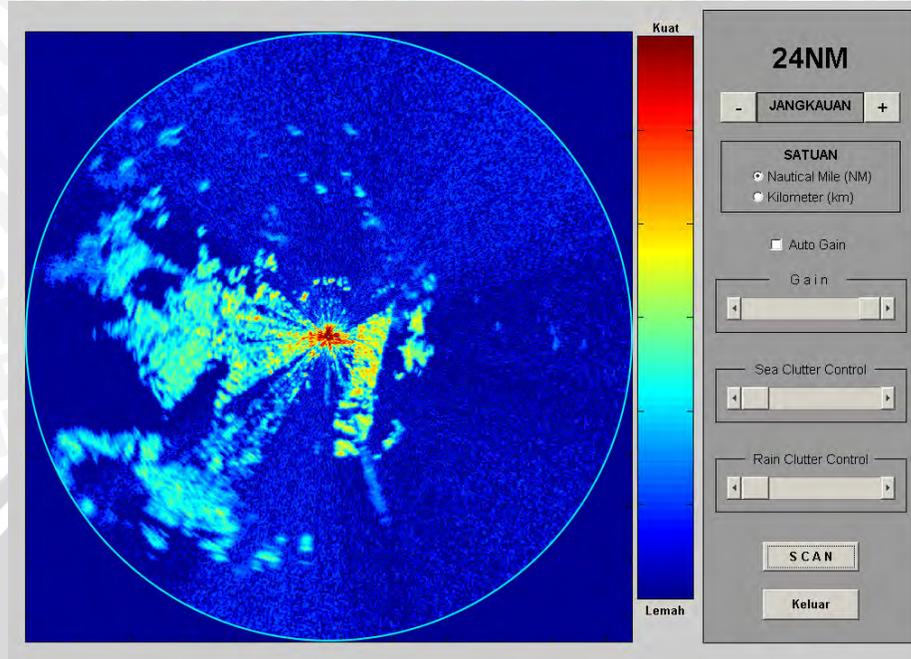
Gambar 5.17. Hasil Pengujian Gain Control Dengan Gain 1/4 Skala



Gambar 5.18. Hasil Pengujian *Gain Control* Dengan *Gain* ½ Skala



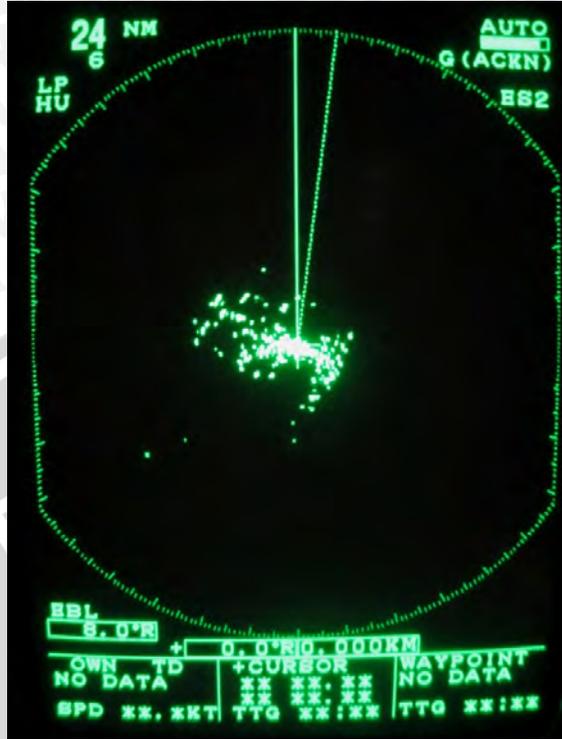
Gambar 5.19. Hasil Pengujian *Gain Control* Dengan *Gain* ¾ Skala



Gambar 5.20. Hasil Pengujian *Gain Control* Dengan *Gain* 1 Skala Penuh

Pada gambar 5.16 hingga 5.20 tampak bahwa jika nilai *gain* semakin besar, maka akan semakin banyak target yang muncul pada *radar map* termasuk juga *clutter*. Bahkan pada gambar 5.20 *rain clutter* tampak begitu jelas.

Tampilan *radar map* pada display radar Furuno pada kondisi *gain* minimum, seperempat skala, setengah skala, tiga perempat skala, dan satu skala penuh ditunjukkan pada gambar 5.21 hingga 5.25. Tampilan *radar map* ini jika dibandingkan dengan *radar map* hasil simulasi sistem *display* radar memiliki beberapa persamaan dan juga perbedaan.



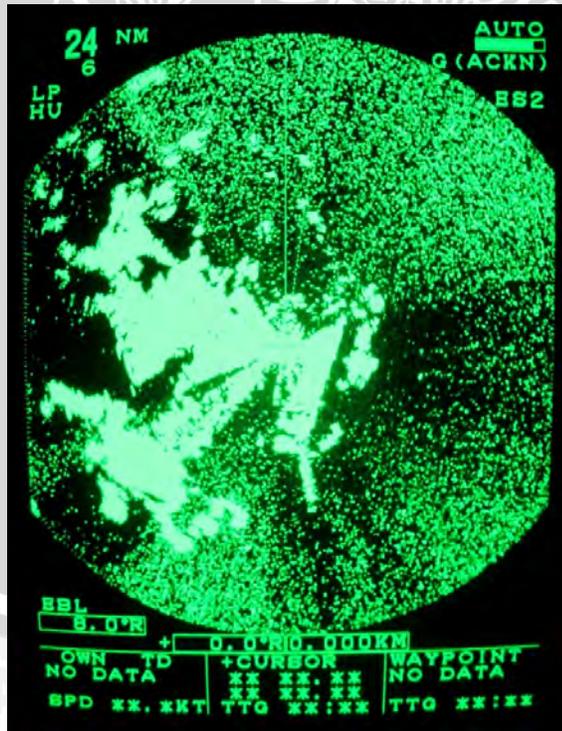
Gambar 5.21. Tampilan *Display* Radar Furuno Dengan *Gain* Minimum



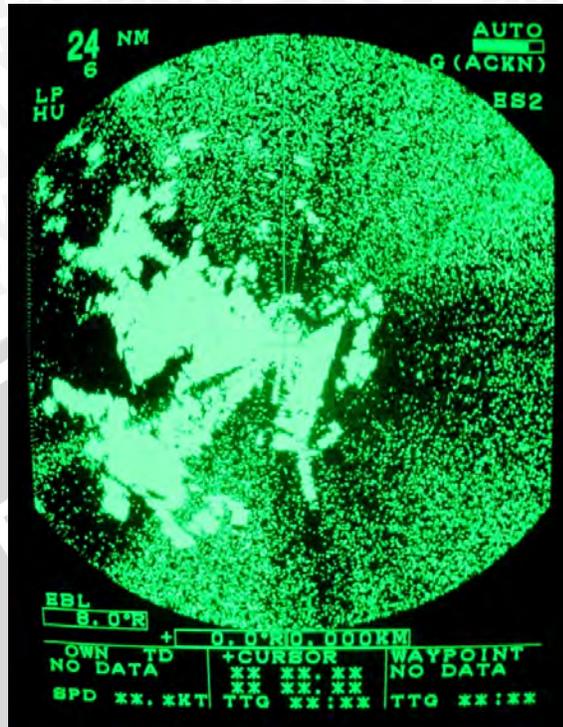
Gambar 5.22. Tampilan *Display* Radar Furuno Dengan *Gain* ¼ Skala



Gambar 5.23. Tampilan *Display* Radar Furuno Dengan *Gain* 1/2 Skala



Gambar 5.24. Tampilan *Display* Radar Furuno Dengan *Gain* 3/4 Skala

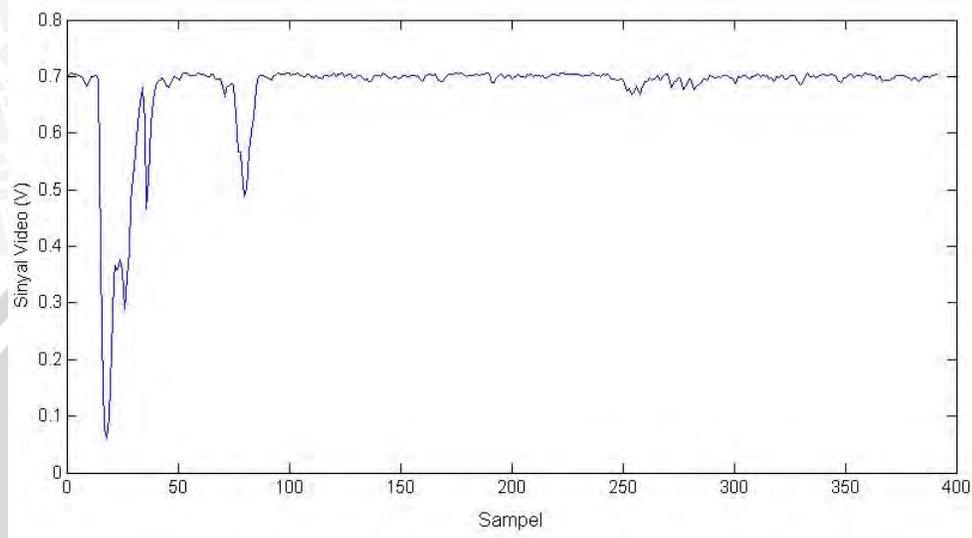


Gambar 5.25. Tampilan *Display* Radar Furuno Dengan *Gain* 1 Skala Penuh

Secara sekilas *radar map* hasil simulasi memiliki pola sebaran target yang sama dengan *radar map* pada *display* radar Furuno. Perbedaan *radar map* hasil simulasi dengan tampilan *display* radar Furuno adalah gradasi warna target sesuai kuat sinyal yang dipantulkannya. Namun ada perbedaan lain antara *display* radar buatan dengan *display* radar Furuno yaitu dalam skala *gain control*. Nilai *gain control* minimum pada *display* radar buatan lebih rendah dibanding nilai *gain control* minimum *display* radar Furuno.

Radar map hasil simulasi dengan *gain* setengah skala mirip dengan *radar map* pada *display* radar Furuno dengan *gain* minimum. *Radar map* hasil simulasi dengan *gain* tiga perempat skala mirip dengan *radar map* pada *display* radar Furuno dengan *gain* setengah skala. Sedangkan *radar map* hasil simulasi dengan *gain* satu skala penuh mirip dengan *radar map* pada *display* radar Furuno dengan *gain* tiga perempat skala.

