

TUGAS AKHIR - TE 145561

OPTIMALISASI GRAPHICAL USER INTERFACE BERBASIS C# DENGAN VISUAL STUDIO PADA OPERATOR CONTROL UNIT RHINO ROBOT

Tata Tanjung Tamara NRP. 1031150000079

Pembimbing Dr. Ir. Achmad Affandi DEA Imam Arifin, S.T., M.T. Muhammad Fajar Adityo, S.T.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya 2018



TUGAS AKHIR - TE 145561

OPTIMALISASI GRAPHICAL USER INTERFACE BERBASIS C# DENGAN VISUAL STUDIO PADA OPERATOR CONTROL UNIT RHINO ROBOT

Tata Tanjung Tamara NRP. 1031150000079

Pembimbing Dr. Ir. Achmad Affandi DEA Imam Arifin, S.T., M.T. Muhammad Fajar Adityo, S.T.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya 2018



FINAL RPOJECT - TE 145561

GRAPHICAL USER INTEFACE OPTIMALIZATION BASED C# WITH VISUAL STUDIO TO OPERATOR CONTROL UNIT RHINO ROBOT

Tata Tanjung Tamara NRP. 1031150000079

Supervisors Dr. Ir. Achmad Affandi DEA Imam Arifin, S.T., M.T. Muhammad Fajar Adityo, S.T.

DEPARTMENT OF ELECTRICAL AUTOMATION ENGINEERING Faculty of Vocations Institut Teknologi Sepuluh Nopember Surabaya 2018



-----Halaman ini sengaja dikosongkan-----

OPTIMALISASI *GRAPHIC USER INTERFACE* BERBASIS C# DENGAN VISUAL STUDIO PADA *OPERATOR CONTROL UNIT* RHINO ROBOT DI PT. BHIMASENA RESEARCH AND DEVELOPMENT

TUGAS AKHIR

Disusun oleh:

Tata Tanjung Tamara

NRP. 10311500000079

Menyetujui,

Kepala Human Resources Department, Fadli Tirmissi NIK. 0492016

Pembimbing Perusahaan,

<u>M. Fajar Adityo, ST.</u> NIK. 120322016

Chief Executive Officer,

2 Aris Budiyarto

-----Halaman ini sengaja dikosongkan-----

OPTIMALISASI GRAPHICAL USER INTERFACE BERBASIS C# DENGAN VISUAL STUDIO PADA OPERATOR CONTROL UNIT RHINO ROBOT

Nama	: Tata Tanjung Tamara
Pembimbing	: Dr. Ir. Achmad Affandi, DEA
-	Imam Arifin, S.T., M.T.

ABSTRAK

Sistem komunikasi dan penyimpanan data robot pada *Graphical User Interface* (GUI) versi lama tidak dapat dikembangkan atau dimodifikasi. Hal tersebut disebabkan karena tidak dibuat secara modular sehingga mempersulit pengembangan Rhino robot. Selain itu, kapasitas memori pada GUI versi lama cukup besar karena terdapat fungsi XNA Framework dan menu-menu pengendali robot yang berlebihan.

Berdasarkan permasalahan tersebut perlu optimalisasi GUI pada Operator Control Unit (OCU) Rhino robot. bahasa pemrograman menggunakan C# dengan Integrated Development Environment (IDE) Visual Studio. Sistem komunikasi dan penyimpanan data robot dikelompokkan menjadi 3 bagian program dengan metode inheritance class atau kelas turunan. Pembuatan GUI menggunakan salah satu fitur IDE yaitu Windows Presentation Foundation (WPF).

Hasil dari penelitian ini adalah program sistem komunikasi dan penyimpanan data lebih terstruktur sehingga dapat dikembangan untuk penelitian selanjutnya. Setelah itu, kapasitas memori pada GUI versi baru sebesar 57,4 MB dengan rata-rata penggunaan memori saat dijalankan sekitar 513,5 MB serta jumlah halaman GUI sebanyak 8 dengan rincian 3 halaman utama dan 5 halaman pendukung untuk mengendalikan gerak robot.

Kata kunci: C#, Graphical User Interface (GUI), Integrated Development Environment (IDE), Operator Control Unit (OCU), Windows Presentation Foundation (WPF), Visual Studio, XNA Framework -----Halaman ini sengaja dikosongkan---

GRAPHICAL USER INTERFACE OPTIMALIZATION BASED C# WITH VISUAL STUDIO TO OPERATOR CONTROL UNIT RHINO ROBOT

Name Supervisors : Tata Tanjung Tamara : Dr. Ir. Achmad Affandi, DEA Imam Arifin, S.T., M.T.

ABSTRACT

The system of communication and data storage of robots on the old version Graphical User Interface (GUI) can't be developed or modified. because it isn't made in a modular way and mae it difficult for developer. In addition, the memory capacity in old versions of GUI is quite large because there are XNA Framework function and excessive robot control menus.

Based on these problems we need to optimize the GUI on the Operator Control Unit (OCU) Rhino robot. programming language using C # with Integrated Development Environment (IDE) Visual Studio. Communications systems and data storage robots are grouped into 3 parts of the program with the method of inheritance class. GUI creation uses one of the IDE features of the Windows Presentation Foundation (WPF).

The result of this research is program of communication system and data storage more structured so that can be developed for further research. After that, the memory capacity of the new version of GUI is 57.4 MB with an average memory usage of about 513.5 MB and the number of GUI pages as many as 8 with 3 main page details and 5 supporting pages to control robot motion.

Keywords : C#, Graphical User Interface (GUI), Integrated Development Environment (IDE), Operator Control Unit (OCU), Windows Presentation Foundation (WPF), Visual Studio, XNA Framework -----Halaman ini sengaja dikosongkan----

KATA PENGANTAR

Puji syukur atas kehadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir dengan judul **"Optimalisasi** *Graphical User Interface* Berbasis C# dengan *Visual Studio* pada *Operator Control Unit* Rhino Robot" untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma 3 di Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya

Dalam proses penyusunan penelitian ini, penulis telah dibantu oleh berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada Mama, Papa, dan Kakak yang selalu memberi dukungan dan doa untuk keberhasilan dan kelancaran. Lalu kepada Bapak Dr. Ir. Achmad Affandi DEA selaku Dosen Pembimbing vang telah memberi arahan serta nasihat kepada penulis. Setelah itu, kepada Bapak Imam Arifin, S.T., M.T. selaku Dosen Pembimbing sekaligus Kepala Laboratorium Sistem Komputer dan Otomasi yang telah membimbing penulis dari awal penelitian hingga saat ini. Lalu kepada Bapak Muhammad Fajar Adityo, S.T. selaku pembimbing di PT Bhimasena Research and Development yang telah memberi arahan disela pekerjaan selama 9 bulan. Selain itu, kepada teman-teman anggota Laboratorium Sistem Komputer dan Otomasi angkatan 2015, 2016 dan 2017 atas pemberian semangat dan bantuan serta semua pihak vang telah membantu secara langsung maupun tidak langsung dalam proses penyelesaian penelitian ini yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa buku Tugas Akhir ini belum sempurna sehingga saran dan masukan sangat diharapkan untuk perbaikan di masa yang akan datang. Semoga buku ini bermanfaat bagi pembaca dan masyarakat pada umumnya.

Surabaya, 2 Agustus 2018

Tata Tanjung Tamara NRP 10311500000079 -----Halaman ini sengaja dikosongkan----

DAFTAR ISI

HALAMAN PENGESAHAN Error! Bookmark n	ot defined.
HALAMAN PENGESAHAN PERUSAHAAN Error! Boo	kmark not
defined.	
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	
1.3 Rumusan Masalah	
1.4 Batasan Masalah	
1.5 Metodologi	
1.6 Sistematika Penulisan	4
BAB II PENUNJANG GUI, SISTEM KOMUNIKASI DAN	1
PENYIMPANAN DATA RHINO ROBOT	5
2.1 Robot Explosive Ordnance Disposal (EOD)	5
2.2 Rhino Robot	6
2.3 Operator Control Unit (OCU)	7
2.4 Graphical User Interface (GUI)	7
2.5 Framework .NET	
2.6 Bahasa Pemrograman C#	9
2.5.1 Namespace	9
2.5.2 Class	9
2.5.3 <i>Method</i>	
2.5.4 Variabel	
2.5.5 Tipe Data	
2.5.6 Statement C#	10
2.5.7 Selection Statement C#	
2.5.8 Iteration Statement C#	
2.5.9 Jump Statement C#	
2.5.10 <i>Object Oriented Programming</i> (OOP)	
2.7 Komunikasi	
2.8 Windows Presentation Foundation (WPF)	
2.9 Visual Studio	15

BAB III PERANCANGAN GUI, SISTEM KOMUNIKASI DAN	
PENYIMPANAN DATA RHINO ROBOT	17
3.1 Design Requirement	17
3.2 Data Flow Diagram Sistem Komunikasi dan Penyimpanan I	Data
GUI Keseluruhan	18
3.3 Protokol Rhino Robot	19
3.4 Rancangan Halaman GUI	20
3.5 Perancangan Desain Tampilan dan Tata Letak Menu	21
3.5.1 Desain Halaman Awal	22
3.5.2 Desain Halaman Verifikasi Robot	23
3.5.3 Desain Halaman Layar Operasi	25
3.5.4 Desain Halaman Pop Up Main Menu	26
3.5.5 Desain Halaman Pop Up Menu Pose	27
3.5.6 Desain Halaman Pop Up Menu Preset Pose	28
3.5.7 Desain Halaman Pop Up Menu Camera	29
3.5.8 Desain Halaman Pop Up Menu Ilumination	31
3.6 Metodologi Pengujian	32
3.6.1 Tata Cara Pengambilan Data Kapasitas Memori GUI	32
3.6.2 Tata Cara Pengujian GUI	33
BAB IV PENGUJIAN DAN ANALISIS	35
4.1 Hasil Pengujian	35
4.2 Analisis Sistem Komunikasi dan Penyimpanan Data Robot (GUI
Versi Lama dan Baru	36
4.3 Uji Memori GUI	37
4.2.1.Kapasitas Memori GUI	37
4.2.2.Data Uji dan Analisis Kinerja GUI	38
4.4 Desain Tampilan dan Penataan Letak Menu	41
BAB V PENUTUP	43
DAFTAR PUSTAKA	45
LAMPIRAN A	49
A-1 Listing Program Transmitter	49
A-2 Listing Program Turunan Transmitter	51
A-3 Listing Program Receiver	53
A-4 Listing Program Turunan Receiver	57
A-5 Listing Program Communication Center	59
A-6 Listing Program Halaman Utama GUI	61
A-7 Listing Program Login Robot GUI	63
A-8 Listing Program Operation Page	67
A-9 Listing Program Pop Up Main Menu	69

A-10 Listing Program Pop Up Pengaturan Pose	73
A-11 Listing Program Pop Up Preset Pose	83
A-12 Listing Program Pop Up Pengaturan Kamera	
A-13 Listing Program Pop Up Penerangan	
LAMPIRAN B	101
B-1 Protokol Mengirim Data Rhino Robot	101
B-2 Protokol Menerima Data Rhino Robot	103
B-3 Memory Usage GUI Versi Lama	105
B-4 Desain Halaman GUI Versi Lama	107
RIWAYAT HIDUP PENULIS	113

-----Halaman ini sengaja dikosongkan---

DAFTAR GAMBAR

Gambar 2.1	Robot Explosive Ordnance Disposal (EOD)[7]	6
Gambar 2.2	Arm Manipulator Robot 6 DoF (i), Chasis (ii)	6
Gambar 2.3	Operator Control Unit (OCU) Robot	7
Gambar 2.4	Contoh GUI	8
Gambar 2.5	Struktur pemrograman Bahasa C# [17]	9
Gambar 2.6	Listing program <i>if</i>	11
Gambar 2.7	Listing program <i>If-Else</i>	11
Gambar 2.8	Listing program <i>switch</i>	11
Gambar 2.9	Listing program while	12
Gambar 2.10	Listing program <i>do</i>	12
Gambar 2.11	Listing program for	12
Gambar 2.12	Listing program return	13
Gambar 2.13	Listing program goto	13
Gambar 2.14	Listing program <i>continue</i>	13
Gambar 2.15	Model Komunikasi Sederhana	15
Gambar 2.16	Tampilan awal Visual Studio	16
Gambar 3.1	Data Flow Diagram Sistem Komunikasi dan Penyimpanan Data	19
Gambar 3.2	Listing program protokol mengirim data	20
Gambar 3.3	Listing program protokol menerima data	20
Gambar 3.4	Skema Rancang Halama GUI Versi Baru	20
Gambar 3.5	Flowchart halaman utama GUI	22
Gambar 3.6	Flowchart halaman menu utama GUI	22
Gambar 3.7	Desain Halaman Awal	23
Gambar 3.8	Listing program Desain Halaman Awal	23
Gambar 3.9	Desain Halaman Verifikasi Robot	24
Gambar 3.10	Listing program Halaman <i>login robot</i>	24
Gambar 3.11	Listing Program Halaman Layar Operasi	25
Gambar 3.12	Desain Layar Operasi	26
Gambar 3.13	Desain halaman Pop Up Main Menu	26
Gambar 3.14	Listing program Halaman Pop Up Menu	27
Gambar 3.15	Desain halaman Pop Up Menu Pose	27
Gambar 3.16	Listing Program Halaman Pengaturan Pose	27
Gambar 3.17	Desain Pop Up Menu Preset Pose	28
Gambar 3.18	Listing program halaman preset pose	28

Gambar 3.19	Desain Pop Up Menu Camera	29
Gambar 3.20	Desain halaman Pop Up Menu Camera 1 Layar	30
Gambar 3.21	Desain halaman Pop Up Menu Camera 2 Layar	30
Gambar 3.22	Desain halaman Pop Up Menu Camera 3 Layar	30
Gambar 3.23	Listing Program Halaman Pengaturan Kamera	31
Gambar 3.24	Desain Pop Up Menu Ilumination	31
Gambar 3.25	5 Listing Program Halaman Pengaturan Lampu	32
Gambar 3.26	Tata cara pengambilan data kapasitas memori 1	32
Gambar 3.27	Tata cara pengambilan data kapasitas memori 2	32
Gambar 3.28	Tata cara pengambilan data kapasitas memori 3	33
Gambar 3.29	Tata Cara Pengujian GUI 1	33
Gambar 3.30	Tata Cara Pengujian GUI 2	34
Gambar 4.1	Sistem Komunikasi dan Penyimpanan Data GUI ve	rsi
	lama	36
Gambar 4.2	Grafik Perbandingan Process Memory GUI	40

DAFTAR TABEL

Tabel 3.1	Design Requirement	1
Tabel 4.1	Hasil Pengujian 35	
Tabel 4.2	Penggunaan Memori GUI	1
Tabel 4.3	Process Memory Halaman GUI versi baru)
Tabel 4.4	Perbanding penggunaan memori GUI versi lama dan baru 40)
Tabel 4.5	Hasil Uji Desain Tampilan pada GUI saat dijalankan41	L

-----Halaman ini sengaja dikosongkan---

BAB I PENDAHULUAN

1.1 Latar Belakang

PT Bhimasena Research and Development merupakan perusahaan Badan Usaha Milik Swasta (BUMS) yang bergerak di bidang riset, pengembangan dan produksi peralatan. Produk utama perusahaan ini vaitu alat untuk Angkatan Darat (AD), Angkatan Laut (AL), dan Angkatan Udara (AU). Bhimasena melayani industri dengan menggabungkan sumber daya manusia top quality, infrastruktur penelitian *high-end* dan akses ke pasar potensial yang luas. Perusahaan ini juga memperluas kemampuannya dengan menciptakan kemitraan internasional dan aliansi global dengan laboratorium dan pusat penelitian di seluruh dunia. Kekuatan korporatnya berpusat pada kemampuan untuk menyediakan solusi teknologi inovatif yang disesuaikan untuk berbagai jenis klien di industri otomotif, dirgantara dan berteknologi tinggi sambil berusaha menjadi pusat teknologi keunggulan terkemuka di negara ini. Instansi ini didirikan pada tahun 2013 oleh pemilik PT Merpati Wahana Raya yang merupakan perusahaan penyediaan alat peralatan pertahanan.

Perusahaan riset dan teknologi ini memiliki beberapa divisi diantaranya *Ground Vehicle* (GV), *Unmanned Aerial Vehicle* (UAV), *Underwater Vehicle* (UV) dan Rhin robot. Pada divisi GV menghasilkan beberapa produk diantaranya *Kitchen Truck*, Jihandak, Dekontaminasi dan *Shop Contact* yang masing-masing didesain untuk memenuhi kebutuhan di lapangan sesuai dengan fungsinya. Pada divisi UAV mempunyai sebuah produk bernama SWGR – 1 UAV. Benda ini merupakan unit pesawat tanpa awak yang dapat diterbangkan tanpa menggunakan area landasan dan dikendalikan dengan pilot secara terintegrasi. Selain itu, pada UV sedang mengembangkan produk bernama Kapal Tempur Bawah Air (KTBA) dan *Diver Propulsion Vehicle* (DPV). Produk ini berfungsi untuk mendukung peralatan perang di dunia militer. Pada divisi Rhino robot mengembangkan sebuah produk robot. Benda ini berfungsi untuk mengambil objek berbahaya dan mengamati lingkungan yang tidak dapat dijangkau oleh manusia.

Rhino robot memiliki 2 bagian utama yaitu *arm* dan *chasis*. Bagian *chasis* berfungsi untuk mobilitas robot. Komponen pada bagian ini terdiri dari beberapa motor, baterai, rangkaian pendukung gerak robot

dan sebagainya. Sedangkan bagian *arm* bertugas untuk mengambil objek yang dianggap berbahaya. Dengan komponen yang terdiri dari *Gripper* (alat pengambil objek), motor, kamera dan sebagainya. Tujuan dikembangkan robot ini untuk meminimalisasi resiko bahaya pada manusia. Robot ini dikendalikan oleh manusia secara jarak jauh dengan alat kendali bernama *Operator Control Unit* (OCU).

