

Perancangan dan Realisasi Alat Demonstrasi PC-based Control untuk Simulasi Keamanan Bangunan menggunakan Embedded PC

Erwani Merry S¹, Indra Maulidin¹, Tara Anggada Putra¹

¹Jurusan Teknik Elektro, Fakultas Teknik, Universitas Krinten Maranatha, Bandung, Jawa Barat.

Informasi Artikel

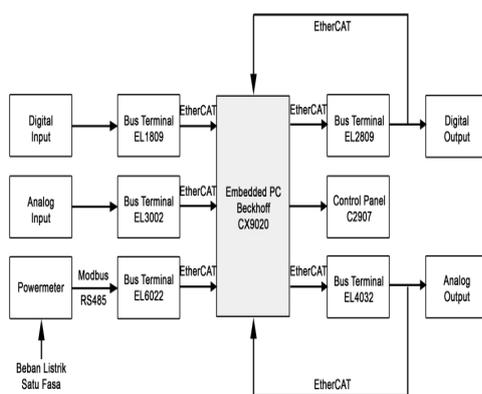
Naskah Diterima : 26 Mei 2018

Direvisi : 22 Juni 2018

Disetujui : 25 Juni 2018

*Korespondensi Penulis :
erwanimerry@gmail.com

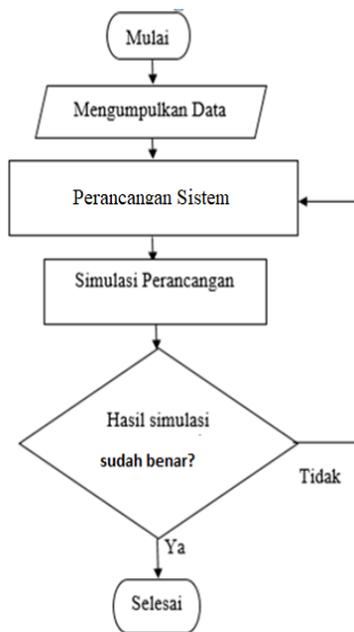
Graphical abstract



Abstract

Demonstration mode is a method used to show a process or work of an object. Demonstration methods can be used to illustrate information to others, predict what will happen, facilitate different types of explanations, and some unfamiliar issues can be asked immediately when a process is demonstrated to be clearly answered. Companies in particular that sell electrical appliances use demo tools to explain the various control systems offered, or how to use the tools they sell. Demo tools are needed to explain the benefits of the control system and the use of the tools. Through automated gate control systems, automatic lights, and alarm functions, the simulation of building security systems becomes the case for the community. The development of industrial technology in the present can not be separated from the role of industrial automation. Industrial automation or numerical control is the use of control systems as well as computers used to control industrial machines and process controls to replace human power carriers. PC based control system is a high-ability control platform. The demonstration tool simulates a security system in the form of a digital output for LED lights whose sequence depends on the given digital input (push button and switch) and smart house applications. Analog input is also implemented, but for analog outputs the voltmeter does not generate the correct voltage value when the potentiometer value changes. This is because there is a loading on the voltmeter so that the output that appears on the voltmeter becomes much smaller than it should be.

Keywords: Simulator, Beckhoff, PC-based control, Embedded PC, TwinCAT 3



Abstrak

Metode demonstrasi adalah metode yang digunakan untuk memperlihatkan sesuatu proses atau cara kerja suatu benda. Metoda demonstrasi dapat dipergunakan untuk memberikan ilustrasi dalam menjelaskan informasi kepada orang lain, memperkirakan apa yang akan terjadi, dan memudahkan berbagai jenis penjelasan. Perusahaan khususnya yang menjual alat listrik menggunakan alat demo untuk menjelaskan berbagai sistem kontrol yang ditawarkan, atau cara penggunaan alat yang dijual. Alat demo dibutuhkan untuk menjelaskan manfaat sistem kontrol dan penggunaan alatnya. Melalui sistem pengontrolan gerbang otomatis, lampu otomatis, dan fungsi alarm, simulasi sistem keamanan bangunan menjadi kasus yang dibutuhkan oleh masyarakat. Perkembangan teknologi industri pada masa kini tidak lepas dari peranan otomasi industri. Otomasi industri atau kontrol numerik merupakan pemanfaatan sistem kontrol seperti halnya komputer yang digunakan untuk mengendalikan mesin-mesin industri dan kontrol proses untuk menggantikan operator tenaga manusia. Sistem kontrol berbasis PC merupakan platform kontrol yang memiliki kemampuan yang tinggi. Alat demonstrasi mensimulasikan sistem keamanan berupa digital output untuk lampu LED yang sequence nya tergantung dengan digital input (push button dan switch) yang diberikan dan aplikasi smart house. Analog input juga diimplementasikan, namun untuk pada analog output yang berupa voltmeter tidak memunculkan nilai tegangan yang seharusnya saat nilai potensiometer berubah. Hal tersebut dikarenakan terjadi pembebanan pada voltmeter sehingga keluaran yang muncul pada voltmeter menjadi jauh lebih kecil dari seharusnya.

Kata kunci: Beckhoff, PC-based control, EmbeddedPC, TwinCAT 3,

© 2017 Penerbit Jurusan Teknik Elektro UNTIRTA Press. All rights reserved

1. PENDAHULUAN

Menurut Syaiful Bahri Djamarah, metode demonstrasi adalah metode yang digunakan untuk memperlihatkan sesuatu proses atau cara kerja suatu benda. Metoda demonstrasi dapat dipergunakan untuk memberikan ilustrasi dalam menjelaskan informasi kepada orang lain, memperkirakan apa yang akan terjadi, memudahkan berbagai jenis penjelasan, dan beberapa persoalan yang belum dimengerti dapat ditanyakan langsung saat suatu proses ditunjukkan sehingga terjawab dengan jelas[1].

Bagi orang awam memahami manfaat sistem kontrol dan penggunaan alatnya tidak mudah dilakukan, selain pengetahuan yang terbatas, perasaan takut akan penggunaan alat listrik masih banyak dialami. Banyak perusahaan khususnya yang menjual alat listrik menggunakan alat demo untuk menjelaskan berbagai sistem kontrol yang ditawarkan, atau cara penggunaan alat yang dijual. Sistem keamanan bangunan merupakan salah satu kasus yang banyak dibutuhkan oleh masyarakat, diantaranya apabila gerbang telah dibuka dalam waktu 10 detik lupa ditutup kembali, maka gerbang akan menutup dengan sendirinya, untuk mencegah hal-hal yang tidak diinginkan pada malam hari terutama ketika tidak ada yang menunggu rumah maka rumah dilengkapi dengan lampu luar yang akan *on/off* dengan sendirinya, dan fungsi alarm menyala bila pintu atau jendela ada yang membuka paksa.