Sinyal video dari data potong 550.000 dengan mengambil sampel dari urutan ke-305.200 hingga data ke-305.590, yaitu di sekitar *bearing* 200^0 , ditunjukkan pada gambar 5.26.



Gambar 5.26. Sinyal Video pada *Bearing* 200^0 .

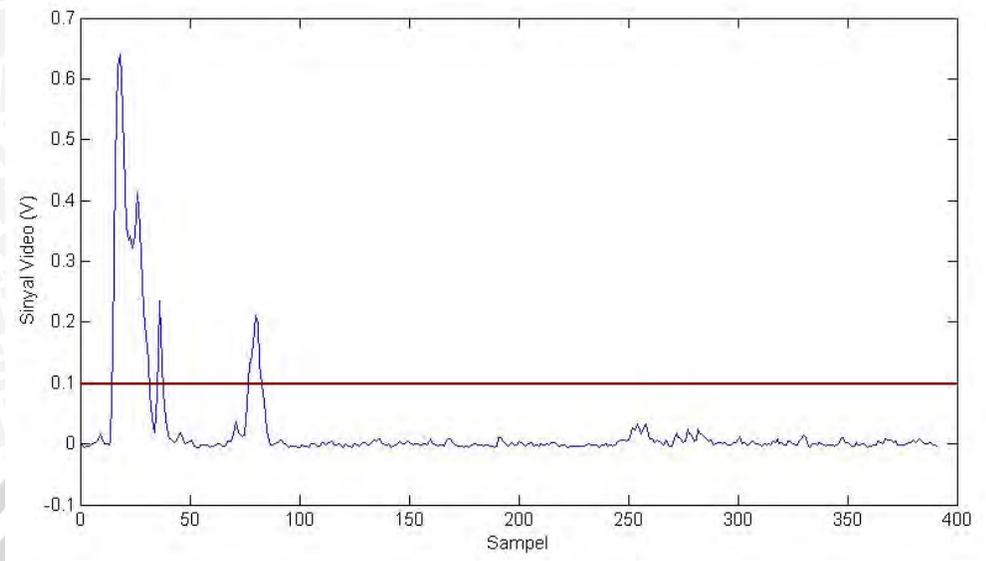
Seperti dijelaskan pada pembuatan *gain control* dalam bab 4, bahwa *gain* digunakan untuk mengatur *threshold* sinyal video. Berdasarkan perintah dalam program pengolah sinyal radar yang menyatakan hubungan *threshold* dengan *gain control*:

$$\text{threshold} = (1 - \text{prethd}) * 0.5;$$

maka misalkan nilai *gain control* adalah 0,8 maka *threshold* bernilai:

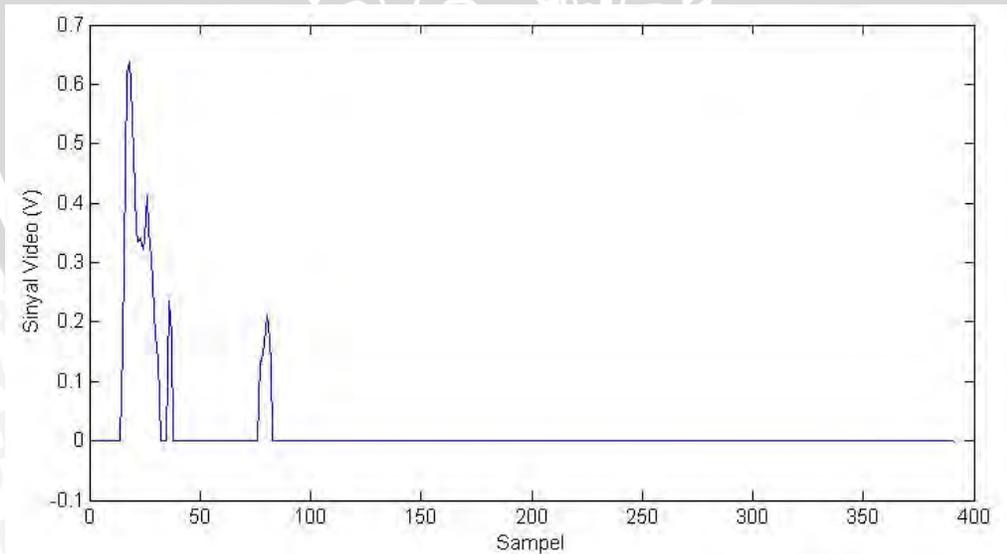
$$\text{threshold} = (1 - 0,8) \times 0,5 = 0,1$$

Sinyal video pada gambar 5.26 setelah dibalik (dimodifikasi hingga menjadi matriks praz) beserta posisi *threshold* dengan *gain control* 0,5 ditunjukkan pada gambar 5.27.



Gambar 5.27. Sinyal Video Dan *Threshold* 0,1

Pada gambar 5.27, sinyal video dengan amplitudo di atas *threshold* akan ditampilkan sebagai target pada *radar map*, sedangkan yang di bawah *threshold* tidak ditampilkan. Karena itu amplitudo sinyal video yang di bawah *threshold* akan diubah amplitudonya menjadi 0 V seperti terlihat pada gambar 5.28.

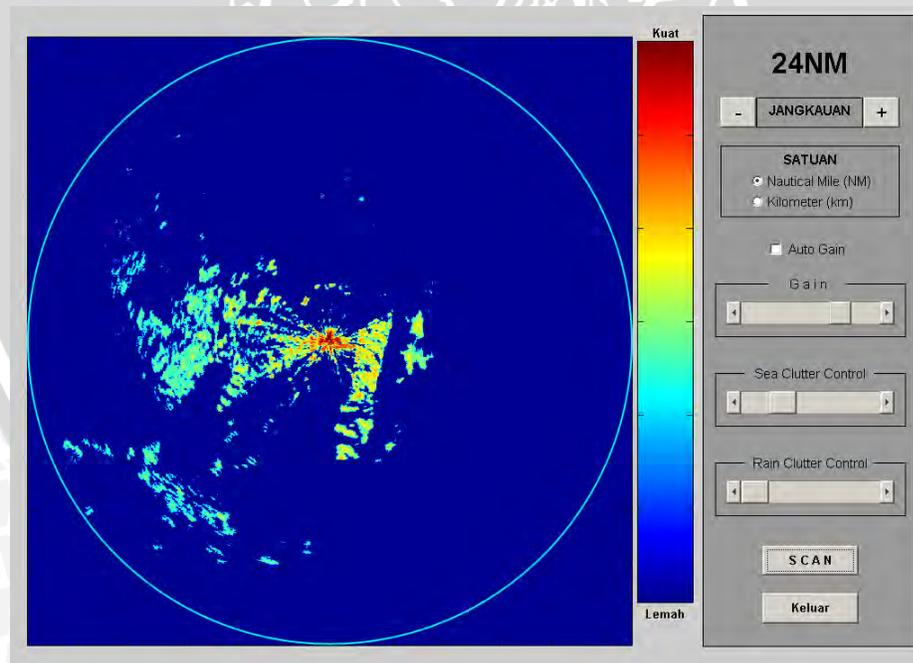


Gambar 5.28. Sinyal Video Hasil Seleksi *Threshold* 0,1

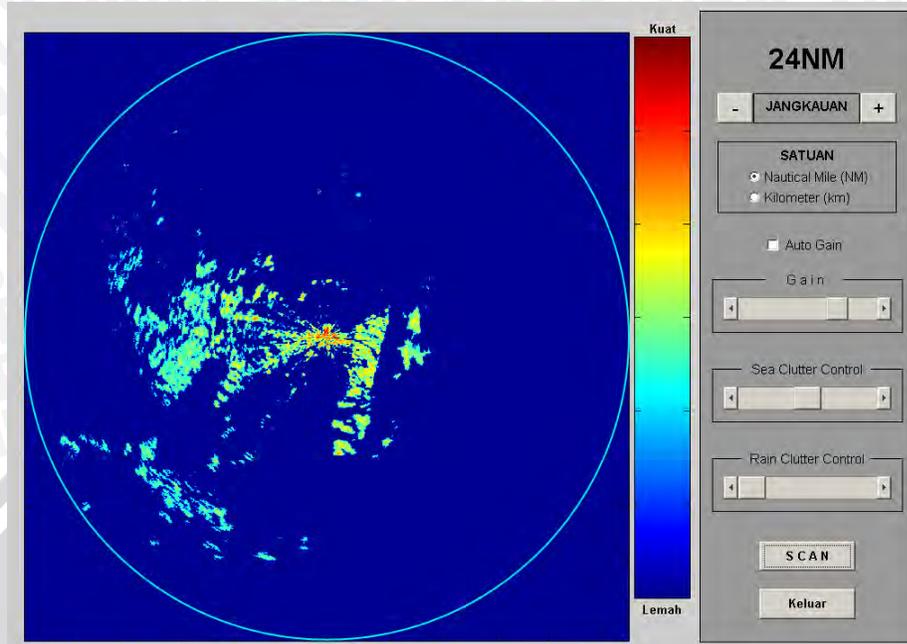
5.2.3 Pengujian *Sea Clutter Control*

Pengujian fitur *sea clutter control* dilakukan dengan simulasi sistem *display* radar menggunakan *sea clutter control* pada empat kondisi, yaitu pada seperempat skala, setengah skala, tiga perempat skala, dan satu skala penuh. Pengujian *sea clutter control* menggunakan jangkauan radar 24 NM. Hasil simulasi dengan empat kondisi *sea clutter control* ditunjukkan pada gambar 5.29 hingga 5.32.