OCU Rhino robot berbentuk seperti laptop *toughbook* yang memiliki *Joystick* dan *Graphical User Interface* (GUI) dengan *operate system* Windows 10. Masing-masing berfungsi sebagai alat pengendali robot untuk manusia. Pada *Joystick*, Pengendalian robot dilakukan menggunakan kombinasi-kombinasi tombol. Lalu, pengguna dapat mengamati lingkungan secara langsung dan menentukan menu-menu fungsi pengendali melalui GUI. Dengan bentuk alat kontrol seperti ini, mempersulit pengguna untuk mengendalikan robot secara jarak jauh. Hal tersebut disebabkan karena pengguna harus mengingat kombinasi-kombinas tombol pada *Joystick*. Lalu pada GUI terdapat halaman menu fungsi pengendali robot sebanyak 20 buah dengan menghasilkan kapasitas memori sebesar 92.8 MB yang dapat mempengaruhi kinerja kecepatan saat GUI dijalankan. Oleh karena itu, perlu dibuat OCU dengan tombol-tombol fungsi robot, optimalisasi GUI dan pembuatan *supply* daya untuk OCU agar sistem dapat bekerja

GUI yang telah dibuat terdapat beberapa masalah diantaranya sistem komunikasi dan penyimpanan data robot pada GUI sulit untuk dikembangkan dan dimodifikasi karena tidak dibuat secara modular. Padahal Rhino robot masih memungkinkan untuk dikembangkan menjadi beberapa versi robot yang berbeda. Selain itu, terdapat menumenu fungsi pengendali yang berlebihan dan menggunakan salah satu fitur XNA Framework pada GUI. Hal tersebut mempengaruhi kinerja kecepatan saat GUI dijalankan. Dengan permasalahan tersebut perlu optimalisasi GUI pada OCU Rhino robot.

Pada Tugas Akhir ini dibuat optimalisasi GUI berbasis bahasa pemrograman C# dengan Integrated Development Environment (IDE) Visual Studio. Perancangan desain tampilan dan tata letak menu menggunakan salah satu fitur Visual Studio yaitu Windows Presentation Foundation (WPF). Halaman GUI dirancang seminimal mungkin guna untuk memperkecil kapasitas memori sehingga penulis memberi target dibuat 8 halaman GUI dengan rincian 3 halaman utama dan 5 halaman pendukung. Pada struktur program komunikasi dan penyimpanan data robot menggunakan metode inheritance class atau kelas turunan. Serta penggunaan kelas turunan ini untuk mengelompokkan komunikasi menjadi 3 yaitu *transmitter*, *receiver* dan *communication center*.

1.2 Tujuan

Optimalisasi GUI pada Rhino robot bertujuan untuk mempermudah pengembangan robot selanjutnya. Selain itu, mempercepat kinerja GUI saat dijalankan serta pengguna dapat mengendalikan robot secara mudah dengan halaman GUI yang efisien.

1.3 Rumusan Masalah

Sistem komunikasi dan penyimpanan data robot GUI pada OCU Rhino robot sulit dikembangkan dan dimodifikasi. Hal tersebut terjadi karena tidak dibuat secara modular. Padahal Rhino robot masih memungkinkan mengalami pengembangan. Selain itu, kapasitas memori pada GUI versi lama cukup besar karena terdapat menu-menu pilihan pengendali robot yang berlebihan sehingga mempengaruhi kinerja kecepatan saat GUI dijalankan.

1.4 Batasan Masalah

Tugas Akhir ini menggunakan bahasa pemrograman C#. Perangkat lunak untuk menjalankan pemrograman yang telah di buat menggunakan Visual Studio 2015. Pembuatan halaman GUI menggunakan WPF. Pengujian dilakukan menggunakan laptop Dell Aspiron 3560. Data uji menggunakan fitur *Diagnostic Tool* Visual Studio dengan mengambil bagian *memory usage*.

1.5 Metodologi

Dalam pelaksanaan Tugas Akhir berupa optimalisasi *Graphical* User Interface berbasis C# dengan Visual Studio pada Operator Control Unit Rhino robot, beberapa kegiatan yang dapat diuraikan yaitu yang pertama adalah tahap persiapan. Pada tahap ini dilakukan studi literatur mengenai dasar-dasar dari bahasa pemrograman C#, Visual Studio dan Windows Presentation Foundation (WPF). Tahap kedua yaitu perancangan sistem komunikasi dan penyimpanan data dan GUI. Kegiatan yang akan dilakukan adalah membuat rancangan sistem komunikasi antara robot dengan OCU dan konsep tampilan layar pada GUI. Tahap selanjutnya yaitu pembuatan sistem yang dilakukan adalah mengaplikasikan rancangan sistem komunikasi antar robot pada GUI dan konsep tampilan layar pada GUI dan konsep tampilan layar pada GUI yang telah dibuat dengan perangkat

lunak Visual Studio. Setelah itu, tahap pengujian dan analisis akan dilakukan yaitu menjalankan GUI yang telah dibuat lalu mengambil data penggunaan memori. Setelah itu menganalisis dengan membandingkan nilai GUI versi lama dengan baru. Tahap terakhir adalah tahap penyusunan laporan. Dalam tahap ini diperlukan referensi atau literatur pendukung dan analisis data dari semua percobaan yang telah dilakukan.

1.6 Sistematika Penulisan

Untuk pembahasan lebih lanjut, laporan Tugas Akhir ini disusun dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan, sistematika laporan, serta sistematika penulisan Tugas Akhir yang dibuat.

BAB II PENUNJANG GUI, SISTEM KOMUNIKASI DAN PENYIMPANAN DATA RHINO ROBOT

Menjelaskan teori tentang Rhino robot, OCU, GUI, bahasa pemrograman C#, perangkat lunak Visual Studio dan WPF sebagai penunjang pembuatan Tugas Akhir.

BAB III PERANCANGAN GUI, SISTEM KOMUNIKASI DAN PENYIMPANAN DATA RHINO ROBOT

Membahas perencanaan alat yang akan dibuat. Dimulai dari perancangan GUI dengan meminimalisasi fungsi pengendali yang berlebihan pada versi lama dan membuat *flowchart* program sistem komunikasi dan penyimpanan data robot

BAB IV PENGUJIAN DAN ANALISIS

Menjelaskan tata cara pengujian alat yang telah dibuat. mengambil dan menganalisis data *memory usage* saat GUI dijalankan. menganalisis perbandingan sistem komunikasi dan penyimpanan data robot versi lama dengan versi baru

BAB V KESIMPULAN DAN SARAN

Membahas kesimpulan dilakukannya pengujian dan analisis dan saran membangun untuk alat yang telah dibuat

BAB II PENUNJANG GUI, SISTEM KOMUNIKASI DAN PENYIMPANAN DATA RHINO ROBOT

Beberapa teori yang dijelaskan dalam bab ini adalah penunjang untuk mengerjakan Tugas Akhir. penjelasan yang dibahas tetang Robot EOD, Rhino robot, OCU, GUI, bahasa pemrograman C#, WPF dan Visual Studio. Teori-teori yang telah diambil berasal dari berbagai macam literatur.

2.1 Robot Explosive Ordnance Disposal (EOD)

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu (kecerdasan buatan) [1]. Kemajuan dalam teknologi robotik telah memungkinkan untuk melakukan hal-hal yang biasanya hanya dilakukan oleh manusia. Salah satu kemajuannya adalah terciptanya Robot penjinak bom. Robot ini dapat menggantikan manusia untuk mengintai dan menangani peledak atau barang berbahaya lingkungan secara langsung [2]. Benda ini ditemukan pada tahun 1972 oleh Letnan Kolonel Peter Miller dengan merancang *chassis* gerobak listrik bertenaga ke perangkat seperti bom mobil sehingga objek berbahaya dengan aman diledakkan tanpa melukai siapapun [3]. Robot ini akhirnya dikembangkan oleh berbagai ahli di seluruh dunia untuk membuat perangkat yang terjangkau dan aman [4], salah satu pengembangannya adalah Robot EOD seperti pada Gambar 2.1

Pada penelitian [5] dibuat sebuah robot EOD dengan harga yang rendah. Hal tersebut didasari karena robot mempunyai harga yang cukup tinggi. Tetapi lengan robot dipilih 2 *Degree of Freedom* (DoF) yang mengakibatkan gerak robot terbatas. Lalu pada penelitian [6] dirancang skema optimasi untuk robot EOD berbasis *supervising control architecture*. Skema optimasi penelitian ini membuat sistem mekanis dan kontrol untuk menangani pendekatan tugas untuk menggenggam objek di lingkungan yang terbatas menggunakan sensor jarak. Tetapi waktu misi yang dihasilkan lebih lama dan menghabiskan energi cukup besar pada robot EOD.



Gambar 2.1 Robot Explosive Ordnance Disposal (EOD)[7]

2.2 Rhino Robot

Indonesia sebagai salah satu negara berkembang juga tidak luput dari bahaya yang mengancam. Oleh karena itu, dibutuhkan robot untuk menangani objek berbahaya dan mengintai kawasan yang tidak dapat dijangkau oleh manusia. Maka PT Bhimasena Research and Development. Mengembangkan sebuah produk bernama Rhino robot. Tujuan perusahaan mengembangkan robot ini adalah untuk menghemat biaya negara. Selain itu, jika robot terjadi hal-hal yang tidak diinginkan maka tidak perlu untuk memperbaiki di luar negeri.

Rhino robot sendiri terdiri dari 2 bagian utama yaitu *arm* dan *chasis*. Pada bagian *arm* berfungsi untuk mengambil objek berbahaya yang dibantu oleh *gripper*. Bagian ini mempunyai 6 *Degrees Of Freedom* (DoF) dapat diilustrasikan seperti pada Gambar 2.2. Komponen yang ada pada bagian ini terdiri dari motor, kamera, dan rangkaian pendukung lainnya. Sedangkan bagian *chasis* berfungsi untuk mobilitas robot yang dapat diilustrasikan pada Gambar 2.2. Bagian ini terdiri dari baterai, roda, motor dan rangkaian pendukung lainnnya.



Gambar 2.2 Arm Manipulator Robot 6 DoF (i), Chasis (ii)

2.3 Operator Control Unit (OCU)

Robot memerlukan perhatian dan kontrol langsung secara terus menerus oleh operator melalui operasi jarak jauh yang bernama *Operator Control Unit* (OCU) [8]. Mengontrol robot melalui jenis OCU ini membutuhkan perhatian penuh oleh operator. Operator biasanya menggunakan kontroler tipe *joystick* untuk mengaktifkan berbagai motor pada robot untuk memindahkannya [9] Perangkat OCU seperti pada Gambar 2.3.

Pada Tugas Akhir ini dirancang OCU versi 2 yang terdiri dari 3 bagian utama yaitu *Power Distribution*, Antarmuka, dan GUI. Bagian *Power Distribution* bertugas untuk membuat *supply* daya untuk OCU agar bisa menyala. Lalu pada bagian Antarmuka bertugas untuk merancang tombol-tombol agar terintergrasi oleh GUI. Dimana penulis mengerjakan bagian GUI untuk OCU Rhino robot yang penjelasannya ada pada Bab III.



Gambar 2.3 Operator Control Unit (OCU) Robot

2.4 Graphical User Interface (GUI)

GUI merupakan antarmuka bergambar untuk sebuah program agar lebih mudah digunakan dengan memberi tampilan yang konsisten seperti tombol, menu dan sebagainya [10]. Selain itu, juga memberikan kemudahan bagi manusia dan komputer untuk berinteraksi dan berkomunikasi. Fungsi ini biasanya bersifat *event-driven* yang artinya bahwa beberapa tugas akan berjalan setiap kali peristiwas terdeksi. Seperti pengguna memberi masukan pada tombol atau item menu [11].

GUI ini memiliki beberapa manfaat diantanya pada penelitian [12]. Penelitian ini dibuat sistem komunikasi grafis manusia dan mesin yang memungkinkan adanya komunikasi menggunakan gambar garis yang dibuat pada layar. Selain itu juga mempunyai kemampuan untuk memperbesar dan memperkecil tampilan. Penelitian lain [13] dirancang model robot *mobile* yang dapat dikontrol menggunakan GUI melalui protokol nirkabel menggunakan perangkat lunak visual basic.



Gambar 2.4 Contoh GUI

Dengan beberapa penelitian ini bahwa GUI penting sebagai visual komunikasi antara manusia dan objek. Oleh karena itu, GUI yang dibuat harus sesuai dengan keperluan dan dibuat efisien agar mempermudah pengguna yang menjalankannya. Pada Gambar 2.4 merupakan salah satu contoh GUI yang dibuat untuk OCU Rhino robot.

2.5 Framework .NET

Framework .NET merupakan suatu komponen Windows yang terintegrasi dan dibuat agar dapat menjalankan berbagai macam aplikasi .NET, termasuk pengembangan aplikasi web service yang menggunakan XML (extensible Markup Language). Bagian ini menyediakan semua tools dan teknologi yang diperlukan untuk membangun sebuah aplikasi. Framework .NET didesain untuk memenuhi tujuan diantaranya menyediakan lingkungan kerja vang konsisten bagi bahasa pemrograman yang berorientasi objek, pengeksekusian kode yang dapat meminimalisasi proses penyebaran perangkat lunak, aman dalam hal pengeksekusian kode termasuk kode yang dibuat oleh pihak ketiga, mengurangi masalah pada persoalan performa kode atau lingkungan intepreternya, lebih mudah mengembangkan berbagai macam jenis aplikasi yang lebih bervariasi serta membangun semua komunikasi yang ada dalam standar industri untuk memastikan bahwa semua kode aplikasi berstandar framework ini dapat berintegrasi [14]. Keunggulan dari Framework .NET adalah mudah yang artinnya memberi kemudahan kepada para pengembang untuk membuat aplikasi yang dijalankan pada fungsi ini. Selain itu efisien, yang berarti saat proses pembuatan aplikasi membutuhkan waktu yang sedikit sehingga menguntungkan bagi para pengembang.

2.6 Bahasa Pemrograman C#

C# (dibaca "See-Sharp") adalah bahasa pemrograman komputer multi-guna yang cocok untuk semua kebutuhan pengembangan dan program berorientasi objek atau OOP (*Object Oriented* Progamming) yang modern [15]. OOP ialah sebuah pendekatan untuk pengembangan perangkat lunak yang didasarkan pada objek yang berinteraksi satu sama lain [16].

Microsoft membuat C# seiring dengan pembuatan *Framework*. NET. C# menjanjikan produktifitas dan kemudahan yang ada di Visual Basic dengan kemampuan dan fleksibilitas yang ada di C/C++ [17]. Berikut adalah struktur pemrograman bahasa C# pada Gambar 2.5.



Gambar 2.5 Struktur pemrograman Bahasa C# [17]

2.5.1 Namespace

Bagian ini dapat diartikan sebagai ruang lingkup tempat *class* atau struktur tersebut diciptakan. Program harus diawali dengan mengetik nama *namespace*-nya. Dua buah *class* (atau struktur) akan dianggap sama jika keduanya memiliki nama *class* dan *namespace* yang sama. Melalui cara ini, peluang terjadinya kesamaan nama antara dua *class* atau struktur yang berbeda menjadi sangat kecil [18].

2.5.2 Class

Fungsi ini adalah sebuah kontruksi yang memungkinkan untuk membuat jenis tipe dengan mengelompokkan berbagai variabel dan

metode menjadi satu [19]. Bagian ini juga sebagai wadah yang menyimpan data dan kode program untuk melakukan pekerjaan tertentu [20]

2.5.3 Method

Bagian fungsi ini mempunyai kemampuan untuk melakukan proses tertentu. Semua *method* yang dimiliki objek hanya dapat diakses melalui program. Akses fungsi ini melalui program dapat dilakukan dengan mengetikkan nama *method* yang diinginkan di belakang nama objek [18].

2.5.4 Variabel

Variabel dapat didefinisikan sebagai tempat untuk menyimpan data yang memiliki suatu tipe data, variabel ini akan diwakili oleh suatu lokasi di memori komputer [21]. *Identifier* adalah suatu nama yang mewakili hal-hal di dalam program. *Identifier* dapat mewakili suatu variabel, konstanta, metoda atau fungsi, *properties, class, enum* atau *struct*.

2.5.5 Tipe Data

Ada 3 jenis tipe data dalam C# yaitu tipe nilai (value types), tipe referensi (reference types), dan tipe petunjuk (pointer types). Variabel pada tipe nilai dapat diberi nilai secara langsung dan variabel ini juga berasal dari kelas System.ValueType. Tipe ini secara langsung berisi beberapa data diantaranya *int*, *char*, *float* dan sebagainya. Ketika mendeklarasikan tipe *int*, sistem mengalokasikan memori untuk menyimpan nilai [22]

2.5.6 Statement C#

Ekspresi terbentuk dari rangkaian operator dan operand. Operator yang terdapat dalam suatu ekpresi menyatakan proses apa yang akan dilakukan pada suatu operand. C# memiliki tiga macam operator yaitu operator *unary*, *binary* dan *ternary* [23]. Operator *unary* memiliki satu buah operand dan menggunakan notasi *prefix* atau *postfix* (misalnya --x atau x++). Operator *binary* menggunakan dua buah operand dan menggunakan notifikasi infix (misalnya x + y). C# hanya memiliki satu buat operator *ternary* yaitu, ? :, operator ini memerlukan tiga buah operand dan menggunakan notasi infix (misalnya x ? y : z) [24].

2.5.7 Selection Statement C#

Selection statement digunakan untuk menentukan bagian mana dari program yang akan dieksekusi selanjutnya. C# menyediakan dua jenis selection statement, yaitu *if* dan *switch* statement [25].

• If

If statement digunakan untuk mengeksekusi kode program jika kondisi tertentu terpenuhi. Misalnya dalam hal melakukan operasi pembagian, program harus mencek terlebih dahulu apakah pembagi bernilai 0 atau tidak agar tidak terjadi *runtime error* [25]. Pada Gambar 2.6 bentuk *statement if* C#.

```
If (expression){
   //Body
   Statement 1
   Statement 2 }
```

Gambar 2.6 Listing program if

• If-Else

Satu variasi dari *if statement* adalah *if-else*. Dalam *statement* ini, selain menentukan langkah yang harus dilakukan jika suatu kondisi terpenuhi [25]. Pada Gambar 2.7 bentuk listing program *if-else* C#

```
If (expression){
Statement 1
Else
Statement 2}
```

Gambar 2.7 Listing program If-Else

• Switch

Statement switch-case berfungsi mengeksekusi blok program dengan membandingkan nilai suatu variabel sebagai pilihan dalam menentukan blok yang akan dieksekusi [26]. Pada Gambar 2.8 bentuk listing program *switch* C#.