Perkembangan teknologi industri pada masa kini tidak lepas dari peranan otomasi industri. Otomasi industri atau kontrol numerik merupakan pemanfaatan sistem kontrol seperti halnya komputer yang digunakan untuk mengendalikan mesin-mesin industri dan kontrol proses untuk menggantikan *operator* tenaga manusia. Penggunaan otomasi industri dalam bidang industri juga akan membuat proses produksi maupun proses lain di lingkungan industri menjadi lebih efektif dan efisien[2]. *PC-based Control*—sistem kontrol berbasis *PC (Personal Computer)*—merupakan platform kontrol yang memiliki kemampuan yang tinggi. *PC-based Control* sendiri merupakan solusi untuk masalah biaya pada sistem kontrol yang menerapkan *PLC*, dimana pada *PLC* semakin tinggi performa *PLC* tersebut maka semakin tinggi pula biaya yang dikeluarkan. *Beckhoff Automation GmbH* merupakan salah satu pengembang teknologi otomasi di bidang *PC-based Control*, yang telah mengembangkan bisnisnya di berbagai negara, salah satunya Indonesia, dengan nama *Beckhoff Automation Pte Ltd*[3].

Berdasarkan fakta di lapangan, industri-industri di Indonesia masih menggunakan *PLC* pada sistem kontrolnya[4]. *PC-based Control* yang merupakan salah satu solusi untuk menghadapi permasalahan tersebut masih belum mampu menggantikan atau bekerja sama dengan *PLC* untuk mengatasi masalah tersebut. Hal tersebut mungkin muncul karena dapat dikatakan bahwa *PC-based Control* masih cukup baru di bidang industri di Indonesia[5]. Maka pada paper ini akan menjelaskan perancangan dan realisasi dari alat demonstrasi sistem kontrol berbasis *PC* untuk simulasi keamanan bangunan menggunakan *Embedded PC*.

2. METODE PENELITIAN

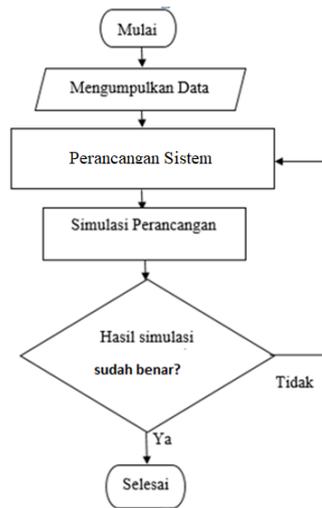
2.1 Metode Penelitian

Berikut ini adalah beberapa tahap yang dilakukan dalam melakukan penelitian ini:

- a) Identifikasi masalah yaitu proses pengenalan dan perumusan masalah sebagai latar belakang hingga tercapainya tujuan dalam penelitian ini.
- b) Studi literatur, yaitu mengumpulkan data-data dari berbagai literatur, buku referensi, dan jurnal sesuai dengan topik penelitian.
- c) Perancangan dan pengujian, yaitu dengan merancang dan merealisasi alat demo.

2.2 Diagram Alir Penelitian

Secara sederhana proses penelitian dijelaskan melalui diagram alir pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

2.3 Perancangan Penelitian

2.3.1 Sistem Keamanan Bangunan

Alat demonstrasi yang digunakan untuk melaksanakan demonstrasi kontrol proses berupa simulasi sistem keamanan bangunan. Sistem keamanan bangunan yang diinginkan adalah sebuah rumah yang dikontrol menggunakan PLC. Aktuator-aktuator yang akan dikontrol adalah sebagai berikut:

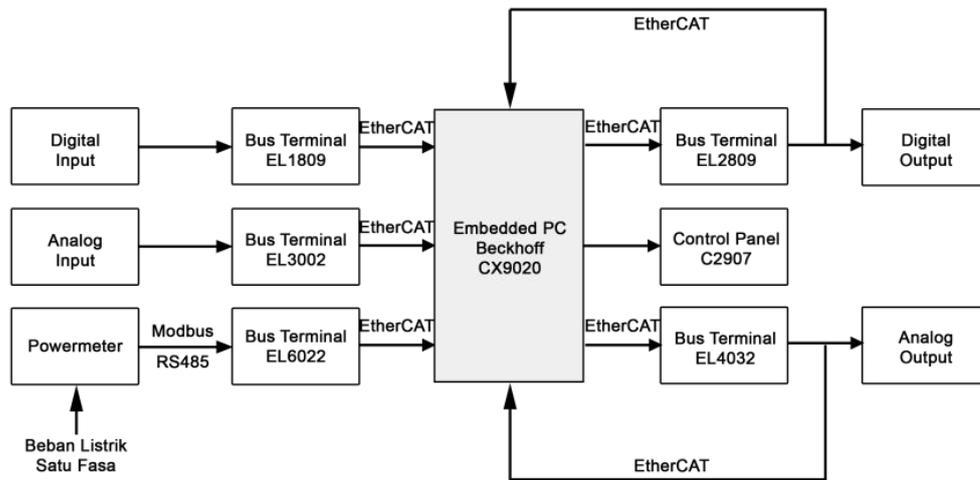
1. Gerbang depan rumah dapat dibuka/ditutup secara otomatis menggunakan *motor* dan *switch*.
2. *Gate Motor* 1 aktif jika *Open Gate* aktif dan akan non aktif jika *Limit Switch Gate* 1 aktif.
3. *Gate Motor* 2 aktif jika *Close Gate* aktif dan akan non aktif jika *Limit Switch Gate* 2 aktif.
4. Apabila *Gate* setelah dibuka dan dalam waktu 10 detik lupa ditutup kembali , maka gate akan otomatis menutup sendiri.
5. Untuk mencegah hal – hal yang tidak diinginkan pada malam hari terutama ketika tidak ada yang menunggu rumah maka rumah dilengkapi oleh lampu luar yang diminta agar dapat *on/off* secara otomatis berdasarkan cahaya matahari. Lampu luar aktif jika Sensor Cahaya aktif (hari sudah gelap) dan lampu luar akan non aktif jika sensor cahaya non aktif (mulai terang).
6. Untuk melengkapi sistem keamanan maka disediakan pula dua buah tipe *alarm* yaitu *alarm* lampu berkedip dengan *on* selama 1 detik dan *off* selama 1 detik. Setiap kali 3 siklus *on-off* alarm tercapai, *alarm sirene* akan *on* selama 3 detik kemudian mati. Siklus berlangsung terus sampai ada tombol yang mematikan *alarm* lampu.
7. *Alarm* Lampu akan aktif jika sensor *proximity (PS) ON* selama 5 detik atau Sensor *Switch (SWI)* aktif. *Alarm* lampu kembali non aktif jika *SWI* kembali non aktif.
8. Sensor *Switch (SWI)* adalah sensor yang dipasang pada pintu atau jendela untuk merespon apabila ada yang masuk secara paksa.
9. *Alarm Sirene* akan aktif jika *Sensor Switch (SWI)* aktif dan *Alarm Sirene* non aktif jika tombol *acknowledge* aktif.

2.3.2 Alat Demonstrasi Sistem Keamanan Bangunan

Alat demonstrasi *PC-based Control* yang dirancang merupakan alat demonstrasi yang dapat digunakan untuk melaksanakan demonstrasi kontrol proses. Untuk memenuhi kebutuhan tersebut, alat ini dilengkapi dengan fungsi *digital input*, *digital output*, *analog input*, dan *analog output*, maupun fungsi lainnya yang dapat digunakan untuk mengoptimalkan demonstrasi kontrol proses[6]. Untuk

mencoba demonstrasi kontrol proses ini, pada alat demonstrasi ini akan dilengkapi dengan program untuk melaksanakan demonstrasi urutan proses dan simulasi sistem keamanan bangunan.

