

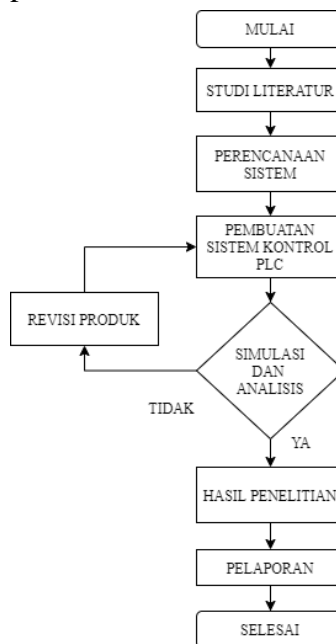
BAB III METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang digunakan adalah R&D (Research and Development). Metode penelitian dan pengembangan ini merupakan metode yang digunakan untuk menghasilkan produk tertentu, dan menyempurnakan suatu produk yang sesuai dengan acuan dan kriteria dari produk yang telah dibuat (Haryati, 2012). Produk tersebut tidak hanya meliputi perangkat keras, tetapi juga perangkat lunak. Sistem kontrol yang dibuat pada penelitian ini berguna sebagai data pendukung dalam pembuatan produk perangkat keras (Mulyatiningsih, 2012). Perancangan sistem kontrol PLC pada alat pengangkut sampah ini dibuat dengan tujuan untuk menghasilkan suatu produk perangkat lunak yang valid melalui berbagai tahapan tertentu sebagaimana mengacu kepada metode penelitian R&D.

3.2 Prosedur Penelitian

Penelitian ini memerlukan sebuah diagram alir agar mudah dipahami untuk mencapai sebuah tujuan penelitian. Berikut tahapan prosedur penelitian yang dilakukan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram alir penelitian.

Berdasarkan diagram alir penelitian pada Gambar 3.1 berikut adalah penjelasan singkat mengenai tahapan diagram alir tersebut:

1. Studi Literatur

Kegiatan studi literatur yaitu kegiatan yang dilakukan untuk mengkaji, menyelidiki, dan mengumpulkan informasi sebagai dasar teori dalam pelaksanaan penelitian. Dasar teori yang dikaji berasal dari buku, jurnal, maupun data observasi langsung di lapangan.

2. Perencanaan Sistem

Sebelum melakukan perancangan dalam membuat sistem, maka dilakukan perencanaan/*planning* terlebih dahulu. Peneliti membuat perancangan kontrol dengan PLC meliputi program seperti apa yang akan dibuat, bagaimana hasil program yang telah dibuat, siapa pengguna yang akan mengontrol PLC tersebut, mengapa perancangan kontrol tersebut dianggap penting.

3. Pembuatan Sistem Kontrol PLC

Pada tahap ini peneliti membuat sistem kontrol dengan menggunakan *software* CX-Designer untuk keperluan perancangan HMI (*Human Machine Interface*) serta *software* CX-Programmer untuk keperluan kontrol PLC (*Programmable Logic Control*) yang mana HMI dan PLC adalah suatu kesatuan dalam sistem SCADA.

4. Simulasi dan Analisis

Pada tahap ini peneliti melakukan simulasi dan menganalisis mengenai hasil perancangan sistem kontrol yang sudah dibuat. Simulasi ini menggabungkan antara program yang dibuat pada PLC dengan HMI yang dibuat dengan *software* CX-Designer dan CX-Programmer.

5. Revisi Produk

Revisi produk ini merupakan perbaikan dan penyempurnaan terhadap perancangan kontrol PLC pada sistem SCADA berdasarkan hasil simulasi, hasil diskusi serta saran-saran yang diberikan dari dosen pembimbing.

6. Hasil Penelitian

Setelah melewati beberapa proses revisi dan sudah sesuai dengan apa yang dicapai maka dilakukan hasil akhir atau mendapatkan simpulan dari perancangan sistem kontrol yang telah dibuat.

7. Pelaporan

Semua kegiatan yang telah dilakukan dari awal hingga akhir selanjutnya dilakukan pembuatan laporan sesuai Pedoman Penulisan Karya Ilmiah UPI Tahun 2019.

3.3 Pengumpulan Data

Dalam penelitian ini, pengumpulan data yang berkaitan dengan penelitian dilakukan dengan tahapan yang memiliki tujuan untuk mencari dan menjawab permasalahan.

1. Studi Literatur

Mengkaji teori sebagai penunjang penelitian yang berasal buku, internet, jurnal nasional, jurnal internasional, dan artikel ilmiah yang berhubungan dengan penelitian ini, khususnya dalam perancangan kontrol PLC pada alat pengangkut sampah otomatis terintegrasi sistem SCADA.

2. Diskusi

Diskusi dilakukan dengan tim peneliti pada ajang PIMNAS (Pekan Ilmiah Mahasiswa Nasional) serta dosen pembimbing berkaitan dengan topik skripsi ini.

3. Observasi

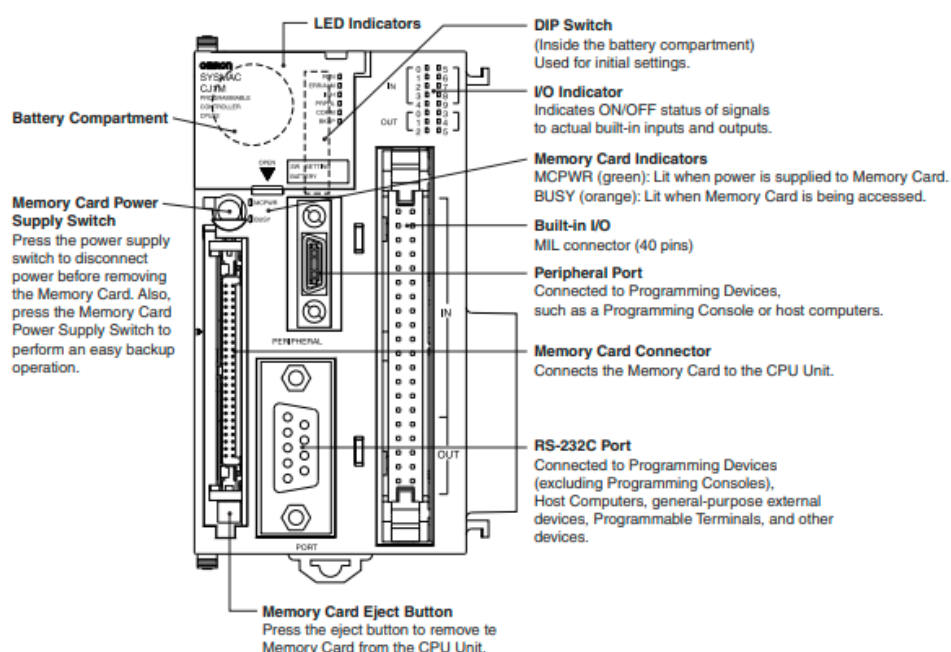
Observasi dilakukan dengan pengumpulan data di lapangan tepatnya di Sungai Citarum, Bandung untuk mengidentifikasi permasalahan yang dihadapi sewaktu kegiatan KKN (Kuliah Kerja Nyata).

