|  |  |  |
| --- | --- | --- |
| Logo Polman Bandung Original PNG Terbaru - Psikolif | **POLITEKNIK MANUFAKTUR BANDUNG**  Jalan Kanayakan 21 Tlp/Fax. +62 (022) 7312564 Bandung 40275  Telp. (022)2500241 Faks. (022)2502649 https://polman-bandung.ac.id/ |  |

**LAPORAN**

**AE**

**PRAKTIK PLC**





FOTO

MAHASISWA

**Muhammad Defval**

**1-AEC-2 / 223443040**

**PROGRAM STUDI TEKNOLOGI REKAYASA INFORMATIKA INDUSTRI**

**JURUSAN OTOMASI MANUFAKTUR DAN MEKATRONIKA**

**POLITEKNIK MANUFAKTUR BANDUNG**

**2024**

# KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat, karunia, dan hidayah-Nya sehingga kami dapat menyelesaikan praktikum PLC (Programmable Logic Control) ini dengan baik. Shalawat serta salam senantiasa tercurah kepada Nabi Muhammad SAW, yang telah membimbing umatnya ke jalan yang lurus.

Laporan praktikum ini disusun sebagai bagian dari tugas akademik dalam mata kuliah "Programmable Logic Control" pada semester ini. Kami ingin mengucapkan terima kasih yang sebesar-besarnya kepada dosen pengajar dan pembimbing mata kuliah yang telah memberikan arahan dan bimbingan sehingga kami dapat menyelesaikan praktikum ini dengan lancar.

Praktikum PLC ini bertujuan untuk memperdalam pemahaman kami terhadap konsep-konsep dasar pengendalian logika yang dapat diprogram, serta aplikasi dan implementasinya dalam dunia industri. Melalui praktikum ini, kami berharap dapat mengembangkan keterampilan dalam merancang, memprogram, dan menguji sistem kontrol yang menggunakan PLC.

Laporan ini merupakan hasil kerja keras dari diri kami sendiri. Kami menyadari bahwa dalam perjalanan menyelesaikan praktikum ini, terdapat berbagai hambatan dan tantangan yang harus kami hadapi. Namun, dengan semangat dan dedikasi yang tinggi, kami berhasil mengatasi setiap tantangan tersebut.

Kami juga menyadari bahwa laporan ini masih memiliki kekurangan dan ruang untuk perbaikan. Oleh karena itu, kami sangat mengharapkan masukan, kritik, dan saran yang membangun dari pembaca untuk perbaikan di masa mendatang.

Akhir kata, kami berharap laporan praktikum ini dapat memberikan manfaat dan wawasan yang lebih luas bagi pembaca yang tertarik dalam bidang pengendalian logika yang dapat diprogram.

Terima kasih.

|  |
| --- |
| Bandung,28 Februari 2024 |
| Muhammad Defval |

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc160052026)

[DAFTAR ISI ii](#_Toc160052027)

[BIODATA MAHASISWA iii](#_Toc160052028)

[PENDAHULUAN iv](#_Toc160052029)

[PRAKTIKUM 1 1-1](#_Toc160052030)

[1.1 Teori Dasar 1-1](#_Toc160052031)

[1.2 Alat Dan Bahan 1-2](#_Toc160052032)

[1.3 Petunjuk Praktikum 1-2](#_Toc160052033)

[1.4 Praktikum 1-3](#_Toc160052034)

[1.4.1 Percobaan 1: Input Dan Output 1-3](#_Toc160052035)

[1.5 Soal Latihan 1-10](#_Toc160052036)

[2 PRAKTIKUM 2 INSTRUKSI DASAR PLC – TIMER 2-15](#_Toc160052037)

[2.1 Teori Dasar 2-15](#_Toc160052038)

[2.2 Alat Dan Bahan 2-16](#_Toc160052039)

[2.3 Petunjuk Praktikum 2-16](#_Toc160052040)

[2.4 Praktikum 2-17](#_Toc160052041)

[2.4.1 Percobaan 1: Timer 2-17](#_Toc160052042)

[2.5 Soal Latihan 2-17](#_Toc160052043)

[3 PRAKTIKUM 3 INSTRUKSI DASAR PLC – COUNTER 3-1](#_Toc160052044)

[3.1 Teori Dasar 3-1](#_Toc160052045)

[3.2 Alat Dan Bahan 3-2](#_Toc160052046)

[3.3 Petunjuk Praktikum 3-2](#_Toc160052047)

[3.4 Praktikum 3-3](#_Toc160052048)

[3.4.1 Percobaan 1: Counter 3-3](#_Toc160052049)

[3.4.2 Percobaan 2: Counter Up/Down 3-3](#_Toc160052050)

[3.5 Soal Latihan 3-5](#_Toc160052051)

[4 PRAKTIKUM 4 INSTRUKSI DASAR PLC – DIFU & DIFD 4-1](#_Toc160052052)

[4.1 Teori Dasar 4-1](#_Toc160052053)

[4.2 Alat Dan Bahan 4-2](#_Toc160052054)

[4.3 Petunjuk Praktikum 4-2](#_Toc160052055)

[4.4 Praktikum 4-3](#_Toc160052056)

[4.4.1 Percobaan 1: DIFU & DIFD 4-3](#_Toc160052057)

[4.5 Soal Latihan 4-4](#_Toc160052058)

[5 PRAKTIKUM 5 INSTRUKSI DASAR PLC – SET & RESET 5-7](#_Toc160052059)

[5.1 Teori Dasar 5-7](#_Toc160052060)

[5.2 Alat Dan Bahan 5-7](#_Toc160052061)

[5.3 Petunjuk Praktikum 5-8](#_Toc160052062)

[5.4 Praktikum 5-9](#_Toc160052063)

[5.4.1 Percobaan 1: SET & RESET 5-9](#_Toc160052064)

[5.5 Soal Latihan 5-9](#_Toc160052065)

[6 PRAKTIKUM 6 INSTRUKSI DASAR PLC – PERBANDINGAN, PERPINDAHAN, DAN PERGESERAN DATA 6-15](#_Toc160052066)

[6.1 Teori Dasar 6-15](#_Toc160052067)

[6.2 Alat Dan Bahan 6-15](#_Toc160052068)

[6.3 Petunjuk Praktikum 6-15](#_Toc160052069)

[6.4 Praktikum 6-16](#_Toc160052070)

[6.4.1 Percobaan 1: Compare 6-16](#_Toc160052071)

[6.4.2 Percobaan 2: Mov 6-17](#_Toc160052072)

[6.4.3 Percobaan 3: SET 6-18](#_Toc160052073)

[6.4.4 Percobaan 4: ++B 6-18](#_Toc160052074)

[6.4.5 Percobaan 5: --B 6-19](#_Toc160052075)

[6.5 Soal Latihan 6-19](#_Toc160052076)

[7 PRAKTIKUM 7 INTRUKSI DASAR PLC-ARITMATIKA 7-1](#_Toc160052077)

[7.1 Teori Dasar 7-1](#_Toc160052078)

[7.2 Alat Dan Bahan 7-3](#_Toc160052079)

[7.3 Petunjuk Praktikum 7-3](#_Toc160052080)

[7.4 Praktikum 7-4](#_Toc160052081)

[7.4.1 Percobaan 1: +B 7-4](#_Toc160052082)

[7.4.2 Percobaan 1: -B 7-4](#_Toc160052083)

[7.4.3 Percobaan 1: \*B 7-5](#_Toc160052084)

[7.5 Soal Latihan 7-6](#_Toc160052085)

[8 PENUTUP 8-1](#_Toc160052086)

[9 DAFTAR PUSTAKA v](#_Toc160052087)

# BIODATA MAHASISWA

Nama : Muhammad Defval

NIM : 223443040

Tanggal dan Tempat Lahir : Cimahi 24 Maret 2005

Agama : Islam

Program Studi : Teknologi Rekayasa Informatika Industri

Jurusan : Teknik Otomasi Manufaktur Dan Mekatronika

Alamat : Jl.Kerkof Kihapit Timur

Nomor Telepon : 082120100897

Email : defvalnahdan@gmail.com

# PENDAHULUAN

1. Definisi

PLC adalah sebuah perangkat keras elektronik yang dirancang untuk mengendalikan berbagai jenis proses dan mesin di lingkungan industri. Ini memungkinkan pengguna untuk memprogram operasi dan logika kontrol melalui software.

1. Komponen Utama

* CPU (Central Processing Unit): Unit pemrosesan utama yang menjalankan program PLC.
* Input/Output (I/O) Modules: Menghubungkan PLC dengan perangkat input (sensor) dan output (aktuator) di lapangan.
* Memory: Menyimpan program dan data yang diperlukan untuk kontrol.
* Power Supply: Memberikan daya listrik ke semua komponen PLC.

1. Fungsi Utama

* Pengolahan Sinyal Input: Menerima sinyal input dari sensor atau perangkat lapangan.
* Eksekusi Program: Menjalankan program kontrol yang telah diprogram oleh pengguna.
* Mengendalikan Output: Mengirimkan sinyal output ke aktuator atau perangkat lapangan.
* Komunikasi: Berkomunikasi dengan perangkat eksternal seperti komputer, HMI (Human-Machine Interface), atau perangkat lain.

1. Programmable

PLC dapat diprogram ulang untuk mengubah atau menyesuaikan fungsi kontrol tanpa mengganti perangkat kerasnya.Program umumnya dibuat menggunakan bahasa pemrograman grafis atau teks yang disesuaikan dengan vendor tertentu.

1. Aplikasi

Digunakan dalam otomasi industri untuk mengontrol mesin, sistem produksi, dan proses lainnya.Ditemukan dalam berbagai industri seperti manufaktur, energi, kendali bangunan, dan lainnya.

1. Struktur Program PLC:

* Rung: Program PLC terstruktur dalam rung-rung atau baris-baris logika. Setiap rung mewakili suatu kondisi atau aksi yang akan diambil.
* Instruksi Logika: Program PLC menggunakan instruksi logika seperti AND, OR, NOT untuk membentuk logika kontrol. Ini menciptakan alur logika yang diterapkan pada input dan output.

1. Bahasa Pemrograman:

* Ladder Logic: Mirip dengan diagram rangkaian listrik kawat keras, ini adalah bahasa pemrograman grafis yang umum digunakan dalam PLC.
* Structured Text, Function Block Diagram, Sequential Function Chart: Beberapa PLC mendukung bahasa pemrograman lainnya.

1. Input dan Output:

* Input Modules: Mengonversi sinyal dari perangkat lapangan menjadi format yang dapat dipahami oleh PLC.
* Output Modules: Mengubah hasil logika kontrol menjadi sinyal keluaran untuk mengendalikan perangkat lapangan.

# PRAKTIKUM 1

**INSTRUKSI DASAR PLC – INPUT OUTPUT**

**Tujuan :**

1. Dapat memahami konsep input dan output pada PLC
2. Belajar menggunakan instruksi dasar PLC seperti instruksi logika (AND, OR, NOT)
3. Melakukan simulasi program yang telah dibuat untuk memastikan bahwa logika

kontrol berfungsi dengan benar

## Teori Dasar

**Instruksi Input**

Instruksi input merupakan instruksi yang membentuk logika, kondisi *onn/off* input menunjukkan kondisi *on/off* bit yang merupakan alamat input tersebut. Beberapa jenis *input* adalah:

**Load: LD**

*Input* dilambangkan dengan perintah LD, akan *on* atau *off* sesuai dengan kondisi *input*tersebut. Simbol instruksi LD adalah:



**Load Not: LD NOT**

*Input* dilambangkan dengan perintah LD NOT, akan *on* atau *off* sesuai dengan kondisi *input* tersebut. Perintah LD NOT mempunyai kondisi yang berkebalikan dengan perintah LD. Simbol instruksi LD NOT adalah:



**AND** dan **AND NOT: AND** dan **AND NOT**

Menggabungkan dua atau lebih *input* secara seri. Dapat divariasikan dengan perintah AND NOT.

**OR** dan **OR NOT: OR** dan **OR NOT**

Menggabungkan dua atau lebih *input* secara paralel. Dapat divariasikan dengan perintah OR NOT.

**Instruksi *Output***

Adalah instruksi yang merupakan hasil dari proses logika dari *input*-*input* yang ada,dan akan menjadikan *bit*-*bit* tertentu yang merupakan alamat *output* akan *on* atau *off*.

**OUTPUT: OUT**

Instruksi ini akan mejadikan sebuah *bit* menjadi *on* selama *input* diaktifkan. Simbol instruksiOUT sebagai berikut:



**OUTPUT NOT: OUT NOT**

Instruksi ini akan menjadikan sebuah *bit* menjadi *off* selama *input* diaktifkan. Simbol instruksiOUT NOT sebagai berikut:

## Alat Dan Bahan

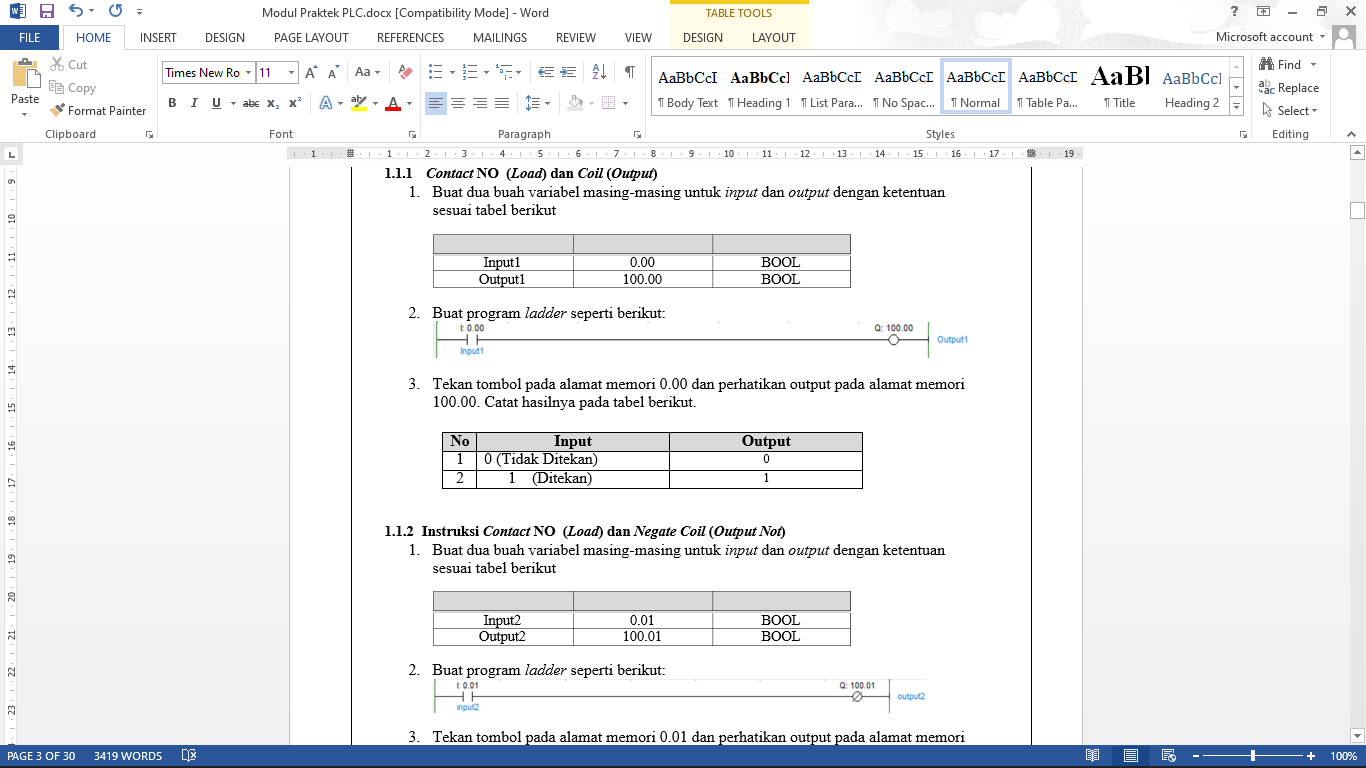
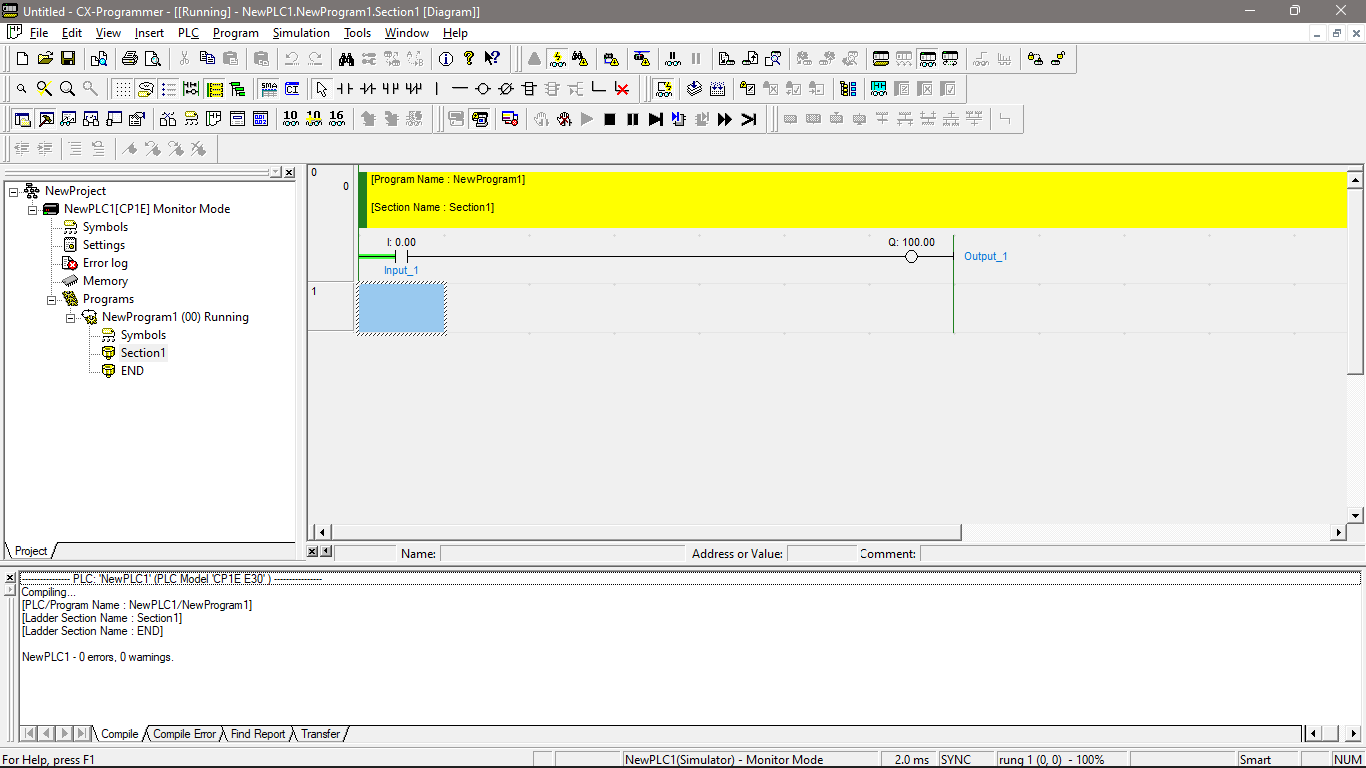
* PLC Omron CP1E-N30
* Laptop
* Kabel jumper
* Kabel USB