Alat demonstrasi *PC-based Control* yang dirancang memiliki kemampuan untuk melaksanakan fungsi *digital input*, *digital output*, *analog input*, dan *analog output* serta dilengkapi dengan *special function* berupa komunikasi serial *ModbusRTU* untuk membaca besaran listrik 1 fasa[7]. Gambar 2 menunjukkan diagram blok alat demonstrasi *PC-based Control* yang dirancang, yang terdiri dari *embedded PC CX9020* sebagai kontroler utama yang dilengkapi dengan modul-modul *input/output* berupa *bus terminal* untuk masing-masing fungsi *digital input (EL1809)*, *digital output (EL2809)*, *analog input (EL3002)*, *analog output (4032)*, dan *special function (EL6002)*.



Gambar 2. Diagram blok alat demonstrasi *PC-based Control*

Masukan pada *embedded PC Beckhoff CX9020* berupa *digital input*, *analog input*, besaran listrik 1 fasa yang dibaca oleh *powermeter*, serta umpan balik dari *digital output* maupun *analog output*. Fungsi *digital input* merupakan *input* berupa sinyal digital yang hanya memiliki 2 keadaan, yaitu *HIGH* atau *LOW*, sedangkan *analog input* merupakan *input* berupa sinyal analog dengan nilai yang bervariasi sesuai dengan pembacaan *input/sensor* analog tersebut. Keluaran pada *embedded PC* pun berupa nilai *digital* dan *analog* yang nilainya sesuai dengan *input* yang diberikan yang dapat ditampilkan pada sebuah layar (*Control Panel*)[8].

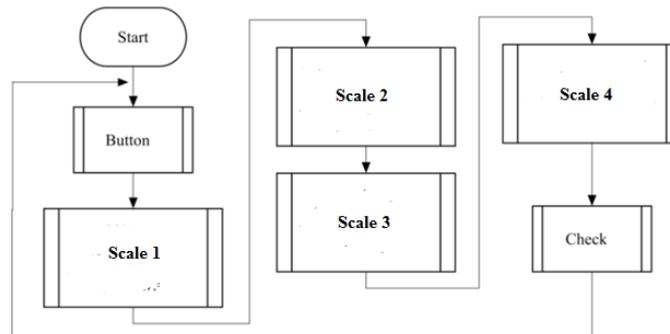
Seluruh masukan maupun keluaran pada sistem dihubungkan pada sebuah *bus terminal* sesuai dengan fungsinya agar dapat membaca nilai *input* maupun mengeluarkan nilai *output*. Masing-masing *bus terminal* tersebut sudah memiliki teknologi *EtherCAT (Ethernet Control Automation Technology)* sehingga pembacaan maupun pengeluaran nilai *input/output* dapat dilaksanakan dengan sangat cepat.

Pada alat yang dirancang, fungsi *digital input* dan *digital output* disediakan sebanyak 16 buah (*channel*) sedangkan untuk *analog input* disediakan sebanyak 4 *channel* dan *analog output* disediakan 2 *channel*, serta masing-masing sebuah *powermeter* untuk membaca besaran listrik 1 fasa dan *Control Panel* untuk fungsi *Human-Machine Interface (HMI)*.

2.3.3 Perancangan Perangkat Lunak

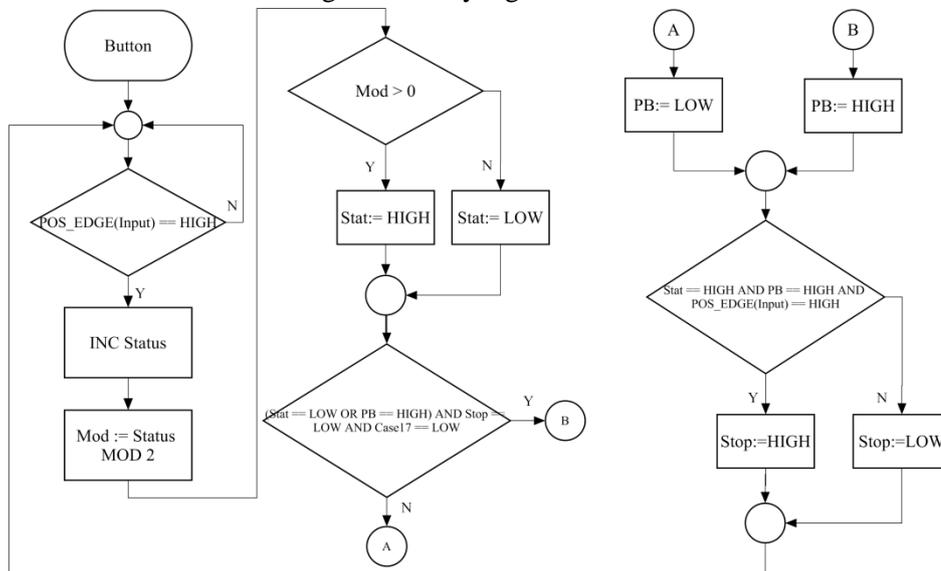
Perangkat lunak yang dirancang untuk melengkapi alat demonstrasi adalah program berupa simulasi keamanan bangunan dan 16 urutan yang berbeda untuk kasus-kasus pada kontrol proses. Program ini dilengkapi dengan *Human-Machine Interface (HMI)* yang menampilkan keadaan *input* dan *output* dari variabel yang digunakan[9].

Program yang dirancang merupakan simulasi keamanan bangunan dan 16 sekuens proses *input/output*. Tiap sekuens yang dibuat diwakili oleh 1 tombol *digital input* dan sekuens lain tidak dapat dijalankan jika sekuens yang sudah dijalankan belum dimatikan dengan menekan tombol yang sama (*interlock*). Untuk menjalankan simulasi keamanan sistem bangunan, *channel 9*, *channel 10*, *channel 11*, dan *channel 12* pada *digital input* harus berada dalam keadaan *HIGH*. Gambar 3 menunjukkan program utama yang dirancang yang terdiri dari fungsi *Button* untuk mendapatkan kondisi nilai *digital input*, penskalaan pada masing-masing *analog input*, dan percabangan proses pada subprogram *Check* berdasarkan kondisi *input*.