Gambar 2.8 Listing program switch

2.5.8 Iteration Statement C#

Statement jenis ini digunakan untuk menentukan bagian mana dari program yang akan dieksekusi berulang-ulang dan apa kondisi yang menentukan perulangan tersebut. Dalam C# ada empat *iteration statement* yang dapat digunakan, yaitu:

• While

While statement berguna untuk melakukan perulangan selama kondisi bernilai true. Karena pengecekan kondisinya dilakukan di awal maka ada kemungkinan badan loop tidak akan dijalankan sama sekali. Pada Gambar 2.9 bentuk listing program *while* C#.

```
While (kondisi)
{
Statements; }
```

Gambar 2.9 Listing program while

• Do

Sama seperti *while* yaitu untuk mengulang eksekusi program. Yang membedakan dengan while adalah pengecekan kondisi dilakukan di akhir blok sehingga blok minimal satu kali dijalankan. Pada Gambar 2.10 bentuk listing program *Do* C#.

```
Do {
[perintah] }
While (kondisi);
```

Gambar 2.10 Listing program do

• For

Fungsi konstruksi ini sama dengan *for* dalam bahasa-bahasa lain yang diturunkan dari C. *For* digunakan untuk melakukan perulangan yang didasarkan atas nilai diskrit, misalnya integer. Salah satu penggunaan *for* yang paling umum adalah dalam menelusuri suatu *array*. Pada Gambar 2.11 bentuk listing program *for* C#.

```
For (expression11)
{ statements 1;
```

Gambar 2.11 Listing program for
2.5.9 Jump Statement C#

Jump statement digunakan untuk mentransfer kontrol eksekusi dari suatu bagian ke bagian lain dalam program [27]. Beberapa statement yang termasuk dalam jenis ini sebaiknya diminimalkan penggunaaanya, contohnya goto, continue, dan break. Alasannya adalah karena penggunaan statement-statement tersebut mengurangi kejelasan flow program. Selain itu pula, kemungkinan terjadinya bug juga semakin besar terutama jika bagian-bagian yang akan dilompati tersebut memiliki bagian inisiasi dan terminasi.

• Return

Seperti halnya dalam bahasa turunan C lainnya, perintah *retur*n digunakan untuk mengembalikan kontrol eksekusi ke pemanggil [27]. Pada Gambar 2.12 bentuk listing program return C#.

```
Expresion11 ()
{
Double area;
Return area;
```

Gambar 2.12 Listing program return

• Goto

Fungsi *Goto* digunakan untuk melanjutkan eksekusi program di label yang sudah didefinisikan sebelumnya [27]. Pada Gambar 2.13 bentuk listing program *Goto* C#.

```
Goto identifier;
Goto case const-expression;
Goto default;
```

Gambar 2.13 Listing program goto

• Continue

Continue adalah sebuah pernyataan yang digunakan untuk melewati kendali kontrol ke perulangan atau iterasi berikutnya [28]. Pada Gambar 2.14 bentuk listing program *continue* C#.

```
Foreach (expresion1)
{
    If (kondisi){}
    If (kondisi2) {}
    Continue;
```

Gambar 2.14 Listing program continue

2.5.10 Object Oriented Programming (OOP)

OOP merupakan pendekatan untuk pengembangan perangkat lunak dimana struktur *software* berdasarkan objek yang berinteraksi satu sama lain untuk menyelesaikan tugas. Dalam upaya untuk mengelola ukuran program yang terus meningkat, maka diperkenalkan pemrograman terstruktur untuk memecah kode menjadi fungsi atau prosedur [16].

Konsep OOP mulai muncul pada pertengahan 1960-an tetapi perkembangannya tidak terlalu maju. Pada tahun 1980-an ada kemajuan dalam bahasa pemrograman berorientasi objek. Secara khusus, bahasa OOP seperti C++. Selanjutnya pada tahun 1990-an, terutama saat munculnya java OOP lebih diminati hingga hari ini. Era saat ini OOP merupakan andalan pemrograman yang modern [16].

Karakteristik dari OPP diantaranya Encapsulation, Polymorphism, Inheritance, dan Objects. Encapsulation merupakan proses akses secara tidak langsung yang diberikan ke data atau menyembunyikan semua detail informasi dan mekanisme [14]. Polymorhism merupakan kemampuan dua objek yang berbeda dalam menanggapi pesan permintaan yang sama. Inheritance atau turunan ialah metode untuk mengklasifikasikan objek dalam program sesuai dengan karakteristik umum dan fungsi. Objects adalah struktur untuk memasukkan data dan prosedur bekerja [16].

2.7 Komunikasi

Kata "Komunikasi" merupakan suatu kata yang dapat diartikan sebagai cara untuk menyampaikan atau menyebarluaskan data dan informasi [29]. Komunikasi juga dapat didefiniskan sebagai proses transmisi informasi dan pemahaman umum dari satu orang ke orang lain [30]. Elemen-elemen model komunikasi diantaranya *source* (sumber), *transmitter* (pengirim), *transmitter system* (sistem transmisi), *receiver* (penerima) dan *destination* (tujuan) [31]. Berikut blok diagram model komunikasi sederhana pada Gambar 2.15.

Transmitter berfungsi untuk mengubah informasi yang akan dikirim menjadi bentuk yang sesuai dengan media penerima. Sistem transmisi merupakan jalur transmisi tunggal atau jaringan transmisi kompleks yang menghubungkan sistem sumber dengan sistem tujuan (*receiver* dan *destination*). Sistem tujuan adalah sistem yang berfungsi untuk menerima sinyal dari sistem transmisi dan menggabungkan ke dalam bentuk tertentu [29].



Gambar 2.15 Model Komunikasi Sederhana

2.8 Windows Presentation Foundation (WPF)

WPF adalah kerangka kerja GUI yang digunakan untuk membuat aplikasi *windows* sama seperti *windows form*. Aplikasi ini sebuah GUI yang digunakan untuk membuat aplikasi sisi klien untuk *windows* serta aplikasi berbasis web. Selain itu, mampu merancang *user interface* dengan kontrol pengguna [32]. WPF merupakan salah satu fitur pada Visual Studio.

Keuntungan dari WPF diantaranya kerangka kerja jauh lebih pintar dan membutuhkan lebih sedikit kode di belakang untuk melakukan halhal umum. Kelemahan dari WPF, jika dibandingkan dengan *windown form* masih ada beberapa kontrol dan definisi yang hilang dan teknologi yang diciptakan cukup baru sehingga untuk pemula tingkat pembelajarannya cukup sulit [32].

Pada penelitian [33] dirancang desain panel instrumen menggunakan WPF untuk membuktikan keunggulan aplikasi ini dalam mengendalikan modul akuisisi data HITPE101 melalui driver perangkat keras berdasarkan Visa. Hasil dari penelitian ini adalah sistem berhasil dibangun dengan cara memverifikasi panel yang telah didesain.

2.9 Visual Studio

Merupakan suatu perangkat lunak yang dapat digunakan untuk pengembangan berbagai macam aplikasi yang memiliki berbagai macam tipe antara lain aplikasi desktop (*Windows Form*, *CommandLine*, Aplikasi Web, *Windows Mobile*). Perangkat lunak ini memiliki lebih dari satu kompiler yaitu Visual Basic, C#, C++, InterDev, J++, F#, *Souce Safe* dan sebagainya [34]. Berikut adalah tampilan awal saat Visual Studio dijalankan pada Gambar 2.16.

			7 34	Lor Laption	
Visual Studio	Discover what's new in Enterp Lean about not features in Enterproc 2011 See relationed to ALT Features Of Explore what is new in Yourd States Team Services	rise 2015	10	5 H F	
Start New Project. Open From Science Control.	Reedy to Direct grown your experience? Connect to Azure 👁	New on Microsoft Platforms			
Recent NGU/PS Windowillionmulopekadaan T	News Mobilizing Existing NET LOB Apps (Video)	Add rep 1 con and true Macrosoft Office GraveFeet Druetopment			
Ratuys Ratuys Raturation Rescaled Rescaled Raturation	Using the Retry statem to make your JNET Azure doubt application more realient The portwas advently lossifiering, Microsoft Azure MIP. Recent, init all botion Code Comp catching and them of Microsoft and Coding to land.	Featured Videos	v 54	East Luciere Team Luplay	Decrem
nor Lill	Sector States and States and States				

Gambar 2.16 Tampilan awal Visual Studio

BAB III PERANCANGAN GUI, SISTEM KOMUNIKASI DAN PENYIMPANAN DATA RHINO ROBOT

Perancangan sistem pada penelitian ini fokus optimalisasi *Graphic User Interface* Rhino robot. Pengoptimalan terdiri dari sistem komunikasi dan penyimpanan data robot dibuat modular berbasis bahasa pemrograman C# serta desain tampilan dan tata letak menu pada GUI dibuat menggunakan *Windows Presentation Foundation* (WPF) pada Visual Studio.

3.1 Design Requirement

Sesuai dengan penjelasan sub bab 2.3, bahwa penulis mengerjakan perancangan OCU versi 2 bagian GUI. Pengerjaan ini terbagi atas 3 bagian dengan rincian tugas yang dikerjakan, fungsi yang diinginkan dan kebutuhan sistem atau target untuk penulis yang ada pada Tabel 3.1.

No	Tugas yang akan dikerjakan	Fungsi yang diinginkan	Kebutuhan sistem
1	Perancangan sistem	Membuat sistem komunikasi dengan pembagian 5 kelompok yaitu <i>transmitter</i> ,	Fungsi komunikasi dapat berfungsi tanpa ada eror ketika dijalankan atau <i>run</i>
1.	penyimpanan data robot	receiver, communication center, robot hub dan transceiver	Dapat menambahkan fungsi baru tanpa harus merubah sistem utama
2.	Desain halaman GUI	Membuat desain	Desain halaman GUI
		menarik	dijalankan atau <i>run</i>
		Membuat 8 halaman	Halaman GUI bisa berpindah ketika diberi masukan
3	Penataan halaman dan menu kendali GUI	GUI dengan menu- menu kendali yang lebih variatif	Ketika menu dipilih maka ada reaksi seperti berganti halaman atau kembali ke lavar operasi

 Tabel 3.1
 Design Requirement

Perancangan sistem komunikasi dan penyimpanan data robot akan dibuat secara modular sehingga penambahan fungsi tidak memerlukan perubahan pada sistem komunikasi utama. Lalu desain halama GUI menggunakan gambar yang telah dibuat oleh pihak desain perusahaan PT Bhimasena Research and Development. Mengingat GUI akan dipasarkan oleh pengguna, jadi desain untuk latar belakang tidak boleh dibuat sembarangan. Dan pada penataan halaman dan menu kendali GUI dibuat 8 halaman GUI yang sudah mencakup semua menu-menu kendali robot.

3.2 Data Flow Diagram Sistem Komunikasi dan Penyimpanan Data GUI Keseluruhan

Pengelompokan struktur program sangat diperlukan teruntuk fungsi yang mempunyai perkembangan di kemudian hari. Pada Gambar 3.1 merupakan struktur program komunikasi dan penyimpanan data keseluruhan GUI versi baru. Dimana terdapat pembagian, bagian kotak nomor 1 merupakan bagian yang penulis kerjakan. Input Panel dan Gamepad merupakan masukan dari pengguna. Data dari Input Panel dan Gamepad diberikan masing-masing kepada Arduino Controller dan Gamepad Controller. Setelah diproses, data-data tersebut diberikan kepada OCU Control Center. OCU Control Center berfungsi sebagai menerima data dari Arduino Controller dan Gamepad Controller serta mengirim data dari Page Input System untuk diberikan kepada LED Controller. Setelah data diolah, OCU Control Center mengirim data masing-masing kepada Page Input System dan Robot Input System. Data diproses pada Page Input System untuk diberikan kepada Page. Pada Robot Input System data diproses dan diberikan kepada Robot Hub. Pada Page Input System mengirim data LED Action kepada OCU Control Center. Setelah diproses data diberikan kepada LED Controller dan mengirimkan ke LED Circuity.

Sistem komunikasi pada GUI terbagi menjadi 5 bagian utama yaitu Receiver, Transmitter, Communication Center, Robot Hub dan Transceiver. Receiver pada sistem ini bertugas sebagai tempat pesan yang dikirim oleh Communication Center dan menyusun report untuk dikirim kepada Communication Center. Lalu tugas dari Transmitter yaitu sebagai tempat instruksi yang dikirim oleh Communication Center dan menyusun perintah untuk dikirim ke Communication Center. Tugas Communication Center sendiri yaitu mengirim report yang telah diberi oleh Receiver untuk Robot Hub, menerima instruksi dari Robot Hub, menerima pesan dari Transmitter dan mengirim perintah yang telah diberi Transmitter untuk Transceiver.



Keterangan: Gambar 1: Sistem dibuat oleh Tata Tanjung Tamara Gambar 2: Sistem dibuat oleh Livian Tjandra

Gambar 3.1 Data Flow Diagram Sistem Komunikasi dan Penyimpanan Data

Bagian Page berfungsi untuk menerima Page Input dari Page Input System, mengirim Page Info untuk Page Manager lalu data diberikan kepada Page Displayer. Lalu menerima Robot Info dari Robot Hub dan mengirim data Robot Action untuk Robot Hub. Bagian Robot Hub memiliki fungsi sebagai penerima data Robot Input dari Robot Input System, menerima data Robot Action dari Page serta mengirim Robot Info kepada Page dan Page Manager.

3.3 Protokol Rhino Robot

Protokol dalam ilmu komputer berarti seperangkat peraturan atau prosedur untuk mengirimkan data antara perangkat elektronik. Agar komputer satu dan komputer lain dapat mempertukarkan informasi, harus sudah ada persetujuan sebelumnya antarperangkat bagaimana struktur informasi dipertukarkan (dikirim dan diterima) [35]. Pada Rhino robot sendiri mempunyai protokol mengirim dan menerima data. Fungsi dari protokol ini untuk bagian pemrograman pada GUI. Seperti pada Pada Gambar 3.3 merupakan salah satu fungsi program untuk mengirim data protokol untuk robot selengkapnya bisa dilihat pada Lampiran A-1 dan tabel protokol mengirim data salah satu fungsi program B-1. Sedangkan pada Gambar 3.3 merupakan salah satu fungsi program

untuk protokol menerima data robot selengkapnya bisa dilihat pada Lampiran A-3 dan tabel protokol menerima data pada Lampiran B-2.

```
public void BeginTransmission(TimeSpan
TransmitInterval)
{
if (convertInstruction == null)
{
tokenSource = new CancellationTokenSource();
cToken = tokenSource.Token;
convertInstruction = Task.Run(async () =>
t
```

Gambar 3.2 Listing program protokol mengirim data

```
public override void ComposeCommand(object Instruction)
{
  latestInstruction = (RH_DataPacket)Instruction;
  int idxCRC = IdxPayload + latestInstruction.PayloadSize;
  this[IdxID] = latestInstruction.ID;
  this[IdxSize] = latestInstruction.PayloadSize;
  this[IdxSequence] = latestInstruction.Sequence;
```

Gambar 3.3 Listing program protokol menerima data

3.4 Rancangan Halaman GUI

Halaman GUI pada versi lama terdapat 20 halaman. Hal tersebut mempersulit pengguna untuk mengendalikan robot secara cepat karena harus melewati beberapa serangkaian halaman terlebih dahulu. Maka penulis membuat rencana pada Gambar 3.4 dimana dirancang halaman GUI dengan mempunyai 8 halaman yang bisa memenuhi menu-menu kendali. Ada beberapa menu fungsi yang dihapus. Hal tersebut sudah menjadi keputusan antara penulis dengan pembimbing perusahaan.

	_	
Halaman GUI Versi Lama		Halaman GUI Versi Baru
Halaman Utama	►	Halaman Utama
Pilih Komunikasi		Login Robot
Maintenance		Login Robor
Layar Operasi		Layar Operasi
Pop Up Main Menu		Pop Up Main Menu
Pop Up Pose	► ►	Pop Up Pengaturan Pose
Pop Up Pengaturan Kamera		Pop Up Pengaturan
Pengaturan Kamera Utama		Kamera

Gambar 3.4 Skema Rancang Halama GUI Versi Baru (1)



3.5 Perancangan Desain Tampilan dan Tata Letak Menu

Tampilan dan tata letak menu pada GUI sangat berperan penting dalam mengendalikan dan memantau robot. Oleh karena itu, perlunya optimalisasi antarmuka pengguna secara grafis. Berikut adalah *flowchart* GUI saat dijalankan pada Gambar 3.5. Saat GUI dijalankan halaman yang akan tampil pertama kali ialah halaman utama pada GUI. Pada Halaman utama sendiri terdiri dari 2 pilihan menu yaitu "Pilih Robot" dan "Matikan". Jika pengguna memilih "Pilih Robot" maka layar operasi berganti halaman *Login Robot*. Jika pengguna memilih "Matikan" maka GUI berhenti. Setelah itu pada halaman kedua yaitu *Login Robot*. Jika *user* memilih "Hubungkan" maka layar operasi menampilkan *Operation page*.

Pada layar operasi terdapat beberapa fungsi menu pendukung atau yang disebut menu utama. Pada menu utama terdiri dari 5 pengaturan diantaranya pengaturan kamera, pose *arm* robot, lampu, mode mengemudi dan mode manipulator. Menu-menu pengaturan tersebut pendukung mobilitas robot. Berikut adalah *flowchart* halaman menu utama GUI pada Gambar 3.6.



Gambar 3.5 Flowchart halaman utama GUI



Gambar 3.6 Flowchart halaman menu utama GUI

3.5.1 Desain Halaman Awal

Halaman utama pada GUI dirancang dengan resolusi Height = 768 dan Width = 1024 atau sesuai dengan monitor pada OCU. Desain pada halaman utama dibuat dengan latar belakang sama seperti GUI versi lama yaitu gambar manusia, alat pengendali dan robot. Lalu menu GUI dibuat 2 menu fungsi yaitu Pilih Robot dan Matikan. Fungsi dari menu Pilih Robot untuk memilih robot yang ingin dikendalikan oleh pengguna. Jika pilihan menu ini dipilih maka halaman akan berganti menjadi *Login Robot*. Lalu fungsi dari menu matikan untuk menutup atau menonaktifkan antarmuka. Latar gambar, simbol Rhino, dan simbol perusahaan pada GUI versi baru tidak berubah atau sama seperti latar

gambar versi sebelumnya. Perbedaan hanya terletak pada pilihan menu Pilih Robot dan Matikan. Berikut desain tampilan dan tata letak menu halaman utama pada Gambar 3.7 dan listing program salah satu fungsi membuat desain halaman awal pada Gambar 3.8 yang selengkapnya dapat lihat pada Lampiran A-6.