3.4 Perancangan Kontrol PLC (*Programmable Logic Controller*)

Dalam sistem SCADA komponen yang perlu ditunjang ada dua bagian, yaitu pembuatan program PLC (*Programmable Logic Controller*) dan pembuatan desain HMI (*Human Machine Interface*). *Software* pada pembuatan kontrol PLC ini menggunakan aplikasi CX-Programmer 9.6 dengan menggunakan bahasa program *Ladder diagram*. Pada tahap ini terdiri dari beberapa bagian, diantaranya:

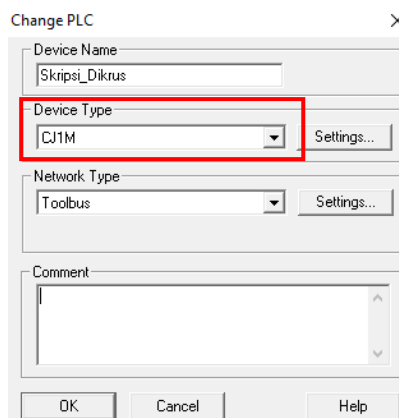
3.4.1 Konfigurasi PLC Tipe CJ1M

PLC tipe CJ1M-CPU23 merupakan bagian yang berfungsi sebagai *Control Processing Unit* dari PLC OMRON seri CJ. CJ1M-CPU23 merupakan tipe PLC modular dimana jumlah I/O nya bisa ditambah sesuai kebutuhan. Untuk komunikasi dengan komputer digunakan komunikasi serial RS-232. Tipe ini memiliki kapasitas memori 5Ksteps dan I/O poin sebanyak 160 poin/10 unit. *Input* tegangan sebesar 17-24 VDC dan arus 6.2mA-10mA serta *output* tegangan sebesar 5-24 VDC dan arus 0.1mA-1.8A (Ardi, S., 2017). Berikut contoh gambar serta spesifikasi dari PLC CJ1M-CPU23 dapat dilihat pada Gambar 3.2.



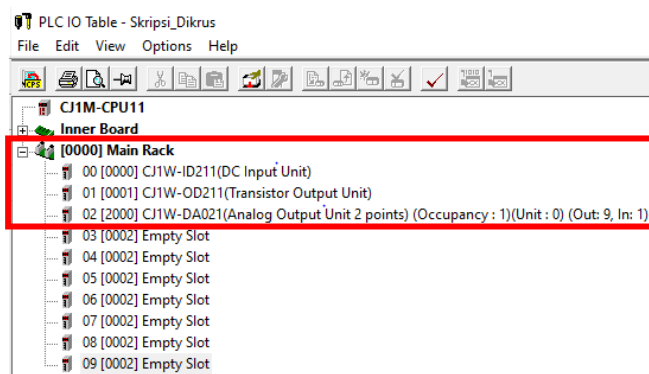
Gambar 3.2. Datasheet CJ1M-CPU23

Dalam aplikasi CX-Programmer tipe PLC ditentukan pertama kali pada saat membuat file atau proyek baru dengan cara mengklik *File*→*New* kemudian akan muncul menu untuk memilih tipe PLC, atur *Device Type* ke CJ1M seperti contoh pada Gambar 3.3 berikut.



Gambar 3.3. Memilih tipe PLC

Setelah memilih tipe PLC, selanjutnya mengatur *rack I/O* sesuai kebutuhan dengan mengklik menu *Project*→*IO Table and Unit Setup*. Dalam penelitian ini terdapat 3 rack dari 10 rack yang dibutuhkan yaitu rack DC *Input Unit* tipe CJ1W-ID211 (16 poin), *Transistor Input Unit* tipe CJ1W-OD211 (16 poin), dan *Analog Output Unit* tipe CJ1W-DA021 (2 poin) seperti yang diperlihatkan pada Gambar 3.4.



Gambar 3.4. Mengatur *rack I/O Table*

3.4.2 Konfigurasi Alamat *Input* dan *Output*

Setelah mengatur jumlah *rack* dan *I/O Table* maka selanjutnya menentukan kebutuhan alamat *I/O* PLC yang akan digunakan. Penentuan alamat *I/O* ini penting dilakukan untuk memudahkan dalam pembuatan *ladder diagram* agar tidak terjadi kesalahan dalam pemrograman, selain itu penentuan alamat *I/O* ini memudahkan dalam menentukan *write address* ketika disambungkan dengan HMI. Cara menentukan kebutuhan *I/O* yaitu dengan mengklik menu *Symbols*→*Klik Kanan*→*Insert Symbol* setelah itu isi data *I/O* yang akan digunakan. Berikut tabel penentuan alamat *I/O* yang telah dibuat.