Gambar 3. Diagram alir fungsi utama

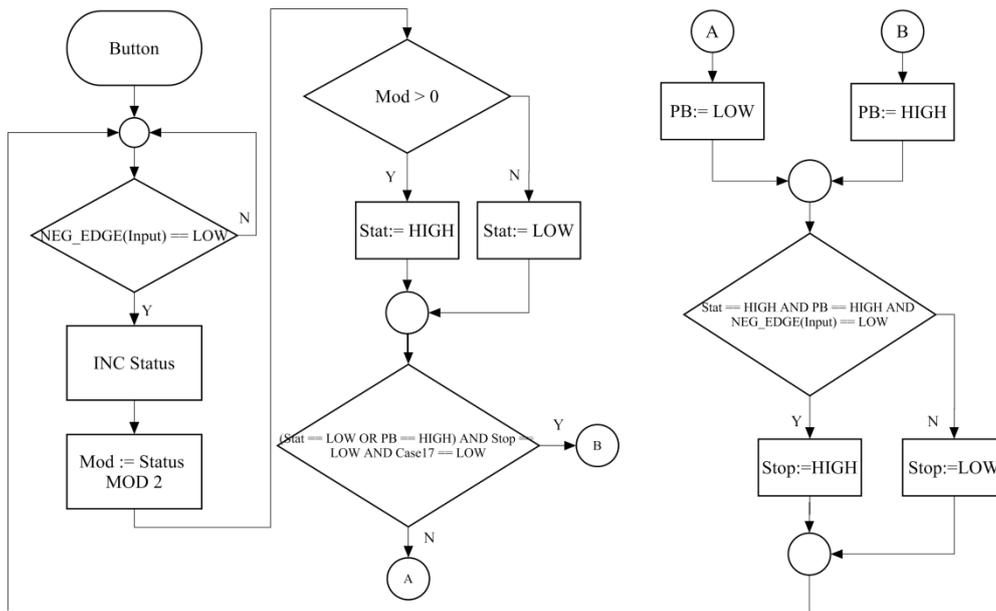
Setelah *CX9020* dinyalakan, *CX9020* akan memanggil fungsi *button* terlebih dahulu untuk mengecek *push button* yang tertekan. Fungsi *Button* tersebut diperlukan untuk membuat fungsi *interlock* karena karakteristik dari *push button* yaitu bersifat *momentary switch*, yang artinya hanya bernilai *HIGH* atau *LOW* pada saat *push button* ditekan saja. Fungsi berikutnya merupakan fungsi penskalaan nilai *analog input* karena nilai yang terbaca pada *bus terminal* merupakan nilai analog 16 bit. Setelah itu, *CX* akan melanjutkan eksekusi dengan memanggil subprogram *Check* yang berfungsi untuk memulai sekuens sesuai dengan tombol yang sudah ditekan.



Gambar 4 Diagram alir fungsi *Button* untuk input dari *push button NO*

Fungsi *button* yang dibuat pada program utama adalah fungsi *interlock* untuk *digital input channel 1* hingga *channel 8* yang komponennya berupa *push button*. Fungsi tersebut memanfaatkan *counter up*, modulus, dan *function block compare*, serta *contact positive edge* dan *rise trigger* atau *contact negative edge* dan *falling trigger* sesuai dengan *push button NO* atau *NC* agar sinyal yang

diberikan ketika *push button* ditekan, hal ini mencegah adanya kebocoran arus dari komponen *push button*.

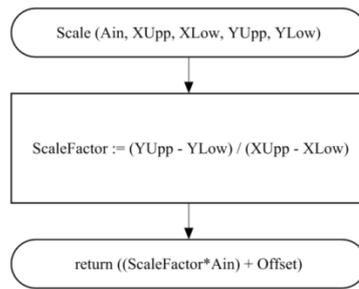


Gambar 5 Diagram alir fungsi *Button* untuk input dari *push button NC*

Gambar 4 merupakan diagram air untuk fungsi *Button* untuk *digital input* dengan komponen *momentary switch*. Algoritma yang dirancang adalah dengan memanfaatkan fungsi modulus yang mengindikasikan banyaknya tombol ditekan. Setiap kali tombol ditekan akan melakukan proses *increment* pada variabel *Status*, variabel tersebut akan dimodulus dengan 2 dan disimpan pada variabel *Mod*. Berdasarkan algoritma tersebut, maka nilai pada variabel *Mod* hanya memiliki 2 nilai: 0 atau 1 berdasarkan genap atau ganjilnya tombol ditekan. Nilai tersebutlah yang menentukan ditekan atau tidaknya *momentary switch* yang disimpan pada variabel *PB*.

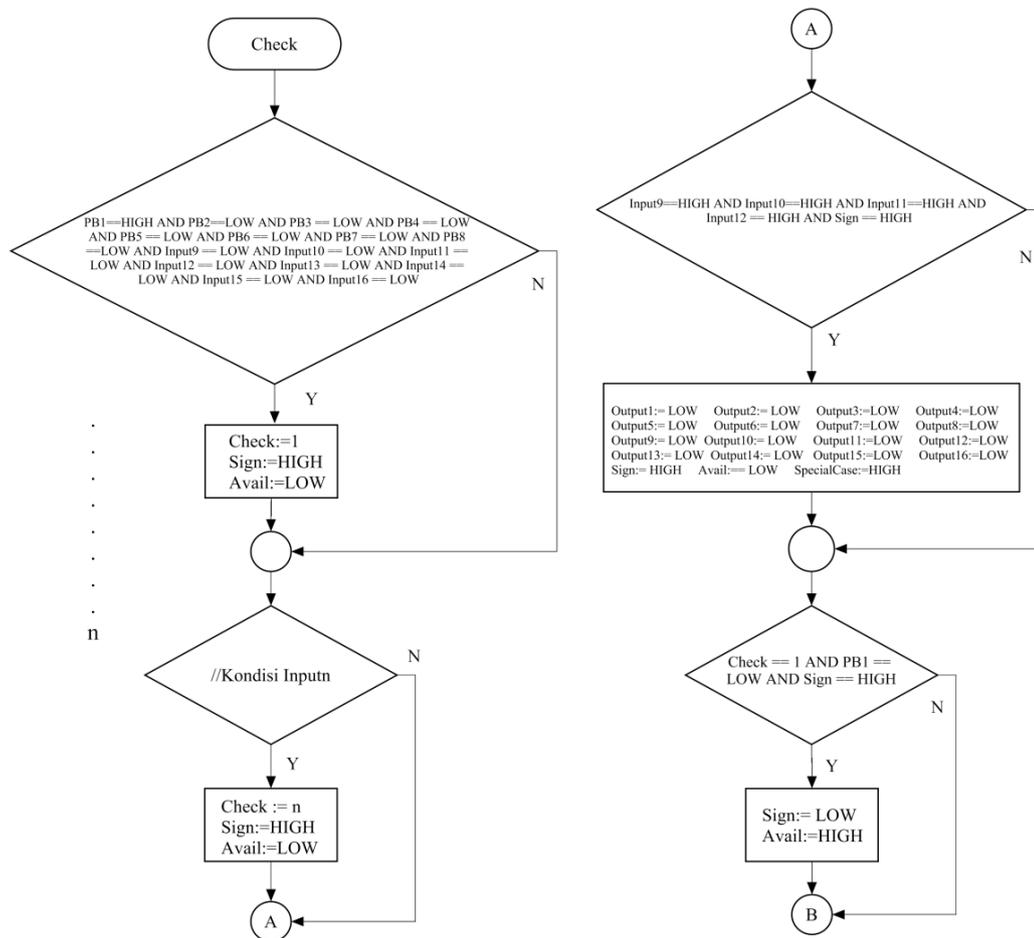
Gambar 5 merupakan diagram air untuk fungsi *Button* untuk *digital input* dengan komonen *momentary switch NC*. Diagram alir tersebut memiliki sedikit perbedaan karena *push button NC* akan *trigger* logika 0 pada saat ditekan, oleh karena itu pada algoritma yang dirancang dilakukan *1's complement* untuk memperoleh sinyal *HIGH* saat *push button* ditekan.