Gambar 3.7 Desain Halaman Awal

```
<Button x:Name="Button_Select" Content="PILIH ROBOT"
Margin="0,0,42,168"
Width="382" Height="60"
FontSize="32" FontFamily="Moire" Foreground="White"
Click="Button_Select_Click" Focusable="False"
HorizontalAlignment="Right"
VerticalAlignment="Bottom">
<Button.Background>
<SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/>
</Button.Background> </Button>
```

Gambar 3.8 Listing program Desain Halaman Awal

Listing program pada Gambar 3.8 menjelaskan salah satu pembuatan menu "Pilih Robot". Dimana menggunakan *button select* dengan ditulis PILIH ROBOT. Penentuan letak *button select* yang telah dibuat bisa ditentukan dengan memasukkan angka pada margin atau menentukan tempat dengan metode *drag and drop*. Pada bagian *Width* dan sebagainya ditentukan berdasarkan GUI versi lama.

3.5.2 Desain Halaman Verifikasi Robot

Halaman verifikasi robot merupakan bagian GUI yang baru yang sebelumnya belum ada. Dengan resolusi yang sama dengan halaman utama. Latar gambar sama seperti halaman kedua pada GUI versi lama.

Tujuan dibuat halaman ini untuk menghubungkan komunikasi antar OCU dan Robot. Halaman ini terdiri dari beberapa menu yaitu Seri Robot, Nomor Robot dan menu Hubungkan. Menu Seri Robot terdiri dari 2 ID dimana pengguna akan memberi masukan ID robot dengan cara *scroll* pilihan. Mengingat OCU berisi berupa tombol-tombol maka dibuat pilihan dengan cara *scroll up* atau *down*.

Menu Nomor Robot pun sama fungsinya dengan Seri Robot. Perbedaannya hanya pada jumlah *input* yang akan dimasukkan oleh pengguna. Setelah itu, fungsi menu Hubungkan adalah menghubungkan OCU dengan Robot yang telah disesuaikan Seri Robot dan Nomor Robot. Jika komunikasi sudah terhubung maka layar operasi akan muncul. Berikut desain tampilan halaman verifikasi robot pada Gambar 3.9 dan salah satu fungsi listing program pada Gambar 3.10 yang selengkapnya bisa dilihat pada Lampiran A-7.



Gambar 3.9 Desain Halaman Verifikasi Robot

```
<Button x:Name="Button_IDSeries" Content="XX"
HorizontalAlignment="Right"
Margin="0,0,654,28" VerticalAlignment="Bottom"
FontSize="64" FontFamily="Moire
Light" Foreground="White" Focusable="False">
<Button.BorderBrush>
<SolidColorBrush Color="#FF707070"
Opacity="0"/>
</Button.BorderBrush>
<Button.Background>
<SolidColorBrush Color="#FFDDDDDD"
Opacity="0"/> </Button.Background></Button>
Gambar 3.10 Listing program Halaman login robot
```

Listing program pada Gambar 3.10 menjelaskan salah satu pembuatan menu "ID Series" atau Seri robot. Dimana menggunakan *button select* dengan ditulis XX. Penentuan letak *button select* yang telah dibuat bisa ditentukan dengan memasukkan angka pada margin atau menentukan tempat dengan metode *drag and drop*. Pada bagian *Width* dan sebagainya ditentukan berdasarkan GUI versi lama. Warna diatur pada listing *opacity*.

3.5.3 Desain Halaman Layar Operasi

Layar operasi memiliki fungsi untuk memantau kawasan tempat robot berada dan keadaan robot secara langsung atau *real time*. Resolusi halaman ini sama seperti halaman utama dan halaman *Login Radio*. Halaman ini juga mempunyai beberapa fungsi menu diantaranya Layar Operasi, *Drivetrain*, Manipulator dan Robot 3D. Layar Operasi memiliki fungsi untuk menampilkan gambar yang telah dikirim oleh kamera robot secara langsung sehingga memudahkan para pengguna untuk mengendalikan robot. Lalu *Drivetrain* mempunya fungsi menunjukkan pengguna menggunakan mode-mode pengendalian robot. Setelah itu menu Manipulator berfungsi sebagai penunjuk kepada pengguna sedang menggunakan mode manipulator robot. Pada menu Robot 3D berfungsi untuk menampilkan visual robot 3 dimensi secara *real time* dengan fitur penunjuk arah mata angin agar pengguna mengetahui keadaan robot secara langsung. Pada Gambar 3.12 merupakan desain Layar Operasi GUI yang dibuat

Listing program pada Gambar 3.11 menjelaskan salah satu pembuatan gambar 3 Dimensi robot. Penentuan letak yang telah dibuat bisa ditentukan dengan memasukkan angka pada margin atau menentukan tempat dengan metode *drag and drop*. Pada bagian *Width* dan sebagainya ditentukan berdasarkan GUI versi lama. Warna diatur pada listing *opacity*.

```
<Image x:Name="image_Navigasi"
HorizontalAlignment="Right" Margin="1,0,32,200"
VerticalAlignment="Bottom" Width="945"
Source="/RGUI_PS;component/Image_Backgrounds/LO_Navig
asi.png"/> <Image x:Name="image_DrivetrainIndicator"
HorizontalAlignment="Right" Margin="0,0,168,602"
VerticalAlignment="Bottom" Width="40" Height="26"/>
```

Gambar 3.11 Listing Program Halaman Layar Operasi



Gambar 3.12 Desain Layar Operasi

3.5.4 Desain Halaman Pop Up Main Menu

Pop Up Menu Main Menu merupakan halaman menu utama yang terdiri dari 5 fungsi menu yaitu Pengaturan Pose, Pengaturan Kamera, Mode Mengemudi, Mode Manipulator dan Pengaturan Lampu. Resolusi halaman ini untuk *Height* = 576 dan Width = 960. Jika pengguna memilih Pengaturan Pose atau Pengaturan Kamera maka halaman akan terganti dengan *Pop Up Menu Pose* seperti pada Gambar 3.15 dan *Pop Up Menu Camera* seperti pada Gambar 3.19. Kendali sekunder ialah pilihan menu yang dibuat untuk pilihan yang lain dalam Mode Mengemudi, Mode Manipulator dan Pengaturan Lampu. Berikut desain tampilan *Pop Up Menu Pose* 3.13 serta salah satu fungsi listing program pada Gambar 3.14 yang selengkapnya pada Lampiran A-9.



Gambar 3.13 Desain halaman Pop Up Main Menu

```
<TextBlock x:Name="textBlock_MainMenu"
HorizontalAlignment="Left" Margin="27,24,0,0"
TextWrapping="Wrap" Text="Menu Utama"
VerticalAlignment="Top" Foreground="White"
FontFamily="Moire" FontSize="29.333"/>
<Button x:Name="button_PoseSetting"
Content="Pengaturan Pose"
```

Gambar 3.14 Listing program Halaman Pop Up Menu

3.5.5 Desain Halaman Pop Up Menu Pose

Pop Up Menu Pose merupakan halaman menu pose untuk robot. Resolusi halaman sama seperti Pop Up Main Menu. Fungsi menu pada halaman ini sebanyak 15 pengendali robot. Halaman ini dibuat agar mempermudah pengguna untuk mengendalikan robot secara otomatis hanya dengan memilih pilihan tersebut. Perlu diketahui bahwa robot juga bisa dikendalikan secara manual sehingga penelitian ini membuat menu praktis keadaan robot. Setiap menu masing-masing diberikan *icon* robot untuk memberitahu *user* jika memilih salah satu menu maka bentuk robot seperti pada *icon*. Berikut adalah desain tampilan Pop Up Menu Pose yang ada pada Gambar 3.15 serta salah satu listing program Halaman menu pose pada Gambar 3.16 yang selengkapnya bisa dilihat pada Lampiran A-10.



Gambar 3.15 Desain halaman Pop Up Menu Pose

```
<TextBlock x:Name="textBlock_Pose" HorizontalAlignment="Right"
Margin="0,0,816,507" TextWrapping="Wrap" Text="Pose"
VerticalAlignment="Bottom" Foreground="White" FontFamily="Moire"
FontSize="29.333"/>
<Button x:Name="button_Drive"
HorizontalAlignment="Right"
```

Gambar 3.16 Listing Program Halaman Pengaturan Pose

3.5.6 Desain Halaman Pop Up Menu Preset Pose

Halaman *Pop Up Menu Preset Pose* pada Gambar 3.17 terdiri dari 6 Menu Preset atau Penyimpanan, Menu Simpan Posisi saat ini, Menu Atur Manual, 7 fungsi pengaturan robot dan 2 tampilan 3D Robot. Tampilan 3D Robot yang pertama (sebelah kiri) merupakan tampilan jika pengguna memilih salah satu dari 7 fungsi pengaturan robot. Contohnya jika pengguna memilih *Flipper* maka 3D Robot menampilkan keadaan robot pada saat *Flipper*. Sedangkan pada 3D Robot *Real Time* menunjukkan keadaan robot pada saat itu juga.

Menu Simpan Posisi saat ini memiliki fungsi untuk menyimpan keadaan robot saat ini yang akan disimpan pada preset. Jika pengguna ingin menyimpan keadaan robot langsung secara otomatis akan tersimpan pada menu preset. Menu Atur Manual berfungsi untuk pilihan ketika mode yang diinginkan pengguna tidak ada dalam pilihan-pilihan pada pengaturan pose preset. Pada Gambar 3.18 merupakan salah satu listing program halaman menu pose preset yang selengkapnya bisa dilihat pada Lampiran A-11

Pengaturan	Pose Preset		
			Flipper
			Turntable
			Shoulder
			Elbow
			Wrist.
			CameraTilt
		0	Camera Pan
PRESET 1	PRESET 4	Simpan posisi saat	Atur manual
PRESET 2	PRESET 5	ini	Handal
PRESET 1	PRESET 6		

Gambar 3.17 Desain Pop Up Menu Preset Pose

<Grid x:Name="PresetPosePopupLayoutRoot" Width="960" Height="576" HorizontalAlignment="Left" VerticalAlignment="Bottom"> <Grid.Background> <ImageBrush>

Gambar 3.18 Listing program halaman preset pose

3.5.7 Desain Halaman Pop Up Menu Camera

Pop Up Menu Camera merupakan sebuah halaman yang berfungsi untuk mengatur susunan gambar kamera yang ada pada Rhino robot. Perlu diketahui bahwa robot memiliki 4 kamera yang terletak pada Gripper, Arm, depan chasis dan belakang chasis sehinga dibuat menu pilihan agar pengguna dapat menentukan gambar secara langsung yang diinginkan untuk tampil pada layar operasi. Berikut adalah desain Pop Up Menu Camera yang telah dibuat pada Gambar 3.19.

A MARCE AND		and the second		- Second Second
Pilihan		susunan		Simpan
Jtama :	Status			Preset 1
enggenggam :	Status	Layar1	Layar2	Preset 2
Depan :	Status			Preset 3
elakang :	Status			
		Lavar4	Layar3	

Gambar 3.19 Desain Pop Up Menu Camera

Langkah pertama untuk menyusun tampilan gambar adalah menentukan pilihan kamera yang diinginkan. Pada Gambar 3.19 menunjukkan 4 kamera dengan disertai menu Status pada masing-masing pilihan. Menu Status berfungsi untuk menunjukkan *OFF* atau *ON* kamera. Jika salah satu pilihan kamera hidup maka pada layar operasi menampilkan gambar yang kameranya dalam keadaan on. Susunan gambar dibuat 4 pilihan yaitu menampilkan gambar dari 1 kamera saja seperti pada Gambar 3.20. Lalu ketika 2 kamera dinyalakan maka susunan layar terbagi menjadi 2 seperti pada Gambar 3.21. Setelah itu jika 3 kamera dinyalakan maka susunan layar terbagi menjadi 3 seperti pada Gambar 3.22. Terakhir pada saat 4 kamera menyala maka susunan layar terbagi menjadi 4 seperti pada Gambar 3.19.

Pilihan menu Simpan terdapat 3 menu yaitu Preset 1, Preset 2 dan Preset 3. Menu ini berfungsi sebagai mode penyimpanan susunan layar serta kamera hidup yang disesuaikan oleh keinginan pengguna. Misal pengguna menginginkan susunan layar terbagi menjadi 3 dengan layar pertama menampilkan gambar dari kamera penggenggam, layar 2 menampilkan gambar dari kamera utama dan layar 3 menampilkan gambar dari kamera belakang. Pada Gambar 3.23 merupakan salah satu fungsi listing program halaman pengaturan kamera yang selengkapnya bisa dilihat pada Lampiran A-12.

lihan		Susunan	Simpan
Utama :	CN		Preset 1
Penggenggam :	OFF		Preset 2
Depan :	OFF	and the second sec	Preset 3
Belakang :	OIFF	UTAMA	

Gambar 3.20 Desain halaman Pop Up Menu Camera 1 Layar

Pilihan		Susunan		Simpan
ltama :	ON			Preset 1
enggenggam :	ON			Preset 2
epan :	DIT	(million	DENCOTINGAN	Preset 3
Belakang :	017	UTAMA	PENGGENGGAM	

ambar 3.21	Desain halaman	Pop Up Menu	Camera 2 Layar

Utama : ON			
			Preset 1
Penggenggam : ON		PENGGENGGAM	Preset 2
Depan : ON			Preset 3
Belakang : OIT	UTAMA		

Gambar 3.22 Desain halaman Pop Up Menu Camera 3 Layar

Listing program Gambar 3.23 menjelaskan salah satu pembuatan bagian "Pilihan" beserta isinya. Penentuan letak yang telah dibuat bisa ditentukan dengan memasukkan angka pada margin atau menentukan tempat dengan metode *drag and drop*. Pada bagian *Width* dan sebagainya ditentukan berdasarkan GUI versi lama. Warna diatur pada listing *opacity*. Ukuran tulisan dibuat 29.3 dan 29.1

```
<TextBlock x:Name="textBlock_Camera"
HorizontalAlignment="Left" Margin="54,36,0,0"
Text="Kamera" VerticalAlignment="Top"
Foreground="White" FontFamily="Moire"
FontSize="29.333"/>
<TextBlock x:Name="textBlock_Selection"
HorizontalAlignment="Left" Margin="40,96,0,0"
Text="Pilihan" VerticalAlignment="Top"
Foreground="White" FontFamily="Moire"
FontSize="21.333"/>
```

```
Gambar 3.23 Listing Program Halaman Pengaturan Kamera
```

3.5.8 Desain Halaman Pop Up Menu Ilumination

Pop Up Menu Ilumination merupakan halaman menu penerangan lampu yang terletak di dekat kamera robot. Pada Gambar 3.24 dibuat desain Pop Up Menu Ilumination yang terdiri dari pilihan lampu, menu status menyala dan intensitas cahaya. Pada pilihan lampu terdapat 4 buah yaitu Utama, Penggenggam, Depan, dan Belakang. Perlu diketahui juga penempatan lampu juga bersama dengan kamera pada robot agar pengguna dapat melihat gambar dalam keadaan gelap sekalipun. Lalu disertakan status untuk menunjukkan OFF atau ON lampu dan pengaturan cahaya juga di tentukan dengan menu intensitas. Pada Gambar 3.25 merupakan salah satu fungsi listing program halaman pengaturan lampu yang selengkapnya ada pada Lampiran A-13.

Peneranga	in	
Status :	Dinonaktifkan	Intensitas :
Utama :	Padam	
Penggenggam :	Padam	
Depan :	Padam	
2		

Gambar 3.24 Desain Pop Up Menu Ilumination

```
<Button x:Name="button_Status"
Content="Dinonaktifkan"
HorizontalAlignment="Right" Margin="0,0,181,260"
VerticalAlignment="Bottom"
Width="140" Height="30"
FontSize="16" FontFamily="Moire" Foreground="White"
Focusable="False" Click="button_Status_Click">
<Button.Background="White"
Focusable="False" Click="button_Status_Click">
<Button.Background="White"
Focusable="False" Click="button_Status_Click">
<Button.Background="White"
Focusable="False" Click="button_Status_Click">
<Button.Background="White"
<SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/>
</Button.Background>
```

Gambar 3.25 Listing Program Halaman Pengaturan Lampu

3.6 Metodologi Pengujian

Setelah rancangan dibuat, perlu adanya untuk membuat rencana pengujian. Hal tersebut menjadi akan menjadi bukti bahwa penelitian yang dilakukan dapat berjalan dengan baik. Pada penelitin ini pengujian diambil 2 cara yaitu:

- 1. Mengetahui jumlah kapasitas memori pada GUI versi baru.
- 2. Pengujian GUI dijalankan menggunakan IDE Visual Studio.

3.6.1 Tata Cara Pengambilan Data Kapasitas Memori GUI

Berikut adalah tata cara pengambilan data kapasitas memori GUI:

1. Buka folder pekerjaan GUI seperti pada Gambar 3.26

RGULUS	4/14/2018 1.50 PM	Tilefolde
RhinoGUNS	4/15/30% 921 PM	File Rolder

Gambar 3.26 Tata cara pengambilan data kapasitas memori 1

2. Lalu menuju file GUI seperi pada Gambar 3.27

.vs	4/18/2018 1:59 PM	File folder	
DTAR_08E86330_4835_4B5C_9E5A_61F37A	9/5/2016 9:12 AM	File folder	
📙 obj	4/18/2018 1:59 PM	File folder	
📙 packages	4/18/2018 1:59 PM	File folder	
RGUI_PS	7/4/2018 2:06 PM	File folder	
RGUI_PS.sln	10/30/2017 5:30 PM	Microsoft Visual S	1 KB
	Size: 988 bytes Type: Microsoft Visual St Version: Visual Studio 14 Date modified: 10/30/20	tudio Solution 17 5:30 PM	

Gambar 3.27 Tata cara pengambilan data kapasitas memori 2

3. Klik kiri pada kursor dan tampilan akan seperti pada Gambar 3.28. Lalu pilih properties dan akan muncul rincian datanya.



Gambar 3.28 Tata cara pengambilan data kapasitas memori 3

3.6.2 Tata Cara Pengujian GUI

Berikut adalah tata cara pengujian GUI menggunakan Visual Studio:

- 1. Buka Visual Studio
- 2. Buka file GUI yang akan dijalankan
- 3. Jika sudah terbuka maka tampilan akan seperti pada Gambar 3.29. Lalu klik start seperti pada gambar yang dilingkari.



Gambar 3.29 Tata Cara Pengujian GUI 1

4. Jika berhasil maka akan seperti pada Gambar 3.30. Dimana akan diambil data seperti gambar yang dilingkari. Data ini diambil dari salah satu fitur visual studio bernama *Diagnostic Tool.* Data yang dihasilkan CPU *Usage* dan *Memory Usage*. Tetapi penulis hanya mengambil dan menganalisis bagian *Memory Usage*.