## Petunjuk Praktikum

1. Pertama yaitu download aplikasi CXONE\_v4.40 terlebih dahulu. Ikutilah langkah-langkah penginstalannya.
2. Setelah selesai terinstall,buka lah aplikasinya. Lalu kita pilih file lalu *New...* , isi device name dan device type sesuai PLC yang anda punya. Disini saya menggunakan PLC dengan device type CP1E-N30.
3. Jika sudah maka tekan OK barulah kita bisa membuat rangkaiannya.
4. Pasang PLC pada panel kontrol atau lokasi yang sesuai.
5. Sambungkan PLC ke sumber daya listrik sesuai dengan spesifikasi produsen.
6. Hubungkan laptop dengan PLC menggunakan kabel yang sesuai. Pastikan bahwa koneksi antara laptop dan PLC berfungsi dengan baik.
7. Pasang modul input/output ke slot yang sesuai di PLC.
8. Hubungkan perangkat input dan output di lapangan ke modul input/output.
9. Gunakan fasilitas simulasi jika tersedia untuk menguji program tanpa perlu terhubung ke PLC fisik.
10. Simulasikan kondisi input dan periksa respons output sesuai dengan logika program.
11. Setelah pemrograman dan simulasi selesai, transfer program ke memori PLC menggunakan koneksi laptop. Pastikan transfer program berjalan tanpa kesalahan.
12. Monitor operasi PLC melalui laptop untuk memastikan bahwa program berjalan sesuai dengan harapan.
13. Gunakan fasilitas debugging untuk mengidentifikasi dan memperbaiki kesalahan dalam program.

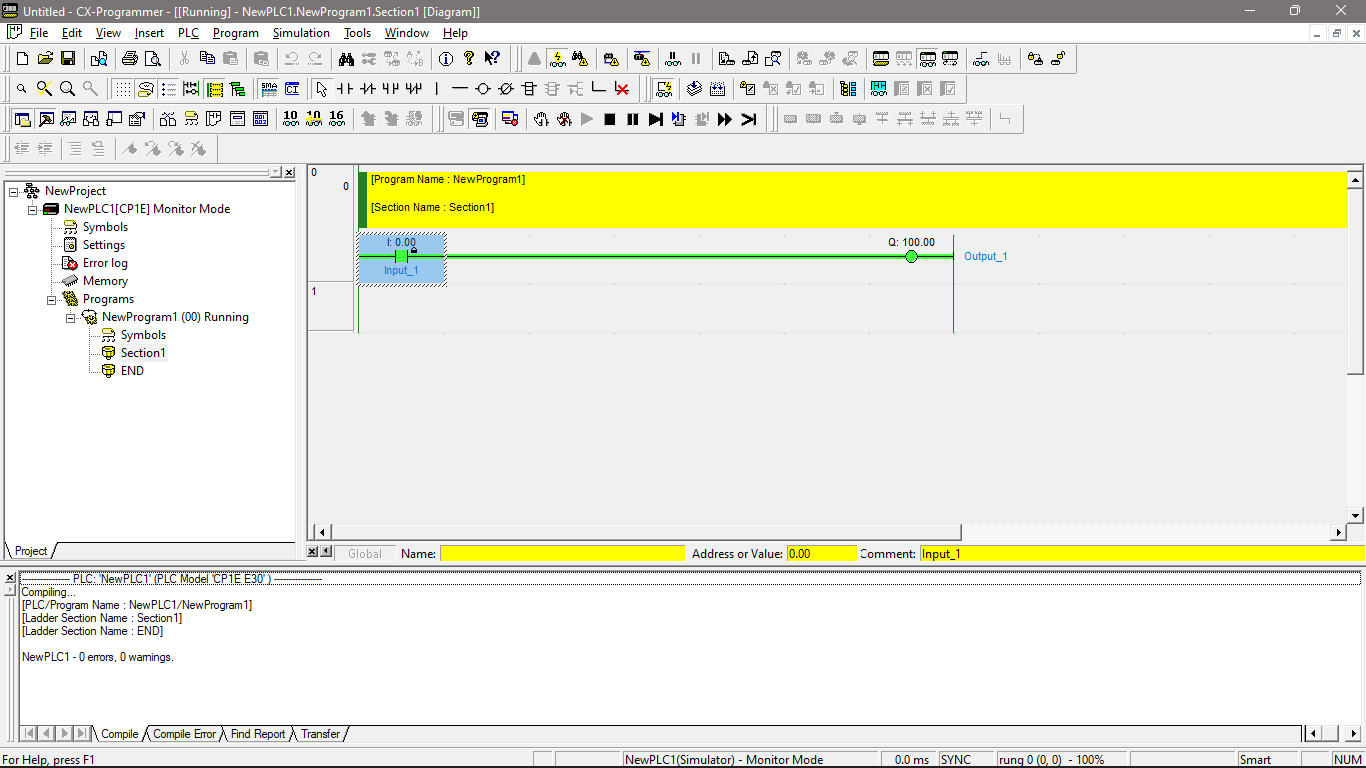
## Praktikum

### Percobaan 1: Input Dan Output

**Contact NO (Load) dan Coil (Output)**

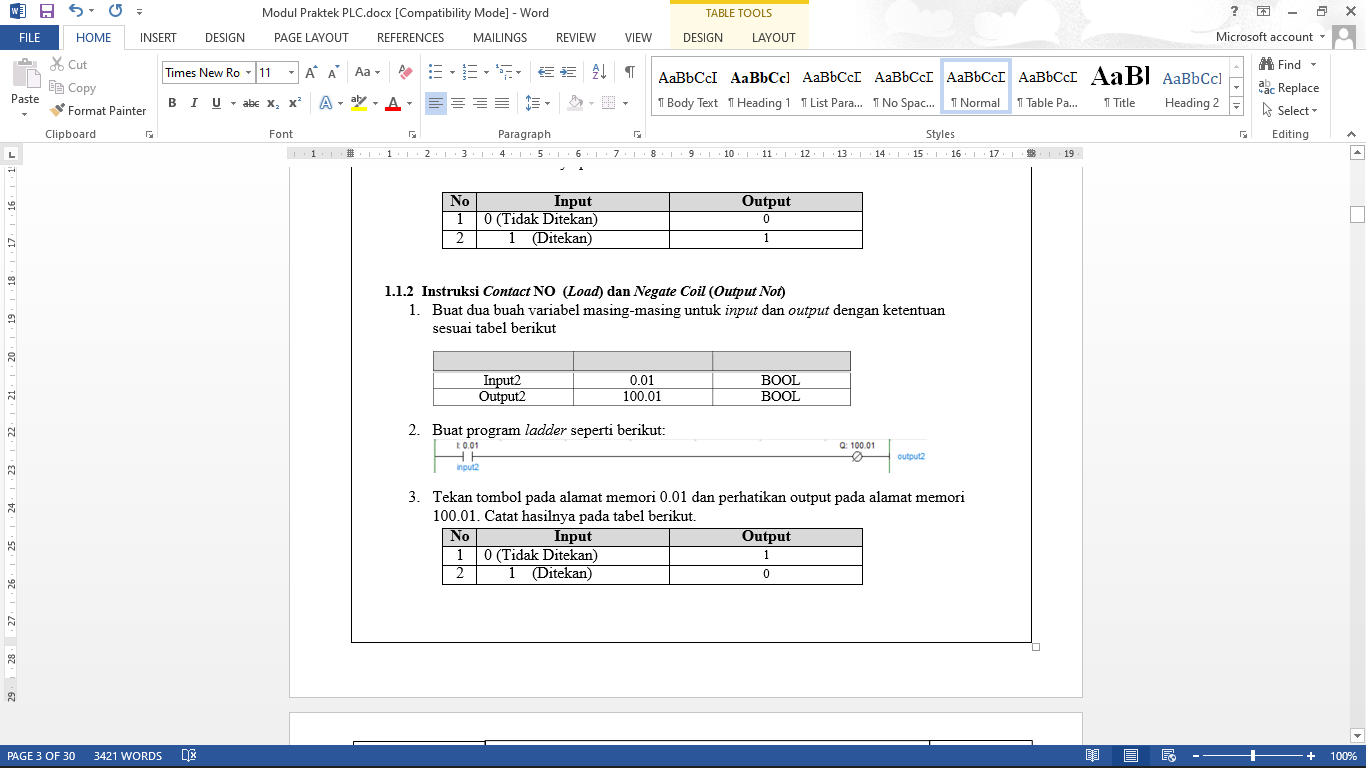
**Hasil dan Analisis**

Dari percobaan diatas, saat menekan tombol dengan alamat 0.00 maka lampu dengan alamat 100.00 akan menyala



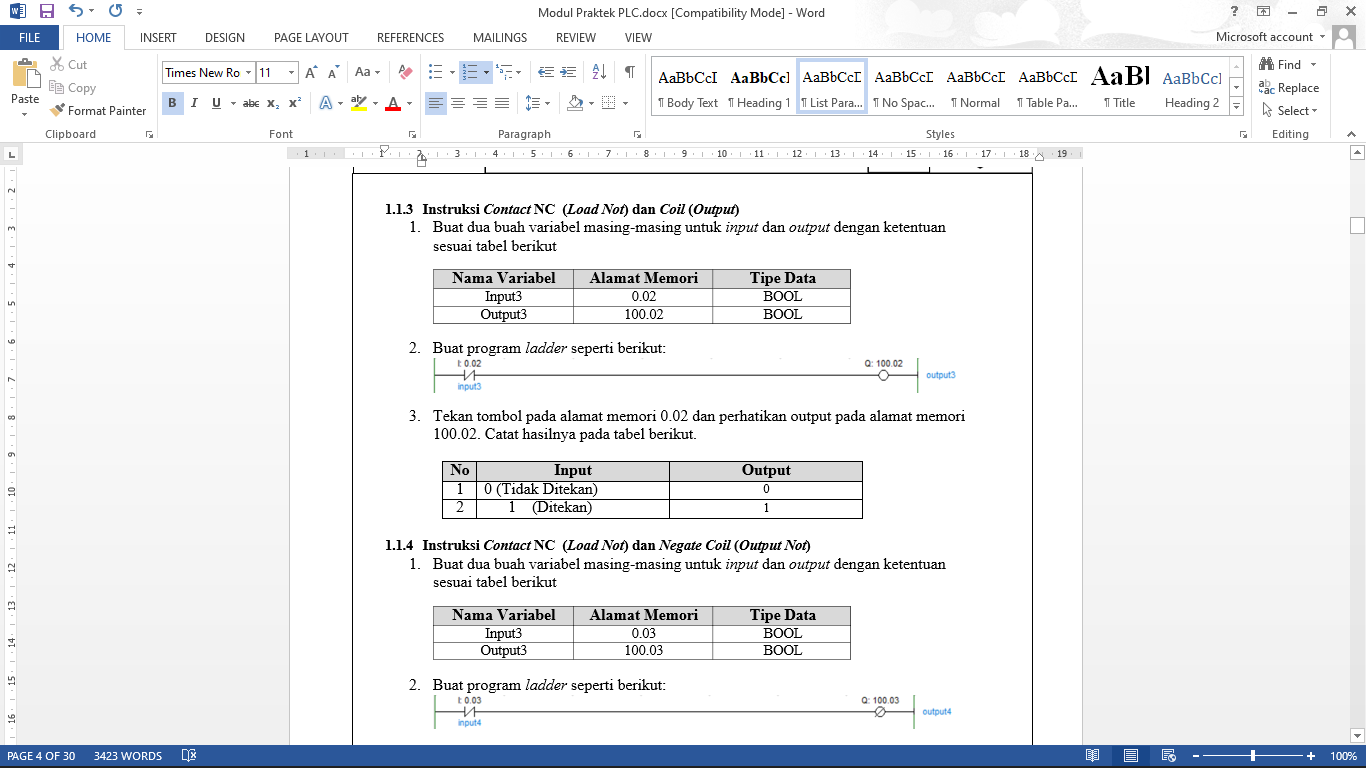
** Instruksi Contact NO (Load) dan Negate Coil (Output Not)**





**Hasil dan Analisis**

Dari percobaan yang saya coba, saat program dijalankan maka output dengan alamat 100.01 otomatis akan menyala.

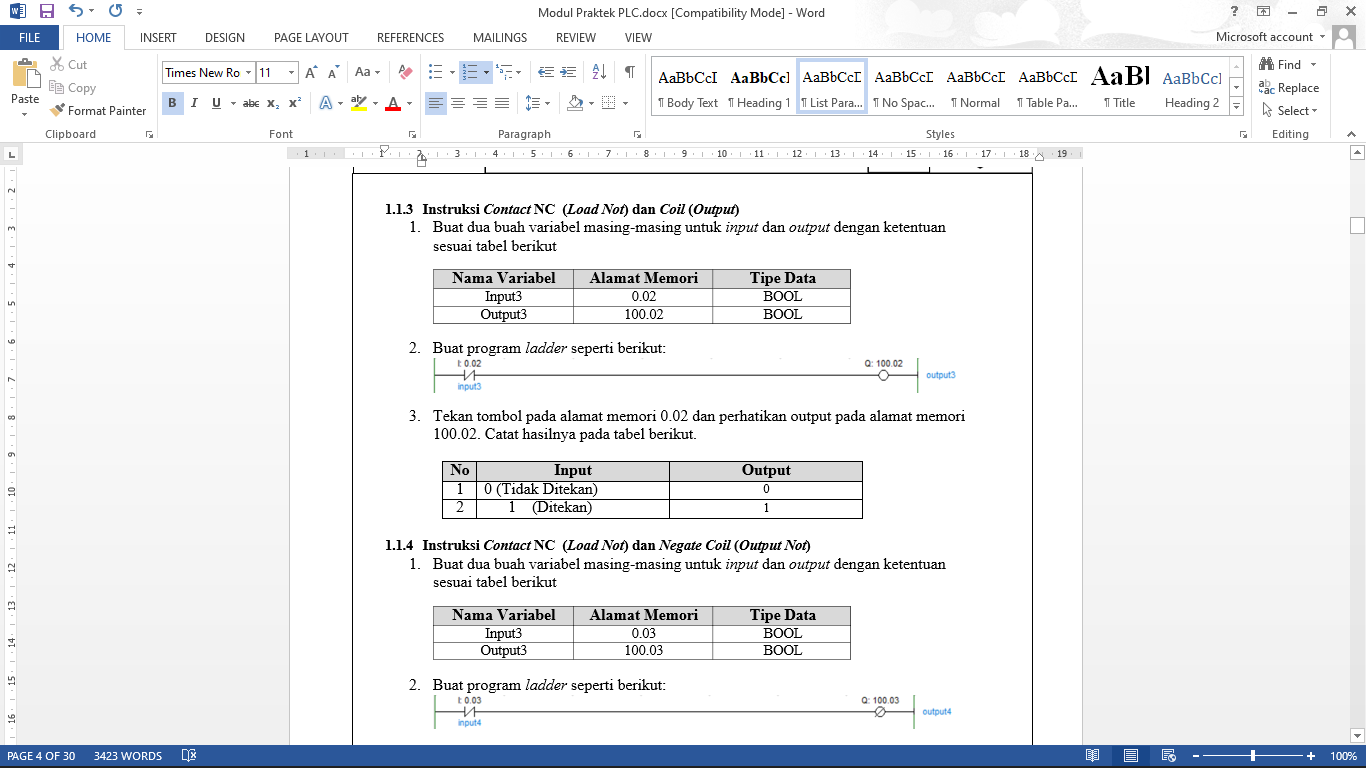
**Instruksi Contact NC (Load Not) dan Coil (Output)**

**Hasil dan Analisis**

Dari percobaan yang saya coba, saat program lalu input 3 ditekan maka output dengan alamat 100.02 otomatis akan menyala.