Fungsi *scale* merupakan fungsi yang mengkonversi nilai analog 16 bit yang terbaca pada *bus terminal EL3002* menjadi nilai tegangan input dengan rentang 0 V hingga 10 V dengan nilai *offset* yang bertujuan agar hasil pembacaan maksimal mendekati nilai 10V. Gambar 6 merupakan fungsi untuk melakukan penskalaan dari nila tegangan *analog input* yang telah diubah ke dalam bentuk desimal 16-bit. Variabel *YUpp* merupakan variabel yang menyimpan nilai maksimum dari hasil keluaran penskalaan yang diinginkan (dalam hal ini 10) dan *YLow* merupakan variabel yang menyimpan nilai minimum dari penskalaan yang diinginkan (0) sedangkan *XUpp* merupakan variabel yang menyimpan nilai input maksimum yang ingin dilakukan penskalaan (dalam hal ini 32767) dan *XLow* merupakan variabel yang menyimpan nilai input minimum dari input yang ingin diskala (0) dan *offset* merupakan nilai yang perlu dijumlahkan agar hasil penskalaan yang diperoleh sesuai dengan nilai yang diinginkan pada saat kondisi *input* sudah mencapai posisi maksimum namun nilai yang dihasilkan tidak sama dengan nilai maksimum seharusnya (*XUpp*), atau digunakan untuk penskalaan dengan nilai keluaran minimum tidak sama dengan nol.

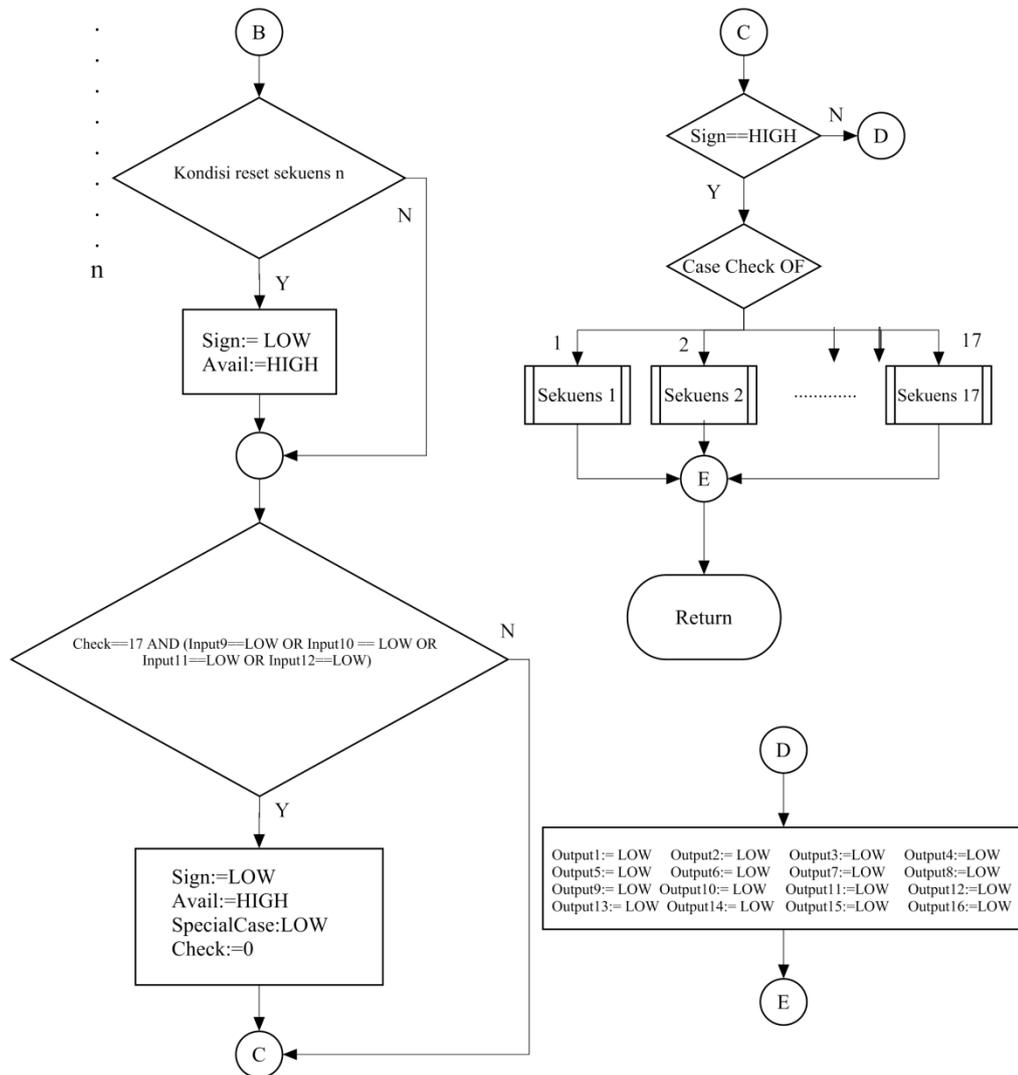


Gambar 6 Diagram alir fungsi *Scale*

Subprogram *Check* merupakan instruksi yang menentukan sekuens mana yang akan dieksekusi. Gambar 7 dan Gambar 8 merupakan diagram alir untuk Algoritma yang dirancang, yaitu dengan memberikan tanda angka setiap kali salah satu tombol atau kombinasi ditekan, dengan syarat tidak ada tombol yang ditekan sebelumnya atau sekuens yang telah dijalankan sebelumnya sudah di-reset, hal ini ditentukan dengan kondisi variabel *Sign*. Sekuens yang dieksekusi sesuai dengan nomor tanda angka setelah tombol ditekan hingga variabel *Sign* bernilai *LOW* yang menandakan sekuens telah di-*reset*.