Gambar 3.30 Tata Cara Pengujian GUI 2

BAB IV PENGUJIAN DAN ANALISIS

Pada bab ini membahas tentang 2 bahasan utama yaitu analisis perbandingan sistem komunikasi dan penyimpanan data antara GUI versi lama dan baru serta pengujian memori. Pada sub bab analisis sistem komunikasi dan penyimpanan data robot menjelaskan struktur sistem GUI versi lama lalu dibandingkan dengan sistem yang telah dibuat. Pada sub bab uji memori GUI terbagi menjadi 2 yaitu kapasitas memori GUI dan uji kinerja kecepatan GUI. Pada kapasitas memori sendiri membandingkan antara memori GUI versi lama dan baru. Sedangkan pada uji kinerja GUI mengambil data melalui fitur *diagnostic tool* pada Visual Studio dan membandingkan dengan GUI versi lama tetapi perbandingan hanya dengan halaman yang sama. Hal tersebut terjadi karena pada GUI versi lama terdapat 20 halaman dan versi baru terdapat 8 halaman.

4.1 Hasil Pengujian

Penelitian telah diuji dengan 2 cara yaitu mengetahui kapasitas memori GUI dan pengujian GUI dijalankan yang menghasilkan data kinerja memori seperti pada Tabel 4.1. Berdasarkan tabel tersebut maka sistem yang telah dirancang pada Bab III telah berhasil dilakukan.

			Uji	
No	Tugas yang akan dikerjakan	Fungsi yang diinginkan	Berhasil	Tidak
				Berhasil
1.	Perancangan sistem komunikasi dan penyimpanan data robot	Membuat sistem komunikasi dengan pembagian 5 kelompok yaitu <i>transmitter</i> , <i>receiver</i> , <i>communication center</i> , <i>robot hub</i> dan <i>transceiver</i>	V	
2.	Desain halaman GUI	Membuat desain halaman GUI lebih menarik	V	
3	Penataan halaman dan menu kendali GUI	Membuat 8 halaman GUI dengan menu- menu kendali yang lebih variatif	V	

Tabel 4.1 Hasil Pengujian

4.2 Analisis Sistem Komunikasi dan Penyimpanan Data Robot GUI Versi Lama dan Baru

Pada Gambar 4.1 merupakan struktur program komunikasi dan data robot GUI versi lama dan Gambar 3.1 adalah struktur program komunikasi dan data robot GUI versi baru. Data komunikasi GUI versi lama berkumpul pada *Page* dimana isi dari halaman ini terdiri dari banyak variabel, *field member*, *object*, dan *class*. Jika pengembang ingin merubah sistem komunikasi harus mendefinisikan atau menghapus variabel baru. Jika tidak ingin menghapus variabel maka fungsi-fungsi komunikasi yang telah dibuat mau tidak mau harus didefinisikan. Sedangkan data komunikasi GUI versi baru ditangani oleh 3 *class* yaitu OCU *Receiver*, OCU *Transmitter* dan OCU *Communication Center*. Jika komunikasi berbeda maka data bisa diturunkan pada *class* sesuai dengan kebutuhan dan fungsinya. Ketika *run time* komunikasi yang berjalan hanya satu saja sesuai dengan keinginan pengguna.

Setelah itu penanganan *message* dari robot pada GUI versi lama diolah oleh *Rhino Link*. Dimana GUI versi baru diproses oleh OCU *Receiver* melalui OCU *Communication Center*. Jika *page* GUI versi lama ingin mendeklarasi macam-macam robot maka definisi harus ada ketika *runtime*. Hal tersebut terjadi karena program tidak dapat diturunkan. Jika GUI versi baru sudah tidak terhubung pada robot dan ingin terhubung dengan robot lain yang telah didefinisi sebelumnya tidak perlu ada ketika *runtime*. Karena *object* OCU *Receiver* dapat bertransformasi sesuai dengan robot yang terhubung.

Transmitter GUI versi lama terdapat pada page yang memiliki akses langsung ke Transceiver robot untuk mengirim data. Sedangkan Transmitter GUI versi baru terdapat pada OCU Transmitter. Akses menuju Transceiver melalui OCU Communication Center terlebih dahulu untuk diolah.



Gambar 4.1 Sistem Komunikasi dan Penyimpanan Data GUI versi lama

Penanganan data robot GUI versi lama terdapat di *Rhino Link* dan pada GUI versi baru terdapat di *RobotHub*. Jika data pada *Rhino Link* berubah atau diambil dengam metode khusus maka harus membuat definisi yang banyak. Sedangkan *RobotHub* bisa bertransformasi pada tempat penyimpanan data robot lain.

4.3 Uji Memori GUI

Pada pengujian memori GUI ini mengambil 2 pengujian yaitu uji kapasitas memori dan uji *memory usage* GUI ketika saat dijalankan dengan Visual Studio.

4.2.1. Kapasitas Memori GUI

Penggunaan memori pada sebuah GUI berpengaruh pada kinerja kecepatan saat GUI dijalankan. Berikut adalah data kapasitas memori GUI yang telah dibuat dan GUI versi lama pada Tabel 4.2.

Dari data tersebut dapat ditarik kesimpulan bahwa kapasitas memori GUI versi baru lebih kecil dibanding dengan memori GUI versi lama. Dapat dilihat pada persamaan 4.1 selisih *size* dan persamaan 4.4 persentase GUI versi lama dan GUI versi baru.

Versi GUI	Size (MB)	Contains		
VUSIOUT		Files	Folder	
Lama	92,8	682	126	
Baru	57,4	255	43	

Tabel 4.2 Penggunaan Memori GUI

Selisih = Jumlah Size GUI versi lama – Jumlah Size GUI versi baru(4.1)Selisih = 92,8 – 57,4(4.2)Selisih = 35,4 MB(4.3)

$$Persentase = \frac{Hasil selisih}{Jumlah Size GUI versi lama} x 100 \%$$
(4.4)

$$Persentase = \frac{35.4}{92.8} x 100 \%$$
(4.5)

$$Persentase = 0.38 x 100 \%$$
(4.6)

$$Persentase = 38\%$$
(4.7)

Dari perhitungan tersebut didapat selisih *size* memori antara GUI versi lama dengan GUI versi baru sebesar 35,4 MB atau persentase 38% lebih ringan GUI versi baru.

4.2.2. Data Uji dan Analisis Kinerja GUI

Pengujian kinerja kecepatan memori GUI didapat dari *Diagnostic Tool* pada fitur Visual Studio. Pada sub bab berikut ini menjelaskan dan menganalisis data-data *memory usage* GUI versi lama dengan versi baru. Data uji kinerja per halaman GUI versi lama pada Lampiran 0.

Pada saat GUI berjalan, halaman yang pertama kali muncul adalah start menu page atau halaman utama menggunakan memori sebesar 227 MB. Pada halaman utama terdapat 3 menu yaitu Pilih Robot, Maintenance dan Shut Down. Ketika pengguna memilih Pilih Robot maka halaman akan berganti Pilih Komunikasi yang menghasilkan memori sebesar 287 MB. Jika pengguna memilih menu Maintenance maka halama utama berganti halaman maintenance dengan perubahan Process Memory sebesar 289 MB dan ketika user memilih menu Shut Down secara otomatis program akan terhenti. Saat berganti halaman operasi memori mengalami peningkatan memori lavar vang menghasilkan sebesar 541 MB. Faktor penyebab kenaikan memori karena pada layar operasi terdapat tampilan gambar kamera, 3D Robot, Status Baterai, Status Koneksi dan sebagainya sehingga membutuhkan process memory yang besar juga.

Saat *Pop Up Main Menu* ditampilkan pada layar, *process memory* meningkat menjadi 650 MB. Setelah itu saat *Pop Up* Pengaturan Pose dipilih *process memory* yang dihasilkan menjadi 650 MB. Lalu pada menu *Pop Up* Pengaturan Kamera terdiri dari menu Pengaturan Kamera Utama, Pengaturan Kamera 2, Tukar Kamera, Tunjukkan Video *Full Screen*, Rekam Gambar dan Rekam Layar OCU menghasilkan memori yang sama yaitu 650 MB. Memori pada menu Pengaturan Kamera Utama dan Pengaturan Kamera 2 sama sama menghasilkan memori 650 MB. Lalu menu Tukar Kamera menghasilkan memori 651 MB, menu Tunjukkan Video *Full Screen* menghasilkan memori 653 MB. Sedangkan menu Rekam Gambar dan Rekam Layar OCU menghasilkan memori yang sama yaitu 655 MB.

Saat *Pop Up* Pengaturan Lampu, Audio, Mode Mengemudi, Mode Inspeksi dan Mode *Gripper* berdasarkan Tabel 13 masing-masing menghasilkan memori yang sama yaitu 660 MB. Setelah itu untuk *process memory* menu *Pop Up* Mode lanjutan sebesar 662 MB dan menu *Pop Up* Keluar sebesar 665 MB. Pada persamaan 4.8 rata-rata penggunaan *memory* saat GUI versi lama dijalankan.

$$rata - rata = \frac{\sum process \ memory}{4.8}$$

$$Rata - rata = \frac{1}{20}$$
(4.9)

$$Rata - rata = 590,9 MB$$
(4.10)

Berikut adalah data *process memory* halaman GUI versi baru pada Tabel 4.3.

No	Halaman	Process Memory (MB)
1	Start Menu Page	198
2	Login Robot	256
3	Operation Page	496
4	Pop Up Main Menu	528
5	Pop Up Pengaturan Pose	655
6	Pop Up Preset Pose	716
7	Pop Up Pengaturan Kamera	630
8	Pop Up Ilumination	629

Tabel 4.3 Process Memory Halaman GUI versi baru

Pada saat GUI berjalan, halaman yang pertama kali muncul adalah start menu page atau halaman utama. Process Memory yang dihasilkan sebesar 198 MB. Lalu ketika berganti halaman menjadi Login Robot, process memory yang dihasilkan meningkat menjadi sebesar 256 MB. Faktor terjadinya peningkatan memori karena halaman login mempunyai menu dan fungsi untuk verifikasi seri robot dan nomor robot untuk menghubungkan alat kendali dengan robot. Ketika pengguna sudah memasukkan seri dan nomor robot, maka halaman akan berganti ke Operation Page. Pada layar operasi process memory yang dihasilkan meningkat menjadi 496 MB. Penyebab terjadinya peningkatan memori adanya fungsi pendukung seperti 3D Robot dan tampilan gambar.

Saat Pop Up Main Menu ditampilkan pada layar, process memory meningkat menjadi 528 MB. Namun peningkatan ini hanya bersifat sementara. Artinnya peningkatan memori terjadi hanya saat menu dipilih. Ketika sudah terpilih, maka process memory kembali turun. Setelah itu saat Pop Up Pengaturan Pose dipilih process memory yang dihasilkan meningkat menjadi 655 MB. Faktor penyebab peningkatan memori karena halaman ini mempunyai 15 fungsi menu gerak robot dan 12 icon robot. Kemudian Pop Up Preset Pose yang menjadi bagian dari pengaturan pose. Saat dipilih, process memory yang dihasilkan sebesar 716 MB. Hal tersebut terjadi adanya fungsi pendukung 3D Robot berjumlah 2, menu-menu *Preset Pose* dan menu atur manual sehingga memakan memori yang cukup besar. Setelah itu saat memilih *Pop Up* Pengaturan Kamera memori mengalami penurunan memori menjadi 630 MB dan *Pop Up Ilumination* atau penerangan juga mengalami penurunan *process memory* menjadi 629 MB. Pada persamaan 4.11 rata-rata pemakaian memori saat GUI versi baru dijalankan.

$$rata - rata = \frac{\sum process memory}{\circ}$$
(4.11)

$$Rata - rata = \frac{4108}{8} \tag{4.12}$$

$$Rata - rata = 513,5 MB \tag{4.13}$$

Berikut adalah data-data perbandingan *memory usage* halaman yang sama antara GUI versi lama dengan versi baru pada Tabel 4.4 dan ditampilkan dalam bentuk grafik pada Gambar 4.2. Halaman-halaman yang sama pada 2 GUI ini terdiri dari *Start Menu Page, Operation Page, Pop Up Main Menu, Pop Up* Pengaturan Pose, *Pop Up* Pengaturan Kamera, dan *Pop Up Ilumination* atau penerangan.

No	Halaman	GUI Versi Lama	GUI Versi Baru	
1	Start Menu Page	227	198	
2	Operation Page	541	496	
3	Pop Up Main Menu	633	528	
4	Pop Up Pengaturan Pose	650	655	
5	Pop Up Pengaturan Kamera	650	630	
6	Pop Up Ilumination	660	629	

Tabel 4.4 Perbanding penggunaan memori GUI versi lama dan baru



Gambar 4.2 Grafik Perbandingan Process Memory GUI

4.4 Desain Tampilan dan Penataan Letak Menu

Hasil pada desain tampilan dan penataan letak menu GUI dilihat saat GUI dijalankan pada IDE Visual Studio. Pada Tabel 4.5

No	Halaman	Desain dan Tata Letak Menu Uji		
			Berhasil	Tidak
1	Start Menu Page		V	
2	Login Robot	Masakkan ID Robot	V	
3	Operation Page	Robot Name Devention Arm Mode	V	
4	Pop Up Main Menu	Menu Utama Kendali Sekunder Pengaturan Pose Mode Mengemudi Rengaturan Kamera Mode Manjpulator Pengaturan Lampu	V	

Tabel 4.5 Hasil Uji Desain Tampilan pada GUI saat dijalankan

No	Halaman	Desain dan Tata Letak Menu	Uji	
			Berhasil	Tidak
5	<i>Pop Up</i> Pengaturan Pose	Pose Integrand Money Integrand Money Integran	V	
6	Pop Up Preset Pose	Pengaturan Pose Preset	V	
7	<i>Pop Up</i> Pengaturan Kamera	Kamera Pilhan Susunan Simpan Durum : Saari Progorogan : Saari Dopae : Saari Bedulang : Saari Layar4 Layar3	V	
8	<i>Pop Up</i> Ilumination	Penerangan Status : Dinonaktikan Intensitas : Utama : Padam Penggenggam : Padam Depan : Padam	V	

Tabel 4.5 Hasil Uji Desain Tampilan pada GUI saat dijalankan (Lanjutan)

BAB V PENUTUP

Setelah melakukan perencanaan, perancangan dan pengujian alat, dapat diambil kesimpulan dan memberikan saran demi penyempurnaan penelitian ini. Sistem komunikasi dan penyimpanan data robot pada GUI versi baru telah dibuat secara modular dengan 3 bagian sistem komunikasi diantaranya Transmitter, Receiver dan Control Center. Selain itu, Penggunaan memori yang dihasilkan oleh GUI versi baru sebesar 57,4 MB atau 38% lebih ringan dibandingkan dengan penggunaan memori GUI versi lama sebesar 92,8 MB. Faktor penyebab GUI versi baru lebih ringan karena hanya menggunakan WPF. Setelah itu, Memory Usage yang digunakan GUI versi baru saat dijalankan ratarata memakai 513,5 MB. Sedangkan untuk GUI versi lama saat dijalankan rata-rata penggunaan memori sebesar 590,9 MB. Hal tersebut membuktikan bahwa versi baru lebih cepat dan ringan saat dijalankan. Serta desain tampilan dan tata letak menu pada GUI versi baru berjumlah 8 halaman dengan rincian 3 halaman utama dan 5 halaman menu utama. Sedangkan untuk versi lama berjumlah 20 halaman dengan rincian 3 halaman utama dan 17 halaman menu utama.

Terdapat pulan saran dalam pengembangan penelitian ini yaitu mengenai uji coba langsung pada OCU dengan terhubung robot. Dikarenakan sewaktu pengerjaan alat, robot sedang mengalami kerusakan bagian elektrik maka pengujian yang dilakukan berupa menjalankan GUI dengan laptop perusahaan. Hal tersebut berguna untuk mengetahui komunikasi OCU dan Robot berjalan dengan baik atau tidak.

-----Halaman ini sengaja dikosongkan---

DAFTAR PUSTAKA

- A. Nggedho and A. Y. E. Prasetyo, "Prototipe Simulasi Robot Penjinak Bom Dengan Pengendalian Android Berbasis Mikrokontroller Atmega16," Malang.
- [2] M. W. Carey, E. M. Kurz, J. D. Matte, T. D. Perrault, and T. Padir, "Novel EOD Robot Design with Dexterous Gripper and Intuitive Teleoperation," Massachusetts, 2012.
- [3] "The Very First Bomb Disposal Robot," 2014. [Online]. Available: http://www.military.com/video/ammunition-andexplosives/explosive-ordnance-disposal/the-first-bombdisposal-robot/3059244734001. [Accessed: 19-Jul-2018].
- [4] A. K. Bin Motaleb, M. B. Hoque, and A. Hoque, "Bomb disposal Robot Discarding explosive through wireless controlled method," 2016 Int. Conf. Innov. Sci. Eng. Technol. ICISET 2016, 2017.
- [5] M. Fracchia *et al.*, "Low-cost explosive ordnance disposal robot for deployment in Southeast Asia," *2015 IEEE Canada Int. Humanit. Technol. Conf. IHTC 2015*, 2015.
- [6] W. Zhang, J. Yuan, J. Li, and Z. Tang, "The optimization scheme for eod robot based on supervising control architecture," 2008 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2008, no. 051111015, pp. 1421–1426, 2008.
- [7] SDR Tactical Robots, "EOD Robots with Arms.".
- [8] N. Checka, S. Schaffert, D. Demirdjian, J. Falkowski, and D. H. Grollman, "Handheld operator control unit," *Proc. seventh Annu. ACM/IEEE Int. Conf. Human-Robot Interact. HRI '12*, p. 137, 2012.
- [9] J. Crossman, R. Marinier, and E. B. Olson, "A hands-off, multi-robot display for communicating situation awareness to operators," *Proc. 2012 Int. Conf. Collab. Technol. Syst. CTS* 2012, pp. 109–116, 2012.
- [10] R. Y. Al Ashi and A. Al Ameri, "Introduction to Graphical User Interface (GUI) MATLAB 6.5," *IEEE UAEU Student Branch UAE Univ.*, pp. 1–35, 2004.
- [11] W. L. Martinez, "Graphical user interfaces," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 3, no. 2, pp. 119–133, 2011.
- [12] I. E. Sutherland, "Sketchpad: A man machine graphical

communication system," no. 574, 2003.