Tabel 3.1 Tabel *input* CJ1W-ID211 yang digunakan

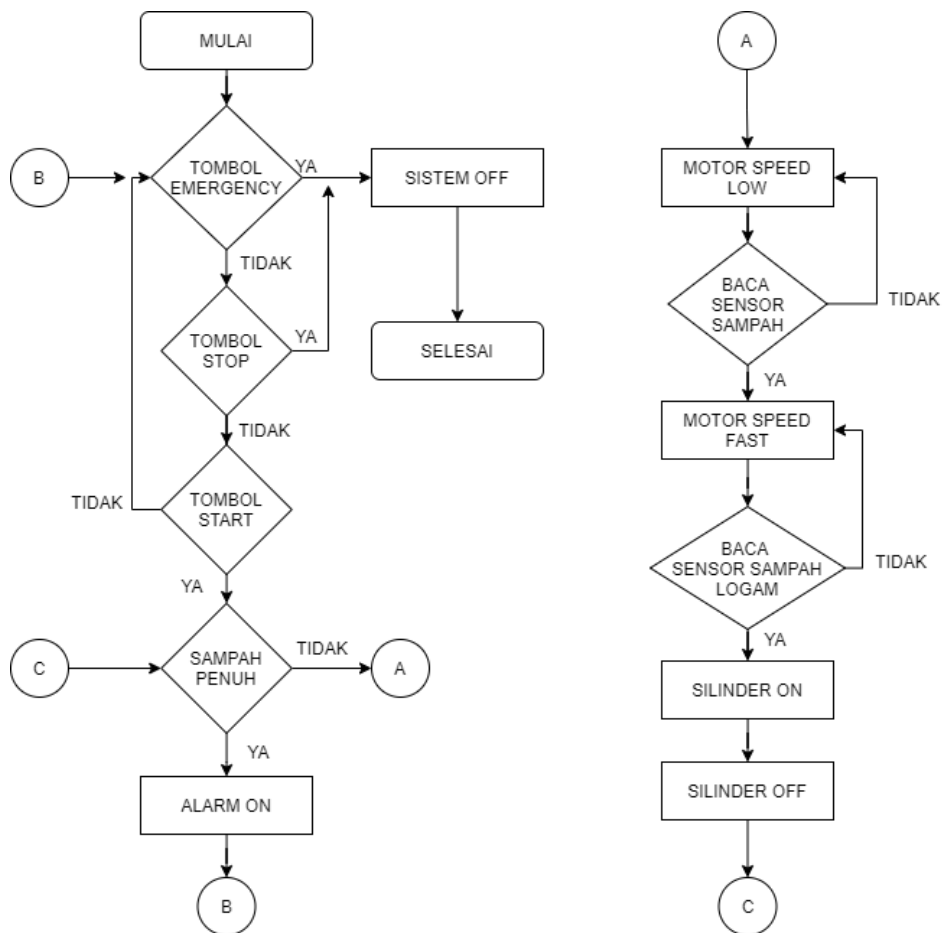
No	<i>Input</i>	Keterangan
1.	0.00	<i>System ON</i>
2.	0.01	<i>System OFF</i>
3.	0.04	<i>Emergency</i>
4.	0.05	<i>Conveyor Speed 1</i>
5.	0.06	<i>Conveyor Speed 2</i>
6.	0.07	<i>Manual Pneumatic</i>
7.	0.08	Sensor Benda
8.	0.09	Sensor Logam
9.	0.10	Sensor Sampah Logam
10.	0.11	Sensor Sampah Non Logam

Tabel 3.2 Tabel *output* CJ1W-OD211 yang digunakan

No	<i>Output</i>	Keterangan
1.	1.00	<i>Lamp System</i>
2.	1.01	<i>Lamp Sensor</i>
3.	1.02	<i>Lamp Pneumatic</i>
4.	1.03	<i>Lamp Sampah Logam</i>
5.	1.04	<i>Lamp Sampah Non Logam</i>
6.	1.05	<i>Silinder Pneumatic</i>
7.	1.06	<i>Lamp Emergency</i>