Gambar 7 Diagram alir subprogram *Check* bagian 1

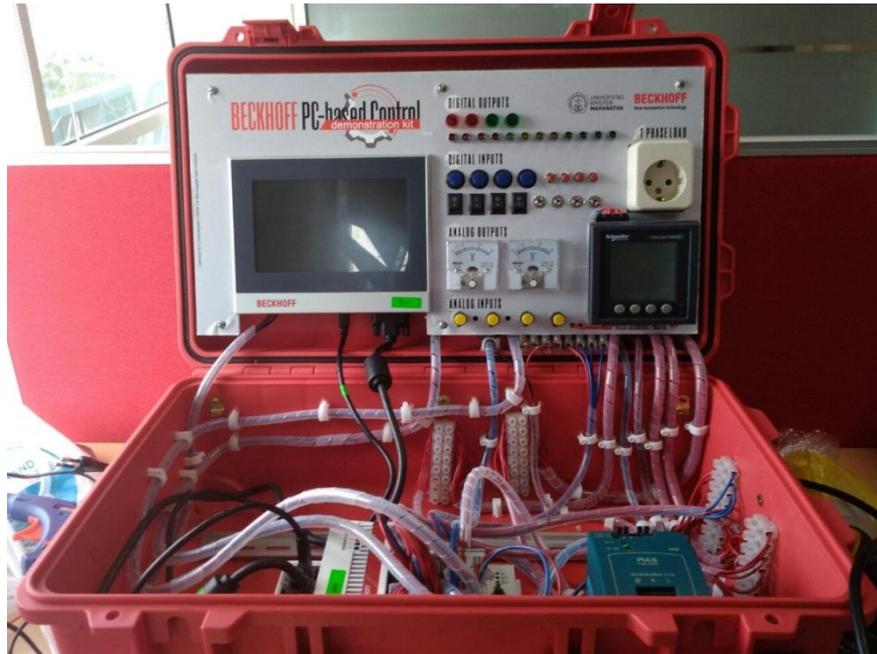


Gambar 8 Diagram alir subprogram *Check* bagian 2

2.3.4 Realisasi Alat Demonstrasi

Alat demonstrasi direalisasikan menggunakan komponen-komponen yang sesuai. Komponen-komponen tersebut dipasang pada sebuah akrilik sehingga membentuk sebuah panel *input/output* dan algoritma-algoritma untuk sekuens proses dan simulasi sistem keamanan bangunan beserta masing-masing *HMI*-nya direalisasikan dengan menggunakan perangkat lunak *TwinCAT 3 XAE*. Program tersebut di-*upload* menuju *CX9020* sebagai kontroler utama untuk menjalankan masing-masing demonstrasi kontrol proses tersebut. Komponen-komponen *input/output* baik berupa *push button*, *toggle switch*, *LED*, *potensiometer*, *voltmeter*, *powermeter*, dan *Control Panel* dipasang pada sebuah pelat akrilik dengan ketebalan 5mm, pemasangan tersebut sesuai dengan cetak biru pada Gambar 9. Pengkabelan dilakukan setelah semua komponen dipasang pada pelat akrilik tersebut, pengkabelan dilakukan sesuai dengan skematik rangkaian masing-masing fungsi *I/O*. Setelah pengkabelan selesai dilakukan, panel akrilik ditempatkan pada bagian atas koper yang sudah disediakan.

Kontroler, *bus terminal*, *power supply* ditempatkan di bagian bawah koper yang sudah dipasang dua buah *DIN Rail*. Selain komponen-komponen tersebut, beberapa *terminal block* juga ditempatkan pada bagian bawah koper, *terminal block* tersebut digunakan untuk melakukan pengkabelan dengan hubungan paralel, pemilihan tersebut berdasarkan skematik rangkaian yang cenderung banyak memiliki banyak *node* yang terhubung secara paralel.



Gambar 9. Alat demonstrasi yang sudah direalisasikan

2.3.5 Perancangan Human-Machine Interface (HMI)

Human-Machine Interface (HMI) merupakan sebuah antarmuka yang menghubungkan antara pengguna dengan mesin. *HMI* pada alat demonstrasi ini bertujuan untuk melakukan *monitoring* pada sistem yang telah dirancang, dalam hal ini melihat keadaan seluruh *input/output* maupun parameter lainnya. Parameter-parameter yang akan dirancang untuk diamati pada *HMI* adalah berupa status *digital input*, *digital output*, *analog input*, dan *analog output* serta parameter lain berupa *input status* (sedang aktif atau yang terakhir aktif) pada program sekuens proses dan untuk program simulasi sistem keamanan bangunan parameter yang diamati adalah berupa masing-masing kondisi dari alarm, sirine, lampu taman, gerbang, status sensor, waktu, intensitas cahaya, dan waktu *timer* untuk *delay* gerbang.

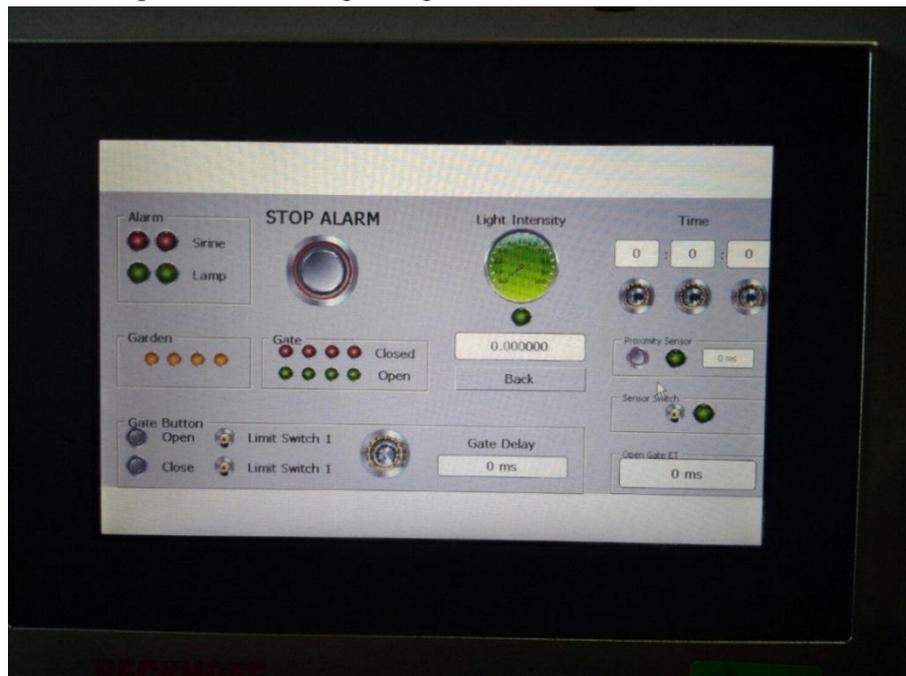
Program yang telah dibuat terdiri 1 buah fungsi *main* dan 29 buah fungsi lainnya: 16 sekuens proses, 1 buah subprogram simulasi keamanan bangunan, 10 fungsi penskalaan, 1 buah fungsi *interlock* dari *input momentary switch*, dan 1 buah subprogram untuk menentukan sekuens yang akan dieksekusi berdasarkan *input* tertentu.

HMI yang telah dibuat terdiri dari 2 jenis: *HMI* untuk 16 sekuens proses dan 1 buah *HMI* khusus untuk simulasi sistem keamanan bangunan. Kedua jenis *HMI* ini pun dibuat dalam perangkat lunak *TwinCAT 3 XAE* pada 1 berkas yang sama dengan programnya. Tampilan pada *HMI* tersebut dapat diubah dengan menekan tombol *Next* pada *HMI* sekuens proses untuk mengubah tampilan menjadi *HMI* simulasi sistem keamanan bangunan. Dengan cara yang sama, tampilan pada *HMI* sistem keamanan bangunan dapat diubah menjadi tampilan *HMI* sekuens proses dengan menekan tombol *Back*. Masing-masing *HMI* dibuat berdasarkan fungsinya untuk melakukan *monitoring* sistem.



Gambar 10. Tampilan *HMI* simulasi sistem keamanan bangunan

Gambar 10 dan Gambar 11 menunjukkan hasil realisasi pembuatan *HMI* untuk simulasi sistem keamanan bangunan. Komponen-komponen pada *HMI* tersebut disesuaikan dengan parameter keamanan gedung yang digunakan, yaitu sirine, lampu alarm, lampu taman, motor gerbang (buka dan tutup), *proximity status*, *sensor switch status* dan intensitas cahaya yang diatur berdasarkan komponen *input* berupa tombol gerbang (buka dan tutup), tombol *stop* alarm, *sensor switch*, *proximity sensor*, waktu, *limit switch*, dan tetapan waktu tunda gerbang.