**Instruksi Contact NC (Load Not) dan Negate Coil (Output Not)**



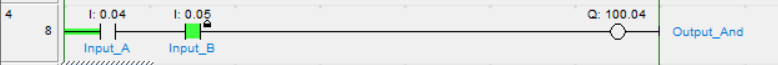


**Hasil dan Analisis**

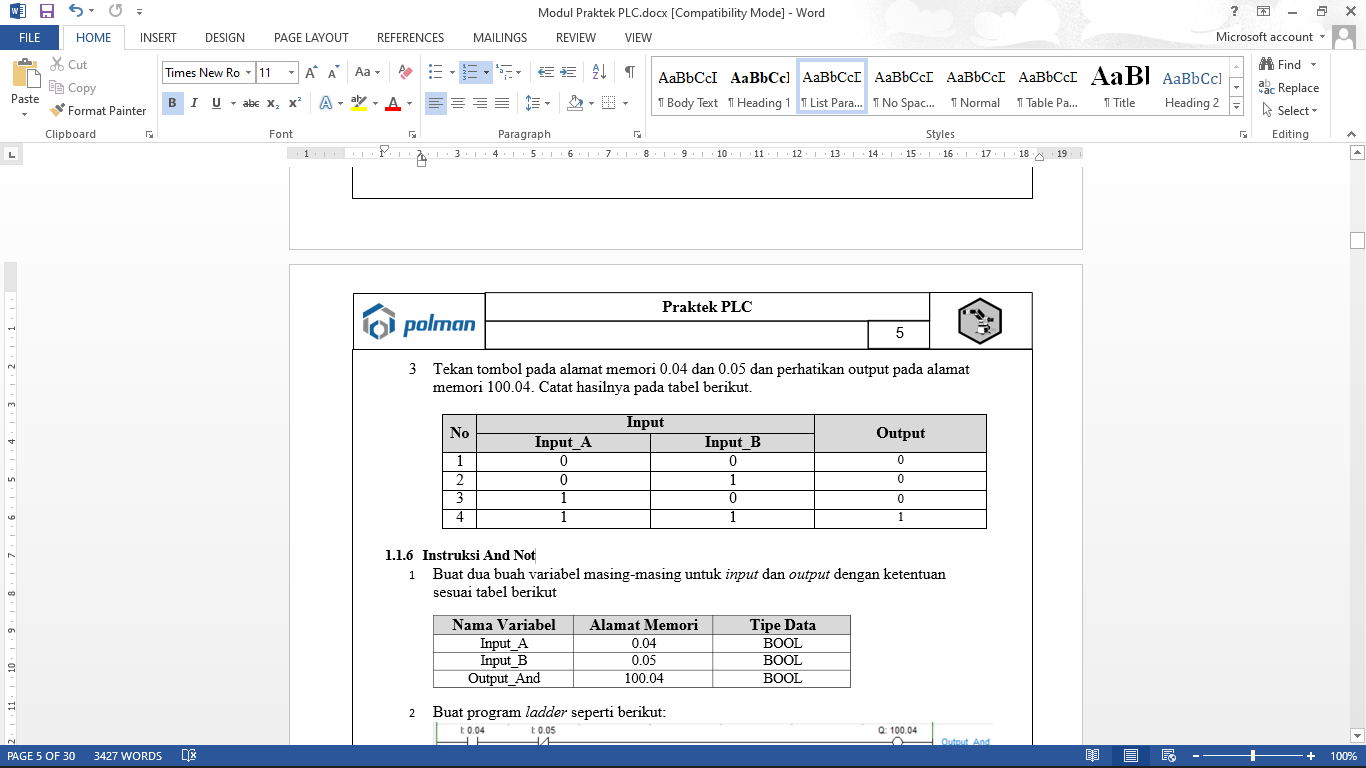
Dari percobaan yang saya coba, saat program dijalankan maka output dengan alamat 100.03 mula-mula dalam posisi mati.Tetapi ketika input 4 ditekan maka output 100.03 akan menyala

**Instruksi And**

Ketika kondisi 0 0

Ketika kondisi 0 1

Ketika kondisi 1 0

Ketika Kondisi 1 1

**Hasil dan Analisis**

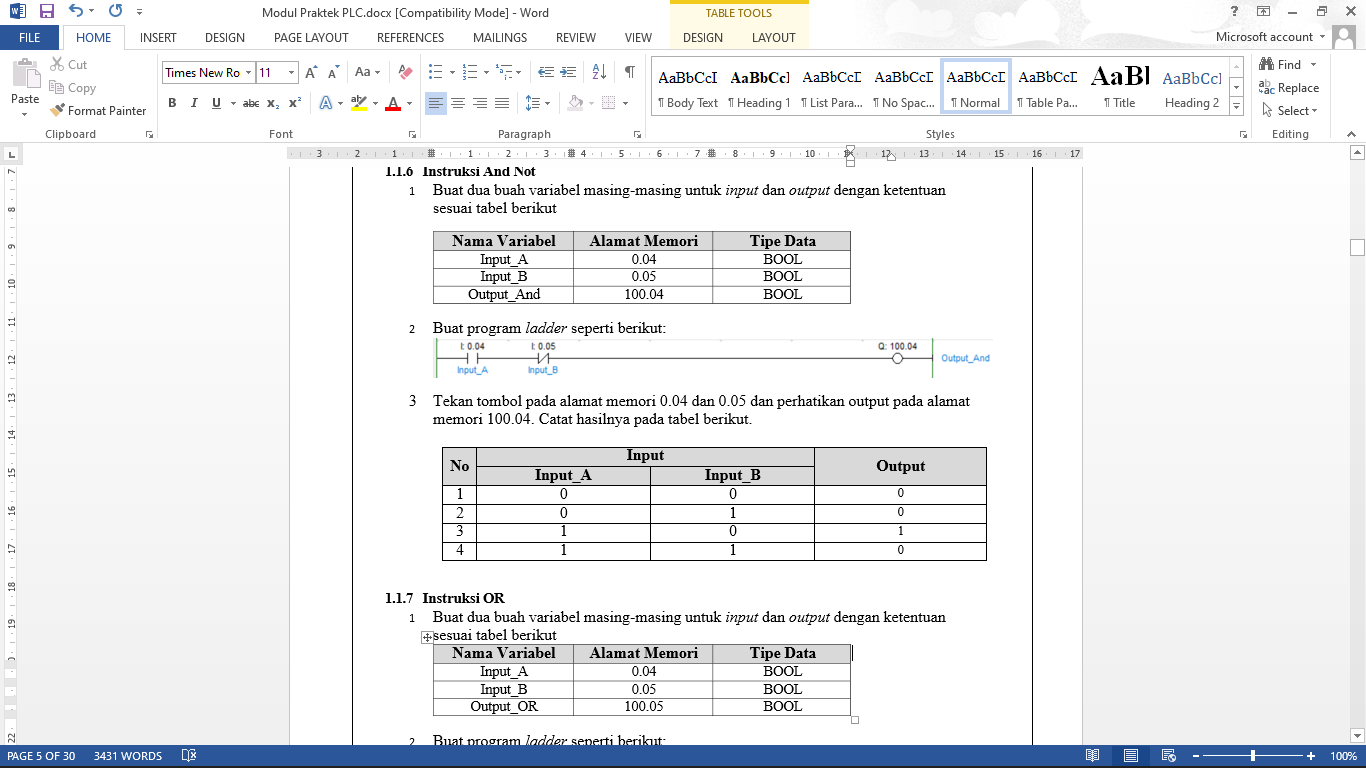
* Ketika input B ditekan maka output 100.04 tidak menyala
* Ketika input A ditekan maka output 100.04 tidak menyala
* Ketika input B dan input A ditekan maka output 100.04 menyala

**Instruksi And Not**

Ketika kondisi 0 0

Ketika kondisi 0 1

Ketika kondisi 1 0

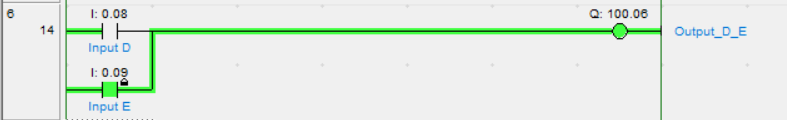
Ketika Kondisi 1 1

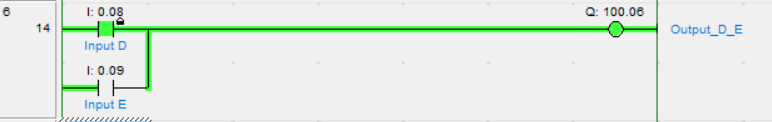
**Hasil dan Analisis**

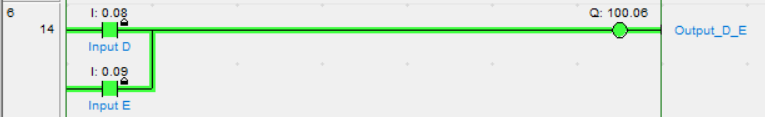
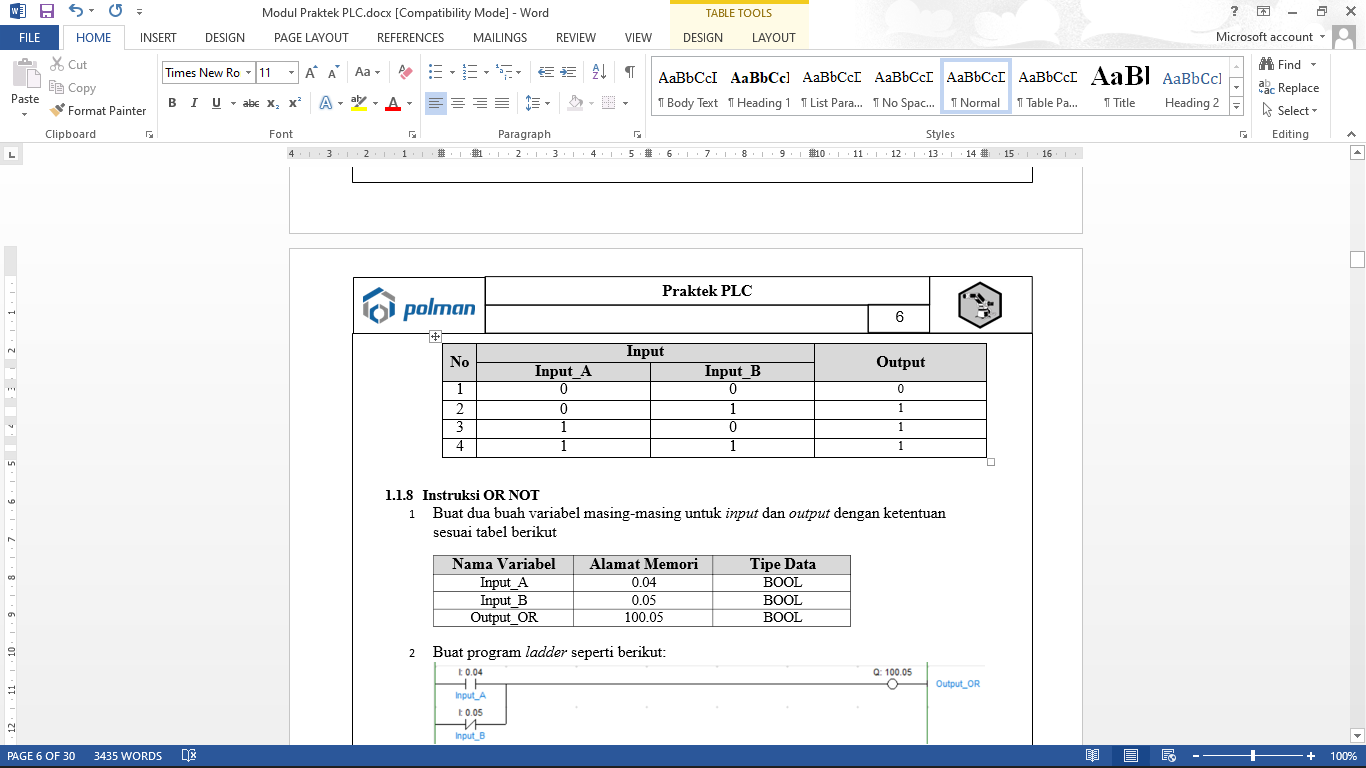
* Ketika input C ditekan maka output 100.05 tidak menyala
* Ketika input B ditekan maka output 100.05 menyala
* Ketika input B dan C ditekan maka output 100.05 tidak menyala

**Instruksi OR**

 Ketika kondisi 0 0

Ketika kondisi 0 1

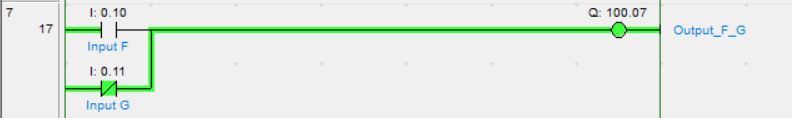
Ketika kondisi 1 0

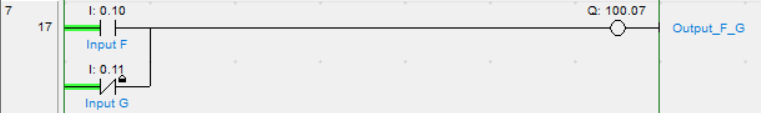
Ketika Kondisi 1 1

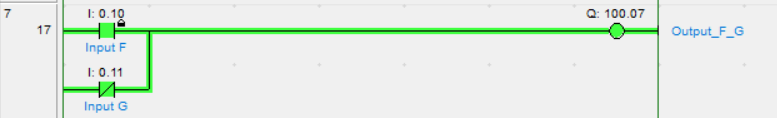
**Hasil dan Analisis**

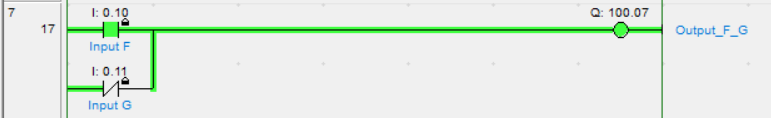
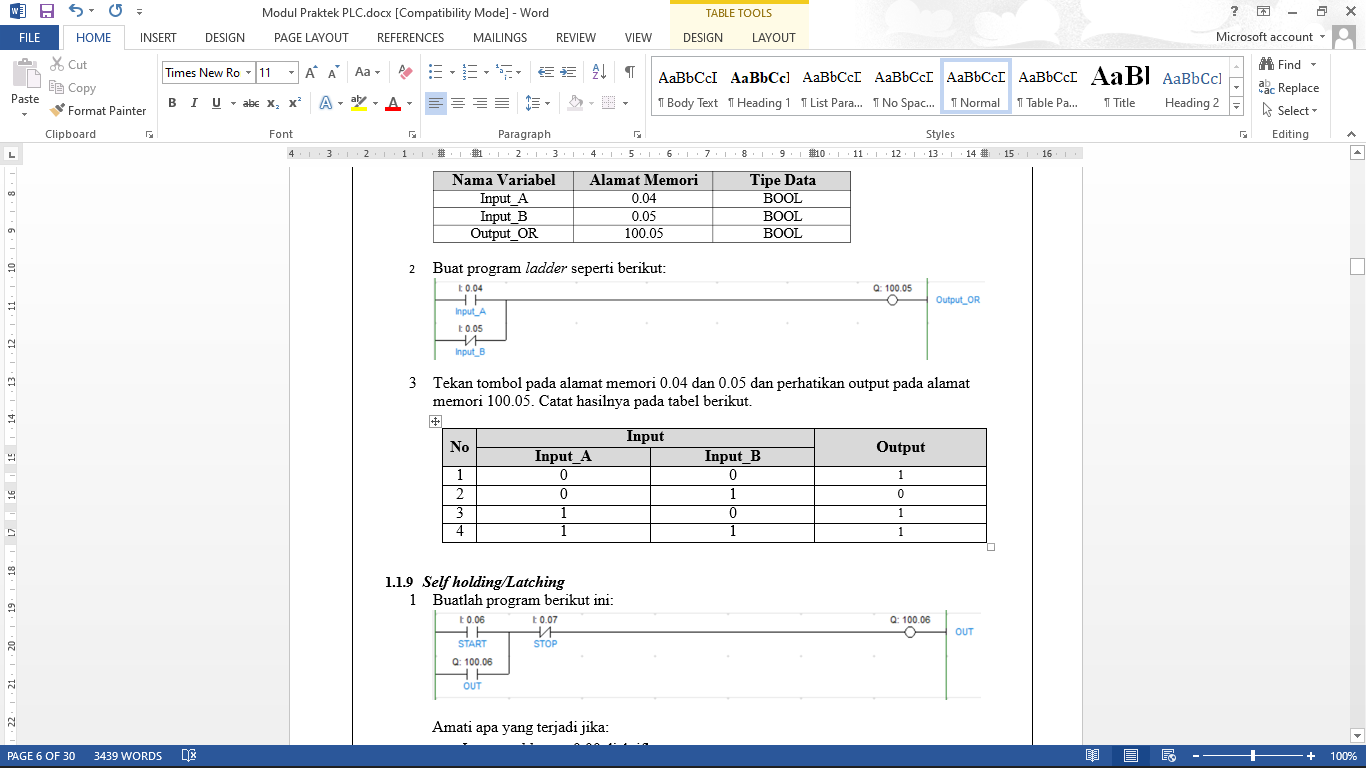
* Jika input E ditekan maka output 100.06 akan menyala
* Jika input D ditekan maka output 100.06 akan menyala
* Jika input E dan D ditekan maka output 100.06 akan menyala