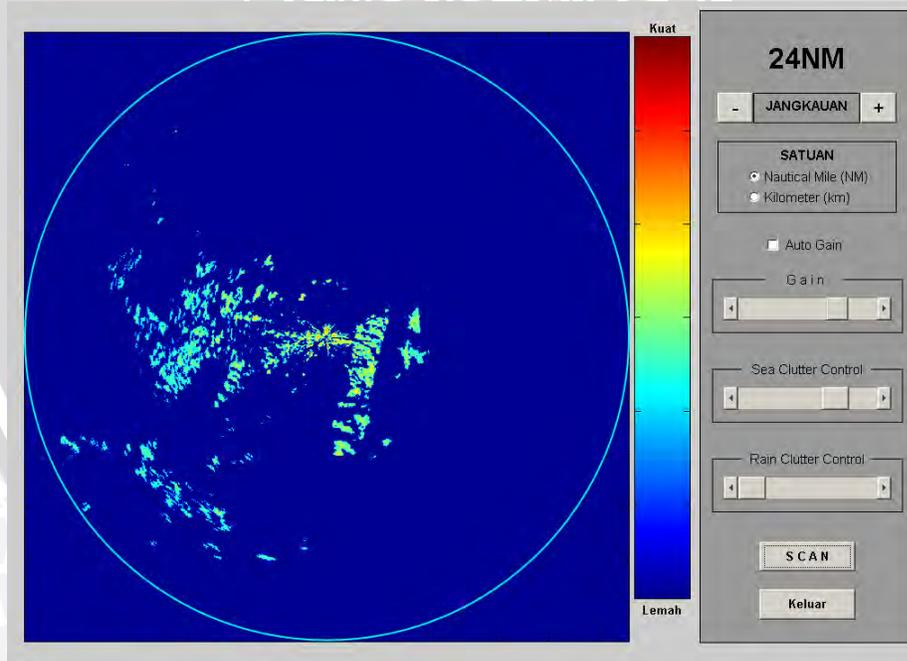
Dari gambar 5.29 hingga 5.32 tampak bahwa jika *sea clutter control* digunakan maka kuat sinyal dari target yang berada di dekat radar akan berkurang. Pengurangan kuat sinyal ini sebanding dengan semakin besarnya nilai *sea clutter control* yang digunakan. Pada simulasi dengan nilai *sea clutter control* yang besar, pengurangan kuat sinyal target juga akan mengenai target pada posisi yang jauh dengan posisi radar.



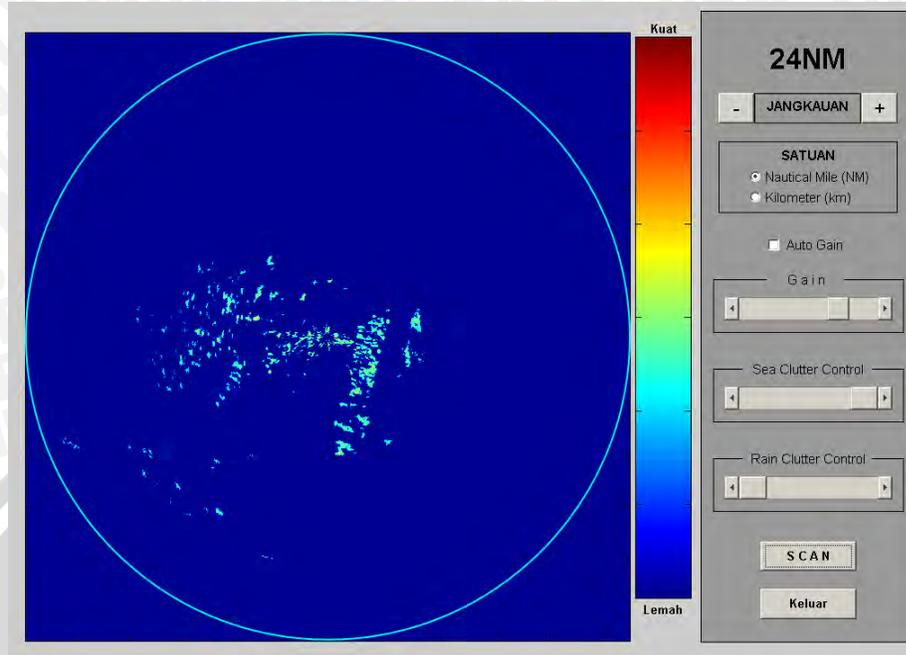
Gambar 5.29. Hasil Pengujian *Sea Clutter Control* Dengan Nilai $\frac{1}{4}$ Skala



Gambar 5.30. Hasil Pengujian *Sea Clutter Control* Dengan Nilai $\frac{1}{2}$ Skala



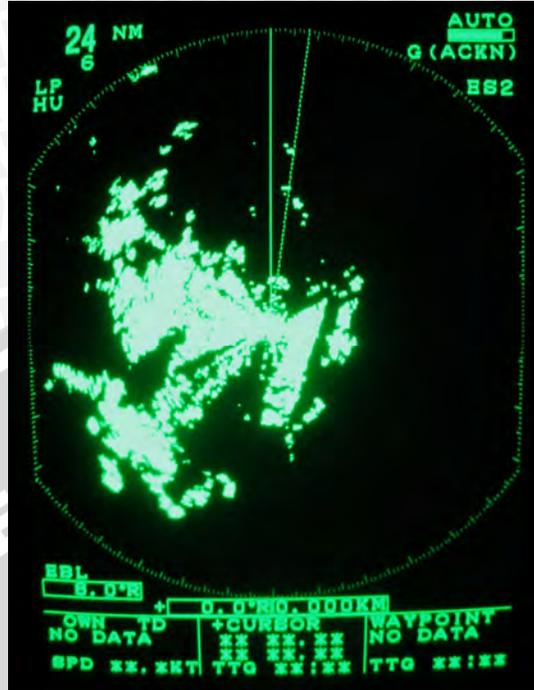
Gambar 5.31. Hasil Pengujian *Sea Clutter Control* Dengan Nilai $\frac{3}{4}$ Skala



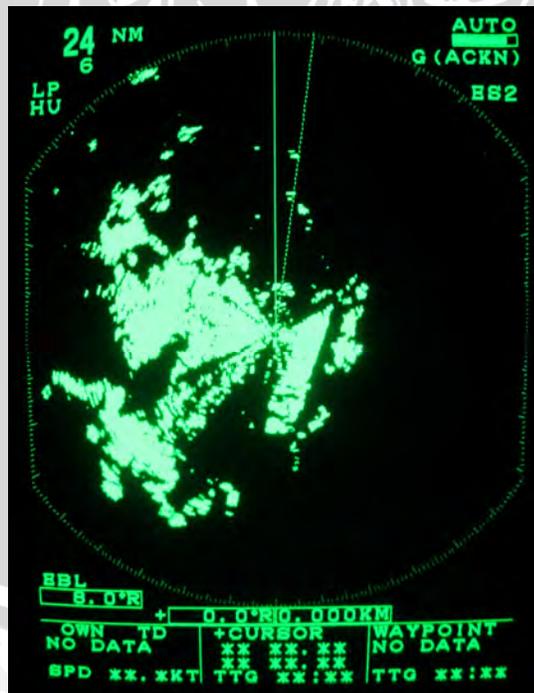
Gambar 5.32. Hasil Pengujian *Sea Clutter Control* Dengan Nilai 1 Skala

Penerapan *sea clutter control* pada *display* radar Furuno tidak menunjukkan efek yang signifikan pada saat pengujian. Ada beberapa kemungkinan yang bisa menyebabkan hal ini terjadi. Di antaranya adalah lokasi pengujian yang berada di wilayah perkotaan sehingga tidak didapatkan pola *clutter* seperti *sea clutter*. Kemungkinan yang lain adalah karena sinyal yang dipantulkan target pada jarak dekat cukup kuat sehingga tidak hilang meskipun dengan penggunaan *sea clutter control*. Pengurangan kuat sinyal target juga tidak bisa diamati karena *display* radar menampilkan *radar map* dengan warna monokrom.

Tampilan *radar map* pada radar Furuno dengan penggunaan A/C *Sea* ditunjukkan pada gambar 5.33 dan 5.34. Karena efek yang ditunjukkan dari penggunaan A/C *Sea* kurang nampak, maka gambar yang diambil hanya pada penggunaan A/C *Sea* setengah skala dan satu skala penuh.



Gambar 5.33. Tampilan *Display Radar Furuno* Dengan A/C Sea Setengah Skala



Gambar 5.34. Tampilan *Display Radar Furuno* Dengan A/C Sea Satu Skala Penuh

Seperti yang telah dijelaskan pada bab 4, bahwa *sea clutter control* mengurangi amplitudo sinyal video dengan menggunakan persamaan 2.3 dan dalam program pengolah sinyal radar, variabel pengurang sinyal video dituliskan dengan:

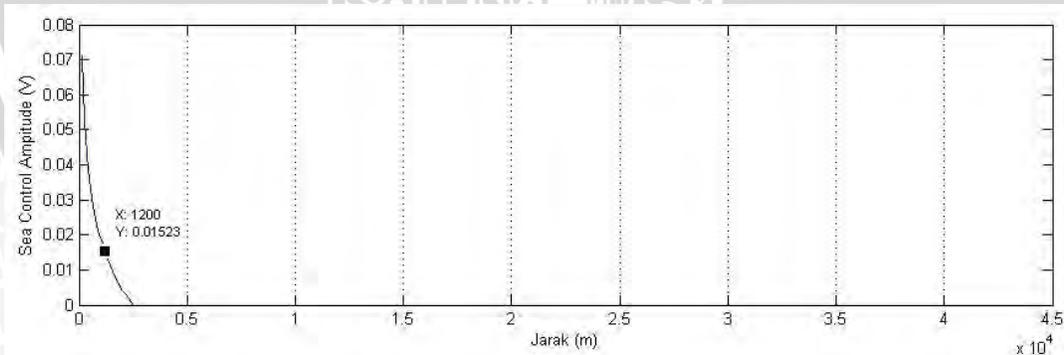
$$scthd = (\text{constsc} / ((\text{radardat}(\text{nodat}, 2) / \text{rangesc})^{\text{sc}})) - (0.25 - 0.5 * \text{sc});$$

Berdasarkan perintah di atas, dengan mengambil contoh pada jarak 1200 m dari posisi radar dan nilai *sea clutter control* (*sc*) = 0,1; konstanta “constsc” = 0,2; dan konstanta “rangesc” = 2500; maka nilai variabel “scthd” (pengurangan amplitudo sinyal video) adalah:

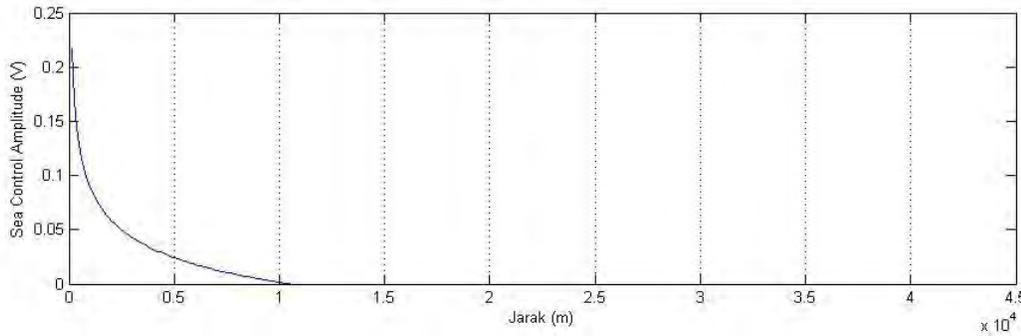
$$scthd = \left(\frac{0,2}{\left(\frac{1200}{2500} \right)^{0,1}} \right) - (0,25 - (0,5 \times 0,1))$$

$$scthd = 0,01523$$

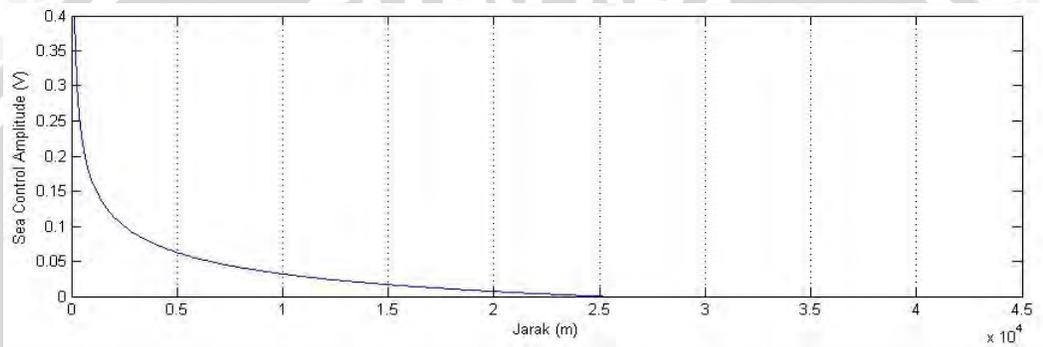
Nilai variabel “scthd” dari posisi radar hingga jarak 45 km jika ditampilkan dalam grafik (atau disebut kurva *sea control*) ditunjukkan pada gambar 5.35.



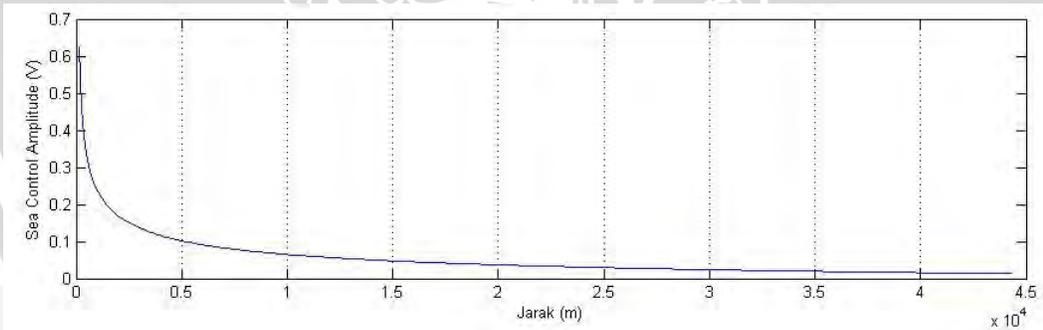
a). dengan nilai *sea clutter control* 0,1



b). dengan nilai *sea clutter control* 0,2

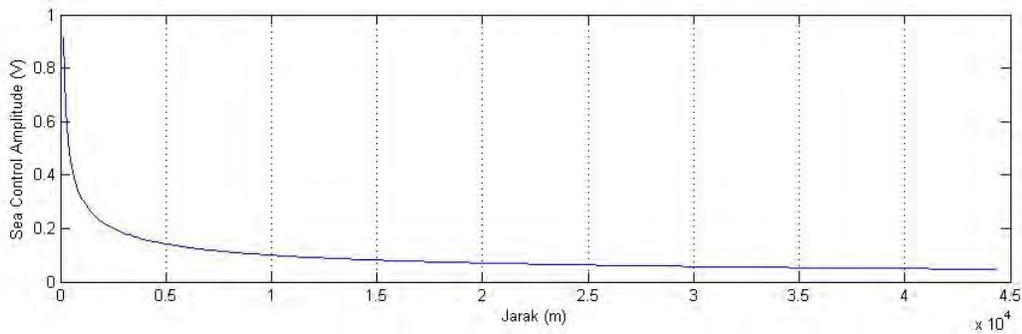


c). dengan nilai *sea clutter control* 0,3



d). dengan nilai *sea clutter control* 0,4





e). dengan nilai *sea clutter control* 0,5

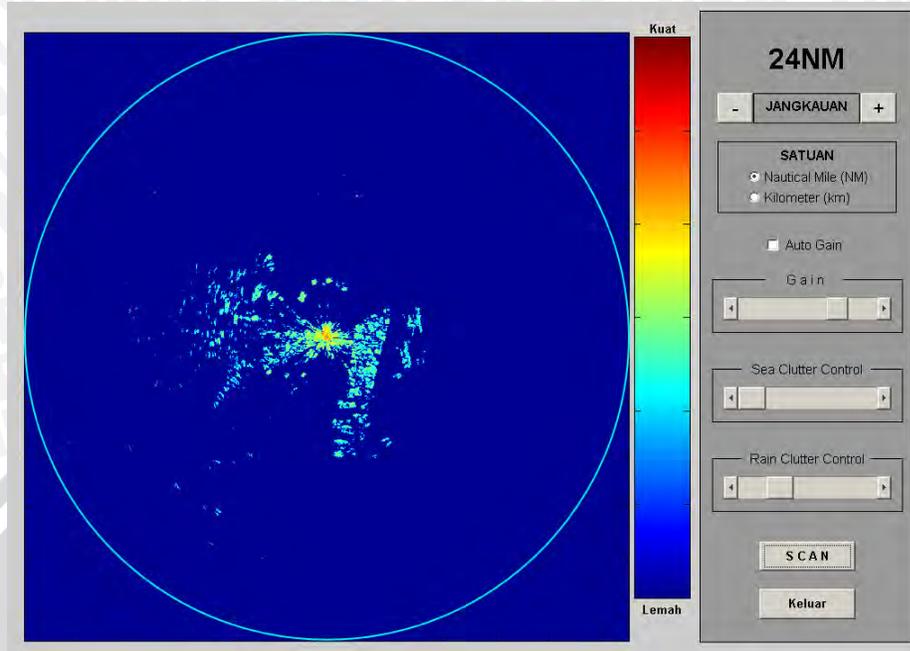
Gambar 5.35. Kurva *Sea Control*

Pada gambar 5.35 dapat dilihat pengurangan amplitudo sinyal video (*sea control amplitude*) bernilai besar pada area di dekat radar dan nilai turun seiring secara eksponensial seiring meningkatnya jarak. Dengan nilai *sea clutter control* yang semakin besar, *sea control amplitude* memiliki nilai yang semakin besar dan jangkauan efeknya juga semakin jauh.

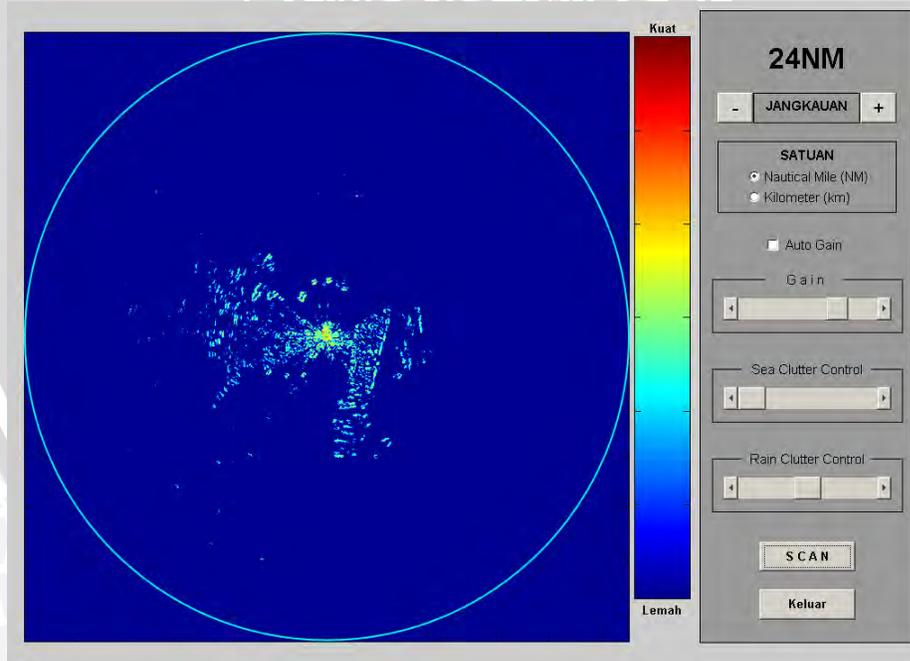
5.2.4 Pengujian *Rain Clutter Control*

Pengujian *rain clutter control* dilakukan dengan simulasi sistem *display* radar menggunakan *rain clutter control* pada empat kondisi. Yaitu pada nilai *rain clutter control* seperempat skala, setengah skala, tiga perempat skala, dan satu skala penuh. Simulasi *display* radar dengan *rain clutter control* menggunakan jangkauan radar 24 NM. Hasil simulasi dengan empat kondisi *sea clutter control* ditunjukkan pada gambar 5.36 hingga 5.39.

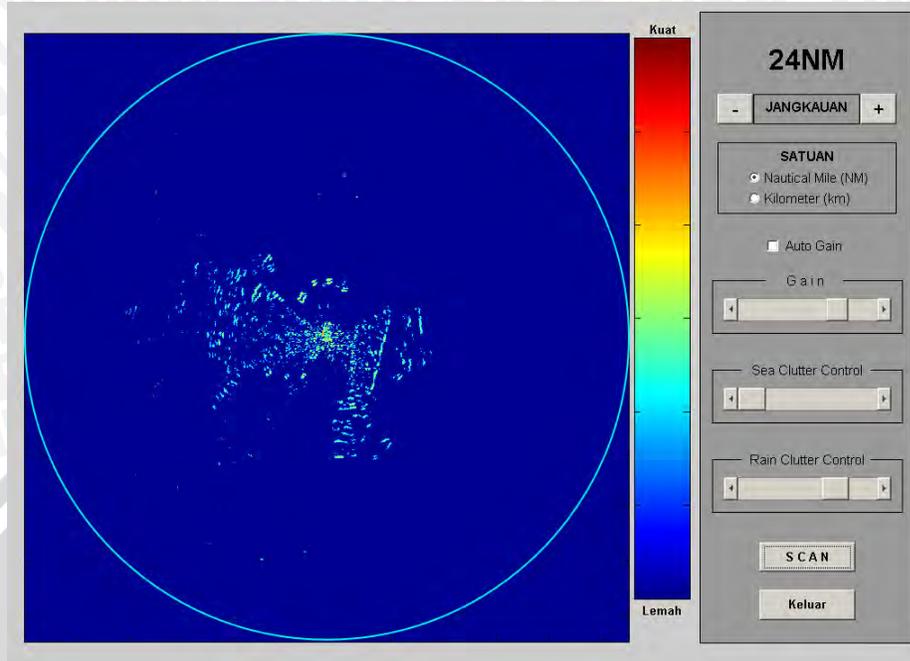
Dengan penggunaan *rain clutter control*, target dengan amplitudo yang hampir sama pada posisi yang berdekatan (sehingga membentuk seperti bidang) akan terkena efek *rain clutter control*. Yaitu target di tengah bidang akan dikurangi amplitudonya sedangkan yang di tepi bidang akan memiliki amplitudo yang tetap. Dengan demikian bidang target akan tampak seperti berlubang. Semakin besar nilai *rain clutter control*, maka pengurangan amplitudo pada target di tengah bidang akan semakin besar dan lubang pada tengah bidang juga akan bertambah lebar.