Gambar 11 Realisasi *HMI* simulasi sistem keamanan bangunan pada *Control Panel*

3. HASIL DAN PEMBAHASAN

A. Pengujian *digital input* dan *digital output*

Pengujian ini dilakukan untuk memvalidasi perubahan *digital input* saat belum ditekan/dinyalakan ataupun saat ditekan/dinyalakan dan keluaran dari *digital output* saat dalam keadaan *ON* atau *OFF* dengan melihat perubahannya pada *TwinCAT 3*. *Digital input* berhasil mengeluarkan *logic high* (1) atau *logic low* (0) sesuai dengan sifatnya sebagai *naturally open* (*NO*)

atau *naturally closed (NC)* dan *logic digital output* sudah berhasil berubah menjadi *high (1)* bila diberi *logic high (1)* atau *logic low (0)* sesuai dengan *input* yang terhubung dengan *output*.

B. Pengujian *analog input* dan *analog output*

Pengujian ini dilakukan untuk memvalidasi perubahan *analog input* saat terjadi perputaran potensiometer atau saat tidak terjadi perputaran potensiometer dan keluaran dari *analog output* dengan melihat perubahan *voltmeter* saat terjadi perputaran potensiometer atau saat tidak terjadi perputaran potensiometer. Nilai pada *analog input* dan *analog output* direpresentasikan dalam tegangan dan *bilangan decimal* yang dimunculkan pada *TwinCAT 3*.

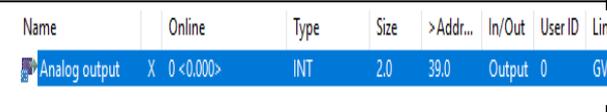
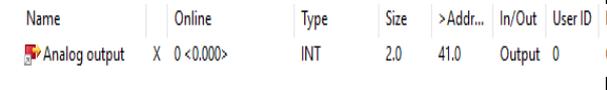
Tabel 1 Antarmuka *analog input* saat tidak terjadi perputaran potensiometer

<i>Analog Input</i>	Gambar								Keterangan
Potensio meter 1	Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to	<i>Input</i> bernilai $(-23)_{10}$ atau -0.007 $V \approx 0$ V
	Status	0x0000 (0)	Status_761...	2.0	57.0	Input	0		
	Value	X -23 <-0.007>	INT	2.0	59.0	Input	0	GVL_IO.Ain	
Potensio meter 2	Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to	<i>Input</i> bernilai $(-22)_{10}$ atau -0.007 $V \approx 0$ V
	Status	0x0000 (0)	Status_761...	2.0	61.0	Input	0		
	Value	X -22 <-0.007>	INT	2.0	63.0	Input	0	GVL_IO.Ain2.F	
Potensio meter 3	Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to	<i>Input</i> bernilai $(-11)_{10}$ atau -0.003 $V \approx 0$ V
	Status	0x0000 (0)	Status_761...	2.0	65.0	Input	0		
	Value	X -11 <-0.003>	INT	2.0	67.0	Input	0	GVL_IO.Ain3.F	
Potensio meter 4	Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to	<i>Input</i> bernilai $(7)_{10}$ atau 0.002 $V \approx 0$ V
	Status	0x0000 (0)	Status_761...	2.0	69.0	Input	0		
	Value	X 7 <0.002>	INT	2.0	71.0	Input	0	GVL_IO.Ain4	

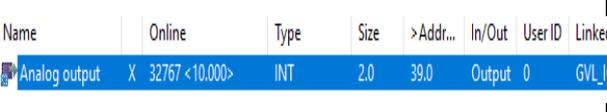
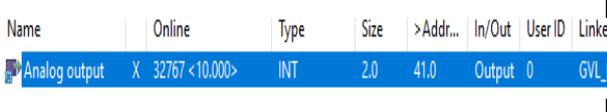
Tabel 2 Antarmuka *analog input* saat terjadi perputaran potensiometer

<i>Analog Input</i>	Gambar								Keterangan
Potensio meter 1	Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to	<i>Input</i> bernilai $(9180)_{10}$ atau 2.802 V
	Status	0x0000 (0)	Status_761...	2.0	57.0	Input	0		
	Value	X 9180 <2.802>	INT	2.0	59.0	Input	0	GVL_IO.Ain1	
Potensio meter 2	Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to	<i>Input</i> bernilai $(17882)_{10}$ atau 5.457 V
	Status	0x0000 (0)	Status_761...	2.0	61.0	Input	0		
	Value	X 17882 <5.457>	INT	2.0	63.0	Input	0	GVL_IO.Ain	
Potensio meter 3	Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to	<i>Input</i> bernilai $(26535)_{10}$ atau 8.098 V
	Status	0x0000 (0)	Status_761...	2.0	65.0	Input	0		
	Value	X 26535 <8.098>	INT	2.0	67.0	Input	0	GVL_IO.A	
Potensio meter 4	Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to	<i>Input</i> bernilai $(32717)_{10}$ atau 9.985 V
	Status	0x8000 (32768)	Status_761...	2.0	69.0	Input	0		
	Value	X 32717 <9.985>	INT	2.0	71.0	Input	0	GVL_IO.Ain4	

Tabel 3 Antarmuka *analog output* saat tidak terjadi perputaran potensiometer

Analog Input	Gambar	Keterangan
Voltmeter 1		Output bernilai 0 bit atau 0 V
Voltmeter 2		Output bernilai 0 bit atau 0 V

Tabel 4 Antarmuka *analog output* saat terjadi perputaran potensiometer

Analog Input	Gambar	Keterangan
Voltmeter 1		Output bernilai 32767 bit atau 10 V
Voltmeter 2		Output bernilai 32767 bit atau 10 V

Pada Tabel 1, Tabel 2, Tabel 3, dan Tabel 4 dapat dilihat bahwa *analog input* dan *analog output* sudah dapat menghasilkan nilai dengan benar maka dari data tersebut dapat disimpulkan bahwa *analog input* dan *analog output* sudah berfungsi dengan baik. *Analog input* berhasil mengeluarkan nilai berupa tegangan saat nilai potensiometer diubah dan nilai keluarannya sebanding dengan nilai potensiometer yang diatur. *Analog output* berhasil menampilkan tegangan yang dihasilkan dari perubahan nilai potensiometer.