**Instruksi OR NOT**

Ketika kondisi 0 0

 Ketika kondisi 0 1

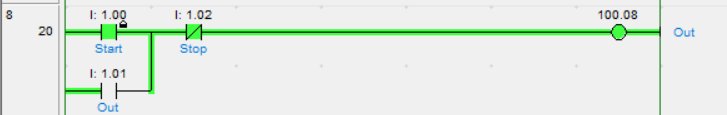
 Ketika kondisi 1 0

 Ketika Kondisi 1 1

**Hasil dan Analisis**

* Jika dijalankan maka output 100.07 akan menyala
* Jika input G ditekan maka output 100.07 tidak menyala
* Jika input F ditekan maka output 100.07 menyala
* Jika input F dan G ditekan maka output 100.07 menyala

**Self holding/Latching**

Ketika kondisi 0 0 (Input pushbutton 1.00 diaktifkan)

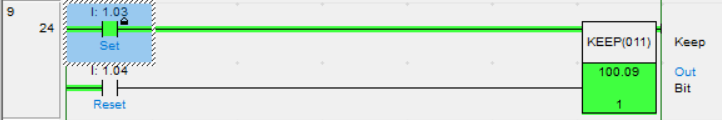
Ketika kondisi 0 1 (Input pushbutton 1.02 diaktifkan)



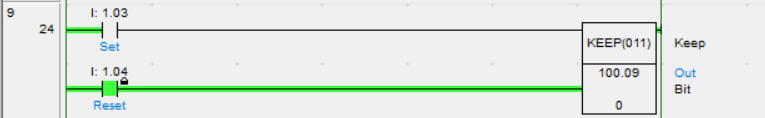
**Hasil dan Analisis**

* Ketika input start ditekan maka output 100.08 akan menyala
* Ketika input stop ditekan maka output 100.08 tidak menyala

**KEEP: KEEP(011)**

Input pushbutton 1.03 diaktifkan kemudian dinonaktifkan



Input pushbutton 0.04 diaktifkan kemudian dinonaktifkan



Kedua input tersebut diaktifkan bersamaan

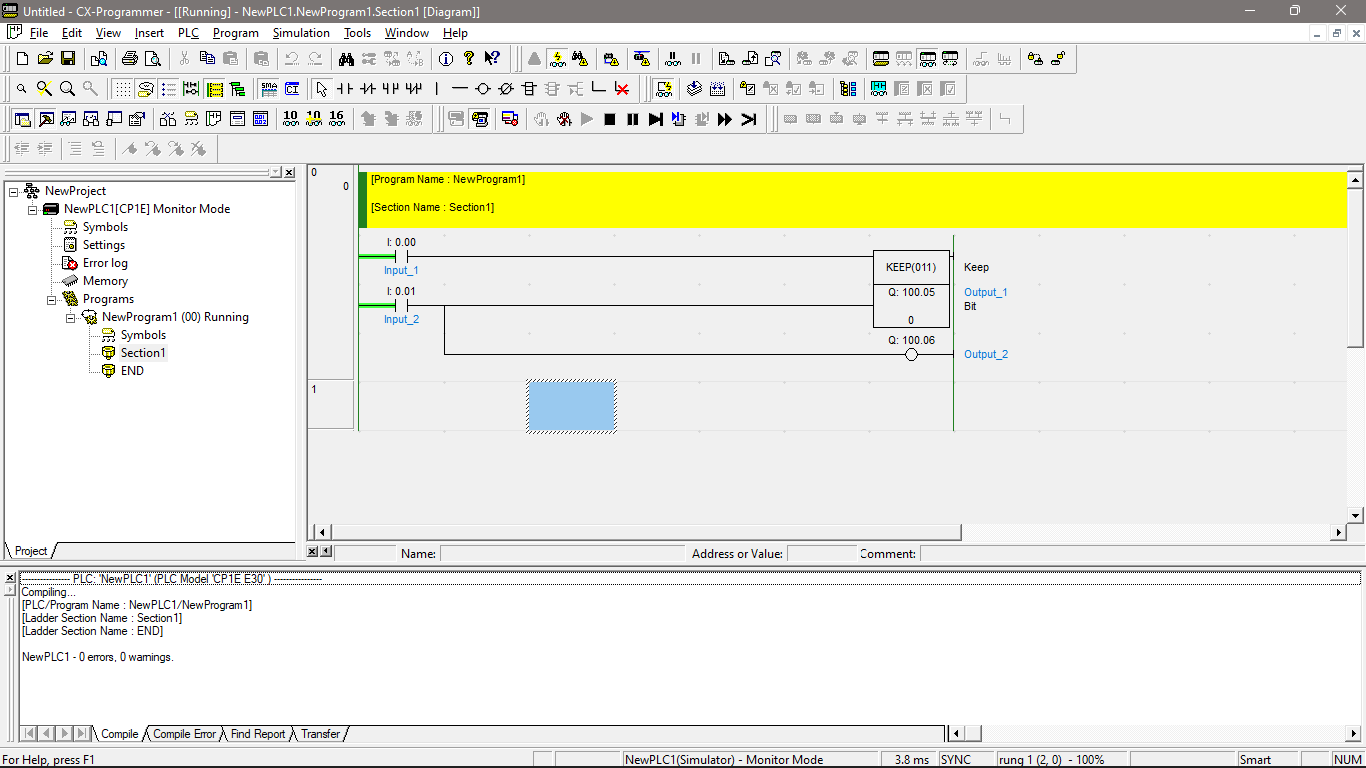
**Hasil dan Analisis**

* Ketika input pushbutton 1.03 diaktifkan kemudian dinonaktifkan maka output 100.09 akan menyala
* Ketika input pushbutton 0.04 diaktifkan kemudian dinonaktifkan maka output mati
* Ketika kedua input tersebut diaktifkan bersamaan maka output mati

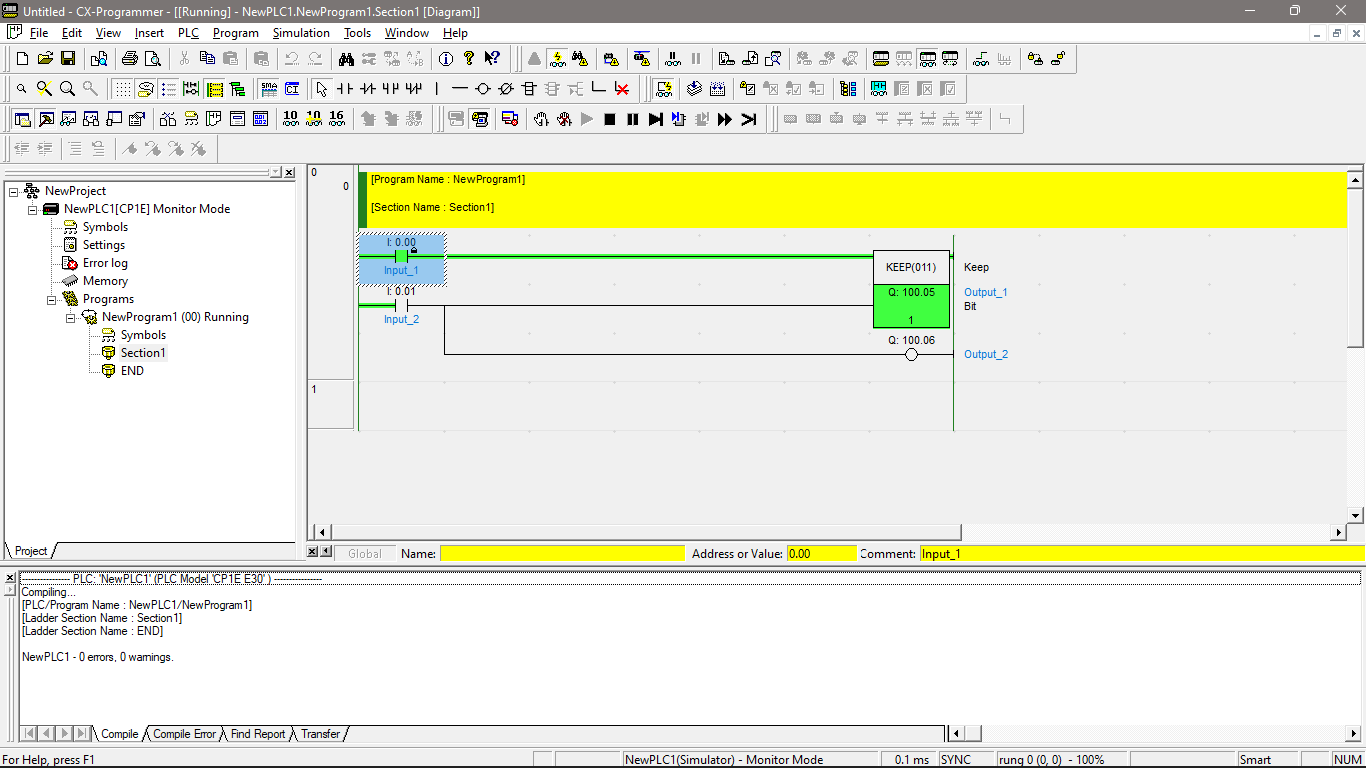
## Soal Latihan

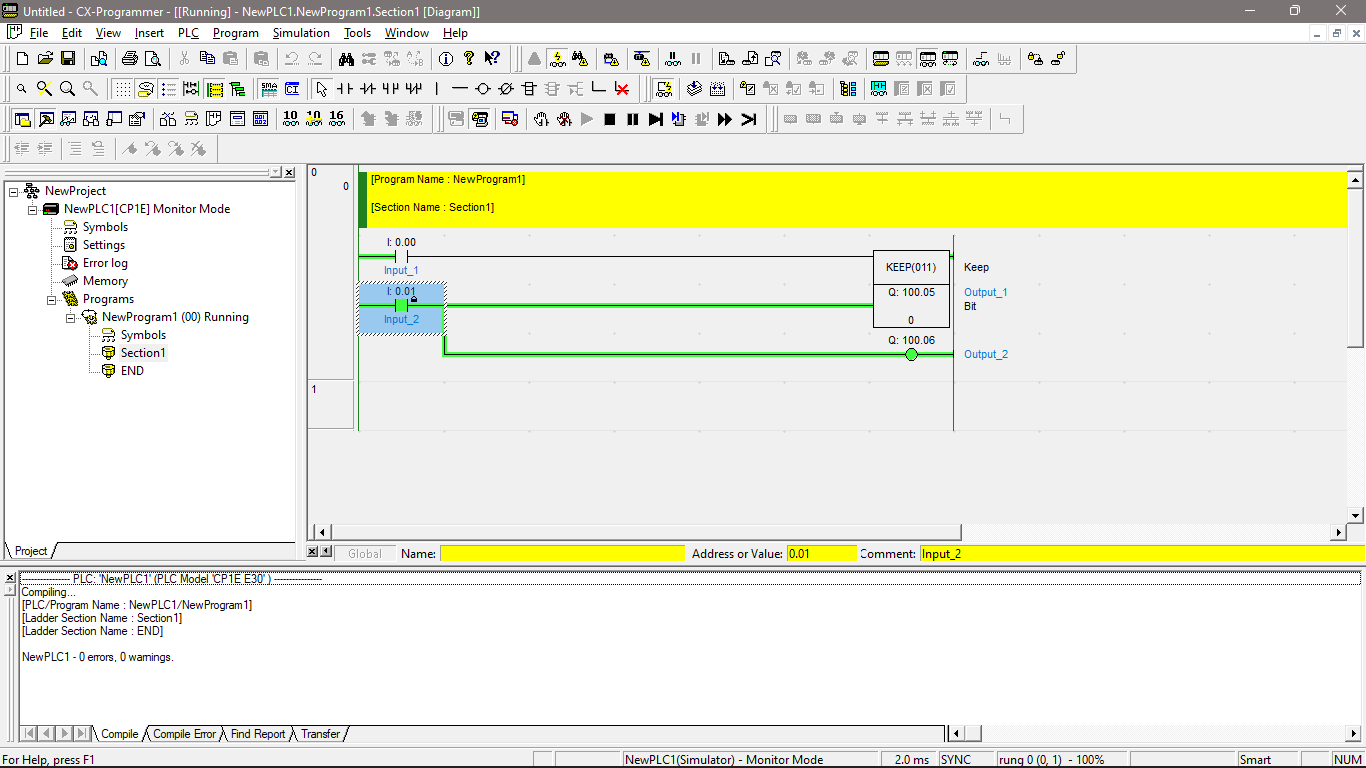
*Buatlah program untuk soal berikut:*

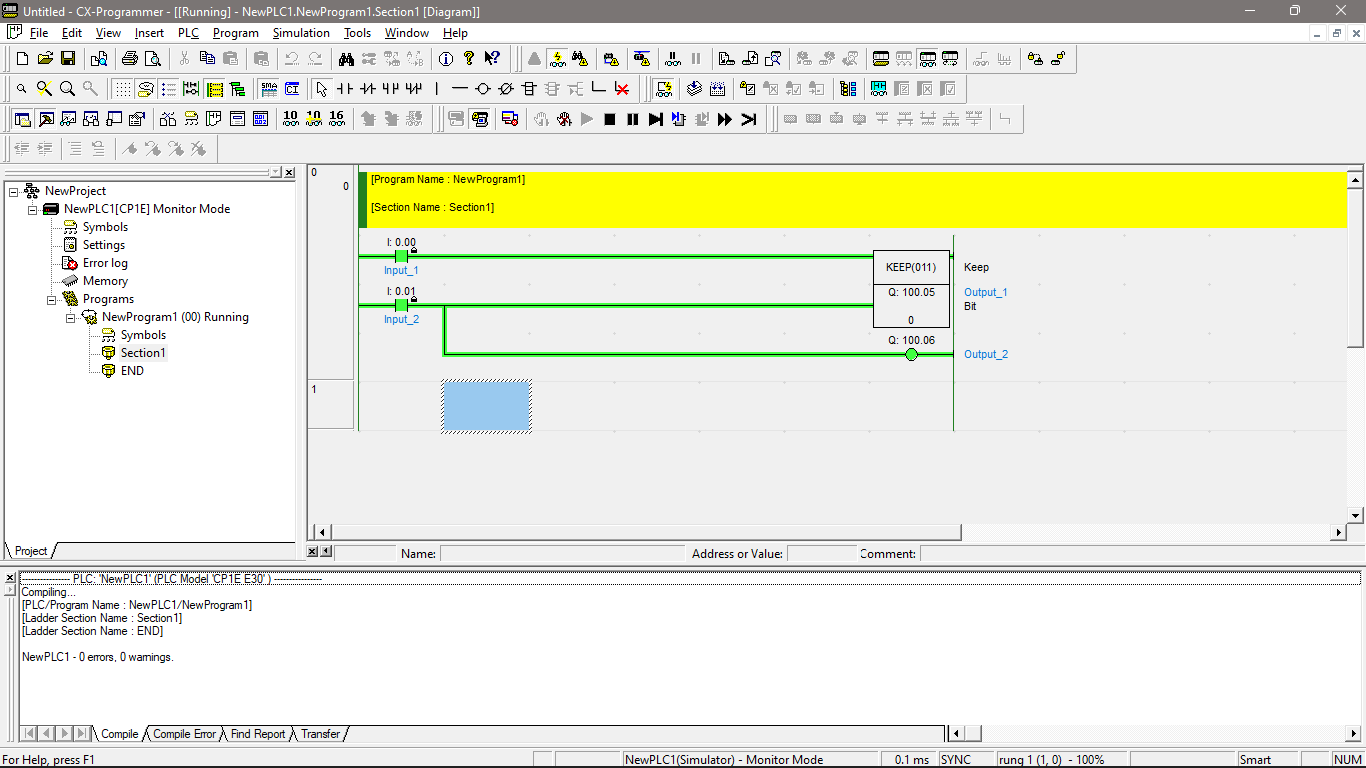
*Jika Input\_1 di-on-kan maka output\_1 akan on, demikian sebaliknya jika Input\_2 di-on-kan maka output\_2 akan on. Kedua output tidak boleh on bersamaan.*