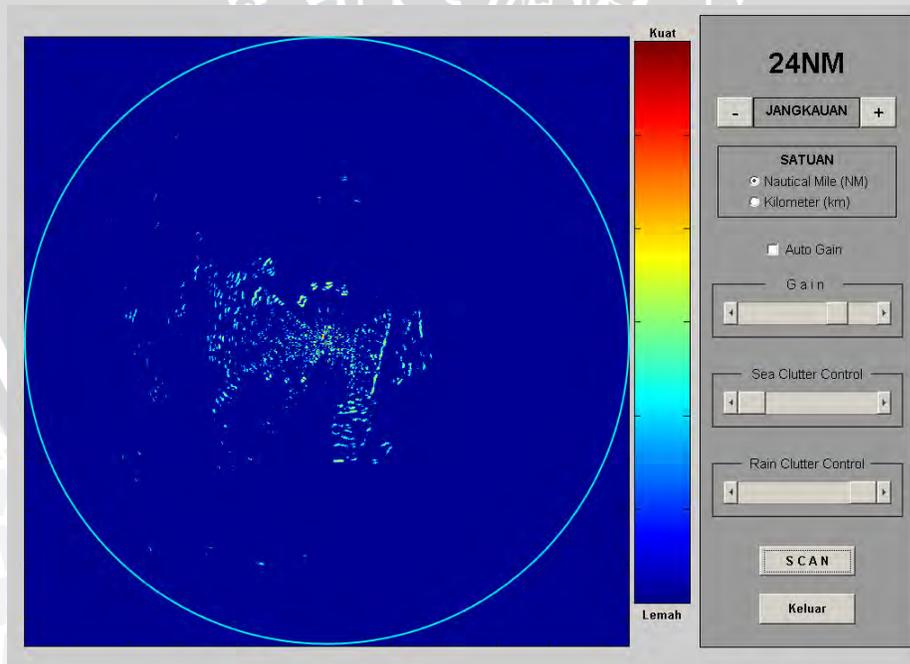
Gambar 5.36. Hasil Pengujian *Rain Clutter Control* Dengan Nilai $\frac{1}{4}$ Skala



Gambar 5.37. Hasil Pengujian *Rain Clutter Control* Dengan Nilai $\frac{1}{2}$ Skala



Gambar 5.38. Hasil Pengujian *Rain Clutter Control* Dengan Nilai $\frac{3}{4}$ Skala



Gambar 5.39. Hasil Pengujian *Rain Clutter Control* Dengan Nilai $\frac{3}{4}$ Skala

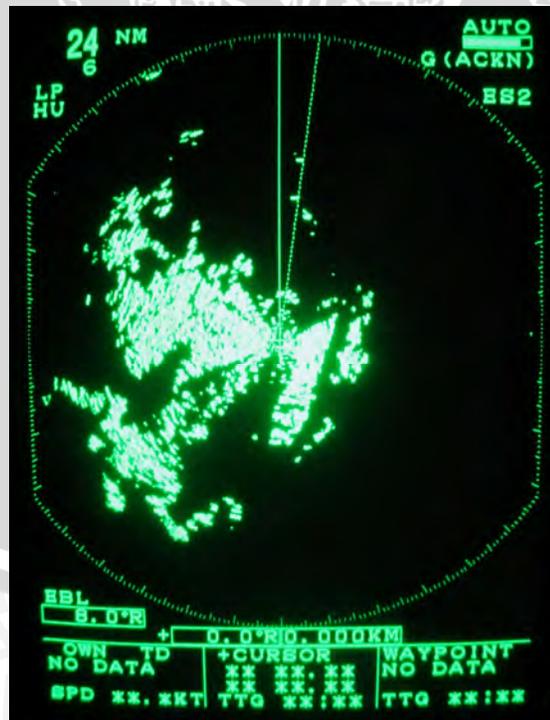
Pada *display* radar Furuno, penggunaan *A/C Rain* juga memberikan efek yang sama dengan penggunaan *rain clutter control* pada *display* radar buatan seperti yang dijelaskan sebelumnya. Hanya saja efek ini kurang begitu tampak jika *display* radar menggunakan jangkauan 24 NM. Tampilan *radar map* pada *display* radar Furuno dengan penggunaan *A/C Rain* ditunjukkan pada gambar 5.40 hingga 5.43.



Gambar 5.40. Tampilan *Display* Radar Furuno Dengan *A/C Rain* ¼ Skala



Gambar 5.41. Tampilan *Display Radar Furuno* Dengan A/C *Rain 1/2* Skala

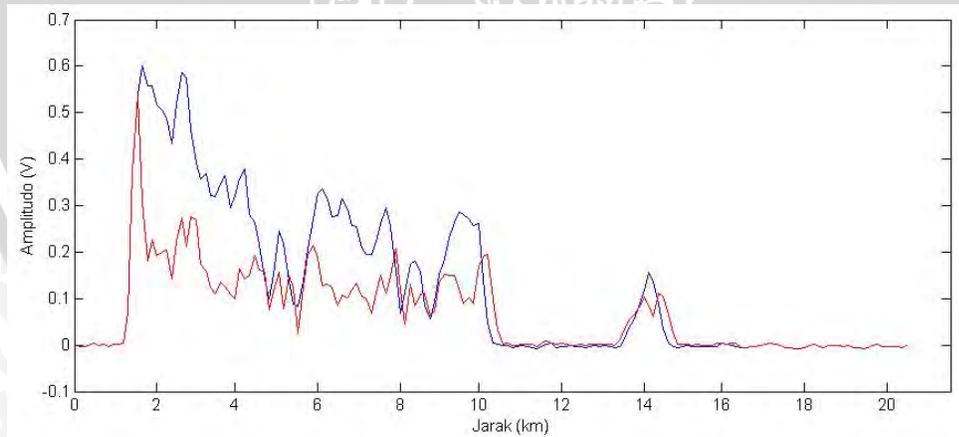


Gambar 5.42. Tampilan *Display Radar Furuno* Dengan A/C *Rain 3/4* Skala



Gambar 5.43. Tampilan *Display Radar Furuno Dengan A/C Rain 1 Skala*

Contoh bentuk sinyal video hasil penggunaan *rain clutter control* ditunjukkan pada gambar 5.44. Sinyal tersebut diambil pada *bearing 90°* dengan penggunaan *gain control 0,8 skala* dan *rain clutter control 0,5 skala*.



— : Sinyal Video — : Sinyal Video dengan *Rain Clutter Control*

Gambar 5.44. Sinyal Video Hasil Penggunaan *Rain Clutter Control*

5.2.5 Waktu Simulasi

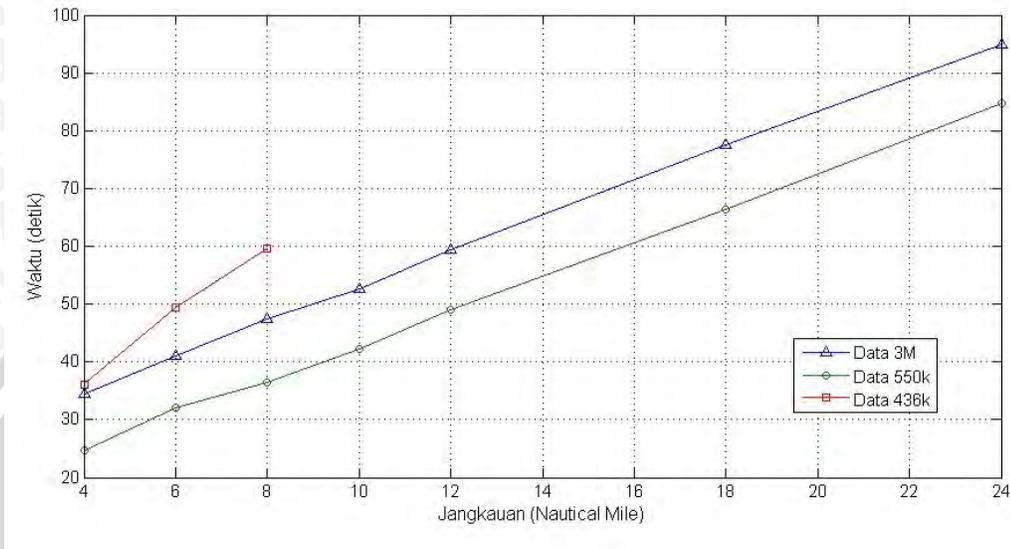
Simulasi sistem *display* radar menggunakan laptop dengan spesifikasi *processor* Intel Core Duo T2400 (2 MB L2 *cache*, 1.80 GHz, 667 MHz FSB) dan *memory* 2 GB DDR2. Waktu yang diperlukan untuk satu simulasi ditunjukkan pada tabel 5.1.

Tabel 5.1

Waktu Yang Diperlukan Untuk Simulasi *Display* Radar

Jumlah Sampel	Jangkauan (NM)	Frekuensi <i>Sampling</i>	Waktu Yang Diperlukan (detik)
3.000.000	24	1,25 MHz	94,89
	18		77,5
	12		59,43
	10		52,65
	8		47,53
	6		41,11
	4		34,47
550.000 (data potong)	24	1,25 MHz	84,82
	18		66,39
	12		49,07
	10		42,33
	8		36,53
	6		31,02
	4		24,6
436.000 (data potong)	8	2,5 MHz	59,68
	6		49,45
	4		36,12

Tabel 5.1 jika ditampilkan dalam grafik seperti diperlihatkan pada gambar 5.45.