Dalam sistem terjadi konversi nilai dimana nilai potensiometer yang berupa *analog* dikonversikan ke nilai *digital* berupa nilai *decimal*. Konversi tersebut menggunakan prinsip *Analog to Digital Converter (ADC)* yang memiliki persamaan :

$$\frac{\text{Resolution of the ADC}}{\text{System Voltage}} = \frac{\text{ADC Reading}}{\text{Analog Voltage Measured}}$$

Dengan :

Resolution of the ADC = Nilai $(2)^N$, N adalah jumlah *bit* yang digunakan sistem

System Voltage = Nilai tegangan yang digunakan sistem

ADC Reading = Nilai konversi dari *analog* ke *digital*

Analog Voltage Measured = Nilai tegangan *analog* yang diukur

Pada alat demonstrasi jumlah bit yang digunakan sistem ialah *15-bit* maka *Resolution of the ADC* bernilai $(2)^{15}$ atau 32768 dan nilai tanganan yang digunakan adalah 10V. Maka perhitungan untuk mencari nilai tegangan yang diukur pada *analog input* saat nilai potensiometer diubah berdasarkan nilai *ADC Reading* yang ditunjukkan pada Tabel 2 :

- Potensiometer 1

$$\text{Analog Voltage Measured} = \frac{9180 \times 10}{32768} = 2.8 \text{ V}$$

- Potensiometer 2

$$\text{Analog Voltage Measured} = \frac{17882 \times 10}{32768} = 5.457 \text{ V}$$



- Potensiometer 3

$$\text{Analog Voltage Measured} = \frac{26535 \times 10}{32768} = 8.098 \text{ V}$$

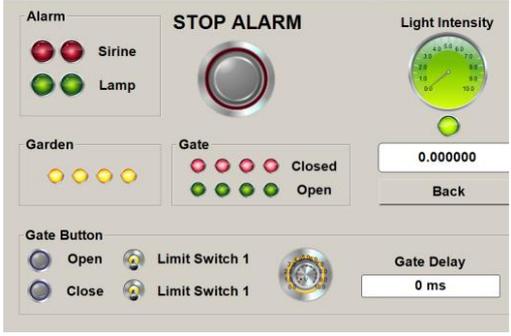
- Potensiometer 4

$$\text{Analog Voltage Measured} = \frac{32717 \times 10}{32768} = 9.985 \text{ V}$$

C. Percobaan program dan HMI

Percobaan ini dilakukan untuk memvalidasi program *visualization* yang telah dirancang dengan menampilkannya melalui HMI pada TwinCAT 3 dan juga waktu yang dibutuhkan HMI untuk muncul pada *Control Panel*.

Tabel 5 Program yang ditampilkan melalui HMI

Percobaan	Gambar	Keterangan
1		Program berupa aplikasi keamanan rumah yang mengatur lampu, gerbang dan alarm pada rumah.

Tabel 6 Waktu yang dibutuhkan HMI untuk muncul pada *Control Panel*

Percobaan	Waktu (detik)
1	23.7
2	23.5
3	23.3
4	23.3
5	23.5
6	23.7
7	23.7
8	23.2
9	23.3
10	23.3

Pada Tabel 5 dan Tabel 6 dapat dilihat bahwa HMI sudah berhasil memunculkan *visualization* dari *demo kit* dan HMI berhasil ditampilkan pada *Control Panel*, maka dari data tersebut dapat disimpulkan bahwa HMI berfungsi dengan baik.

4. KESIMPULAN

4.1 Kesimpulan

Hasil perancangan dan pengujian yang dilakukan dapat disimpulkan yaitu:

1. Sistem *demo kit* ini dibuat untuk membantu memperkenalkan *PC-based Control* pada bidang industri di Indonesia. *Panel PC* menampilkan HMI pada *demo kit*.
2. Rata-rata jeda waktu untuk menampilkan HMI pada *Panel PC* adalah 23.45 detik.
3. Sistem sudah bisa mengimplementasikan sebuah program aplikasi dari *smart house* melalui *digital input* dan *digital output*. *Analog input* berupa potensiometer sudah bisa menghasilkan tegangan pada sistem dengan rentan nilai dari 0 – 9.985 V. *Analog output* berupa *voltmeter*



belum berhasil menampilkan nilai tegangan yang dihasilkan dari perubahan nilai potensiometer.

4.2 Saran

Dalam penelitian ini masih banyak sekali kekurangan dan ketidaksempurnaan. Untuk itu, perlu dilakukan pengembangan agar ke depannya menjadi sempurna ataupun lebih baik lagi sehingga memiliki beberapa saran, diantaranya:

1. Mengganti *voltmeter* dengan *voltmeter* yang memiliki nilai tahanan dalam yang besar agar tidak terjadi pembebanan pada *voltmeter*.
2. Menambahkan sistem *IoT (Internet of Things)* pada *demo kit* agar dapat mengimplementasikan *Industry 4.0* yang merupakan kecenderungan saat ini dalam bidang otomasi dan pertukaran data pada teknologi manufaktur.

REFERENSI

- [1] Syaiful Bahri Djamarah dan Aswan Zain, "Strategi Belajar Mengajar-." Rineka Cipta, Jakarta, 2010.
- [2] N. C. Erwani Merry S, T. Rudi Sarjono, "Otomasi pada Simulator Green House dengan SCADA Berbasis Web," *Kendali-tenaga-elektronika-telekomunikasi-komputer, Sist.*, vol. 6, no. 2, pp. 205–214, 2017.
- [3] "beckhoff Automation." 2017.
- [4] H. Maghfiroh, "in Speed Control f DC Motor Base on PLC," no. Figure 1, pp. 233–238, 2013.
- [5] B. Solution, "PC-Based Controls vs. PLC-Based Controls for Machine Automation." p. 2013.
- [6] S. Zheng and F. Liu, "The design of elevator control system based on PLC and configuration," *Adv. Intell. Soft Comput.*, vol. 104, pp. 743–748, 2011.
- [7] X. Wu and H. Zhou, "Control and Simulation of Head Temperature Field of Vascular 3D Based on Beckhoff," pp. 626–631, 2016.
- [8] G. Bonney, H. Höfken, B. Paffen, and M. Schuba, "ICS/SCADA security analysis of a beckhoff CX5020 PLC," *ICISSP 2015 - 1st Int. Conf. Inf. Syst. Secur. Privacy, Proc.*, pp. 137–142, 2015.
- [9] A. Nagar and S. Mehta, "Steam turbine lube oil system protections using SCADA & PLC," *Proc. 2017 Int. Conf. Intell. Comput. Control Syst. ICICCS 2017*, vol. 2018–January, pp. 1376–1381, 2018.
- [10] Beckhoff Automation. 2017. *CX9020 | Basic CPU Module*. https://www.beckhoff.com/english.asp?embedded_pc/cx9020.htm. [Diakses 3 November 2017]
- [11] Beckhoff Automation. 2017. *The Beckhoff Bus Terminal* system. [https://infosys.beckhoff.com/english.php?content=../content/1033/bx8000/2792573963.html&id=.](https://infosys.beckhoff.com/english.php?content=../content/1033/bx8000/2792573963.html&id=..)[Diakses 22 November 2017]
- [12] Beckhoff Automation. 2017. *EtherCAT Terminal*. [https://infosys.beckhoff.com/english.php?content=../content/1033/fieldbusinfosys/3149086475.html&id=.](https://infosys.beckhoff.com/english.php?content=../content/1033/fieldbusinfosys/3149086475.html&id=..) [Diakses 22 November 2017]
- [13] Beckhoff Automation. 2017. *TwinCAT 3 | eXtended Automation (XA)*. <https://www.beckhoff.com/english.asp?twincat/twincat-3.htm>. [Diakses 10 November 2017]
- [14] Johnson, Curtis D. 2003. *Process Control Instrumentation Technology*. Seventh Edition. New Delhi : Prentice-Hall of India.
- [15] Bryan, L.A & E.A Bryan. 1997. *Programmable Controller : Theory and Implementation*. Second Edition. United States of America : Industrial Text Company.