Saat kondisi tidak dinyalakan

**Hasil dan Analisis**

Jika input-1 di on kan output-1 menyala

Jika input-2 di on kan output-2 menyala

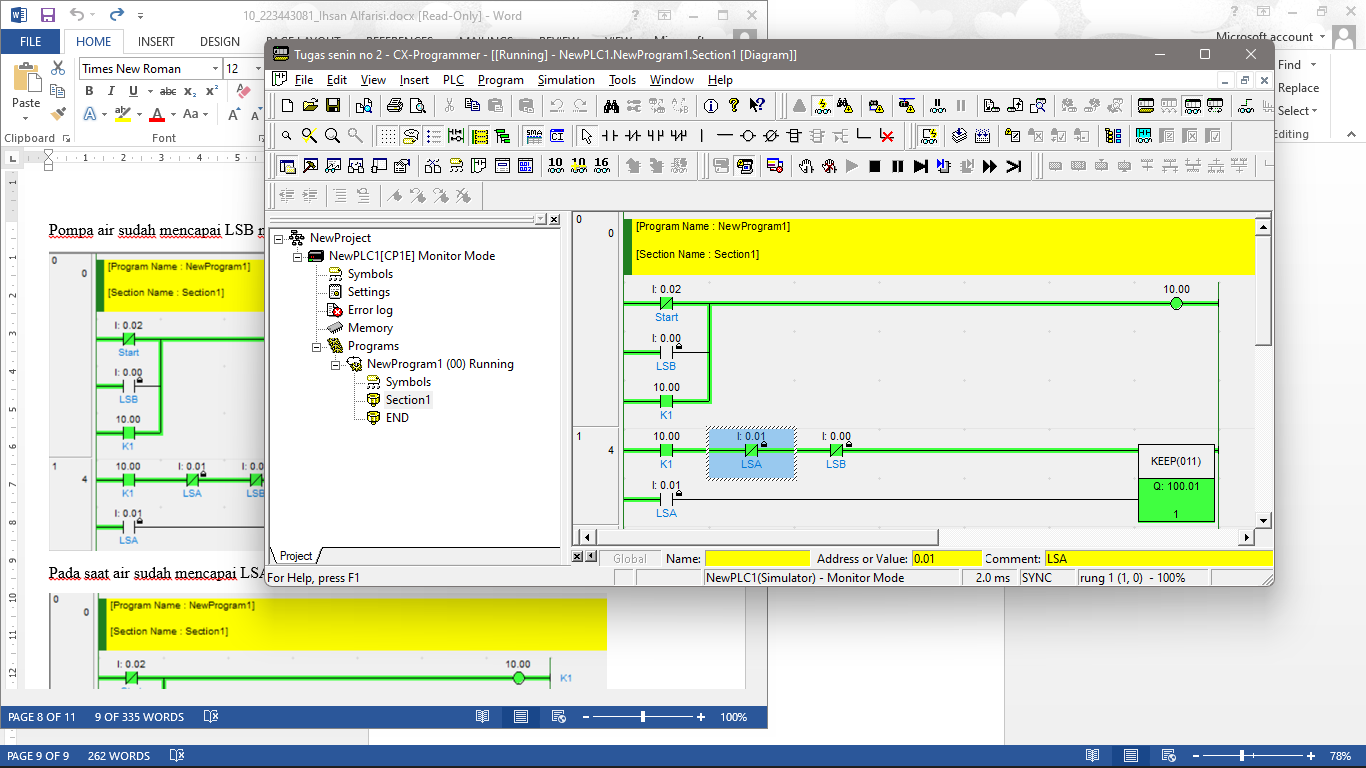
Jika input-1 dan input-2 di on kan syaratnya kedua output tidak boleh menyala bersama,jadi hanya output-2 yang menyala

Buatlah program untuk soal berikut:

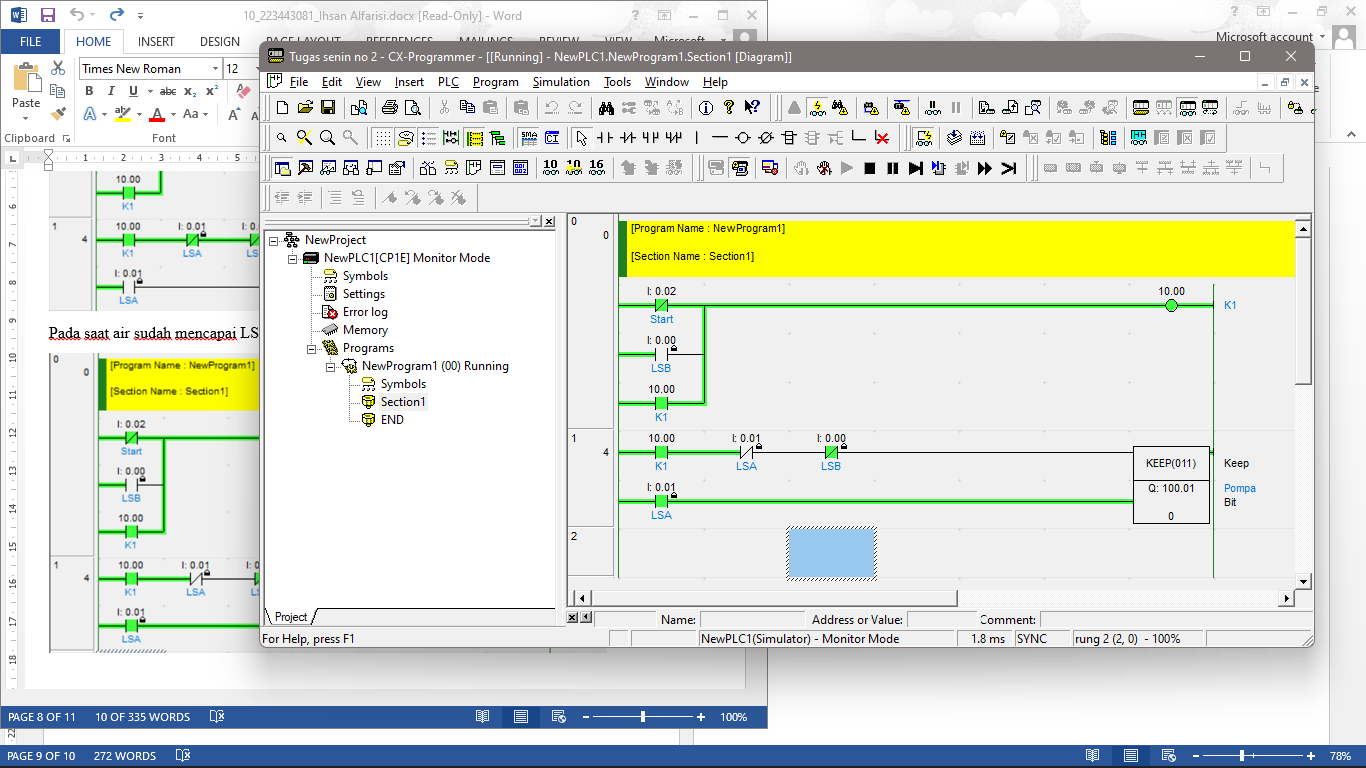
Sebuah instalasi tangki penyimpan airsederhana seperti pada gambar berikut. Asumsi awal tangki dalam keadaan kosong. Jika tangki kosong maka pompa akan bekerja memompa air dari sumur ke dalam tangki. Jika LSB (*Level Switch* Bawah) terkena bahkan terendam air, pompa masih bekerja. Pompa akan mati jika LSA (*Level Switch* Atas) tersentuh air. Karena air digunakan terus, maka level air akan turun. Karena level air turun maka ketika LSB tidak terendam airpompa akan bekerja kembali.

**Hasil dan Analisis**

Pada Saat awal semuanya sudah menyala

Pompa air sudah mencapai LSB maka pompa akan menyala

Pada saat air sudah mencapai LSA maka pompa akan mati

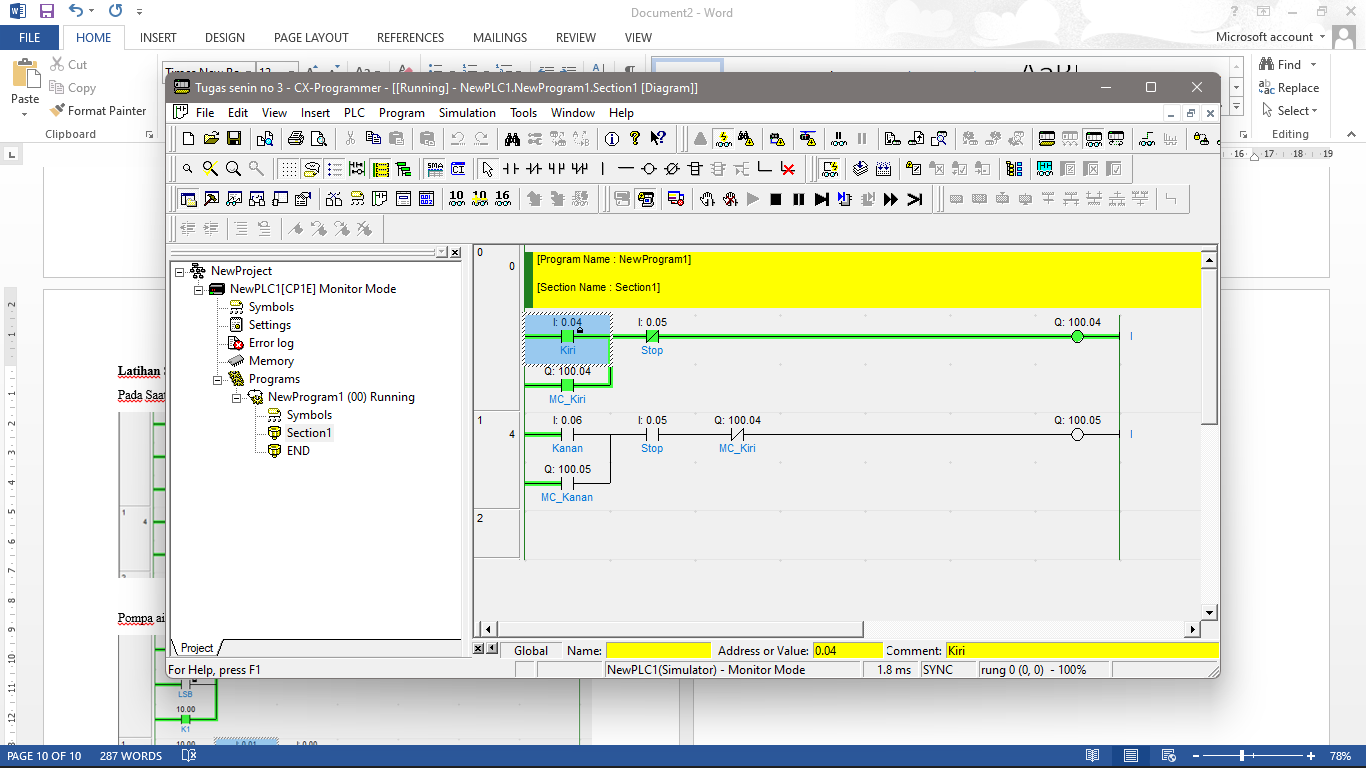


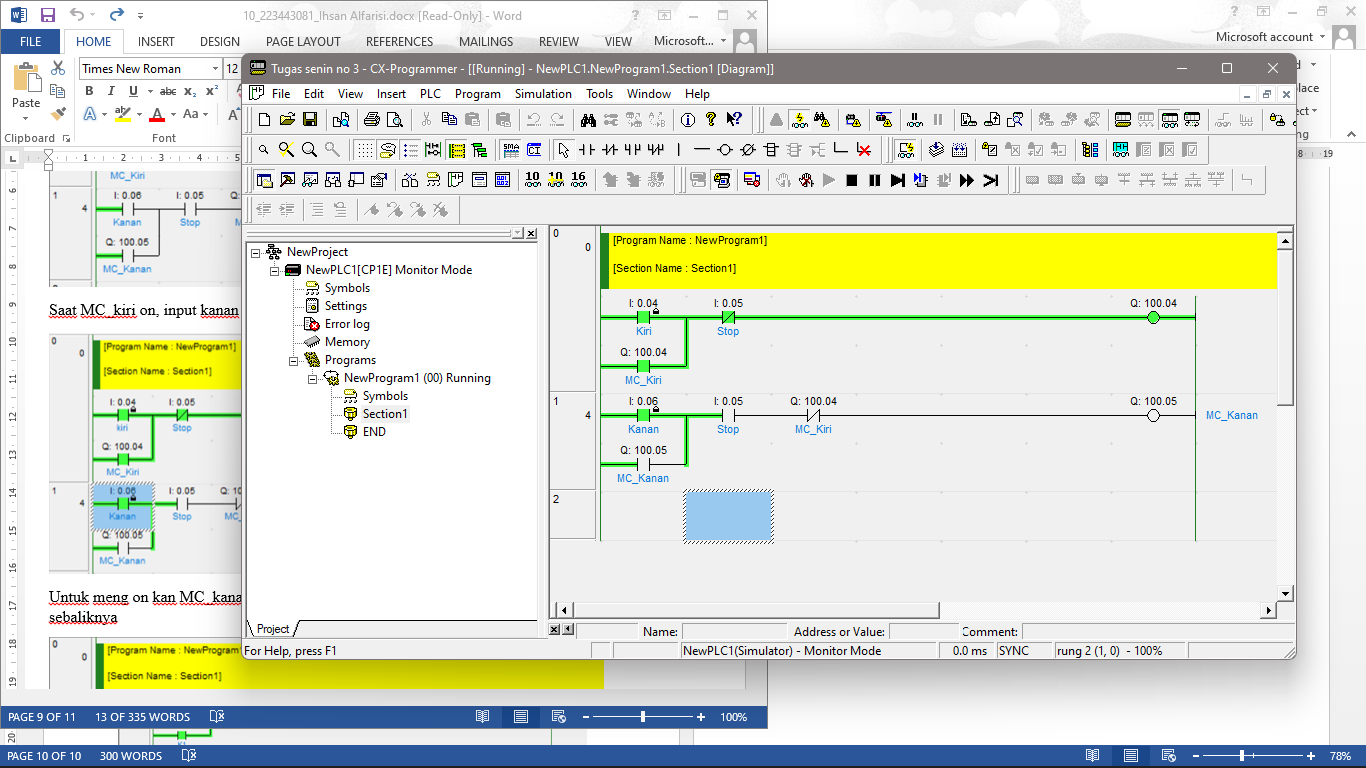
Buatlah program untuk soal berikut:

Terdapat sebuah mesin *milling* sederhana, ketika *Input* kiri di- *on*-kan maka MC\_kiri (*Magnetic Contactor*\_kiri) akan*on*. Saat MC\_kiri *on*, *input* kanan di-*on*-kan tidak akan mengakibatkan MC\_kanan *on*. Untuk meng-*on*-kan MC\_kanan, tombol *stop* harus di- *on*-kan lebih dulu. Demikian juga sebaliknya

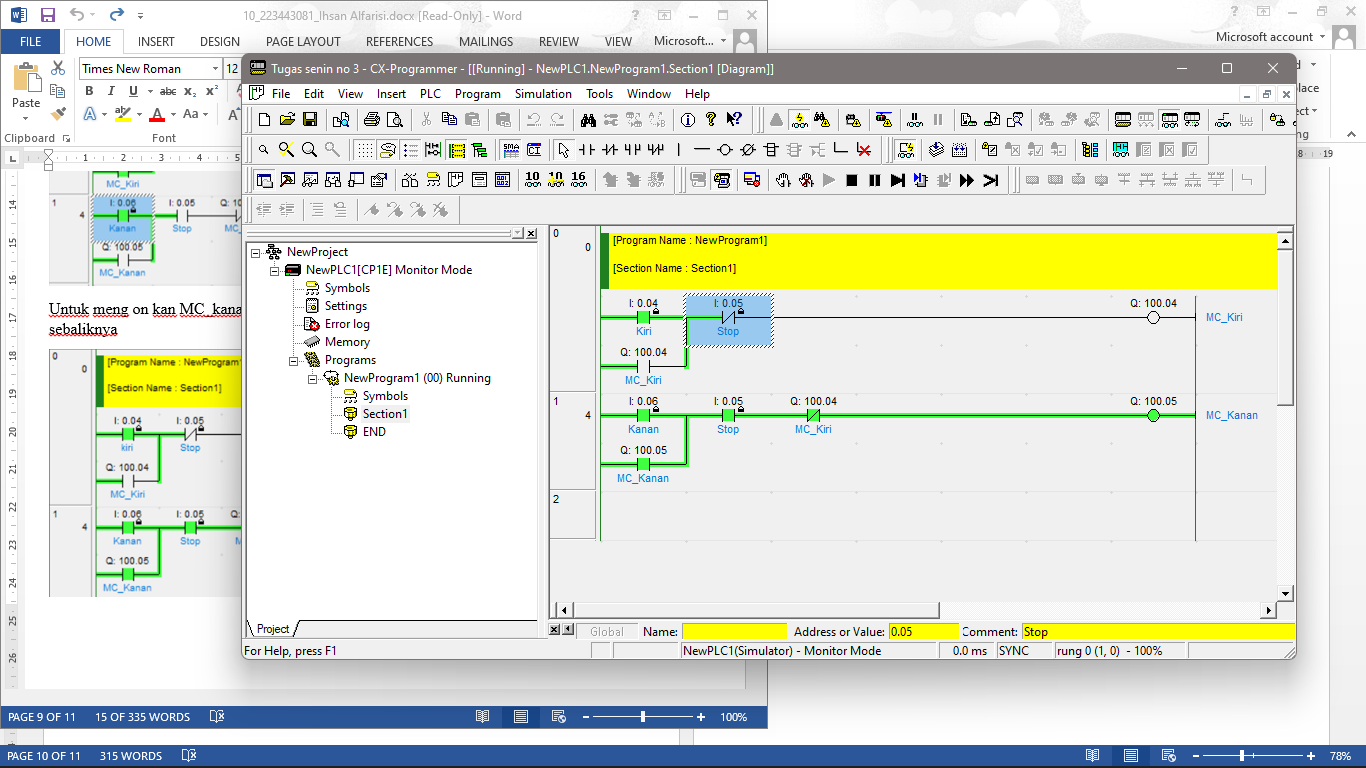
**Hasil dan Analisis**

Ketika Input kiri di on kan maka MC\_kiri (Magnetic Contactor\_kiri) akan on



Saat MC\_kiri on, input kanan di on kan tidak akan mengakibatkan MC\_kanan on.