Gambar 5.45. Grafik Waktu Simulasi

Berdasarkan grafik pada gambar 5.45, waktu yang diperlukan untuk simulasi sistem *display* radar sebanding dengan jangkauan radar yang digunakan. Semakin besar jangkauan radar maka waktu yang diperlukan semakin lama. Dengan menggunakan data yang dipotong, waktu yang diperlukan untuk simulasi menjadi semakin singkat. Perbandingan besar data yang telah dipotong dengan data asli adalah 550.000 : 3.000.000 atau jika disederhanakan menjadi 1 : 5,45. Dengan perbandingan tersebut waktu simulasi menjadi lebih cepat pada tiap jangkauan dengan selisih rata-rata 10,26 detik.

BAB VI PENUTUP

6.1 KESIMPULAN

Berdasarkan analisis hasil simulasi yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

1. *Display* radar yang dibuat dengan Matlab dapat menampilkan citra *radar map* dengan gradasi warna sesuai kuat sinyal radar yang dipantulkan target. Hal ini dapat membantu pengenalan target, terutama target dengan sinyal yang lebih kuat dibanding *clutter* di sekitarnya.
2. *Sampling* sinyal radar dapat dilakukan dengan frekuensi *sampling* 1,25 MHz dan jumlah sampel 3 juta. Hasil sampel berupa variabel data yang merupakan matriks dua kolom.
3. Metode pengurangan *clutter* pada *marine radar* adalah menggunakan *gain control*, *sea clutter control* dan *rain clutter control*.
4. *Gain control* dapat digunakan untuk mengatur *radar map* yang dihasilkan oleh *display* agar menampilkan target dengan baik tanpa menyertakan *clutter*. *Gain control* mempengaruhi nilai *threshold* sinyal. Jika *gain* semakin besar maka *threshold* akan semakin kecil sehingga akan banyak target yang muncul termasuk juga *clutter*.
5. *Sea clutter control* dapat mengurangi amplitudo sinyal pantul pada area dekat radar di mana pada area ini *sea clutter* paling banyak muncul. Sayangnya karena simulasi dilakukan di perkotaan, hasil simulasi kurang *reliable*. Namun penggunaan *sea clutter control* untuk mengurangi amplitudo sinyal pantul berdasarkan jarak dari radar dapat berfungsi sesuai dengan yang diharapkan.
6. *Rain clutter control* dapat meredam *rain clutter* dengan mengurangi amplitudo sinyal pantul pada daerah dengan amplitudo yang hampir seragam. Hasil penggunaan *rain clutter control* akan membuat target yang luas pada radar map akan terlihat berlubang di tengahnya.

7. Waktu yang diperlukan untuk simulasi *display* radar hingga menghasilkan satu *radar map* penuh cukup lama dibandingkan waktu *scanning* radar satu rotasi penuh. Hal ini dikarenakan keterbatasan kemampuan perangkat keras. Sehingga simulasi hanya bisa dilakukan secara *non real-time*.

6.2 SARAN

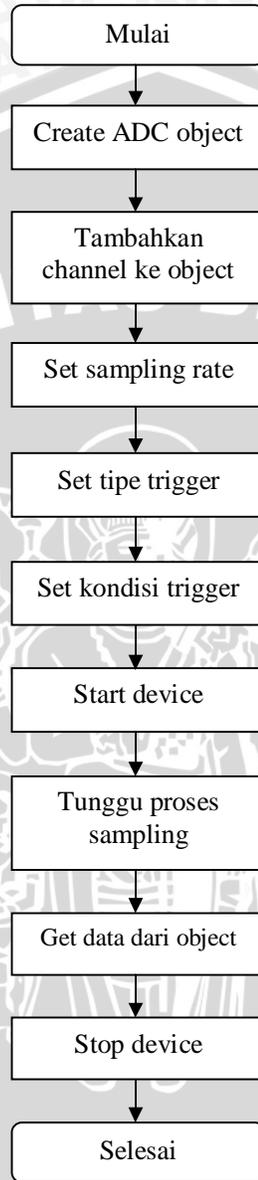
Saran yang diberikan adalah:

1. Perlu dikembangkan sistem *display* radar dengan spesifikasi *hardware* yang tinggi, misalnya ADC dengan kecepatan *sampling* yang lebih tinggi, komputer dengan spesifikasi yang lebih tinggi, agar sistem *display* dapat bekerja dengan lebih cepat atau bahkan secara *real time*.
2. Membuat mekanisme *clutter control* yang lebih baik, misalnya dengan mengembangkan persamaan yang digunakan pada *clutter control*, atau menggunakan sistem *clutter control* adaptif dan mengujinya hingga dapat memberikan hasil yang lebih baik atau pun dapat mempercepat proses *mapping*.
3. Mencoba membuat GUI *display* radar dengan perangkat lunak yang lain yang dapat dieksekusi sendiri.
4. Melakukan pengujian di laut, pantai, atau pelabuhan agar data yang didapat lebih bervariasi. Untuk mendapat data radar dengan *sea clutter* yang nyata pun sebaiknya dilakukan pengujian di wilayah laut. Dengan demikian kualitas pengujian akan lebih baik.

DAFTAR PUSTAKA

- Awaludin, Asif, 2013. *Desain dan Implementasi Sistem Pengolahan Sinyal Radar Cuaca Berbasis Radar Pulsa Standar*, Bandung: Tesis Program Studi Teknik Elektro STEI-ITB.
- Furuno, 2001. *Furuno Operator's Manual, Marine Radar Model 1832/1932/1942*. Jepang: Furuno Electric Co., Ltd.
- , 2008. *Furuno Operator's Guide to Marine Radar*. USA: Furuno USA.
- Hayt, William H., John A. Buck. 2001. *Engineering Electromagnetics*. USA: McGraw-Hill.
- Irawan, Feriza A. 2012. *Buku Pintar Pemrograman MATLAB*. Yogyakarta: Mediakom.
- Knight, Andrew, 2000. *Basic of MATLAB and Beyond*. USA: CRC Press.
- Mahafza, Bassem R., 2008. *Radar Signal Analysis and Processing Using MATLAB*. USA: CRC Press.
- Marchand, Patrick. 2003. *Graphics and GUIs with MATLAB*. USA: CRC Press.
- Meikle, Hamish, 2008. *Modern Radar System, Second Edition*. London: Artech House.
- Wagener, Paul, 2011. *Real-time Realistic Radar Simulation*. Belanda: Tesis Eindhoven University of Technology.
- Wang, Bu-Chin, 2008. *Digital Signal Processing Techniques and Application in Radar Image Processing*. USA: John Wiley and Sons, Inc.
- Watts, Simon, 2006. *Sea Clutter: Scattering, The K-distribution and Radar Performance*. London: The Institution of Engineering and Technology.

Lampiran 1. Diagram Alir *Sampling* Sinyal Radar Dengan ADC ADLINK dan MATLAB



Lampiran 2. Source Code Matlab Untuk Sampling Sinyal Menggunakan

Adlink PCI-9810

```
close all;
clear all;
clc;
%Name: AD_AI_HwDigital_Trigger.m
%Description: This sample program shows how to execute a
continuous analog
%   input with the 'HwAnalogChannel' trigger type. The data
%   acquisitionoperation executes while the tigger signal was
%   detected at the external digital trigger pin.
%Support cards: 9111, 9112, 9113, 9114, 9116, 9118, 9810, 9812,
9820, 2005,
%   2006, 2010, 2204, 2205, 2206, 2208, 2213, 2204, 2501,
2502
%Copyright 2005, ADLINK Inc.

ret = 0;

%Create an analog input object. 0 stands for the first ADLINK
device
%installed.
ai_device = analoginput('mwadlink', 0);

%Add channel#0 to ai_device object.
ai_channel = addchannel(ai_device, 0:1);
%videoku=zeros(1500,100);
%headku=zeros(1500,100);

%Set properties of ai_device.
try
    %Set 'samplerate' of ai_device to 1000Hz. The valid value of
    %;¥samplerate;| is hardware-dependent.
    set(ai_device, 'SampleRate', 1250000);

    %Set 'samplesPerTrigger' of ai_device to 999 Sample 1000
    points
    %after each trigger asserted.
    set(ai_device, 'SamplesPerTrigger', 3000000);

    %Set 'TriggerType' of ai_device to 'HwDigital'. Execute data
    %acquisition after the tigger signal was detected at the
    external
    %digital trigger
    pin.
    set(ai_device, 'TriggerType', 'HwDigital');

    %set 'TriggerCondition' of ai_device to 'TriggerPositive'.
    set(ai_device, 'TriggerCondition', 'TriggerNegative');
    %set(ai_device, 'TriggerDelayUnits', 'TriggerNegative');
    %set(ai_device, 'TriggerDelay', 'TriggerNegative');
```

```
%Start the data acquisition operation according to the
setting of ai_device.
start(ai_device);

catch
    %Display a message when error occurs.
    display('System error or this card not support!')

    ret = -1
end
%Wait for the completion of analog input execution.
%for i=1:1:100
%try
%    start(ai_device);
%catch
%    %If timeout occurs, display a error message
%    display('System error or this card not support!')
%    ret = -1;
%end

try
    %Wait for the completion of ai_device object and set timeout
value to 5
    %seconds.
    wait(ai_device, 10);
catch
    %If timeout occurs, display a error message
    display('Timeout!')
    ret = -1;
end

%get and plot data
if(ret == 0)
    %Get the data from ai_device object
    data = getdata(ai_device);
    %videoku(:,i)=data(:,1);
    %headku(:,i)=data(:,2);
    %plot data
    %plot(data);
else
    %If error occurs, stop the execution of ai_device.
    stop(ai_device);
end
%end
%delete objects and free memory.
delete (ai_channel);%delete ai_channel
clear ai_channels;%clear ai_channels
delete (ai_device);%delete ai_device
clear ai_device;%clear ai_device
```

Lampiran 3. Source Code Matlab Untuk GUI Display Radar

```

%GUI untuk display radar
close all
clear all
clc

%Window Utama
win1 = figure('units','points','position',[0 10 770 570],...
    'menubar','none','resize','off','numbertitle','off','name',...
    'Display Radar');

%//////// Grup UI Control////////

grafik1 = axes('parent',win1,'units','pixel','position',...
    [20 30 750 680]);
axis off;
frm_bwh = uicontrol('parent',win1,'units','points','position',...
    [12 18 510 2],'style','frame');
frm_kanan = uicontrol('parent',win1,'units','points', ...
    'position',[520 20 2 516],'style','frame');
frm_kiri =
uicontrol('parent',win1,'units','points','position',...
    [12 20 2 516],'style','frame');
frm_atas =
uicontrol('parent',win1,'units','points','position',...
    [12 534 510 2],'style','frame');
%Grup Colorbar
ket_warna = colorbar('parent',win1,'yticklabel',[]);
clrbar1 = uicontrol('units','points','position',[525 43 46
11],...
    'style','text','string','Lemah','fontsize',10,...
    'fontweight','bold');
clrbar2 = uicontrol('units','points','position',...
    [525 530 46 11],'style','text','string','Kuat',...
    'fontsize',10,'fontweight','bold');

%Frame CONTROL PANEL////////
frm_cp = uicontrol('parent',win1,'units','points','position',...
    [580 20 180 530],'style','frame','backgroundcolor',...
    [0.6 0.6 0.6]);

%Grup Pengatur Jangkauan
tombol_min = uicontrol('parent',win1,'units','points',...
    'position',[595 455 30 26],'style','pushbutton',...
    'string','-','callback','kurangi_range','fontsize',15);
tombol_plus = uicontrol('parent',win1,'units','points',...
    'position',[715 455 30 26],'style','pushbutton','string',...
    '+','callback','tambah_range','fontsize',15);
frm_jangk = uicontrol('parent',win1,'units','points',...
    'position',[625 455 89 26],'style','frame',...
    'backgroundcolor',[0.6 0.6 0.6]);
jarak_teks = uicontrol('units','points','position',...