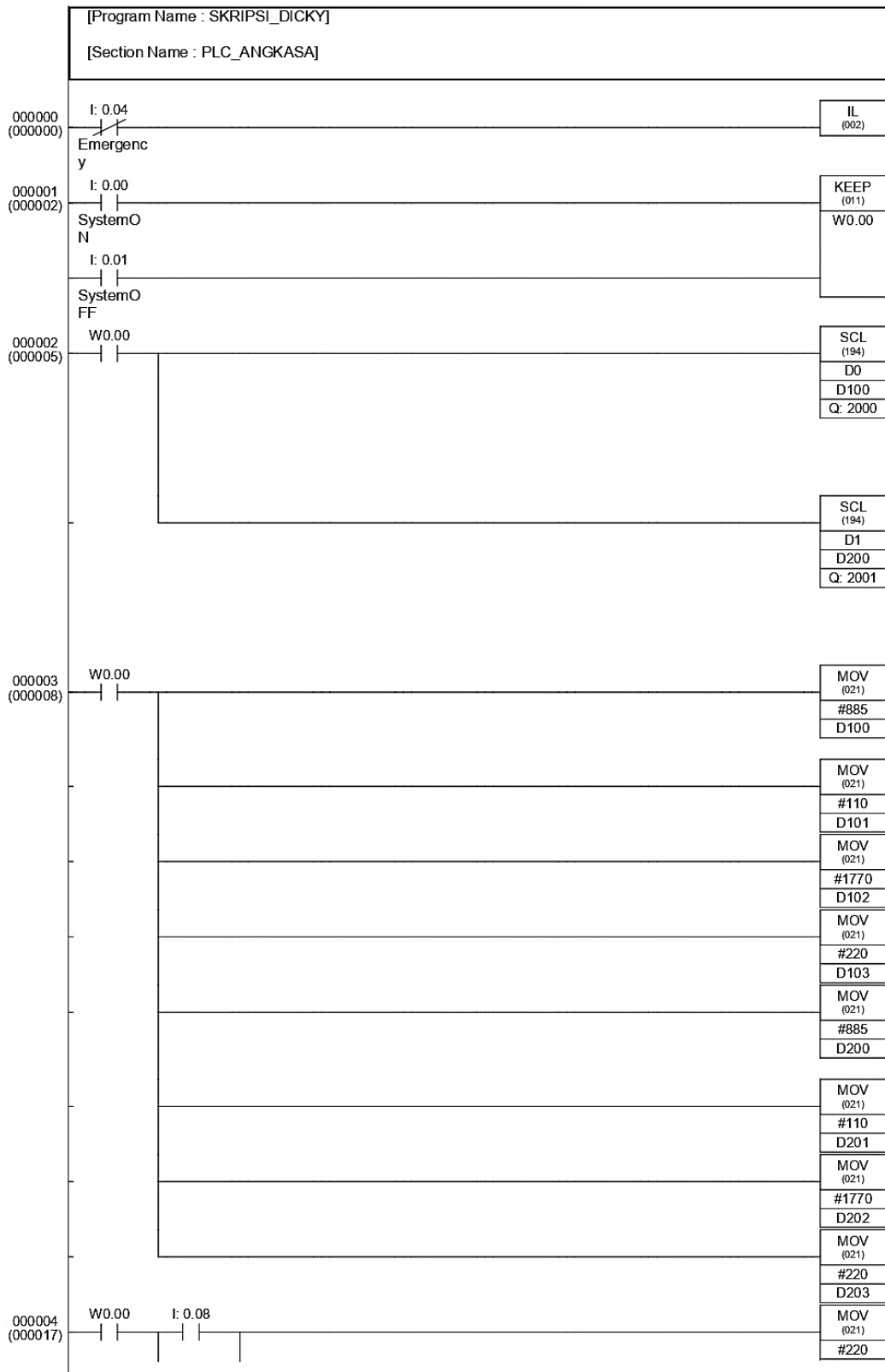
3.4.3 Pembuatan Ladder diagram

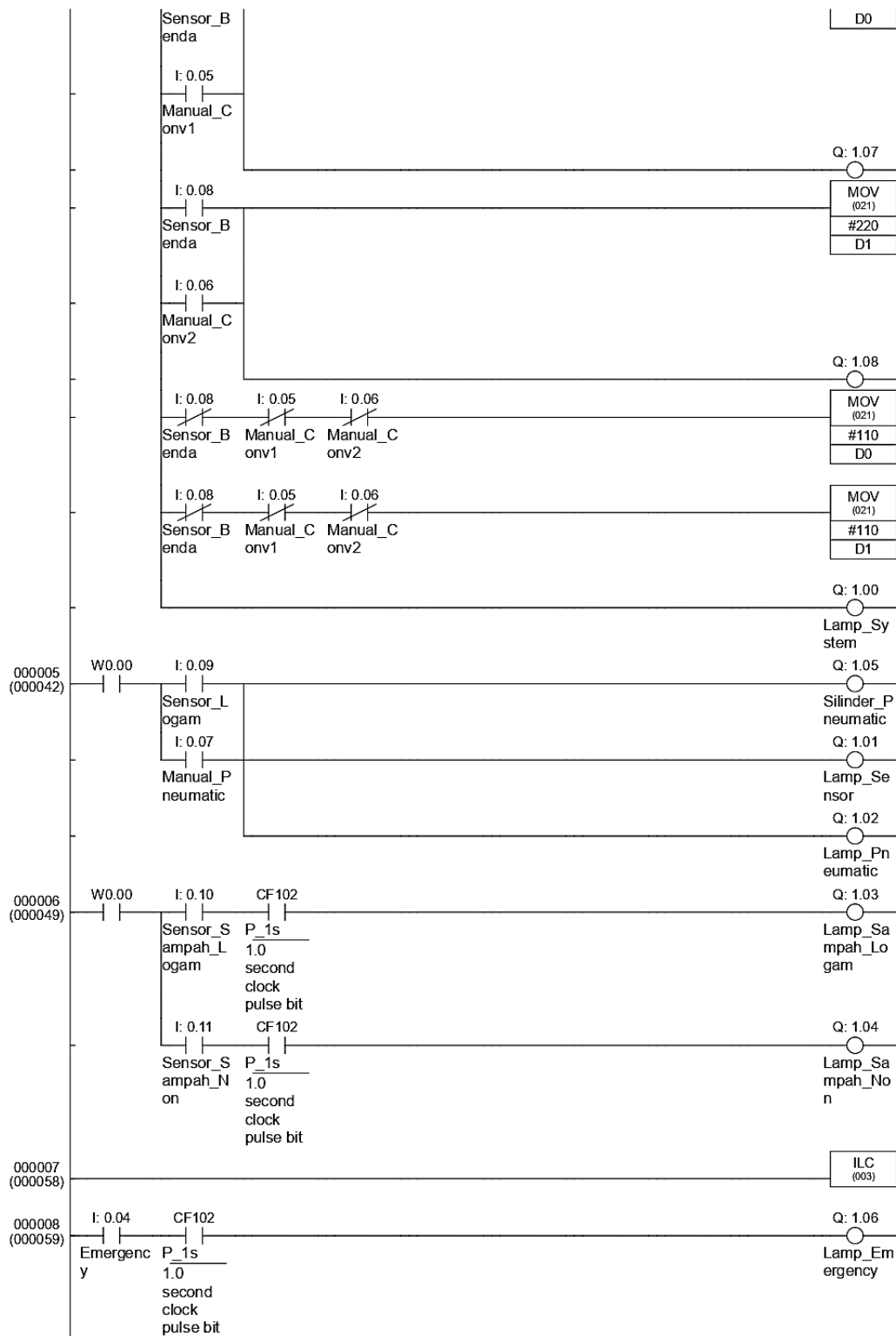
Setelah menentukan I/O yang akan digunakan pada PLC, selanjutnya membuat *flowchart* terlebih dahulu agar memudahkan dalam pembuatan *ladder diagram*. *Flowchart* ini merupakan algoritma cara kerja sistem dari alat pengangkut sampah otomatis di sungai. Berikut *flowchart* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Flowchart Ladder diagram

Berdasarkan *flowchart* di atas dapat dibuat program *ladder diagram* pada CX-Programmer seperti berikut ini:





Gambar 3.6 Pembuatan Ladder diagram

Berikut adalah penjelasan keseluruhan program pada *ladder diagram* diatas. Pada *ladder* tersebut sistem pemrograman alat pengangkut sampah otomatis di sungai ini menggunakan 6 *input*, 4 sensor, 7 *output*, serta menggunakan dua *output* analog untuk mengatur kecepatan motor konveyor.