Untuk meng on kan MC\_kanan, tombol stop harus di on-kan lebih dulu. Demikian juga sebaliknya

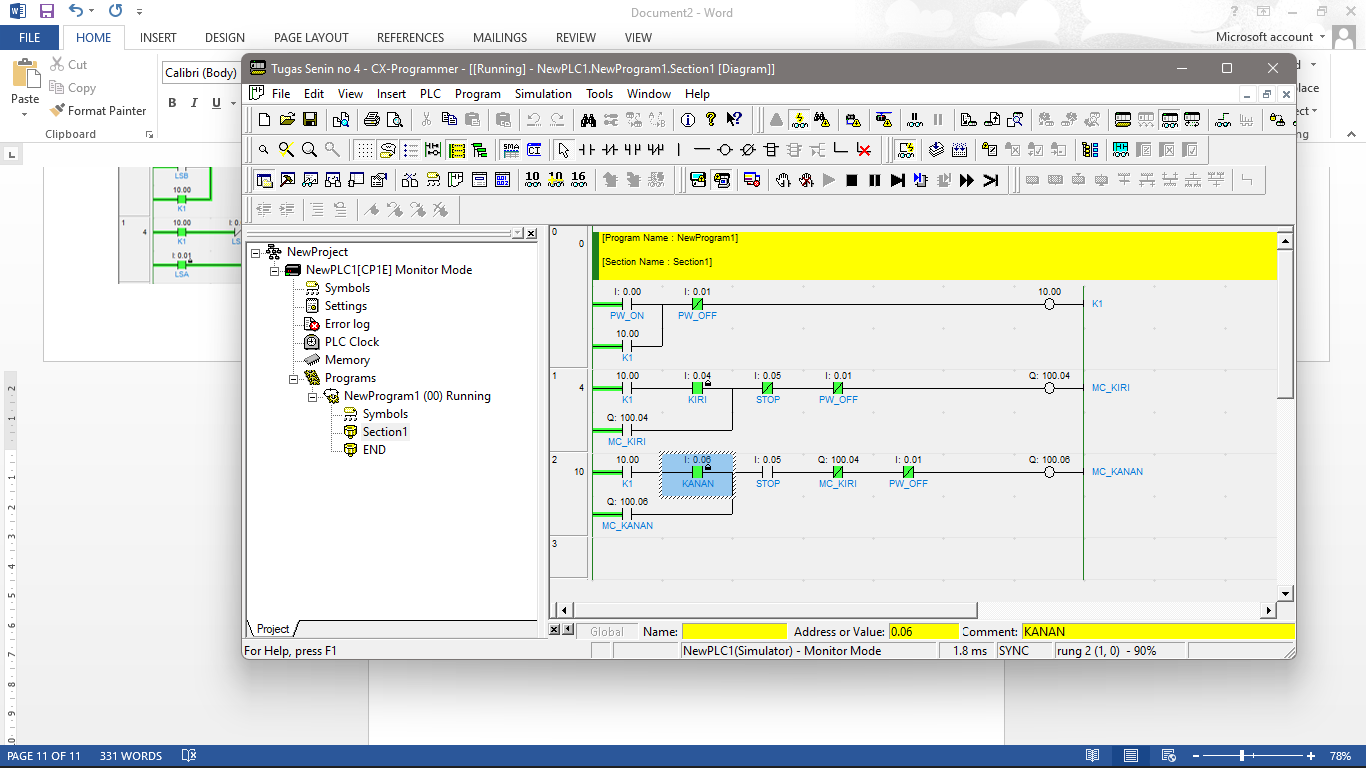


Buatlah program untuk soal berikut:

Sama dengan soal sebelumnya, hanya saja ditambahi dengan *Input* PW\_*ON* (*Power*\_*ON*)dan PW\_*OFF* (*Power*\_*OFF*) *Input PW\_ON* berfungsisebagai saklar *ON*. Jika saklar ini belum di-*on*-kan maka mesin *milling* ini tidak bisa dioperasikan

**Hasil dan Analisis**

Input kiri atau Input kanan di-on-kan tetapi Input PW\_ON belum di on kan



Output MC\_kiri atau MC\_kanan sedang on, Input PW\_OFF di on kan



# PRAKTIKUM 2 INSTRUKSI DASAR PLC – TIMER

**Tujuan :**

1. Memahami bagaimana penggunaan timer
2. Mampu memprogram fungsi timer menggunakan perangkat lunak pemrograman PLC
3. Melakukan simulasi program yang melibatkan fungsi timer untuk memastikan bahwa waktu yang diatur berperilaku sesuai dengan yang diharapkan.

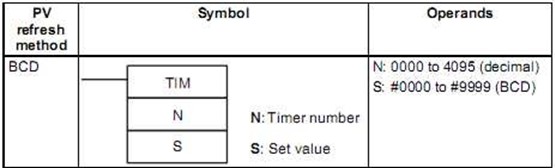
## Teori Dasar

Instruksi Timer Adalah instruksi yang digunakan untuk meng-*handle* pewaktu dan pencacah peristiwa.

**TIMER: TIM(XXXX)**

Adalah instruksi yang digunakan untuk meng-*handle* fungsi pewaktu. Bentuk fungsi

*timer* adalah sebagai berikut:



Ketika *timer* mendapat *input*, maka *set value timer* akan berkurang, sampai menjadi nol. Jika sudah nol maka *timer* tersebut akan aktif.

Nilai *set value timer* diisi dengan bilangan BCD, dimana nilai pewaktunya adalah bilangan BCD tersebut dikali 0.1 detik.

Format penulisan instruksi timer adalah : TIM spasi addres timmer spasi #waktu Misal : TIM 0000 #10

TIM: instruksi timer

0000 : initial value (0000 – 9999) -> N #10 : timer value -> SV

(#10 = 1000ms = 10 x 0,1 s = 1s)

## Alat Dan Bahan

* PLC Omron CP1E-N30
* Laptop
* Kabel jumper
* Kabel USB

## Petunjuk Praktikum

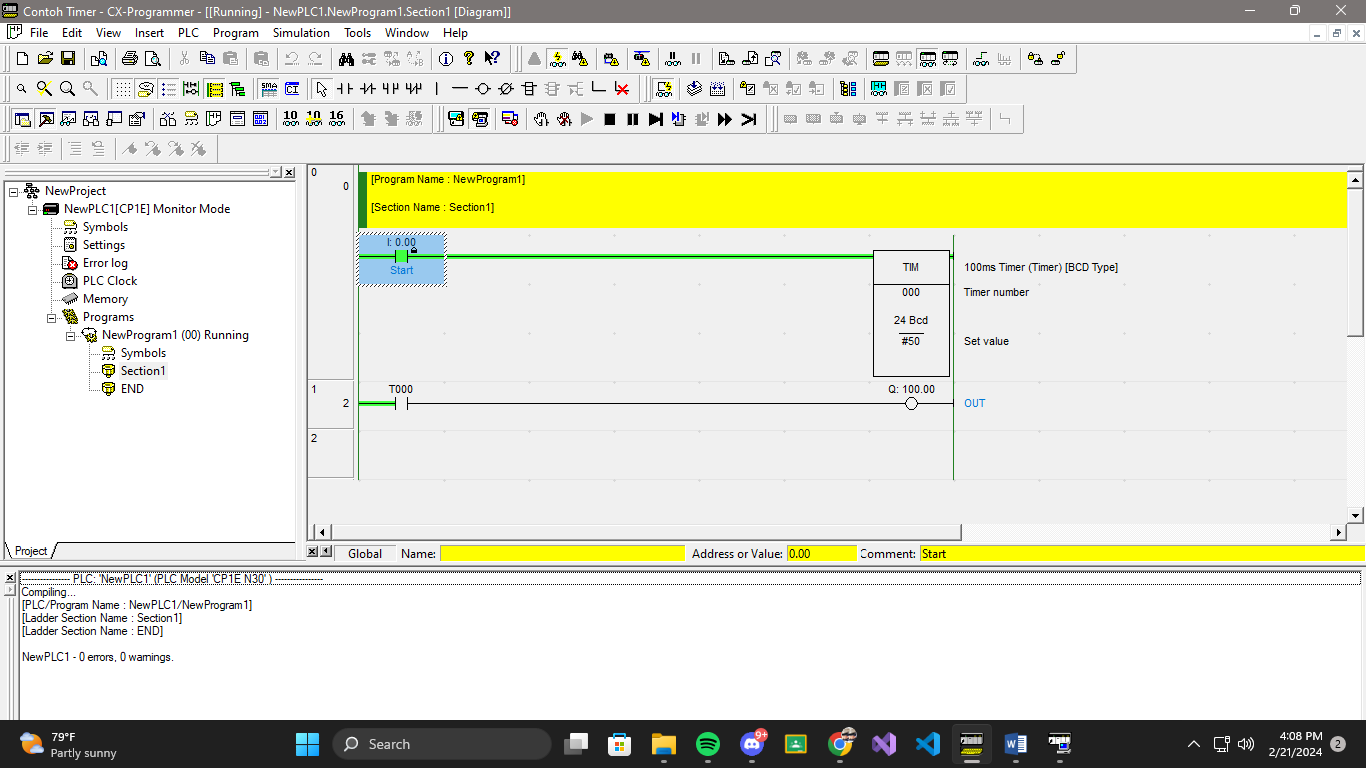
1. Pertama yaitu download aplikasi CXONE\_v4.40 terlebih dahulu. Ikutilah langkah-langkah penginstalannya.
2. Setelah selesai terinstall,buka lah aplikasinya. Lalu kita pilih file lalu *New...* , isi device name dan device type sesuai PLC yang anda punya. Disini saya menggunakan PLC dengan device type CP1E-N30.
3. Jika sudah maka tekan OK barulah kita bisa membuat rangkaiannya.
4. Pasang PLC pada panel kontrol atau lokasi yang sesuai.
5. Sambungkan PLC ke sumber daya listrik sesuai dengan spesifikasi produsen.
6. Hubungkan laptop dengan PLC menggunakan kabel yang sesuai. Pastikan bahwa koneksi antara laptop dan PLC berfungsi dengan baik.
7. Pasang modul input/output ke slot yang sesuai di PLC.
8. Hubungkan perangkat input dan output di lapangan ke modul input/output.
9. Gunakan fasilitas simulasi jika tersedia untuk menguji program tanpa perlu terhubung ke PLC fisik.
10. Simulasikan kondisi input dan periksa respons output sesuai dengan logika program.
11. Setelah pemrograman dan simulasi selesai, transfer program ke memori PLC menggunakan koneksi laptop. Pastikan transfer program berjalan tanpa kesalahan.
12. Monitor operasi PLC melalui laptop untuk memastikan bahwa program berjalan sesuai dengan harapan.
13. Gunakan fasilitas debugging untuk mengidentifikasi dan memperbaiki kesalahan dalam program.

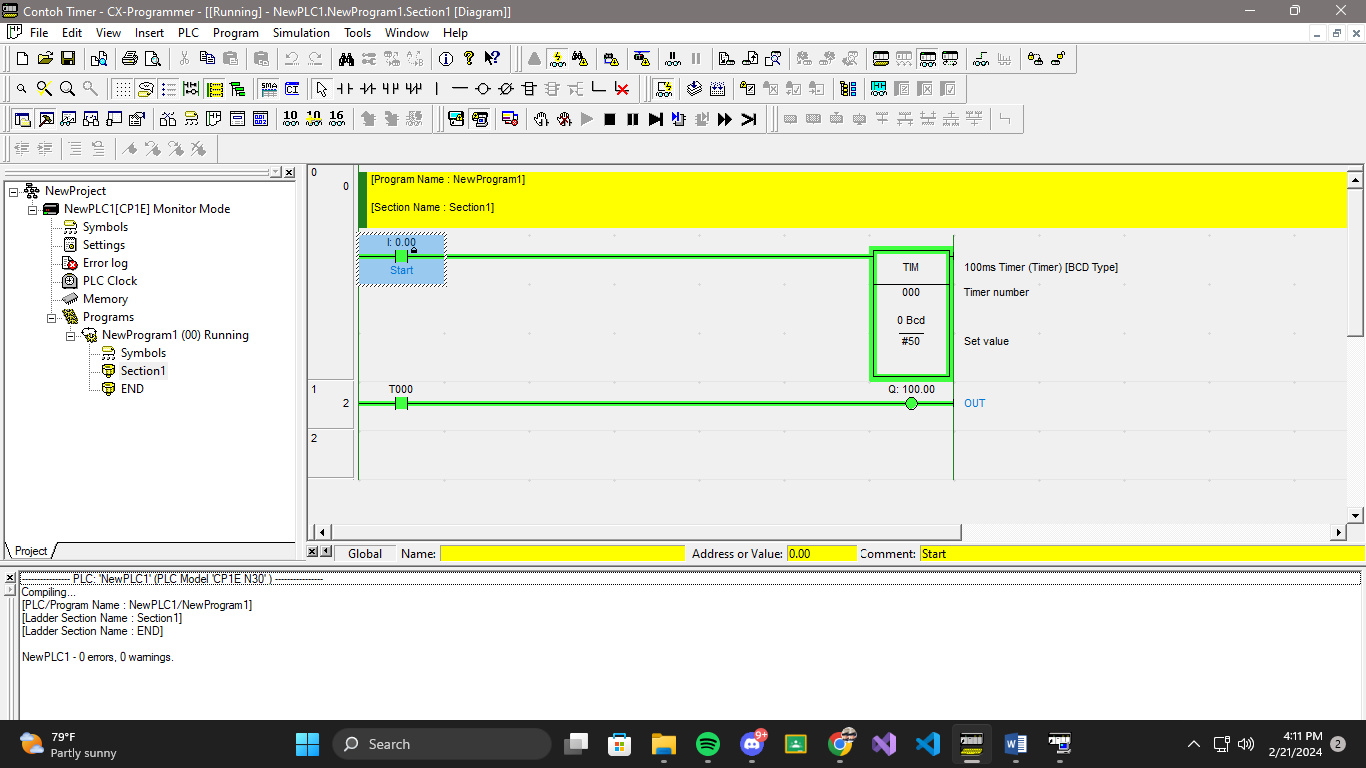
## Praktikum

### Percobaan 1: Timer



**Hasil dan Analisis**

 Kondisi pada saat start dinyalakan timer akan berjalan

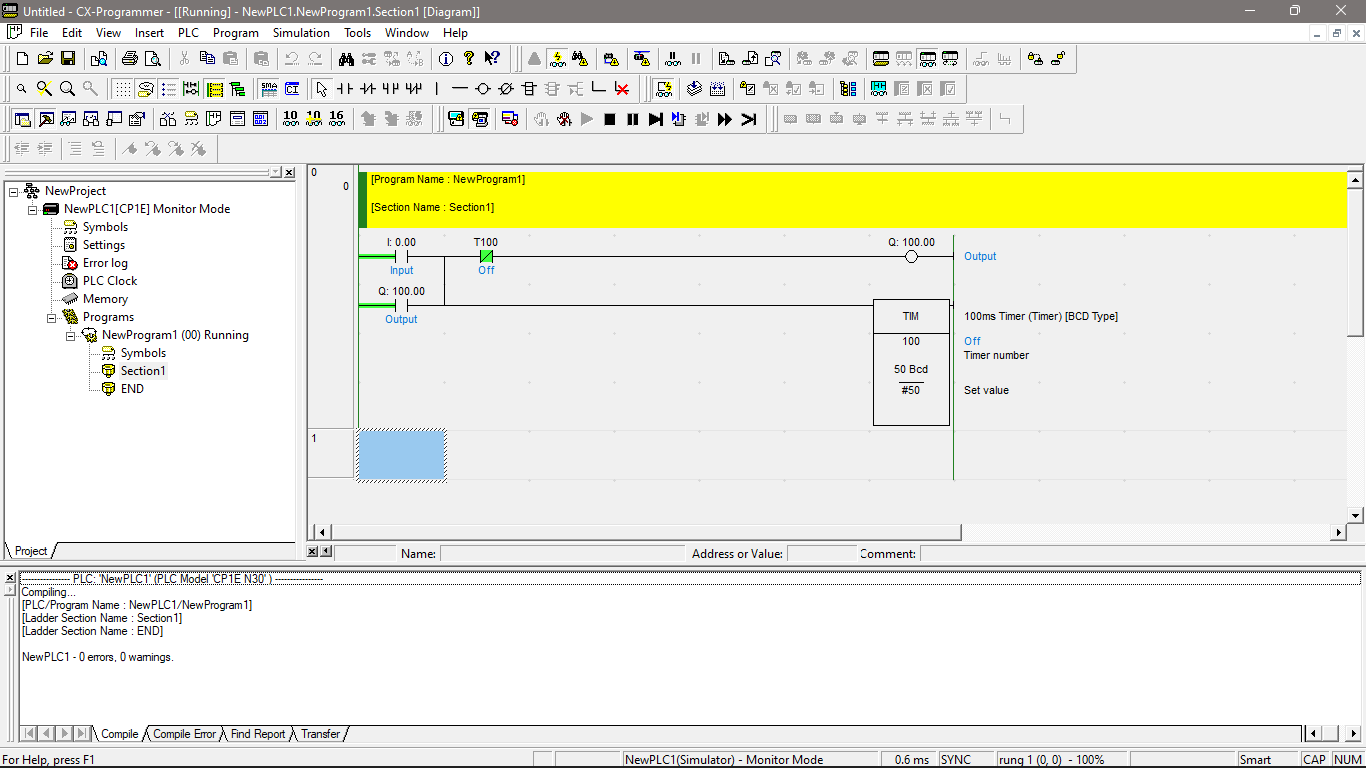
 Kondisi saat timer sudah 0 dan ouput akan menyala