```

```
[629 456 80 20], 'style', 'text', 'string', 'JANGKAUAN', ...
    'fontsize', 11, 'backgroundcolor', [0.6 0.6 0.6], ...
    'fontweight', 'bold');

jangk=4;           %default jangkauan
r_min = 15;
r_max = 170;
r_pengali = 120;
disp_jangk = uicontrol('units','points','position',...
    [629 500 80 25], 'style','text','string','10NM',...
    'fontsize', 24, 'backgroundcolor', [0.6 0.6 0.6], ...
    'fontweight', 'bold');

%satuan jangkauan
frm_gain = uicontrol('parent', win1, 'units', 'points', ...
    'position', [595 380 150 60], 'style', 'frame', ...
    'backgroundcolor', [0.6 0.6 0.6]);
h_satuan = uicontrol('units', 'points', 'position', ...
    [624 410 90 25], 'style', 'text', 'string', 'SATUAN', ...
    'fontsize', 11, 'backgroundcolor', [0.6 0.6 0.6], ...
    'fontweight', 'bold');
sat_nm = 1; %satuan default nm
h_nm = uicontrol('units', 'points', 'position', [620 402 115 17], ...
    'string', 'Nautical Mile (NM)', 'fontsize', 11, ...
    'style', 'radiobutton', 'value', 1, 'backgroundcolor', ...
    [0.6 0.6 0.6], 'callback', 'satuan_nm');
h_km = uicontrol('units', 'points', 'position', [620 385 115 17], ...
    'string', 'Kilometer (km)', 'fontsize', 11, 'style', ...
    'radiobutton', 'value', 0, 'backgroundcolor', [0.6 0.6 0.6], ...
    'callback', 'satuan_km');

%Auto Gain
aut_gain = uicontrol('units', 'points', 'position', ...
    [635 345 115 17], 'string', ' Auto Gain', 'fontsize', 11, ...
    'style', 'checkbox', 'value', 0, 'backgroundcolor', [0.6 0.6
0.6]);

%GAIN Control Group
frm_gain = uicontrol('parent', win1, 'units', 'points', ...
    'position', [590 280 160 45], 'style', 'frame', ...
    'backgroundcolor', [0.6 0.6 0.6]);
gain_sld = uicontrol('parent', win1, 'units', 'points', ...
    'position', [600 290 140 20], ...
    'style', 'slider', 'sliderstep', [.05 .2], 'max', 1, 'min', 0);
gain_teks = uicontrol('units', 'points', 'position', ...
    [635 316 65 15], 'style', 'text', 'string', 'G a i n', ...
    'fontsize', 11, 'backgroundcolor', [0.6 0.6 0.6], ...
    'fontweight', 'demi');

%SEA Clutter Control Group
frm_sc = uicontrol('parent', win1, 'units', 'points', 'position', ...
    [590 205 160 45], 'style', 'frame', 'backgroundcolor', ...
    [0.6 0.6 0.6]);
```

```
sea_sld = uicontrol('parent',win1,'units','points','position',
...
    [600 215 140 20], 'style','slider','sliderstep',[.1 .25], ...
    'max',1, 'min',0);

sea_teks = uicontrol('units','points','position', ...
    [615 241 108 15], 'style','text','string', ...
    'Sea Clutter Control', 'fontsize',11,...
    'backgroundcolor',[0.6 0.6 0.6], 'fontweight','demi');

%RAIN Clutter Control Group
frm_rc = uicontrol('parent',win1,'units','points','position', ...
    [590 130 160 45], 'style','frame', ...
    'backgroundcolor',[0.6 0.6 0.6]);
rain_sld = uicontrol('parent',win1,'units','points', ...
    'position',[600 140 140 20], 'style','slider', ...
    'sliderstep',[.1 .25], 'max',1, 'min',0);
rain_teks = uicontrol('units','points','position', ...
    [615 166 108 15], 'style','text','string', ...
    'Rain Clutter Control', 'fontsize',11,...
    'backgroundcolor',[0.6 0.6 0.6], 'fontweight','demi');

%Tombol eksekusi
tmbldscan = uicontrol('parent',win1,'units','points', ...
    'position',[630 80 80 25], 'style','pushbutton', ...
    'string','S C A N', 'callback','imageproc_total', ...
    'fontweight','bold', 'fontsize',10);
tmbldkeluar = uicontrol('parent',win1,'units','points', ...
    'position',[630 40 80 25], 'style','pushbutton', 'string', ...
    'Keluar', 'callback','close', ...
    'fontweight','bold', 'fontsize',10);
```

Lampiran 4. Source Code Matlab Untuk Pengolah Sinyal Radar

```

%program pemroses sinyal radar
clc;
if jangk<4
    load dataptng_436k.mat           %dengan frek sampling 2,5MHz
else
    load datapotong_550000.mat      %dengan frek sampling 1,25MHz
end
newtrig=0;
num=0;
nodat=1;
jmlbaris=length(data);
radardat=zeros(jmlbaris,2);        %koord polar(sudut, jarak)
praz=zeros(jmlbaris,1);           %matriks utk manipulasi
sinyal video
autogain=get(aut_gain, 'value');
if autogain==1
    ratavid=mean(data(1:jmlbaris,1));
    praz=ratavid-data(1:jmlbaris,1);
    threshold=0.1;
else
    praz=0.7-data(1:jmlbaris,1);
    prethd=get(gain_sld, 'value'); %input gain (0-1)
    threshold=(1-prethd)*0.5;     %threshold yg dipakai dlm
program
end
praz2=praz(:,1);
radz=zeros(jmlbaris,1);          %sinyal video yg akan
ditampilkan
constsc=0.2;                    %konstanta sea control
sc=(get(sea_sld, 'value'))*0.5;  %input sea control (0-0.5)
rangesc=2500;                   %pembagi jarak
inputrc=get(rain_sld, 'value');
rc=inputrc*0.6+0.4;
if inputrc==0
    rc=0;
end
for k=1:1:jmlbaris
    if k > 1
        if data(k,2)-data(k-1,2)>0.2 %mendeteksi tepi naik
            trigger
                if newtrig==0
                    newtrig=1;
                    nodat=nodat+1;
                    radardat(nodat,1)=radardat(nodat-1,1)+0.26666;
                    radardat(nodat,2)=0;
                    radz(nodat,1)=0.5; %posisi radar pada map
                    num=0;
                else
                    num=num+1;

```

```

if num>r_min           %range minimum radar
    nodat=nodat+1;
    radardat(nodat,1)=radardat(nodat-1,1);
    radardat(nodat,2)=(num-r_min)*r_pengali;
    differ=abs(praz2(k+1,1)-praz2(k-2,1));
    praz(k,1)=(praz(k,1)*(1-rc))+differ*rc;

scthd=(constsc/((radardat(nodat,2)/rangesc)^sc))-(0.25-0.5*sc);
if scthd<0
    scthd=0;
end
praz(k,1)=praz(k,1)-scthd;
if praz(k,1)<=threshold
    radz(nodat,1)=0;
else
    radz(nodat,1)=praz(k,1)+0.1;
end
end
else
    num=num+1;
    if num>r_min
        if num<=r_max   %sesuai jangkauan radar yg
dipilih
            nodat=nodat+1;
            radardat(nodat,1)=radardat(nodat-1,1);
            radardat(nodat,2)=(num-r_min)*r_pengali;
            differ=abs(praz2(k+1,1)-praz2(k-2,1));
            praz(k,1)=(praz(k,1)*(1-rc))+differ*rc;

scthd=(constsc/((radardat(nodat,2)/rangesc)^sc))-(0.25-0.5*sc);
if scthd<0
    scthd=0;
end
praz(k,1)=praz(k,1)-scthd;
if praz(k,1)<=threshold
    radz(nodat,1)=0;
else
    radz(nodat,1)=praz(k,1)+0.1;
end
if num==r_max           %border
    radz(nodat,1)=0.4;
end
end
end
newtrig=0;
end
else
    radardat(nodat,1)=0;   %data pertama
    radardat(nodat,2)=0;
    radz(nodat,1)=0.5;
end
end;

```

```
radx=radardat(:,2).*cosd(radardat(:,1)-90);
rady=radardat(:,2).*sind(radardat(:,1)-90);
xmax=max(radx);
xmin=min(radx);
ymin=min(rady);
ymax=max(rady);
[XI,YI]=meshgrid(linspace(xmin,xmax,1152),linspace(ymin,ymax,1152
));
ZI = griddata(radx,rady,radz,XI,YI);
set(win1,'CurrentAxes',grafik1);

clims = [0 0.6]; %range skala warna untuk target
imagesc(ZI,clims);
colormap(jet)
set(gca,'xticklabel',[])
set(gca,'yticklabel',[])
colorbar(ket_warna);
colorbar('yticklabel',[]);
```