- M. H. Zulkefli, K. A. M. Annuar, S. H. Johari, M. R. M. Sapiee, and S. Ahmad, "Graphical User Interface (GUI) Controlled Mobile Robot," *J. Adv. Res. Comput. Appl.*, vol. 1, no. 1, pp. 42–49, 2015.
- [14] A. Nugroho, D. W. Chandra, J. Prestiliano, and Y. R. Beeh, *Cara Mudah dan Cepat Belajar Pemrograman C#.NET*. Yogyakarta: Andi Offset, 2013.
- [15] A. Hejlsberg, "C Sharp Programming Language," p. 704, 2006.
- [16] D. Clark, "Beginning C # Object Oriented Programming, 2nd Ed.," p. 373, 2013.
- [17] A. Rachmatullah, "Mempelajari C#: Bahasa Pemrograman Modern," pp. 1–136, 2002.
- [18] B. Hartanto, *Memahami Visual C#.Net secara mudah*. Yogyakarta: Penerbit Andi, 2008.
- [19] Microsoft, "Classes (C# Programming Guide)," 2018. [Online]. Available: https://docs.microsoft.com/enus/dotnet/csharp/programming-guide/classes-andstructs/classes. [Accessed: 12-Apr-2018].
- [20] R. Miles, C # Programming Yellow Book, no. September. England: Departmen of Computer Science University of Hull, 2015.
- [21] Tutorialspoint, "C# Variables," 2015. [Online]. Available: https://www.tutorialspoint.com/csharp/csharp_variables.htm. [Accessed: 12-Apr-2018].
- [22] Tutorialspoint, "C# Programing tutorial," *Tutorialspoint*, p. 339, 2014.
- [23] "Chapter 3 : Writing C # Expressions Unary Operators," 2001, pp. 1–19.
- [24] Sonisitez, "Ekspresi Operator dan Operand di CSharp," 2015. [Online]. Available: https://www.noos.co.id/web/ekspresioperator-dan-operand-di-csharp-90.html. [Accessed: 13-Apr-2018].
- [25] A. Kurniawan *et al.*, "Pengenalan Bahasa C#," pp. 17–19, 2004.
- [26] Junindar, "C# Dan Dapper,".
- [27] M. R. Faisal, Seri Belajar ASP.NET : Membangun Aplikasi Web Mudah dan Cepat. 2008.

- [28] "Belajar Pemrograman C# Unity 3d Break and Continue Statement," 2017. [Online]. Available: http://www.beelancah.com/2017/02/belajar-pemrograman-cunity-3d-break-and-continue-keywords.html. [Accessed: 13-Apr-2018].
- [29] D. Ariyus and R. A. K. R., *Komunikasi Data*. Yogyakarta: Penerbit Andi, 2008.
- [30] F. C. Lunenburg, "Communication : The Process, Barriers, And Improving Effectiveness," *Schooling*, vol. 1, pp. 1–11, 2010.
- [31] S. Sukaridhoto, "Komunikasi Data dan Komputer," Surabaya, 2016.
- [32] A. Misra, "Use of Windows Presentation Foundation and Windows Forms in Windows Application Programming," vol. 7, no. 7, pp. 20–24, 2016.
- [33] F. Lei, "Design of the Instrument Soft Panel using Windows Presentation Foundation," pp. 927–929, 2015.
- [34] R. Yesputra, *Belajar Visual Basic*. *Net dengan Visual Studio* 2010, no. October. 2017.
- [35] "Protocol," Encyclopedia Britannica. .

-----Halaman ini sengaja dikosongkan---
LAMPIRAN A

A-1 Listing Program Transmitter

using System; using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading; using System.Threading.Tasks; using System.Windows.Resources; using System.Windows.Threading;

```
namespace RGUI_PS
{
    public class OCUTransmitter
    {
        #region Initialization
        public OCUTransmitter(int TxMaxLength)
        {
            txBuffer = new byte[TxMaxLength];
            instructionBox = new Queue<object>(10);
        }
        #endregion
```

#region Field
private Queue<object> instructionBox;
private byte[] txBuffer;

private Task convertInstruction; private CancellationTokenSource tokenSource; private CancellationToken cToken;

public delegate void CommandReady(byte[] Command, int Length);

private event CommandReady newCommand; #endregion

#region Properties

```
#region Methods
    /// <summary>
    /// Start an asynchronus task that read instructions in instruction
queue and coverts them into commands every Transmit Interval.
    /// </summary>
    public void BeginTransmission(TimeSpan TransmitInterval)
     ł
       if (convertInstruction == null)
       ł
         tokenSource = new CancellationTokenSource();
         cToken = tokenSource.Token;
         convertInstruction = Task.Run(async () =>
         ł
            while (!cToken.IsCancellationRequested)
            ł
              if (instructionBox != null && instructionBox.Count >
0)
                 ComposeCommand(instructionBox.Dequeue());
              await Task.Delay(TransmitInterval);
         }, cToken);
      }
    2
    /// <summary>
    /// Stop the asynchronus task that started by BeginTransmission
Methods
    /// </summary>
    public void EndTransmission() { tokenSource.Cancel(); }
```

A-2 Listing Program Turunan Transmitter

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace RGUI PS
  class RhinoTransmitter : OCUTransmitter
  ł
     #region Initialization
    public RhinoTransmitter(int TxMaxLength, int
PayloadMaxLength) : base(TxMaxLength)
     ł
       latestInstruction = new RH DataPacket(PayloadMaxLength);
       this[0] = Convert.ToByte('R');
       this[1] = Convert.ToByte('O');
     }
     #endregion
     #region Fields
    //Command Index Constant
    private const int IdxID = 2;
    private const int IdxSize = 3;
    private const int IdxSequence = 4:
    private const int IdxPayload = 5;
    private RH DataPacket latestInstruction;
     #endregion
     #region Methods
    public override void ComposeCommand(object Instruction)
     {
```

```
latestInstruction = (RH DataPacket)Instruction;
```

```
int idxCRC = IdxPayload + latestInstruction.PayloadSize;
```

//Load Command Buffer this[IdxID] = latestInstruction.ID; //Command ID this[IdxSize] = latestInstruction.PayloadSize; //Size this[IdxSequence] = latestInstruction.Sequence; //Sequence for (int i = IdxPayload; i < idxCRC; i++)</pre>

{
 this[i] = latestInstruction.Payloads[i - IdxPayload];
//Payloads
}

this[idxCRC] = latestInstruction.Checksum; //Checksum

```
NewCommand(BufferTX, idxCRC + 1);

}

#endregion

}
```

A-3 Listing Program Receiver

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
```

```
namespace RGUI_PS
```

{

```
public class OCUReceiver
{
    #region Initialization
    public OCUReceiver()
    {
        messageBox = new Queue<byte[]>(10);
        senderBox = new Queue<IPEndPoint>(10);
    }
    #endregion
```

```
#region Field
private Queue<byte[]> messageBox;
private Queue<IPEndPoint> senderBox;
```

```
private Task translateMessages;
private CancellationTokenSource tokenSource;
private CancellationToken cToken;
```

private int packetLoss; private int packetError; private int packetReceived;

```
public delegate void ReportCompleted(object report);
private event ReportCompleted newReport;
```

#endregion

```
#region Methods
    public void deinit()
     ł
       packetLoss = 0;
       packetError = 0;
       packetReceived = 0;
     }
    /// <summary>
    /// Start an asynchronus task that read messages in message
queue and translates them into reports every 100ms.
    /// If the message being translated has a sender, the task also put
the sender in the report
    /// </summary>
    public void BeginReading()
       if (translateMessages == null)
       ł
         tokenSource = new CancellationTokenSource();
         cToken = tokenSource.Token:
         translateMessages = Task.Run(async () =>
         ł
            while (!cToken.IsCancellationRequested)
            ł
              if (messageBox != null && messageBox.Count > 0)
              ł
                 if (senderBox != null && senderBox.Count > 0)
                   SignReport(senderBox.Dequeue());
                 ComposeReport(messageBox.Dequeue());
              2
              await Task.Delay(100);
            }
        }, cToken);
      }
     2
    /// <summary>
```

```
/// Stop the asynchronus task that started by BeginRead Methods
    /// </summary>
    public void EndReading() { tokenSource.Cancel(); }
    /// <summary>
    /// Put a sender IP End Point in the sender queue
    /// </summary>
    /// <param name="Sender"> sender to be store sender queue
</param>
    public void NoteSender(IPEndPoint Sender)
    ł
      senderBox.Enqueue(Sender);
      if (senderBox.Count > 10)
        senderBox.Dequeue();
    }
    /// <summary>
    /// Put a message in the message queue
    /// </summary>
    /// <param name="Sender"> message to store in message queue
</param>
    public void StoreMessage(byte[] Message)
    ł
      messageBox.Enqueue(Message);
      if (messageBox. Count > 10)
         messageBox.Dequeue();
```

}

-----Halaman ini sengaja dikosongkan---

```
A-4 Listing Program Turunan Receiver
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
```

}

```
namespace RGUI_PS
{
    class RhinoReceiver : OCUReceiver
    {
        #region Initialization
        public RhinoReceiver(int PayloadMaxLength)
        {
            currentReport = new RH_DataPacket(PayloadMaxLength);
        }
        #endregion
```

```
protected override void SignReport(IPEndPoint Sender) {
  currentReport.Sender = Sender; }
```

```
protected override void ComposeReport(byte[] Message)
{
    for (int i = 0; i < Message.Count(); i++)
    {
        translateMessage(Message[i]);
    }
}
#endregion
}</pre>
```

-----Halaman ini sengaja dikosongkan---

A-5 Listing Program Communication Center

using System; using System.Collections.Generic; using System.IO.Ports; using System.Linq; using System.Text; using System.Threading.Tasks;

namespace RGUI_PS

ł

public class OCUCommCenter
{

#region Initialization
public OCUCommCenter() { }
#endregion

#region Field
private RobotHub robot;

//Receiver-Related Field
private bool listening;
private OCUReceiver receiver;

//Transmitter-Related Field private bool talking; private OCUTransmitter transmitter; #endregion

#region Methods
/// Pass an instruction to the transmitter for storing
/// </summary>
/// </param name="newInstruction"> instruction to be passed to
the transmitter </param>
private void storeInstruction(object newInstruction) {
transmitter.StoreInstruction(newInstruction); }

/// <summary>
/// Get OCU to transmit a command

```
/// </summary>
/// <param name="Command"> command to send </param>
protected virtual void sendCommand(byte[] Command, int
Length) { }
#endregion
```

#region Update
public virtual void Connect(object state) { }
public virtual void Disconnect(object state) { }

public virtual void StartListening() { }
public virtual void StopListening() { }

} }

60

public virtual void StartTalking() { }
public virtual void StartTalking(TimeSpan TalkingInterval) { }
public virtual void StopTalking() { }
#endregion

A-6 Listing Program Halaman Utama GUI

<UserControl x:Class="RGUI PS.StartMenu"

- xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markupcompatibility/2006"
- xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
 xmlns:local="clr-namespace:RGUI_PS"
 mc:Ignorable="d"
 d:DesignHeight="768" d:DesignWidth="1024">
 <Grid x:Name="StartMenuLayoutRoot">
 <Grid x:Name="StartMenuLayoutRoot">
 <Grid.Background>
 <ImageBrush
 ImageSource="/RGUI_PS;component/Image_Backgrounds/BG_MEN
 U.png"/>
 </Grid.Background>

```
<Button x:Name="Button_Select" Content="PILIH ROBOT"
Margin="0,0,42,168"
Width="382" Height="60"
FontSize="32" FontFamily="Moire" Foreground="White"
Click="Button_Select_Click" Focusable="False"
HorizontalAlignment="Right" VerticalAlignment="Bottom">
<Button_Select_Click" Focusable="False"
HorizontalAlignment="Right" VerticalAlignment="Bottom">
<Button.Background>
<SolidColorBrush Color="#FFDDDDDDD"
Opacity="0.45"/>
</Button.Background>
</Button>
```

```
<Button x:Name="Button_ShutDown" Content="MATIKAN"
HorizontalAlignment="Right" Margin="0,0,42,103"
VerticalAlignment="Bottom"
Width="382" Height="60"
FontSize="32" FontFamily="Moire" Foreground="White"
Click="Button_ShutDown_Click" Focusable="False">
<Button.Background>
```

<SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background>

</Button>

<Image x:Name="image_RhinoIcon" HorizontalAlignment="Right" VerticalAlignment="Bottom" Source="/RGUI_PS;component/Image_Backgrounds/LG_RhinoIcon2. png" Height="100" Margin="0,0,523,201"/> <Image x:Name="image_BhimasenaIcon" HorizontalAlignment="Right" Margin="0,0,567,76" VerticalAlignment="Bottom" Source="/RGUI_PS;component/Image_Backgrounds/LG_Bhimasena. png" Height="100"/> <Rectangle Fill="#FFF4F4F5" HorizontalAlignment="Left" Height="2" Margin="99,458,0,0" Stroke="#FFC4C4C3" VerticalAlignment="Top" Width="400"/> <Rectangle Fill="#FFF4F4F5" HorizontalAlignment="Left" Height="2" Margin="99,570,0,0" Stroke="#FFC4C4C3" VerticalAlignment="Top" Width="400"/>

</Grid> </UserControl>

A-7 Listing Program *Login* Robot GUI

<UserControl x:Class="RGUI PS.LoginRadio"

- xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markupcompatibility/2006"

```
<Button x:Name="Button_IDSeries" Content="XX"
HorizontalAlignment="Right" Margin="0,0,654,28"
VerticalAlignment="Bottom"
FontSize="64" FontFamily="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrush>
<SolidColorBrush Color="#FF707070" Opacity="0"/>
</Button.Background>
</Button.Background>
</Button>
```

```
<Button x:Name="Button_ID0" Content="0"
HorizontalAlignment="Right" Margin="0,0,581,28"
VerticalAlignment="Bottom"
FontSize="64" FontFamily="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrush>
<SolidColorBrush Color="#FF707070" Opacity="0"/>
```

```
</Button.BorderBrush>
<Button.Background>
<SolidColorBrush Color="#FFDDDDDDD" Opacity="0"/>
</Button.Background>
</Button>
```

```
<Button x:Name="Button_ID1" Content="0"
HorizontalAlignment="Right" Margin="0,0,530,28"
VerticalAlignment="Bottom"
FontSize="64" FontFamily="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrush>
<SolidColorBrush Color="#FF707070" Opacity="0"/>
</Button.Background>
</Button.Background>
</Button.Background>
```

```
<Button x:Name="Button_ID2" Content="0"
HorizontalAlignment="Right" Margin="0,0,479,28"
VerticalAlignment="Bottom"
FontSize="64" FontFamily="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrushy="Moire Light"
Foreground="White" Focusable="False">
<Button.BorderBrush>
<SolidColorBrush Color="#FF707070" Opacity="0"/>
</Button.Background>
</Button.Background>
</Button>
```

```
<Button x:Name="Button_Connect" Content="HUBUNGKAN"
HorizontalAlignment="Right" Margin="0,0,240,38"
VerticalAlignment="Bottom"
Width="200" Height="60"
FontSize="24" FontFamily="Moire" Foreground="White"
Focusable="False" Click="Button_Connect_Click">
<Button.Background>
```

<SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> </Button>

<TextBlock x:Name="textBlock_LoginStatus" Margin="212,121,212,608" TextWrapping="Wrap" Text="Masukkan ID Robot" FontSize="32" Foreground="White" FontFamily="Moire" HorizontalAlignment="Center" VerticalAlignment="Center" Width="600"/>

<TextBlock x:Name="textBlock_RobotSeries" HorizontalAlignment="Right" Margin="0,0,654,115" TextWrapping="Wrap" Text="Seri Robot" VerticalAlignment="Bottom" FontSize="16" Foreground="White" FontFamily="Moire"/> <TextBlock x:Name="textBlock_RobotNumber" HorizontalAlignment="Left" Margin="439,634,0,0" TextWrapping="Wrap" Text="Nomor Robot" VerticalAlignment="Top" FontSize="16" Foreground="White" FontFamily="Moire"/>

<Border x:Name="border_ImageRobot" BorderThickness="1" Width="600" Height="400" Margin="212,184" BorderBrush="White"> <Border.Background> <SolidColorBrush Color="White" Opacity="0.1"/> </Border.Background>

<Image x:Name="image_Robot" Margin="0" HorizontalAlignment="Left" VerticalAlignment="Top" Width="600" Height="400"/> </Border> </Grid> </UserControl> -----Halaman ini sengaja dikosongkan---

A-8 Listing Program Operation Page

<UserControl x:Class="RGUI_PS.OperationPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markupcompatibility/2006"

<Image x:Name="image_Navigasi" HorizontalAlignment="Right" Margin="1,0,32,200" VerticalAlignment="Bottom" Width="945" Source="/RGUI_PS;component/Image_Backgrounds/LO_Navigasi.pn g"/>

<Image x:Name="image_DrivetrainIndicator" HorizontalAlignment="Right" Margin="0,0,168,602" VerticalAlignment="Bottom" Width="40" Height="26"/>

<Image x:Name="image_ManipulatorIndicator" HorizontalAlignment="Right" Margin="0,0,148,531" VerticalAlignment="Bottom" Width="40" Height="26"/>

<TextBlock x:Name="textBlock_RobotName" HorizontalAlignment="Right" Margin="0,87,338,0" TextWrapping="Wrap" Text="Robot Name" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="29.333"/> <TextBlock x:Name="textBlock_Drivetrain" HorizontalAlignment="Left" Margin="698,140,0,0" TextWrapping="Wrap" Text="Drivetrain:" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="21.333"/>

<TextBlock x:Name="textBlock_DrivetrainMode" HorizontalAlignment="Left" Margin="725,171,0,0" TextWrapping="Wrap" Text="Drive Mode" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="29.333"/>

<TextBlock x:Name="textBlock_Manipulator" HorizontalAlignment="Left" Margin="698,211,0,0" TextWrapping="Wrap" Text="Manipulator:" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="21.333"/>

<TextBlock x:Name="textBlock_ManipulatorMode" HorizontalAlignment="Left" Margin="725,242,0,0" TextWrapping="Wrap" Text="Arm Mode" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="29.333"/>

<HelixToolkit:HelixViewport3D ZoomExtentsWhenLoaded="False" x:Name="helixViewPort_Robot" Margin="720,290,50,224" Width="254" Height="254" HorizontalAlignment="Center" VerticalAlignment="Center"> <HelixToolkit:SunLight/> <ModelVisual3D Content="{Binding RobotModel}"/> </HelixToolkit:HelixViewport3D>

<Popup x:Name="PopupMenu" AllowsTransparency="True" Placement="Center" VerticalOffset="192"> <Grid> </Grid> </Popup>

```
<WindowsFormsHost x:Name="VideoTab"
HorizontalAlignment="Center" Height="420"
Margin="49,144,335,204" VerticalAlignment="Center" Width="640"
Background="{x:Null}"/>
</Grid>
</UserControl>
```

A-9 Listing Program Pop Up Main Menu

<UserControl x:Class="RGUI_PS.MainPopup_Rhino"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markupcompatibility/2006"