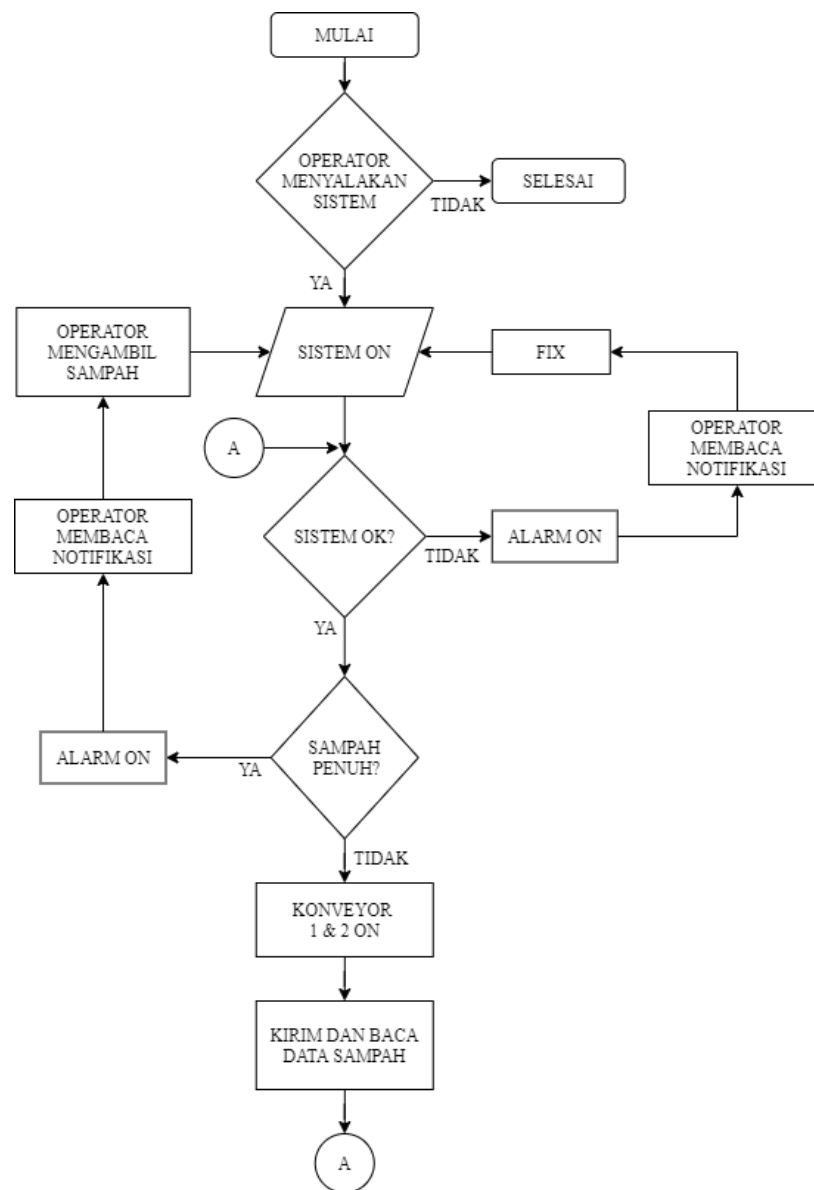
Program diawali dengan kondisi dari tombol *emergency* [0.04] apabila ditekan maka sistem berhenti, begitu juga dengan kondisi tombol *stop* [0.01]. Sistem bisa bekerja apabila tombol *SystemON* [0.00] ditekan, tetapi jika kondisi sensor (*proximity* kapasitif) dari tempat sampah logam penuh [0.10] atau tempat sampah non logam penuh [0.11] maka akan memberikan indikator alarm kepada operator [1.03] atau [1.04]. Apabila tempat sampah belum penuh atau belum terisi maka sistem bekerja kembali dengan motor konveyor berkecepatan minimum [0.05] sampai sensor benda (ultrasonik) mendeteksi sampah yang mengalir menuju konveyor vertikal [0.08], maka kedua motor konveyor berubah menjadi kecepatan maksimum [0.06]. Kemudian sampah tersebut melewati konveyor vertikal dan horizontal dan akan dideteksi oleh sensor logam (*proximity* induktif) apabila sampah tersebut memiliki unsur logam [0.10] maka sampah tersebut dikategorikan sebagai sampah logam dan akan terdorong oleh silinder pneumatik [1.05] ke tempat pembuangan sampah logam. Apabila sensor logam (*proximity* induktif) [0.10] tidak mendeteksi adanya sampah logam, maka sampah tersebut dikategorikan sampah non-logam dan menuju ke tempat pembuangan sampah non-logam. Sistem akan terus bekerja secara *loop* sampai tempat sampah terisi penuh. Apabila tempat sampah tersebut penuh maka sensor (*proximity* kapasitif) akan aktif dan memberikan alarm kepada operator.

Pada proses pemrograman ini juga terdapat fungsi khusus yaitu adanya indikator lampu pada kondisi *input* atau *output* ketika aktif. Fungsinya yaitu memberikan informasi apabila sistem di *field* telah bekerja, tetapi pada HMI tidak muncul indikatornya maka operator harus menekan tombol *emergency* dan harus melakukan pemeriksaan terlebih dahulu. Perbedaan fungsi tombol *emergency* dengan tombol *stop* yaitu pada tombol *emergency* bisa menghentikan semua fungsi atau instruksi pada program tanpa mematikan sistem. Apabila tombol *stop* bisa mematikan sistem sekaligus menghentikan instruksi pada program yang berjalan.

3.5 Perancangan HMI (*Human Machine Interface*)

Software dalam perancangan desain HMI ini menggunakan CX-Designer OMRON yang berfungsi untuk menampilkan, mengawasi, dan mengontrol proses mesin yang sedang berlangsung di lapangan dengan membuat simulasinya terlebih dahulu. HMI juga berguna untuk operator dalam melakukan pengawasan *plant*, pengendalian *plant*, penanganan alarm, serta mendeteksi *error* dalam program sebelum atau sesudah dijalankan (Fiset, 2009).

Berikut *flowchart* HMI secara operasional untuk operator dalam menjalankan sistem kontrol dapat dilihat pada Gambar 3.7.

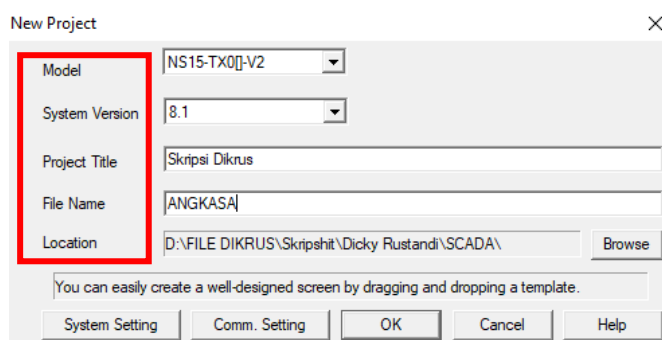


Gambar 3.7 Flowchart operator HMI

Berdasarkan *flowchart* diatas, untuk menunjang semua kebutuhan operator dalam menjalankan sistem kontrol alat pengangkut sampah otomatis ini diperlukan pembuatan perancangan HMI melalui beberapa tahap, yaitu:

3.5.1 Pembuatan Proyek dan *Screen*

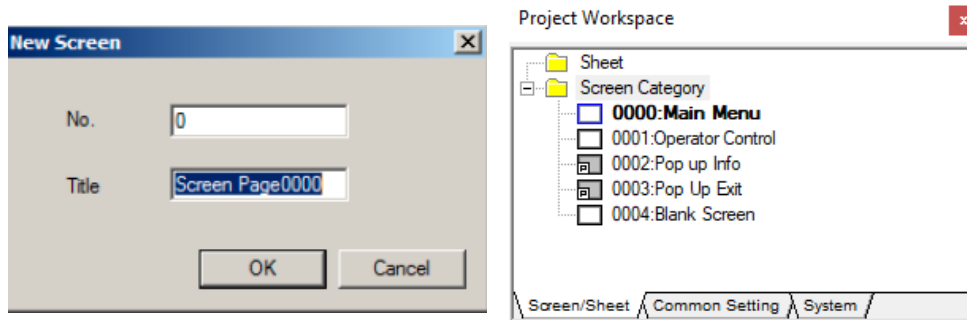
Pada awal proses pembuatan simulasi HMI di CX-Designer ini hampir sama dengan pembuatan pemrograman pada CX-Programmer yaitu dengan membuat file atau proyek baru dengan cara mengklik menu *File*→*New Project* kemudian edit judul proyek serta nama file, lalu untuk *Model* dan *System Version* mengikuti bawaan hasil instalannya seperti yang diperlihatkan pada Gambar 3.8



Gambar 3.8 Pembuatan Proyek di CX-Designer

Setelah membuat proyek di CX-Designer selanjutnya membuat *screen* atau layar yang bertujuan untuk menentukan jumlah halaman yang akan digunakan dalam simulasi HMI. Pembuatan *screen* dapat dilakukan dengan dua cara, yaitu dengan cara mengklik menu *File*→*New Screen* atau proyek sudah berjalan dengan memilih di menu *Project Workspace*→*Screen/Sheet*→*New Screen*.

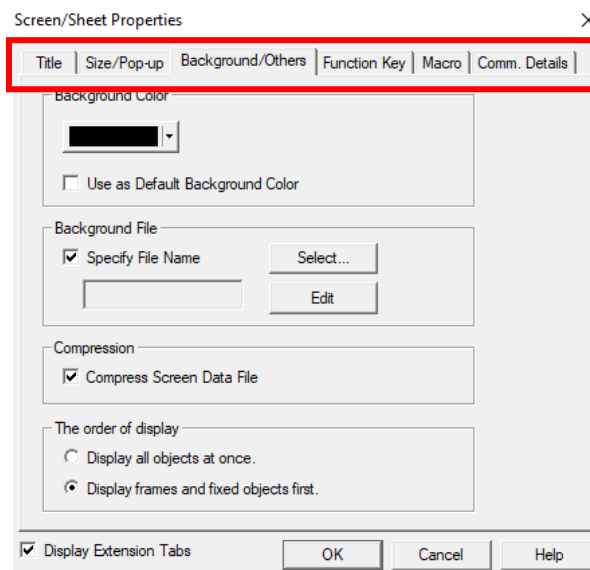
Dalam mendesain HMI dari alat pengangkut sampah otomatis ini peneliti menggunakan 5 layar yang terdiri dari layar pertama untuk tampilan *main menu* [0000], layar kedua untuk tampilan *operator control* [0001], layar ketiga untuk *pop up info* [0002], layar keempat untuk *pop up exit* [0003], dan layar kelima layar *blank screen* [0004]. Berikut tampilan pembuatan layar dapat ditunjukkan pada Gambar 3.9.



Gambar 3.9 Pembuatan layar pada CX-Designer

3.5.2 Pembuatan Objek pada Layar

Sebelum membuat objek pada layar yang akan digunakan sebagai kontrol HMI, bisa terlebih dahulu mengubah warna *background* tampilan layar ke warna dasar atau dengan mengunggah berkas gambar yang diinginkan caranya dengan menekan klik kanan pada layar → *Screen/Sheet Property* → *Background/Others*. Pada menu *Screen/Sheet Property* juga terdapat pilihan untuk mengubah ukuran layar sesuai kebutuhan di bagian *Size/Pop Up* seperti yang diperlihatkan pada Gambar 3.10.



Gambar 3.10 Pembuatan objek pada layar

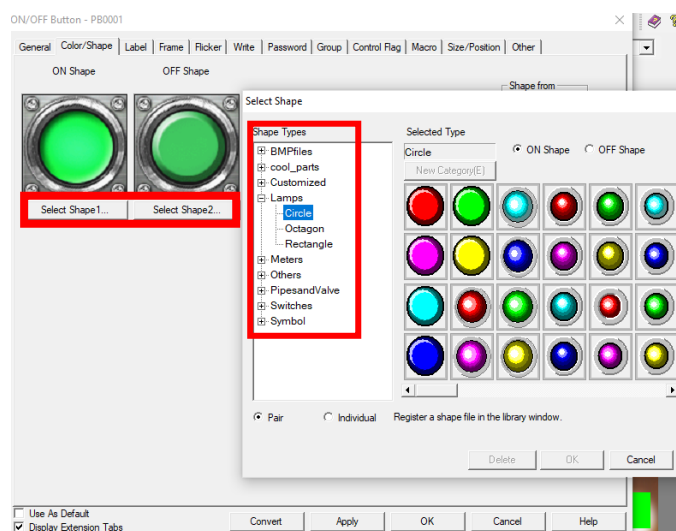
Setelah menentukan *background* pada layar, selanjutnya mengisi layar tersebut dengan objek-objek yang dibutuhkan seperti *push button*, *input sensor*, konstruksi mesin, *numeral display*, *output*, dan alarm. Langkah-langkah memasukkan objek pada layar yaitu dengan mengklik menu *Functional Obejct* seperti *ON/OFF Button* untuk menambahkan *input push button*, *Bit Lamp* untuk

menambahkan *output* serta indikator, *Label* untuk memberi keterangan, *Numeral DisplayInput* untuk menambahkan keterangan angka pada program serta masih banyak fungsi-fungsi yang lainnya seperti yang ditunjukkan pada Gambar 3.11.



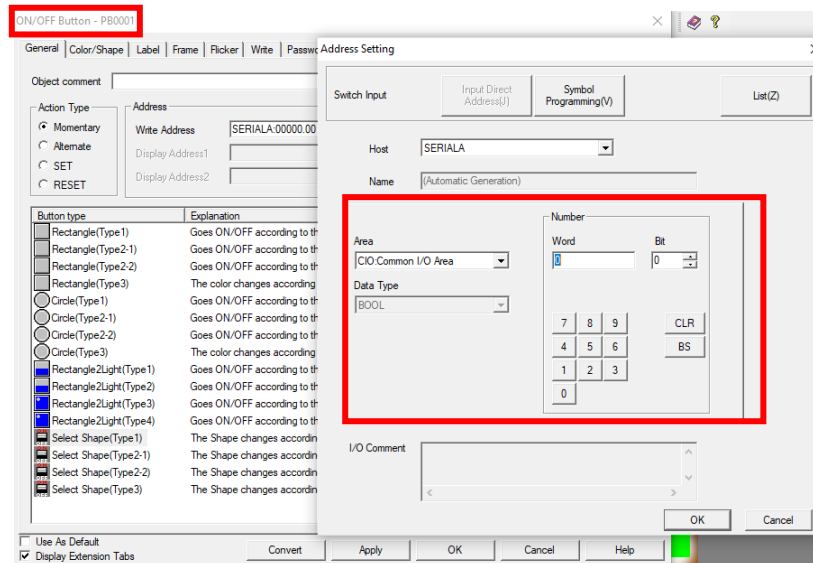
Gambar 3.11 *Functional Object* pada CX-Designer