Lampiran 5. Source Code Matlab Untuk Menambah Jangkauan Radar

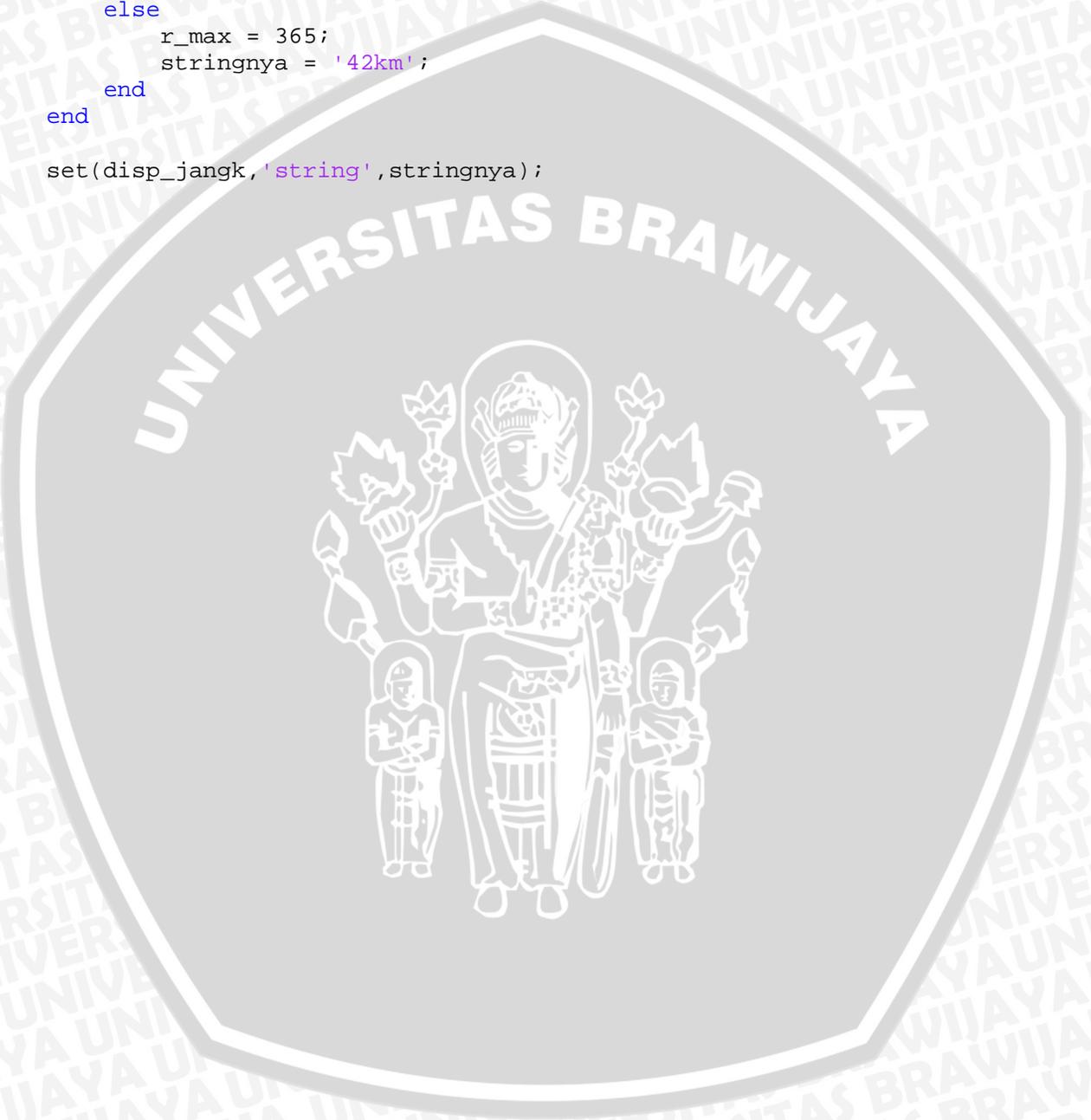
```
%Fungsi Penambah Jangkauan
```

```
if jangk<7
    jangk=jangk+1;
end

if jangk<4
    r_min = 30;
    r_pengali = 60;
else
    r_min = 15;
    r_pengali = 120;
end

if sat_nm == 1
    if jangk==1
        r_max = 155;
        stringnya = '4NM';
    elseif jangk==2
        r_max = 216;
        stringnya = '6NM';
    elseif jangk==3
        r_max = 278;
        stringnya = '8NM';
    elseif jangk==4
        r_max = 170;
        stringnya = '10NM';
    elseif jangk==5
        r_max = 200;
        stringnya = '12NM';
    elseif jangk==6
        r_max = 293;
        stringnya = '18NM';
    else
        r_max = 385;
        stringnya = '24NM';
    end
else
    if jangk==1
        r_max = 164;
        stringnya = '8km';
    elseif jangk==2
        r_max = 198;
        stringnya = '10km';
    elseif jangk==3
        r_max = 251;
        stringnya = '12km';
    elseif jangk==4
        r_max = 141;
        stringnya = '15km';
    end
end
```

```
elseif jangk==5
    r_max = 182;
    stringnya = '20km';
elseif jangk==6
    r_max = 265;
    stringnya = '30km';
else
    r_max = 365;
    stringnya = '42km';
end
end
set(displ_jangk,'string',stringnya);
```



Lampiran 6. Source Code Matlab Untuk Mengurangi Jangkauan Radar

```
%Fungsi Pengurang Jangkauan
```

```
if jangk>1
    jangk=jangk-1;
end

if jangk<4
    r_min = 30;
    r_pengali = 60;
else
    r_min = 15;
    r_pengali = 120;
end

if sat_nm == 1
    if jangk==1
        r_max = 155;
        stringnya = '4NM';
    elseif jangk==2
        r_max = 216;
        stringnya = '6NM';
    elseif jangk==3
        r_max = 278;
        stringnya = '8NM';
    elseif jangk==4
        r_max = 170;
        stringnya = '10NM';
    elseif jangk==5
        r_max = 200;
        stringnya = '12NM';
    elseif jangk==6
        r_max = 293;
        stringnya = '18NM';
    else
        r_max = 385;
        stringnya = '24NM';
    end
else
    if jangk==1
        r_max = 164;
        stringnya = '8km';
    elseif jangk==2
        r_max = 198;
        stringnya = '10km';
    elseif jangk==3
        r_max = 251;
        stringnya = '12km';
    elseif jangk==4
        r_max = 141;
        stringnya = '15km';
    end
end
```

```
elseif jangk==5
    r_max = 182;
    stringnya = '20km';
elseif jangk==6
    r_max = 265;
    stringnya = '30km';
else
    r_max = 365;
    stringnya = '42km';
end
end
set disp_jangk, 'string', stringnya);
```



Lampiran 7. Source Code Matlab Untuk Satuan Nautical Mile (NM)

```
%Satuan Nautical Mile

sat_nm = 1;
sat_km = 0;

set(h_nm,'value',1);
set(h_km,'value',0);

if jangk<4
    r_min = 30;
    r_pengali = 60;
else
    r_min = 15;
    r_pengali = 120;
end

if jangk==1
    r_max = 155;
    stringnya = '4NM';
elseif jangk==2
    r_max = 216;
    stringnya = '6NM';
elseif jangk==3
    r_max = 278;
    stringnya = '8NM';
elseif jangk==4
    r_max = 170;
    stringnya = '10NM';
elseif jangk==5
    r_max = 200;
    stringnya = '12NM';
elseif jangk==6
    r_max = 293;
    stringnya = '18NM';
else
    r_max = 385;
    stringnya = '24NM';
end

set(disj_jangk,'string',stringnya);
```

Lampiran 8. Source Code Matlab Untuk Satuan Kilometer (km)

```
%Satuan Kilometer

sat_nm = 0;
sat_km = 1;

set(h_nm,'value',0);
set(h_km,'value',1);

if jangk<4
    r_min = 30;
    r_pengali = 60;
else
    r_min = 15;
    r_pengali = 120;
end

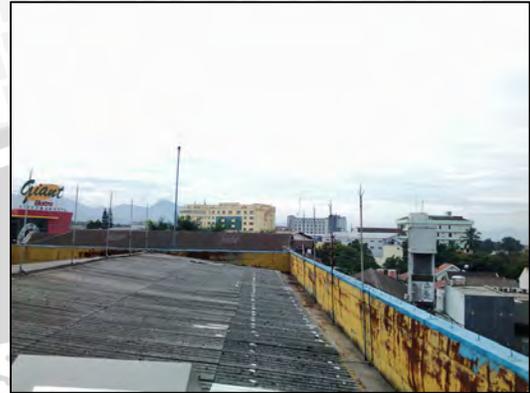
if jangk==1
    r_max = 164;
    stringnya = '8km';
elseif jangk==2
    r_max = 198;
    stringnya = '10km';
elseif jangk==3
    r_max = 251;
    stringnya = '12km';
elseif jangk==4
    r_max = 141;
    stringnya = '15km';
elseif jangk==5
    r_max = 182;
    stringnya = '20km';
elseif jangk==6
    r_max = 265;
    stringnya = '30km';
else
    r_max = 365;
    stringnya = '42km';
end

set(disj_jangk,'string',stringnya);
```

Lampiran 9. Foto *Landscape* dan Foto Satelit Area di Sekitar Lokasi Radar



1. Utara



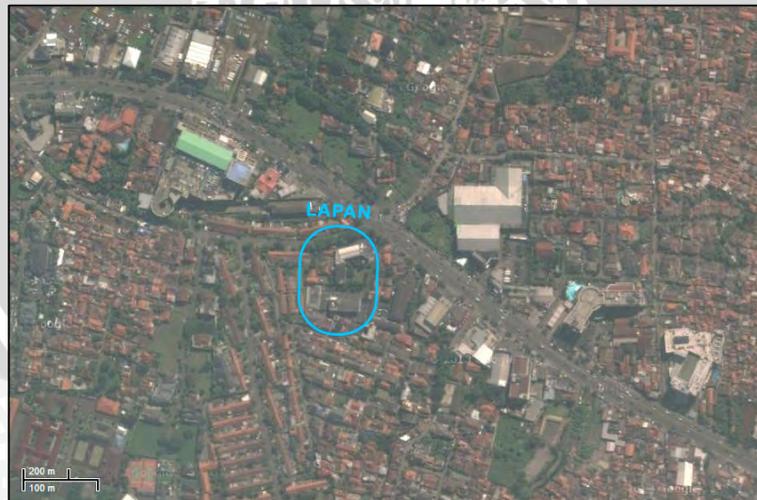
2. Timur



3. Selatan



4. Barat



5. Foto Satelit Area di Sekitar LAPAN Bandung