<TextBlock x:Name="textBlock_MainMenu" HorizontalAlignment="Left" Margin="27,24,0,0" TextWrapping="Wrap" Text="Menu Utama" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="29.333"/>

<Button x:Name="button_PoseSetting" Content="Pengaturan Pose"

HorizontalAlignment="Right" Margin="0,0,325,262" VerticalAlignment="Bottom" Width="160" Height="30" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" Click="button_PoseSetting_Click"> <Button_Background> <SolidColorBrush Color="#FFDDDDDDD" Opacity="0.45"/> </Button.Background> </Button>

```
<Button x:Name="button CameraSetting" Content="Pengaturan
Kamera"
               HorizontalAlignment="Right"
Margin="0,0,325,227" VerticalAlignment="Bottom"
                Width="160" Height="30"
               FontSize="14.667" FontFamily="Moire"
Foreground="White" Focusable="False"
Click="button CameraSetting Click">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <TextBlock x:Name="textBlock SecondaryControl"
HorizontalAlignment="Right" Margin="0,64,47,0"
TextWrapping="Wrap" Text="Kendali Sekunder"
VerticalAlignment="Top" Foreground="White" FontFamily="Moire"
FontSize="18.667"/>
    <Button x:Name="button Drivetrain" Content="Mode"
Mengemudi"
       HorizontalAlignment="Right" Margin="0,92,50,0"
VerticalAlignment="Top"
        Width="160" Height="30"
       FontSize="16" FontFamily="Moire" Foreground="White"
Focusable="False" Click="button Drive Click">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <Button x:Name="button Manipulator" Content="Mode"
Manipulator"
       HorizontalAlignment="Right" Margin="0,127,50,0"
```

```
VerticalAlignment="Top"
```

```
</Button>
<Button x:Name="button_IlluminationSetting"
Content="Pengaturan Lampu"
HorizontalAlignment="Right" Margin="0,0,50,192"
VerticalAlignment="Bottom"
Width="160" Height="30"
FontSize="16" FontFamily="Moire" Foreground="White"
Focusable="False" Click="button_IlluminationSetting_Click">
<Button.Bize="16" FontFamily="Moire" Foreground="White"
Focusable="False" Click="button_IlluminationSetting_Click">
<Button.Background="White"
Focusable="False" Click="button_IlluminationSetting_Click">
<Button.Background="White"
Focusable="False" Click="button_IlluminationSetting_Click">
<Button.Background=
<SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
</Button.Background>
```

</Button>

</Grid> </UserControl> -----Halaman ini sengaja dikosongkan---

A-10Listing Program *Pop Up* Pengaturan Pose

<UserControl x:Class="RGUI PS.PosePopup Rhino"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markupcompatibility/2006"

<TextBlock x:Name="textBlock_Pose" HorizontalAlignment="Right" Margin="0,0,816,507" TextWrapping="Wrap" Text="Pose" VerticalAlignment="Bottom" Foreground="White" FontFamily="Moire" FontSize="29.333"/>

<Button x:Name="button_Drive" HorizontalAlignment="Right" Margin="0,0,729,402" VerticalAlignment="Bottom" Width="160" Height="100" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button_Drive_Click"> <Button_Drive_Click"> <Button_Drive_Click"> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background>

```
<TextBlock x:Name="textBlock_Drive"
HorizontalAlignment="Left" TextWrapping="Wrap"
Text="Mengemudi" VerticalAlignment="Top" Foreground="White"
FontFamily="Moire" FontSize="16"/>
</Button>
<Image x:Name="image_Drive" HorizontalAlignment="Right"
Margin="0,0,757,402" VerticalAlignment="Bottom" Width="104"
Height="60" RenderTransformOrigin="1.5,1.5"
Source="/RGUI_PS;component/Icon_Pose/POSE_Drive.png">
</Image>
```

```
<Button x:Name="button DriveBackward"
               HorizontalAlignment="Right"
Margin="0,0,564,402" VerticalAlignment="Bottom"
               Width="160" Height="100"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Left" VerticalContentAlignment="Top"
Click="button DriveBackward Click">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
      <TextBlock x:Name="textBlock DriveBackward"
HorizontalAlignment="Left" TextWrapping="Wrap"
VerticalAlignment="Top" Foreground="White" FontFamily="Moire"
FontSize="16" Height="45"><Run
Text="Mengemudi"/><LineBreak/><Run
Text="Mundur"/></TextBlock>
    </Button>
    <Image x:Name="image DriveBackward"
HorizontalAlignment="Right" Margin="0,0,564,402"
VerticalAlignment="Bottom" Width="132" Height="91"
Source="/RGUI PS;component/Icon Pose/POSE DriveBackward.pn
g"/>
```

```
<Button x:Name="button DriveUp"
```

HorizontalAlignment="Right" Margin="0,0,399,402" VerticalAlignment="Bottom" Width="160" Height="100" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button DriveUp Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> <TextBlock x:Name="textBlock DriveUp" HorizontalAlignment="Left" TextWrapping="Wrap" Text="Mengemudi Menaiki Tangga" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/> </Button><*Image x:Name="image DriveUp"* HorizontalAlignment="Right" Margin="0,0,417,402" VerticalAlignment="Bottom" Width="104" Height="69" Source="/RGUI PS;component/Icon Pose/POSE DriveUp.png"/> <Button x:Name="button LoadObject" HorizontalAlignment="Right" Margin="0,0,234,402" VerticalAlignment="Bottom" Width="160" Height="100" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button LoadObject Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> <TextBlock x:Name="textBlock LoadObject" HorizontalAlignment="Left" TextWrapping="Wrap" Text="Memuat Benda ke Penggenggam" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/> </Button>

<Button x:Name="button PickObject" HorizontalAlignment="Right" Margin="0,0,69,402" VerticalAlignment="Bottom" Width="160" Height="100" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button PickObject Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> <TextBlock x:Name="textBlock PickObject" *HorizontalAlignment="Left" TextWrapping="Wrap"* Text="Mengangkat atau Menaruh Benda" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/> </Button><Image x:Name="image PickObject" HorizontalAlignment="Right" Margin="0,0,97,402" VerticalAlignment="Bottom" Width="104" Height="60" Source="/RGUI PS;component/Icon Pose/POSE PickObject.png"/> *<Button x:Name="button LookHigh"* HorizontalAlignment="Right" *Margin="0,0,729,297" VerticalAlignment="Bottom"* Width="160" Height="100" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button LookHigh Click"> *<Button.Background>* <SolidColorBrush Color="#FFDDDDDD" *Opacity="0.45"/>* </Button.Background> <TextBlock x:Name="textBlock LookHigh" HorizontalAlignment="Left" TextWrapping="Wrap" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16" Height="45"><Run

Text="Melihat"/><LineBreak/><Run Text="dari Atas"/></TextBlock> </Button> <Image x:Name="image LookHigh" HorizontalAlignment="Right" Margin="0,0,738,297" VerticalAlignment="Bottom" Width="123" Height="100" Source="/RGUI PS;component/Icon Pose/POSE LookHigh.png" RenderTransformOrigin="0.388,0.5"/> <Button x:Name="button LookMiddle" HorizontalAlignment="Right" Margin="0,0,564,297" VerticalAlignment="Bottom" Width="160" Height="100" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button LookMiddle Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> <TextBlock x:Name="textBlock LookMiddle" HorizontalAlignment="Left" TextWrapping="Wrap" Text="Melihat dari Tengah" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/> </Button><Image x:Name="image LookMiddle" HorizontalAlignment="Right" Margin="0,0,592,297" VerticalAlignment="Bottom" Width="104" Height="60"/> <*Button x:Name="button LookLow"*

<SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> <TextBlock x:Name="textBlock LookLow" HorizontalAlignment="Left" TextWrapping="Wrap" Text="Melihat dari Bawah" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/> </Button><Image x:Name="image LookLow" HorizontalAlignment="Right" Margin="0,0,427,297" VerticalAlignment="Bottom" Width="121" Height="80" Source="/RGUI PS;component/Icon Pose/POSE LookLow.png"/> <Button x:Name="button LookUnder" HorizontalAlignment="Right" Margin="0,0,234,297" VerticalAlignment="Bottom" Width="160" Height="100" FontSize="16" FontFamily="Moire" *Foreground="White" Focusable="False"* HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button LookUnder Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> <TextBlock x:Name="textBlock LookUnder" HorizontalAlignment="Left" TextWrapping="Wrap" Text="Melihat Bagian Bawah" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/> </Button><Image x:Name="image LookUnder" HorizontalAlignment="Right" Margin="0,0,250,297" VerticalAlignment="Bottom" Width="132" Height="80" Source="/RGUI PS;component/Icon Pose/POSE LookUnder.png"/> <Button x:Name="button LookDown" HorizontalAlignment="Right" Margin="0,0,69,297"

VerticalAlignment="Bottom"

Width="160" Height="100"

FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button LookDown Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" Opacity = "0.45" /></Button.Background> <TextBlock x:Name="textBlock LookDown" *HorizontalAlignment="Left" TextWrapping="Wrap" VerticalAlignment="Top" Foreground="White" FontFamily="Moire"* FontSize="16" Height="49"><Run Text="Melihat"/><LineBreak/><Run Text="Ke Bawah"/></TextBlock> </Button><Image x:Name="image LookDown" *HorizontalAlignment="Right" Margin="0,0,97,297"* VerticalAlignment="Bottom" Width="104" Height="100" Source="/RGUI PS;component/Icon Pose/POSE LookDown.png"/> <*Button x:Name="button LookClose"* HorizontalAlignment="Right" Margin="0,0,729,192" VerticalAlignment="Bottom" Width="160" Height="100" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button LookClose Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" *Opacity="0.45"/>* </Button.Background> <TextBlock x:Name="textBlock LookClose" *HorizontalAlignment="Left" TextWrapping="Wrap" Text="Melihat* dari Dekat" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/> </Button><Image x:Name="image LookClose" HorizontalAlignment="Right" Margin="0,0,738,192"

VerticalAlignment="Bottom" Width="138" Height="79" Source="/RGUI PS;component/Icon Pose/POSE LookClose.png"/>

<*Button x:Name="button LookSpooler"* HorizontalAlignment="Right" *Margin="0.0.564,192" VerticalAlignment="Bottom"* Width="160" Height="100" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button LookSpooler Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" *Opacity="0.45"/>* </Button.Background> <TextBlock x:Name="textBlock LookSpooler" HorizontalAlignment="Left" TextWrapping="Wrap" Text="Melihat *Alat Penggulung" VerticalAlignment="Top" Foreground="White"* FontFamily="Moire" FontSize="16"/> </Button><Image x:Name="image LookSpooler" HorizontalAlignment="Right" Margin="0,0,592,192" VerticalAlignment="Bottom" Width="104" Height="60"/> <Button x:Name="button StowStorage" HorizontalAlignment="Right" Margin="0,0,399,192" VerticalAlignment="Bottom" Width="160" Height="100" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button StowStorage Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" *Opacity="0.45"/>* </Button.Background> <TextBlock x:Name="textBlock StowStorage" HorizontalAlignment="Left" TextWrapping="Wrap" Text="Lipat

untuk penyimpanan" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/> </Button> <Image x:Name="image_StowStorage" HorizontalAlignment="Right" Margin="0,0,427,192" VerticalAlignment="Bottom" Width="104" Height="60" Source="/RGUI_PS;component/Icon_Pose/POSE_StowStorage.png"/ >

<Button x:Name="button StoreBox" HorizontalAlignment="Right" Margin="0,0,234,192" VerticalAlignment="Bottom" Width="160" Height="100" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button StoreBox Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> <TextBlock x:Name="textBlock Storebox" HorizontalAlignment="Left" TextWrapping="Wrap" Text="Simpan Dalam Kotak" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/> </Button><Image x:Name="image StoreBox" HorizontalAlignment="Right" Margin="0,0,262,192" VerticalAlignment="Bottom" Width="104" Height="60" Source="/RGUI PS;component/Icon Pose/POSE StoreBox.png"/>

<Button x:Name="button_Custom" HorizontalAlignment="Right" Margin="0,0,69,192" VerticalAlignment="Bottom" Width="160" Height="100" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Left" VerticalContentAlignment="Top" Click="button_Custom_Click">

```
<Button.Background>

<SolidColorBrush Color="#FFDDDDDD"

Opacity="0.45"/>

</Button.Background>

<TextBlock x:Name="textBlock_Custom"

HorizontalAlignment="Left" TextWrapping="Wrap" Text="Atur Pose

Preset..." VerticalAlignment="Top" Foreground="White"

FontFamily="Moire" FontSize="16"/>

</Button>

<Image x:Name="image_Custom" HorizontalAlignment="Right"

Margin="0,0,97,192" VerticalAlignment="Bottom" Width="104"

Height="60"/>
```

</Grid> </UserControl>

A-11 Listing Program Pop Up Preset Pose <UserControl x:Class="RGUI PS.PresetPosePopup"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markupcompatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:helix="http://helix-toolkit.org/wpf" xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit" xmlns:HelixToolkit="clrnamespace:HelixToolkit.Wpf;assembly=HelixToolkit.Wpf" xmlns:local="clr-namespace:RGUI PS" mc:Ignorable="d" d:DesignHeight="768" d:DesignWidth="1024"> <Grid x:Name="PresetPosePopupLayoutRoot" Width="960" Height="576" HorizontalAlignment="Left" VerticalAlignment="Bottom"> <Grid.Background> <ImageBrush ImageSource="/RGUI PS;component/Image Backgrounds/BG Popu pMenu.png"/> </Grid.Background>

<TextBlock x:Name="textBlock_PresetPose" HorizontalAlignment="Right" Margin="0,0,533,504" TextWrapping="Wrap" Text="Pengaturan Pose Preset" VerticalAlignment="Bottom" Foreground="White" FontFamily="Moire" FontSize="29.333" Width="376"/>

<HelixToolkit:HelixViewport3D ZoomExtentsWhenLoaded="False" x:Name="helixViewPort_CurrentPose" Margin="87,159,619,163" Width="254" Height="254" HorizontalAlignment="Center" VerticalAlignment="Center"> <HelixToolkit:SunLight/> <ModelVisual3D Content="{Binding CurrentPose}"/> </HelixToolkit:HelixViewport3D>

<*HelixToolkit:HelixViewport3D* ZoomExtentsWhenLoaded="False" x:Name="helixViewPort NewPose" Margin="427,159,279,163" Width="254" Height="254" HorizontalAlignment="Center" VerticalAlignment="Center"> <HelixToolkit:SunLight/> <ModelVisual3D Content="{Binding NewPose}"/> </HelixToolkit:HelixViewport3D> <Button x:Name="button Preset1" Content="PRESET 1" HorizontalAlignment="Right" *Margin="0,0,749,110" VerticalAlignment="Bottom"* Width="160" Height="30" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Click="button Preset1 Click" GotFocus="button Preset1 GotFocus"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" *Opacity="0.45"/>* </Button.Background> </Button><Button x:Name="button Preset2" Content="PRESET 2" HorizontalAlignment="Right" Margin="0,0,749,75" VerticalAlignment="Bottom" *Width="160" Height="30"* FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" *HorizontalContentAlignment="Center"* VerticalContentAlignment="Center" Click="button Preset2 Click" GotFocus="button Preset2 GotFocus"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" Opacity = "0.45" /></Button.Background> </Button>
<Button x:Name="button_Preset3" Content="PRESET 3" HorizontalAlignment="Right" Margin="0,0,749,40" VerticalAlignment="Bottom" Width="160" Height="30" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Click="button_Preset3_Click" GotFocus="button_Preset3_GotFocus"> <Button_Preset3_GotFocus"> <Button_Preset3_GotFocus"> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> </Button>

<Button x:Name="button_Preset4" Content="PRESET 4" HorizontalAlignment="Right" Margin="0,0,584,110" VerticalAlignment="Bottom" Width="160" Height="30" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Click="button_Preset4_Click" GotFocus="button_Preset4_GotFocus"> <Button_Preset4_GotFocus"> <Button_Preset4_GotFocus"> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> </Button>

```
<Button x:Name="button_Preset5" Content="PRESET 5"
HorizontalAlignment="Right" Margin="0,0,584,75"
VerticalAlignment="Bottom"
Width="160" Height="30"
FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
```

```
VerticalContentAlignment="Center" Click="button Preset5 Click"
GotFocus="button Preset5 GotFocus">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <Button x:Name="button Preset6" Content="PRESET 6"
               HorizontalAlignment="Right" Margin="0,0,584,40"
VerticalAlignment="Bottom"
               Width="160" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Click="button Preset6 Click"
GotFocus="button Preset6 GotFocus">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <Button x:Name="button SaveCurrent"
               HorizontalAlignment="Right" Margin="0,0,279,77"
VerticalAlignment="Bottom"
               Width="160" Height="60"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
Click="button SaveCurrent Click">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
       </Button.Background>
      <TextBlock x:Name="textBlock SaveCurrent"
HorizontalAlignment="Left" TextWrapping="Wrap" Text="Simpan
```

posisi saat ini" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="18.667"/> </Button>

<Button x:Name="button SaveManual" HorizontalAlignment="Right" Margin="0,0,50,77" VerticalAlignment="Bottom" *Width="160" Height="60"* FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" Click="button SaveManual Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> <TextBlock x:Name="textBlock SaveManual" *HorizontalAlignment="Left" TextWrapping="Wrap" Text="Atur* manual..." VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="18.667"/> </Button><Button x:Name="button Flipper" Content="Flipper" *HorizontalAlignment="Right" Margin="0,0,50,387"*

VerticalAlignment="Bottom" Width="160" Height="30" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" GotFocus="button_Flipper_GotFocus"> <Button.Background> <SolidColorBrush Color="#FFDDDDDDD" Opacity="0.45"/> </Button.Background> </Button>

```
<Button x:Name="button Turntable" Content="Turntable"
               HorizontalAlignment="Right" Margin="0,0,50,352"
VerticalAlignment="Bottom"
               Width="160" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
GotFocus="button Turntable GotFocus">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <Button x:Name="button Shoulder" Content="Shoulder"
               HorizontalAlignment="Right" Margin="0,0,50,317"
VerticalAlignment="Bottom"
               Width="160" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
GotFocus="button Shoulder GotFocus">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity = "0.45" />
      </Button.Background>
    </Button>
    <Button x:Name="button Elbow" Content="Elbow"
               HorizontalAlignment="Right" Margin="0,0,50,282"
VerticalAlignment="Bottom"
               Width="160" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
GotFocus="button Elbow GotFocus">
```

```
<Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <Button x:Name="button Wrist" Content="Wrist"
               HorizontalAlignment="Right" Margin="0,0,50,247"
VerticalAlignment="Bottom"
               Width="160" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
GotFocus="button Wrist GotFocus">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <Button x:Name="button CameraTilt" Content="CameraTilt"
               HorizontalAlignment="Right" Margin="0,0,50,212"
VerticalAlignment="Bottom"
               Width="160" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
GotFocus="button CameraTilt GotFocus">
```

```
<Button.Background>
```

<SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/>

</Button.Background> </Button>

```
<Button x:Name="button_CameraPan" Content="Camera Pan"
HorizontalAlignment="Right" Margin="0,0,50,177"
VerticalAlignment="Bottom"
```

Width="160" Height="30" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" GotFocus="button_CameraPan_GotFocus"> <Button_CameraPan_GotFocus"> <Button.Background> <SolidColorBrush Color="#FFDDDDDDD" Opacity="0.45"/> </Button.Background> </Button>

</Grid> </UserControl>

A-12Listing Program Pop Up Pengaturan Kamera

<UserControl x:Class="RGUI PS.CameraPopup Rhino"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markupcompatibility/2006"

<TextBlock x:Name="textBlock_Camera" HorizontalAlignment="Left" Margin="54,36,0,0" Text="Kamera" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="29.333"/>

<TextBlock x:Name="textBlock_Selection" HorizontalAlignment="Left" Margin="40,96,0,0" Text="Pilihan" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="21.333"/>

<TextBlock x:Name="textBlock_Layout" HorizontalAlignment="Left" Margin="258,96,0,0" Text="Susunan" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="21.333"/>

<TextBlock x:Name="textBlock_SavePreset" HorizontalAlignment="Left" Margin="592,96,0,0" Text="Simpan" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="21.333"/> <TextBlock x:Name="textBlock_MainCam" HorizontalAlignment="Left" Margin="40,131,0,0" Text="Utama :" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/>

<TextBlock x:Name="textBlock_GripperCam" HorizontalAlignment="Left" Margin="40,166,0,0" Text="Penggenggam :" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/>

<TextBlock x:Name="textBlock_FrontCam" HorizontalAlignment="Left" Margin="40,201,0,0" Text="Depan :" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/>

<TextBlock x:Name="textBlock_RearCam" HorizontalAlignment="Left" Margin="40,236,0,0" Text="Belakang :" VerticalAlignment="Top" Foreground="White" FontFamily="Moire" FontSize="16"/>

```
<Button x:Name="button_MainCam" Content="Status"
HorizontalAlignment="Left" Margin="168,126,0,0"
VerticalAlignment="Top"
Width="80" Height="30"
FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
Click="button_MainCam_Click">
<Button.Background>
<SolidColorBrush Color="#FFDDDDDDD"
Opacity="0.45"/>
</Button.Background>
</Button>
```

```
<Button x:Name="button_GripperCam" Content="Status"
HorizontalAlignment="Left" Margin="168,161,0,0"
VerticalAlignment="Top"
Width="80" Height="30"
FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
```

```
VerticalContentAlignment="Center"
Click="button GripperCam Click">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <Button x:Name="button FrontCam" Content="Status"
               HorizontalAlignment="Left" Margin="168,196,0,0"
VerticalAlignment="Top"
               Width="80" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
Click="button FrontCam Click">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <Button x:Name="button RearCam" Content="Status"
               HorizontalAlignment="Left" Margin="168,231,0,0"
VerticalAlignment="Top"
               Width="80" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
Click="button RearCam Click">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
```

<Button x:Name="button PresetCam1" Content="Preset 1"

```
HorizontalAlignment="Left" Margin="592,127,0,0"
VerticalAlignment="Top"
                Width="120" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
Click="button PresetCam1 Click">
      <Button.Background>
        <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <Button x:Name="button PresetCam2" Content="Preset 2"
               HorizontalAlignment="Left" Margin="592,162,0,0"
VerticalAlignment="Top"
                Width="120" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
Click="button PresetCam2 Click">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity = "0.45" />
      </Button.Background>
    </Button>
    <Button x:Name="button PresetCam3" Content="Preset 3"
               HorizontalAlignment="Left" Margin="592,197,0,0"
VerticalAlignment="Top"
               Width="120" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
Click="button PresetCam3 Click">
      <Button.Background>
```

<SolidColorBrush Color="#FFDDDDDD" *Opacity="0.45"/>* </Button.Background> </Button><Border x:Name="border Screen1" BorderBrush="White" BorderThickness="1" HorizontalAlignment="Left" Height="105" Margin="258,127,0,0" VerticalAlignment="Top" Width="160"> <TextBlock x:Name="textBlock Screen1" HorizontalAlignment="Center" Margin="0" Text="Layar1" *VerticalAlignment="Center" Foreground="White"* FontFamily="Moire" FontSize="18.667"/> </Border> *<Border x:Name="border Screen2" BorderBrush="White"* BorderThickness="1" HorizontalAlignment="Left" Height="105" Margin="417,127,0,0" VerticalAlignment="Top" Width="160"> <TextBlock x:Name="textBlock Screen2" *HorizontalAlignment="Center" Margin="0" Text="Layar2" VerticalAlignment="Center" Foreground="White"* FontFamily="Moire" FontSize="18.667"/> </Border> <Border x:Name="border Screen3" BorderBrush="White" BorderThickness="1" HorizontalAlignment="Left" Height="105" Margin="417,231,0,0" VerticalAlignment="Top" Width="160"> <TextBlock x:Name="textBlock Screen3" HorizontalAlignment="Center" Margin="0" Text="Lavar3" VerticalAlignment="Center" Foreground="White" FontFamily="Moire" FontSize="18.667"/> </Border> <Border x:Name="border Screen4" BorderBrush="White" BorderThickness="1" HorizontalAlignment="Left" Height="105" Margin="258,231,0,0" VerticalAlignment="Top" Width="160"> <TextBlock x:Name="textBlock Screen4" *HorizontalAlignment="Center" Margin="0" Text="Lavar4"* VerticalAlignment="Center" Foreground="White" FontFamily="Moire" FontSize="18.667"/> </Border>

</Grid> </UserControl> -----Halaman ini sengaja dikosongkan---

A-13 Listing Program Pop Up Penerangan

<UserControl x:Class="RGUI PS.IlluminationPopup Rhino"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats.org/markupcompatibility/2006"

<TextBlock x:Name="textBlock_Illumination" HorizontalAlignment="Right" Margin="0,0,295,316" TextWrapping="Wrap" Text="Penerangan" VerticalAlignment="Bottom" Foreground="White" FontFamily="Moire" FontSize="29.333"/> <TextBlock x:Name="textBlock Status"

HorizontalAlignment="Right" Margin="0,0,400,264" TextWrapping="Wrap" Text="Status : "VerticalAlignment="Bottom" Foreground="White" FontFamily="Moire" FontSize="18.667"/> <TextBlock x:Name="textBlock MainLamp"

HorizontalAlignment="Right" Margin="0,0,399,228" TextWrapping="Wrap" Text="Utama : "VerticalAlignment="Bottom" Foreground="White" FontFamily="Moire" FontSize="18.667"/> <TextBlock x:Name="textBlock GripperLamp"

<TextBlock x:Name="textBlock FrontLamp" HorizontalAlignment="Right" Margin="0,0,399,159" *TextWrapping="Wrap" Text="Depan : " VerticalAlignment="Bottom"* Foreground="White" FontFamily="Moire" FontSize="18.667"/> <TextBlock x:Name="textBlock Intensitas" HorizontalAlignment="Right" Margin="0,0,41,264" TextWrapping="Wrap" Text="Intensitas : " VerticalAlignment="Bottom" Foreground="White" FontFamily="Moire" FontSize="18.667"/> <Button x:Name="button Status" Content="Dinonaktifkan" HorizontalAlignment="Right" *Margin="0,0,181,260" VerticalAlignment="Bottom"* Width="140" Height="30" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" Click="button Status Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> </Button><Button x:Name="button MainLamp" Content="Padam" HorizontalAlignment="Right" *Margin="0,0,181,225" VerticalAlignment="Bottom"* Width = "140" Height = "30" FontSize="16" FontFamily="Moire" Foreground="White" Focusable="False" Click="button MainLamp Click"> <Button.Background> <SolidColorBrush Color="#FFDDDDDD" Opacity="0.45"/> </Button.Background> </Button>*<Button x:Name="button GripperLamp" Content="Padam"* HorizontalAlignment="Right" *Margin="0,0,181,190" VerticalAlignment="Bottom"* Width="140" Height="30"

```
FontSize="16" FontFamily="Moire" Foreground="White"
Focusable="False" Click="button GripperLamp Click">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <Button x:Name="button FrontLamp" Content="Padam"
               HorizontalAlignment="Right"
Margin="0,0,181,155" VerticalAlignment="Bottom"
                Width="140" Height="30"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False"
Click="button FrontLamp Click">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.45"/>
      </Button.Background>
    </Button>
    <Button x:Name="button LampIntensity"
               HorizontalAlignment="Right" Margin="0,0,84,155"
VerticalAlignment="Bottom"
               Width="18" Height="104"
               FontSize="16" FontFamily="Moire"
Foreground="White" Focusable="False" BorderBrush="{x:Null}"
Click="button LampIntensity Click">
      <Button.Background>
         <SolidColorBrush Color="#FFDDDDDD"
Opacity="0.0"/>
      </Button.Background>
      <Slider x:Name="slider LampIntensity"
HorizontalAlignment="Center" VerticalAlignment="Center"
Orientation="Vertical" Maximum="255" Height="100"
Width="20"/>
    </Button>
```

</Grid> </UserControl> -----Halaman ini sengaja dikosongkan---

LAMPIRAN B

Tipe	Hea	ıder	System Id	Size	Sequence	Tipe Payload	Byte	CRC
						ID2	1	
Search	R	0	0XFF	3	1	ID1	1	XOR
						ID0	1	
						Manual (L)	1	
						Manual (U)	1	
						Accessories	1	
							2	
Operate Mode	R	0	1, 2, 3, 4	15	1	Analog Left	2	XOR
							2	
							2	
						Analog Right	2	
							2	
						Wrist Position	2	
			5	6	1	Main Camera Tilt	2	XOR
						Main Camera Pan	2	
						Turntable Position	2	
Pose	R	0	5	6	2	Shoulder Position	2	XOR
						Elbow Position	2	
						Flipper Position	2	
			5	6	3	Gripper Grasp Position	2	XOR
						Gripper Rotation Position	2	
Illumination	R	0	6	2	1	Illuminator Setting	1	XOR
Setting K O o 2	1	1	Value	1				

B-1 Protokol Mengirim Data Rhino Robot

Tipe	Hea	ıder	System Id	Size	Sequence	Tipe Payload	Byte	CRC
Emergency Setting	R	0	0X52	1	1	Idle Setting	1	XOR

Keterangan	Kegunaan
Header	Penanda awal data
Size	Jumlah data payload yang dikirim dalam byte
Packet Sequence	Mendeteksi paket data yang hilang, diketahui jika sekuens yang diterima saat ini tidak urut dengan nomor sekuens yang sebelumnya
CRC (Checksum)	XOR setiap byte di payload
Payload	Data yang dikirim, untuk tipe data lebih dari satu byte, LSB dikirim dulu

ID CO	Data Type	Desired Output	BYTE	IDX	Total Byte	
1 : BATTERY						
leftBattery_voltage	Left battery voltage	INT16_T	float	2	0	
leftBattery_current	Left battery current	INT16_T	float	2	2	
leftBattery_capacity	Left battery capacity	INT16_T	float	2	4	
	Left battery temperature	INT16_T	float	2	6	
	Right battery voltage	INT16_T	float	2	8	16
	Right battery current	INT16_T	float	2	10	
	Right battery capacity	INT16_T	float	2	12	
	Right battery temperature	INT16_T	float	2	14	
2 : FEEDBACK MOT	TOR			-		-
leftMotor_current	Left motor current	INT16_T	float	2	0	
	Left motor RPM	INT16_T	float	2	2	
	Right motor current	INT16_T	float	2	4	12
	Right motor RPM	INT16_T	float	2	6	12
	Flipper current	INT16_T	float	2	8	
	Flipper RPM/Pos	INT16_T	float	2	10	
3 : IMU						
roll	Roll	INT16_T	float	2	0	
	Pitch	INT16_T	float	2	2	
	Yaw	INT16_T	float	2	4	
AccelX	X Accel	INT16_T	float	2	6	
	Y Accel	INT16_T	float	2	8	20
	Z Accel	INT16_T	float	2	10	20
Gyrox	X Gyro	INT16_T	float	2	12	
	Y Gyro	INT16_T	float	2	14	
	Z Gyro	INT16_T	float	2	16	
GyroT	Temperature Gyro	INT16_T	float	2	18	
4 : POSITION ARM				-		-
joint_turntable_pos	Joint turntable	INT16_T	float	2	0	
	Joint shoulder	INT16_T	float	2	2	16
	Joint elbow	INT16_T	float	2	4	10
	Joint wrist	INT16_T	float	2	6	

B-2 Protokol Menerima Data Rhino Robot

ID COMPONENT		Data Type	Desired Output	BYTE	IDX	Total Byte
	Joint tilt	INT16_T	float	2	8	
	Joint pan	INT16_T	float	2	10	
	Joint twist	INT16_T	float	2	12	
	Joint grasp	INT16_T	float	2	14	
5 : CURRENT ARM						
joint_turntable_cur	Joint turntable	INT16_T	float	2	0	
	Joint shoulder	INT16_T	float	2	2	
	Joint elbow	INT16_T	float	2	4	
	Joint wrist	INT16_T	float	2	6	16
	Joint pan	INT16_T	float	2	8	10
	Joint tilt	INT16_T	float	2	10	
	Joint twist	INT16_T	float	2	12	
	Joint grasp	INT16_T	float	2	14	
6 : RPM ARM						
joint_turntable_rpm	Joint turntable	INT16_T	float	2	0	
	Joint shoulder	INT16_T	float	2	2	
	Joint elbow	INT16_T	float	2	4	
	Joint wrist	INT16_T	float	2	6	16
	Joint pan	INT16_T	float	2	8	10
	Joint tilt	INT16_T	float	2	10	
	Joint twist	INT16_T	float	2	12	
	Joint grasp	INT16_T	float	2	14	

No	Halaman	Memory Usage (MB)
1	Halaman Utama	227
2	Pilih Komunikasi	287
3	Maintenance	289
4	Layar Operasi	541
5	Pop Up Main Menu	633
6	Pop Up Pose	650
7	Pop Up Pengaturan Kamera	650
8	Pengaturan Kamera Utama	650
9	Pengaturan Kamera 2	650
10	Tukar Kamera	651
11	Tunjukkan Video Full Screen	653
12	Rekam Gambar	655
13	Rekam Layar OCU	655
14	Pop Up Pengaturan Lampu	660
15	Pop Up Pengaturan Audio	660
16	Pop Up Mode Mengemudi	660
17	Pop Up Mode Inspeksi	660
18	Pop Up Move Gripper	660
19	Pop Up Mode Lanjutan	662
20	Pop Up Menu Keluar	665

B-3 Memory Usage GUI Versi Lama

-----Halaman ini sengaja dikosongkan---

No	Halaman	Desain dan Tata Letak Menu
1	Halaman Utama	
2	Pilih Komunikasi	
3	Maintenance	
4	Layar Operasi	SCOLO ADDRESS S NACIONALS Anto Report Anto Report S Solution S Solution S Solution S Solution S Solution S Solution S Solution S Solution S Solution S Solution S S Solution S S Solution S S S S Solution S S S S S S S S S S S S S S S S S S S

B-4 Desain Halaman GUI Versi Lama

No	Halaman	Desain dan Tata Letak Menu
5	Pop Up Main Menu	Main Menu Main Main Main Main Main Main Main Main
6	Pop Up Pose	Poses Morganal Marganal Strangen De Strangen Statu Strangen De Strangen Statu Strangen De Strangen Statu Strangen De Strangen Strangen Strangen Strangen Strangen De Strangen Strangen Strangen Strangen Strangen De Strangen S
7	<i>Pop Up</i> Pengaturan Kamera	Camera Preparate Great Laure L Preparate Kanera . Tular Kanera . Tular Kanera . Ratare Caretaris. Ratare Caretaris. Ratare Caretaris.
8	Pengaturan Kamera Utama	Head Camera Options

No	Halaman	Desain dan Tata Letak Menu
9	Pengaturan Kamera 2	Turret Camera Settings Depin Relande Depin R
10	Tukar Kamera	Turret Camera Settings Kamera Head Turret Gripper
11	Tunjukkan Video Full Screen	
12	Rekam Gambar	Rekam Gambar Mena yan Kenen Jana Meniter dar Karen 2 Usa Drine Ap

No	Halaman	Desain dan Tata Letak Menu
13	Rekam Layar OCU	Nama Viši brire ning Viši brire ning
14	Pop Up Pengaturan Lampu	
15	Pop Up Pengaturan Audio	Audio Sector Sector Notation Synakov Notation Synakov Notation Synakov Notation Synakov
16	Pop Up Mode Mengemudi	Mode Mengemudi Digunskin urukt regereratur Motor dan tipper Werd Her tipper Werd Her tipper O NM O NM O NM O NM O NM O NM O NM O MA O MA

No	Halaman	Desain dan Tata Letak Menu
17	Pop Up Mode Inspeksi	Mode Inspeksi Brigge and the base of
18	Pop Up Move Gripper	Mode Move Gripper
19	<i>Pop Up</i> Mode Lanjutan	Mode Lanjutan Nylakar, Alaszoren Tem Hea Reboot Robot Matikan Robot Control Hotkeys GPS

No	Halaman	Desain dan Tata Letak Menu
20	Pop Up Menu Keluar	Keluar ? Mengapat untuk penyimaase Keluar Program Matkan Laptop USB Drive sit

RIWAYAT HIDUP PENULIS



Nama	: Tata Tanjung Tamara
TTL	: Surabaya, 10 Juli 1997
Agama	: Islam
Jenis Kelaı	nin: Wanita
Alamat	:JL. Gayung Kebonsari No 34 sby
Telp / HP	: 082299911716
E-mail	: Tatatnjngtmr@gmail.com

RIWAYAT PENDIDIKAN

- 2003 2009 : SD Khadijah Surabaya
 2009 2012 : SMP Negeri 32 Surabaya
- 2012 2015 : SMA Negeri 10 Surabaya
- 2015 Sekarang : Departemen Teknik Elektro Otomasi, ITS

PENGALAMAN KEPANITIAN

- Staff Data Center IARC 2016 ٠
- Pengurus Inti Data Center IARC 2017 •
- Organizing Comitte Kestari GERIGI ITS 2017 •