Setelah memasukkan objek pada layar, selanjutnya mengubah desain objek tersebut sesuai kebutuhan agar secara tampilan mudah dimengerti oleh operator. Langkah-langkah dalam mengubah desain objek yaitu dengan melakukan *double click* pada objek yang dipilih → *Color/Shape* → *Select Shape1* dan *Shape2* lalu pilih shape/bentuk sesuai kebutuhan dan klik OK. Berikut tampilan mengubah desain objek dapat dilihat pada Gambar 3.12.

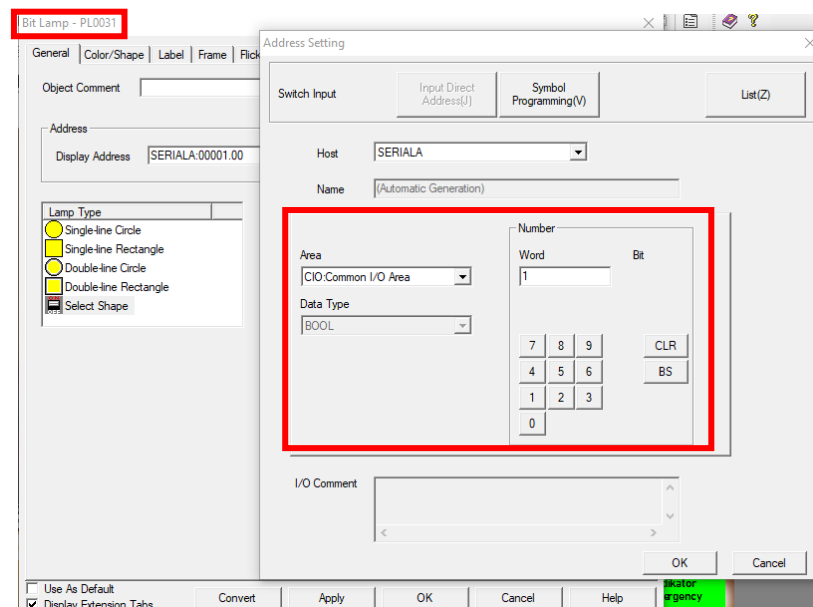


Gambar 3.12 Mengubah objek pada CX-Designer

Langkah selanjutnya yaitu dengan melakukan konfigurasi alamat *input* atau *output* yang disesuaikan dengan kontak penomoran IO pada CX-Programmer caranya dengan melakukan *double click* pada objek yang dipilih → *Write Address* → *Set1* → OK. Dalam mengatur alamat *input* diawali dengan alamat 0.00, sedangkan untuk alamat *output* diawali dengan alamat 1.00 dan seterusnya. Berikut tampilan mengubah alamat objek dapat dilihat pada Gambar 3.13 dan Gambar 3.14.



Gambar 3.13 Mengatur alamat *input*



Gambar 3.14 Mengatur alamat *output*

Setelah melakukan langkah-langkah diatas berikut adalah hasil tampilan HMI yang telah dibuat:

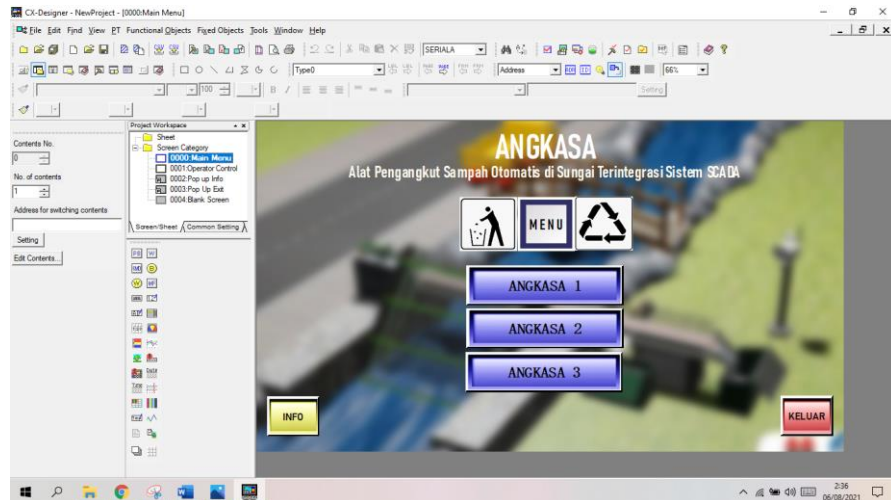
1. *Layar Main Menu*

Pada halaman layar *main menu* (Gambar 3.15) terdapat pilihan menu ANGKASA 1, ANGKASA 2, dan ANGKASA 3 yang berfungsi untuk masuk ke tampilan layar operator kontrol. Sedangkan menu INFO yaitu untuk mengetahui identitas pembuat desain HMI dan menu KELUAR untuk mengakhiri sistem HMI.

Dicky Rustandi, 2021

PERANCANGAN KONTROL PLC PADA ALAT PENGANGKUT SAMPAH OTOMATIS TERINTEGRASI SISTEM SCADA

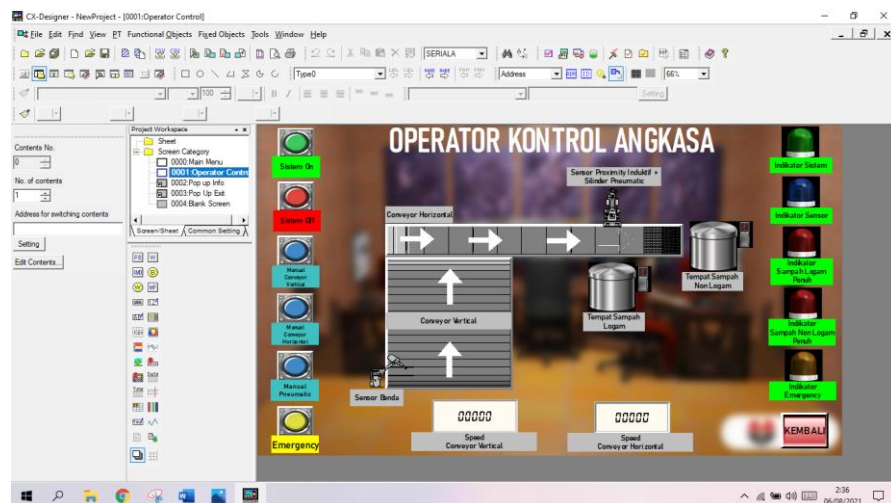
Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu



Gambar 3.15 Layar *Main Menu*

2. Layar Operator Kontrol

Pada halaman layar operator kontrol (3.16) merupakan tampilan desain gambaran dari alat pengangkut sampah otomatis di sungai. Pada layar ini terdapat beberapa tombol *input* untuk menjalankan fungsi pada program, *input* sensor, indikator kecepatan motor konveyor, serta *output* indikator lampu ketika program sedang berjalan.

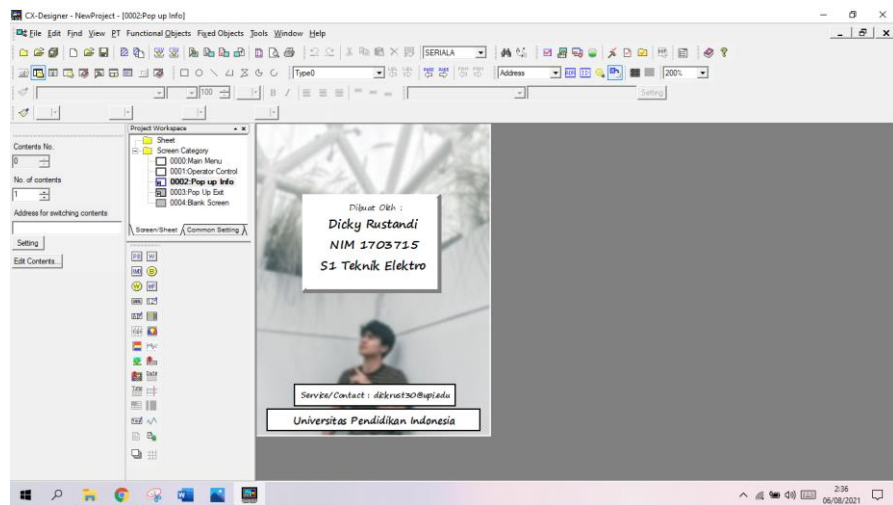


Gambar 3.16 Layar Operator Kontrol

3. Layar *Pop Up Info*

Pada halaman layar *pop up info* (Gambar 3.17) berisikan informasi pembuat sistem kontrol ini, fungsinya apabila terjadi kerusakan atau keadaan darurat pada alat, maka operator bisa menghubungi kontak pembuat yang tertera

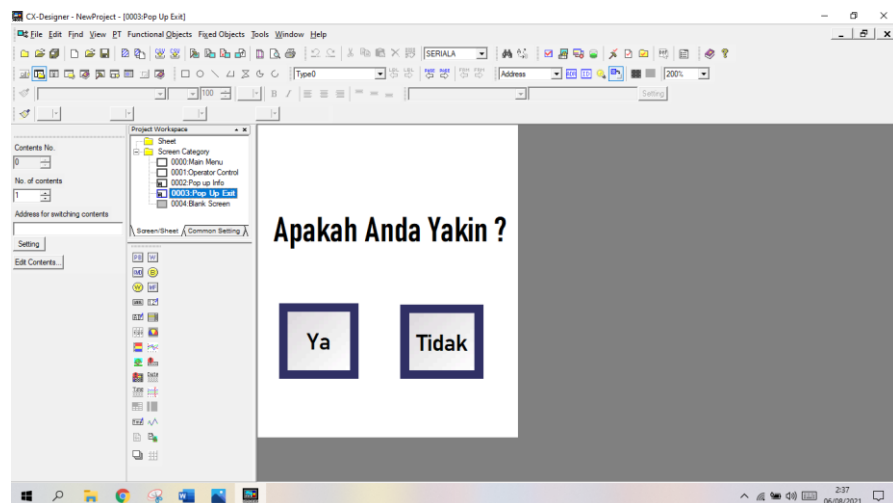
pada *pop up info*. Layar ini muncul ketika menekan menu INFO pada layar *main menu*.



Gambar 3.17 Layar *Pop Up Info*

4. Layar *Pop Up Exit*

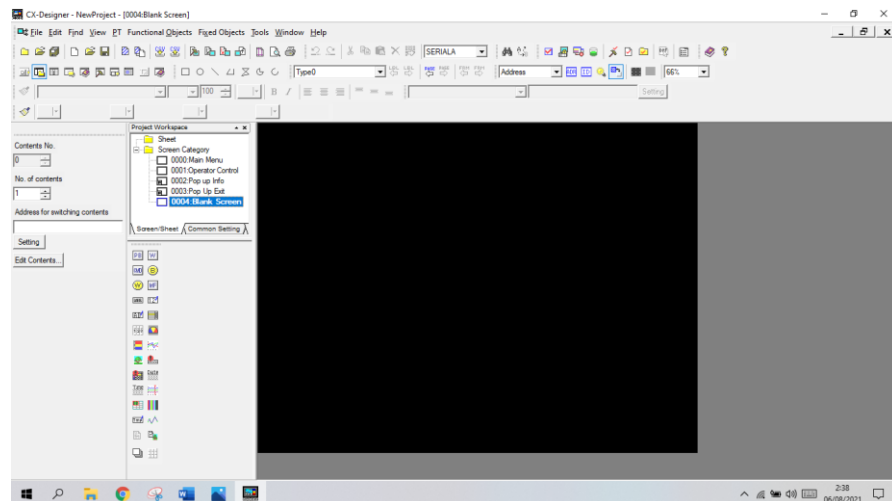
Pada halaman layar *pop up exit* (Gambar 3.18) terdapat kondisi apabila pengguna ingin keluar dan menghentikan sistem pada HMI maka layar ini akan muncul sebagai konfirmasi ulang.



Gambar 3.18 Layar *Pop Up Exit*

5. Layar *Blank Screen*

Pada halaman layar *blank screen* (Gambar 3.19) ini menandakan bahwa sistem sudah berhenti. Layar ini muncul ketika pengguna sudah mengkonfirmasi ingin keluar dari halaman.



Gambar 3.19 Layar *Blank Screen*