## Soal Latihan

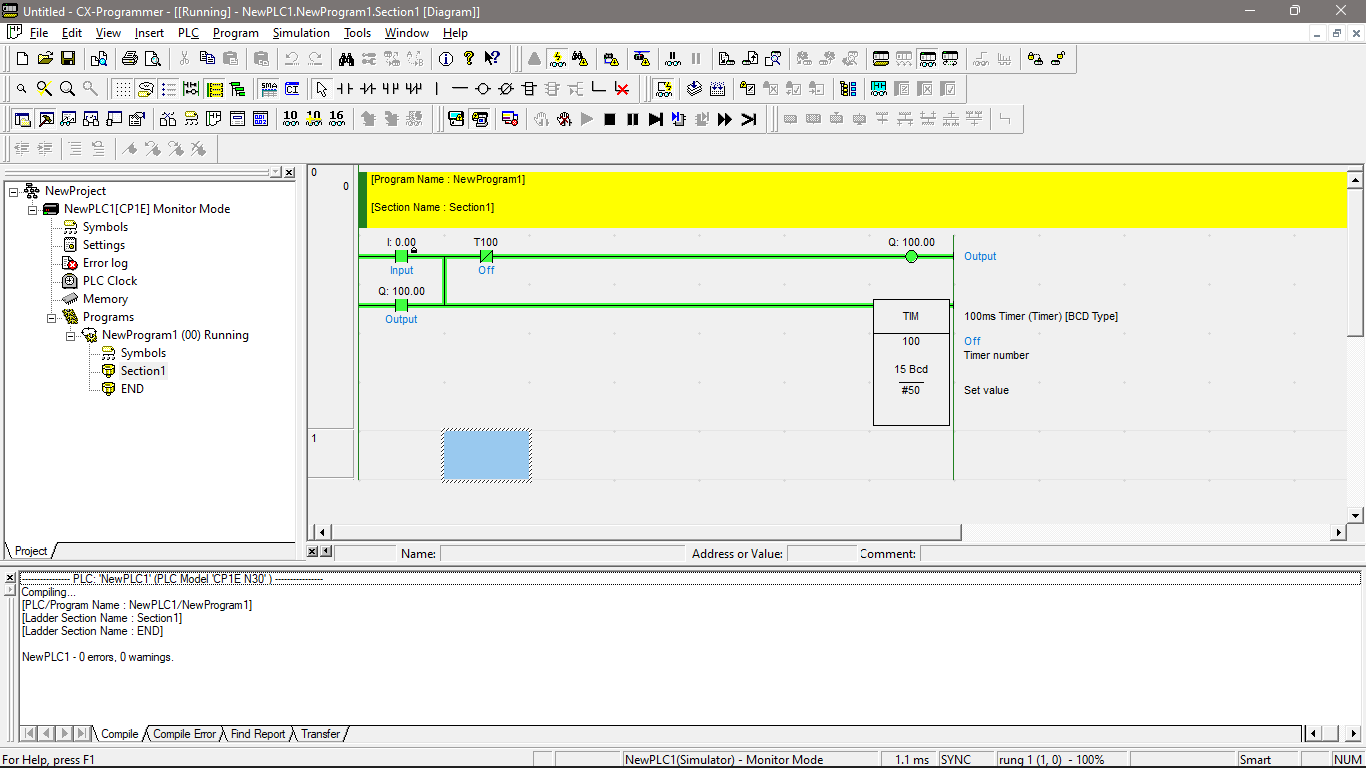
Buatlah program untuk soal berikut:

Jika *Input* di-*on*-kan maka *output* akan *on* selama 5 detik, setelah itu *output* akan mati sendiri.

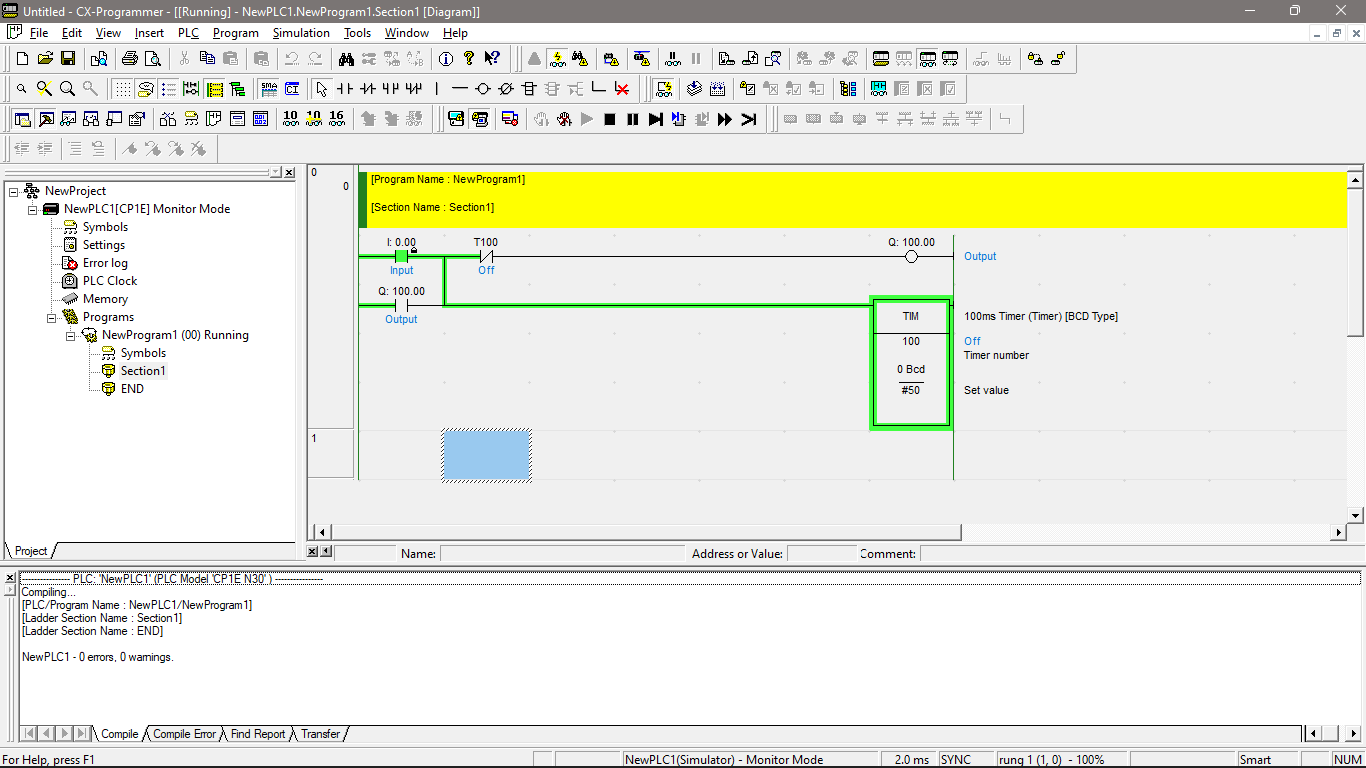
**Hasil dan Analisis**

 Pada saat kondisi dirunning

Pada saat button input ditekan timer akan berjalan sebanyak 5 detik



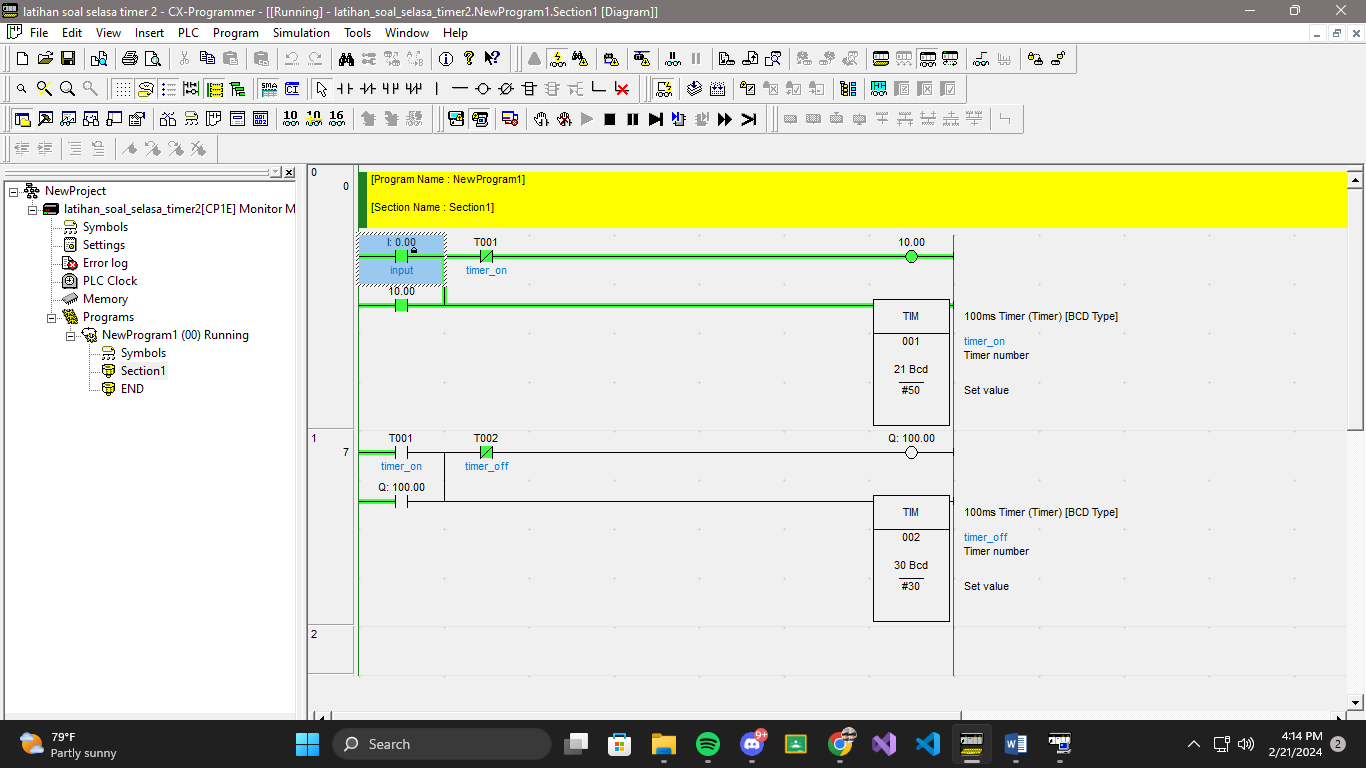
Kondisi saat timer sudah menjadi 0,lalu output akan mati otomatis



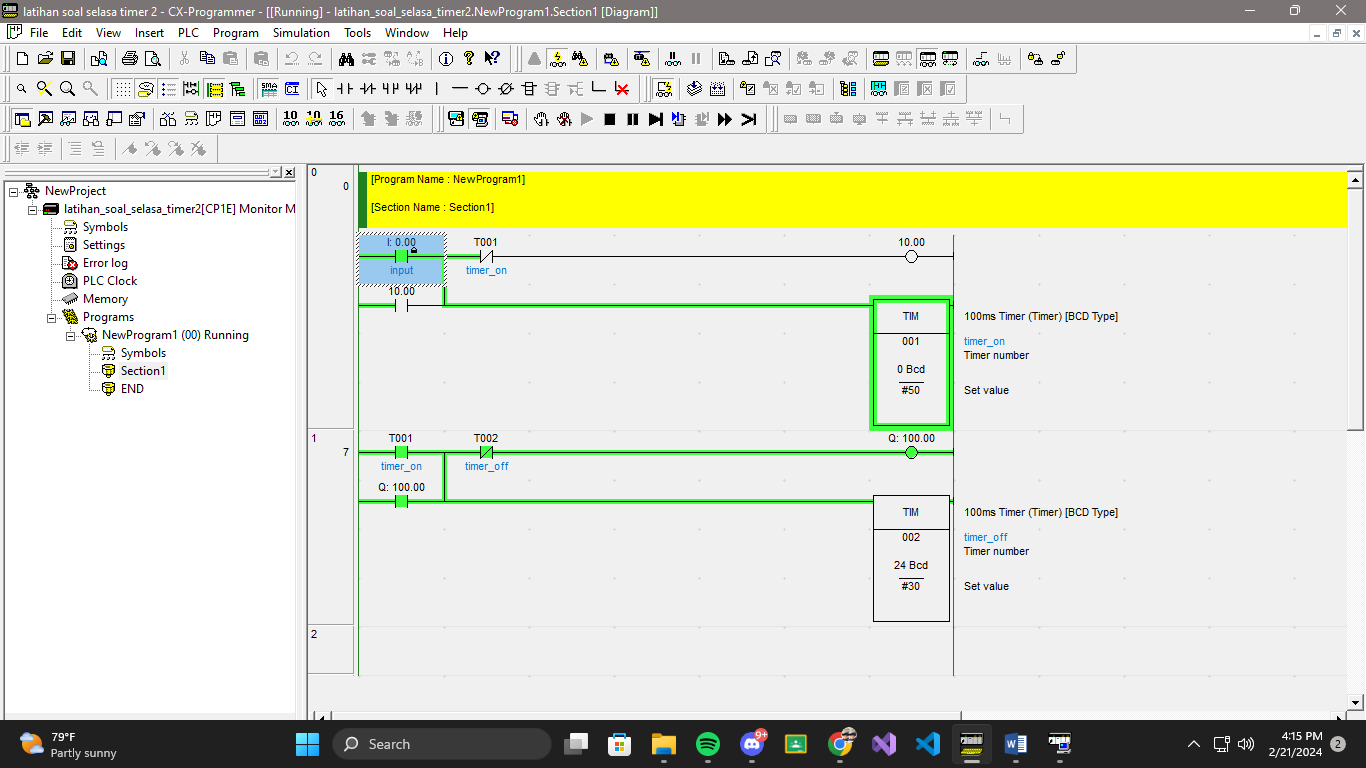
Buatlah program untuk soal berikut:

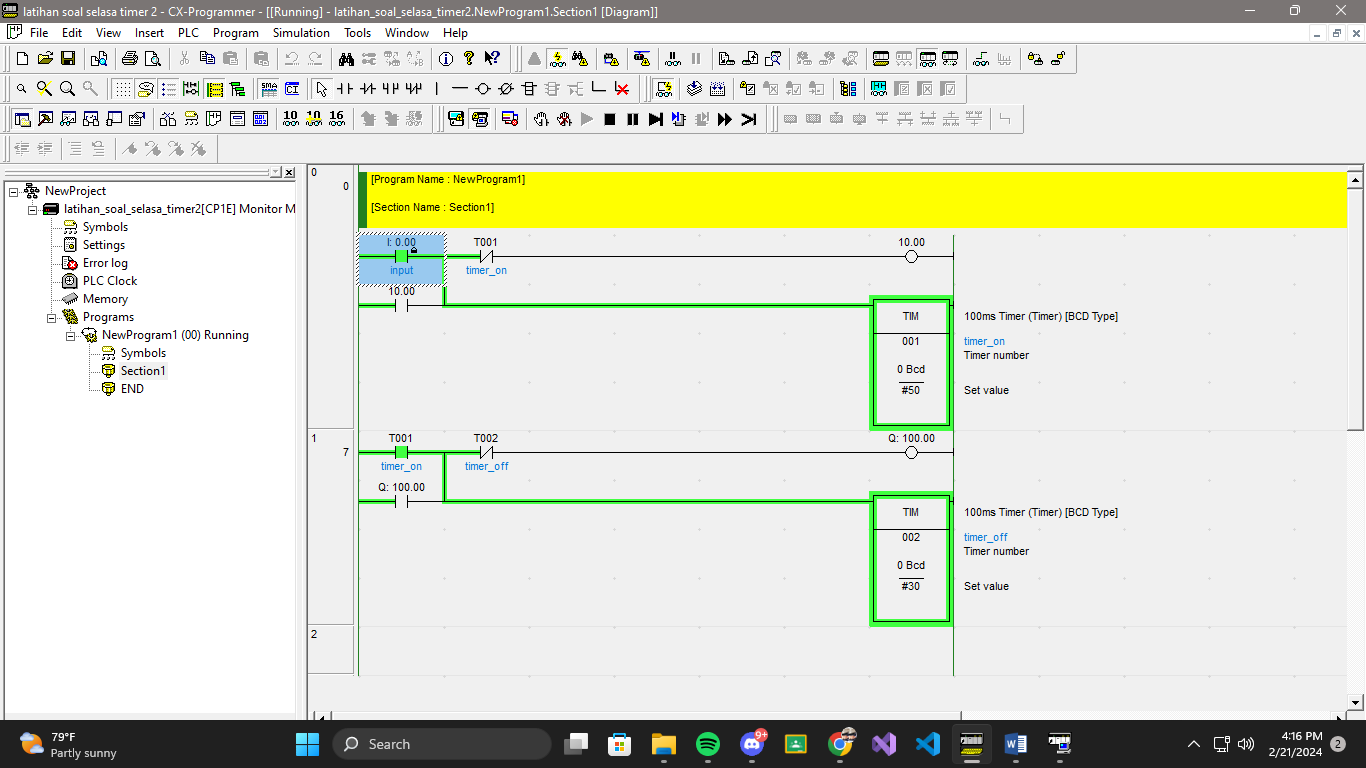
Jika *Input* di-*on*-kan maka *output* akan menunggu selama 5 detik baruakan *on*. *Output on* selama 3 detik, setelah itu *output* akan matisendiri. Gunakan memori *internal* 200.00 untuk membantu membuat program ini.

**Hasil dan Analisis**

 Kondisi saat start dinyalakan dan timer akan berjalan selama 5 detik

Setelah timer 5 detik habis output akan menyala selama 3 detik



 Setelah output menyala selama 3 detik output akan mati otomatis

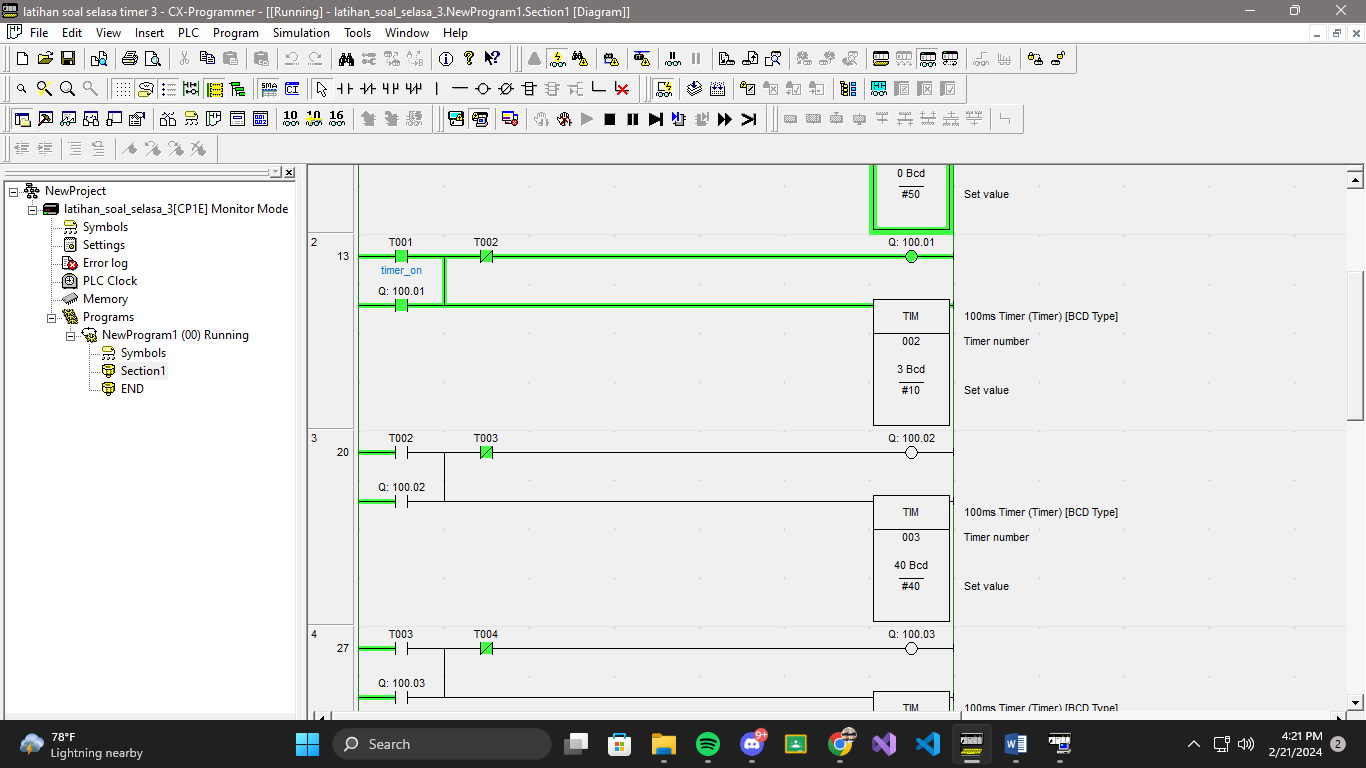
Buatlah program untuk soal berikut:

Terdapat dua *Input* yaitu *start* dan *stop*, serta 5 *output* yaitu *output* A, B,C, D, dan E. Jika *Input start* di-*on*-kan maka *output* A akan langsung *on* selama 5 detik. Setelah 5 detik *output* A akan *off*, kemudian *output* B akan *on* selama 1 detik. Demikian seterusnya sampai *output* E. Setelah *output* E mati, *output* A akan *on* lagi selama 5 detik. Proses tersebut akan terus berulang sampai *Input stop* diaktifkan.

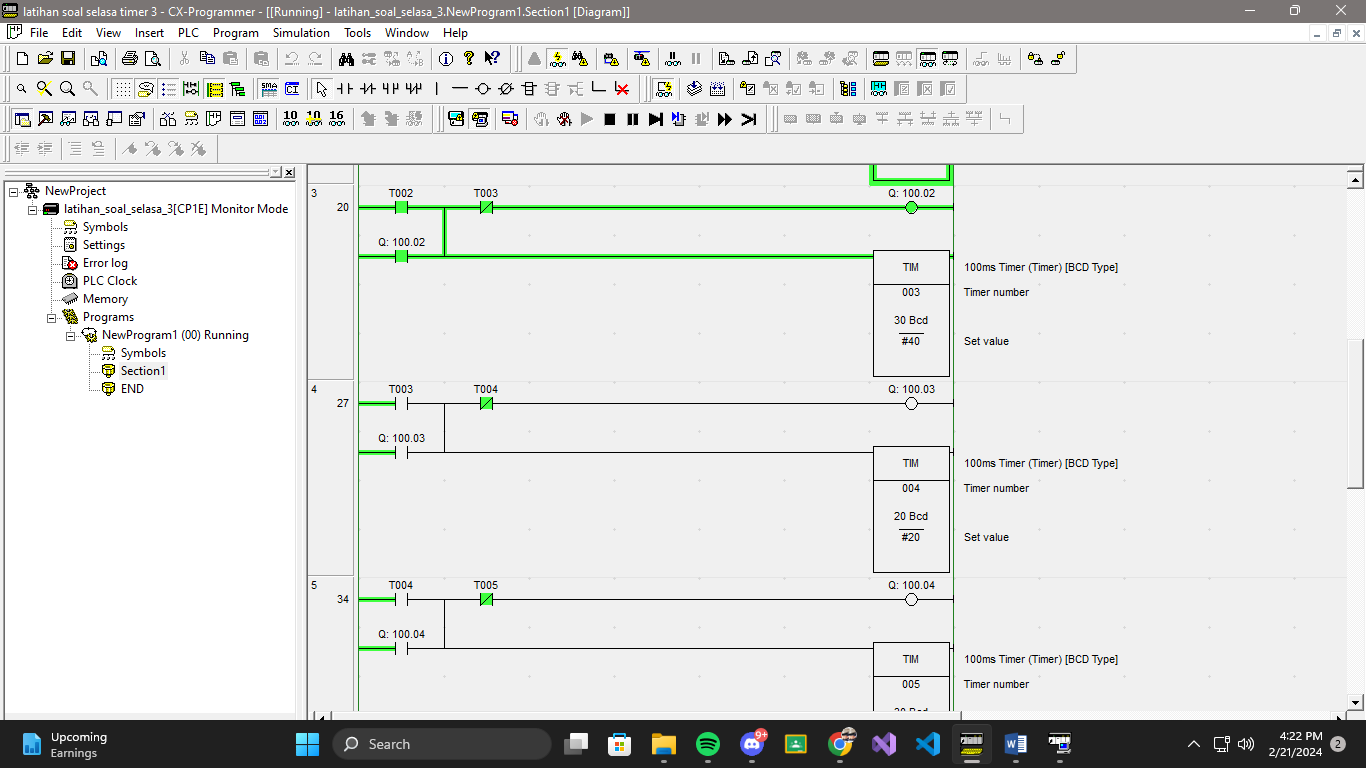
**Hasil dan Analisis**

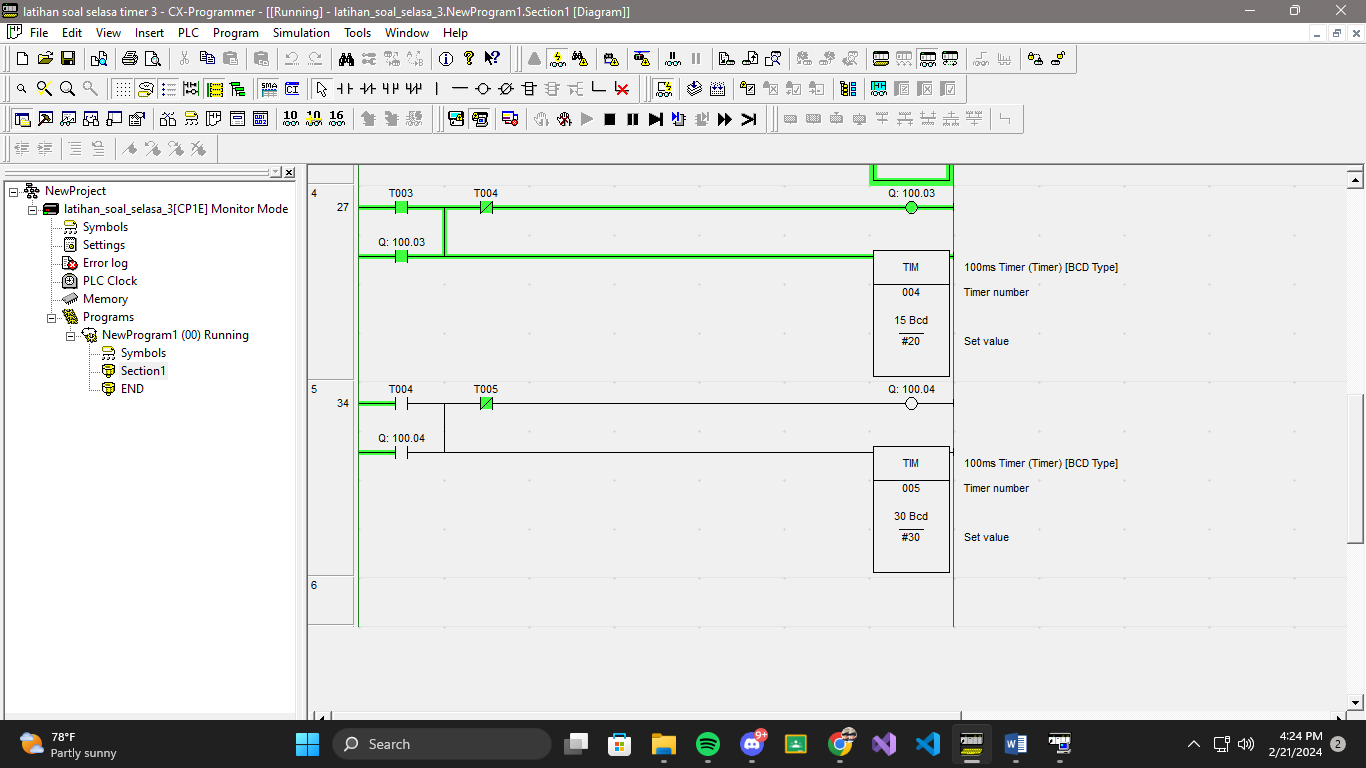
 Jika Input start di-on-kan maka output A akan langsung on selama 5 detik

Setelah 5 detik output A akan off, kemudian output B akan on selama 1 detik

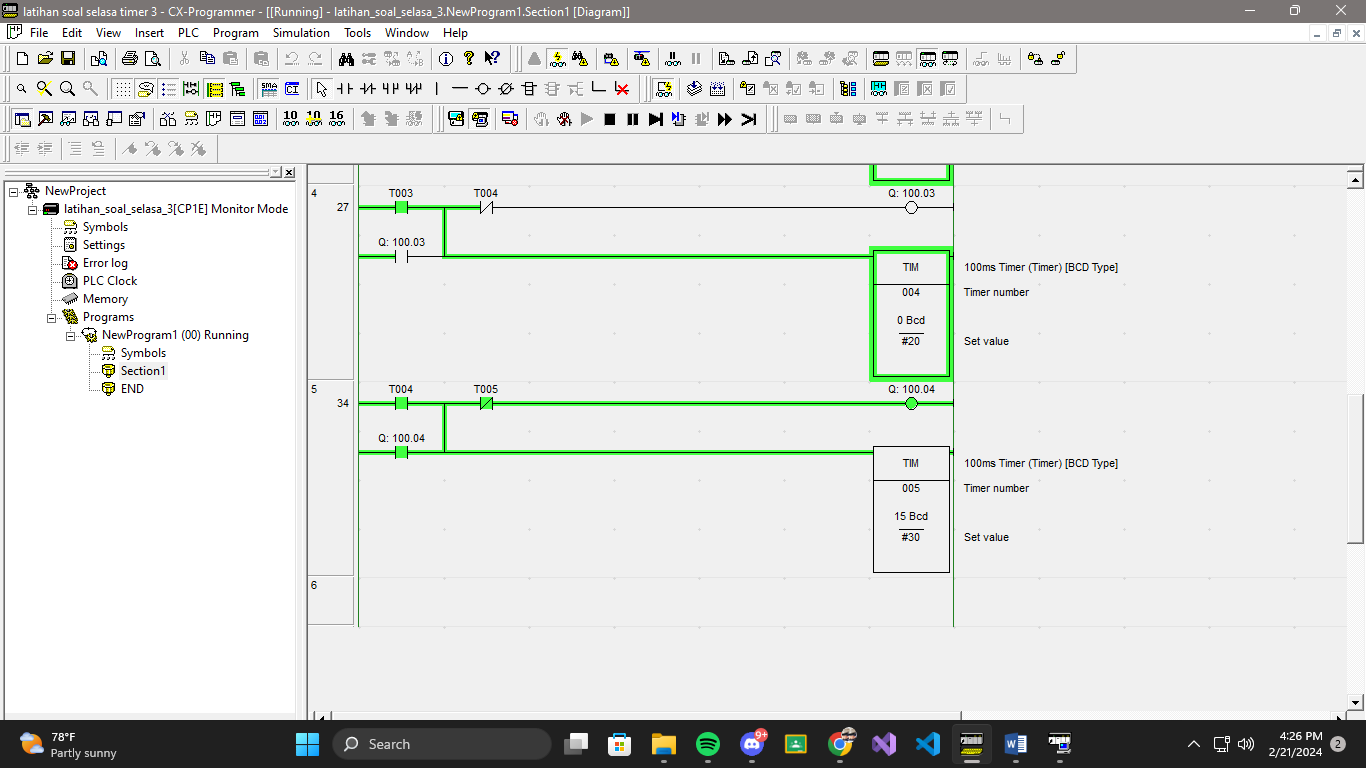


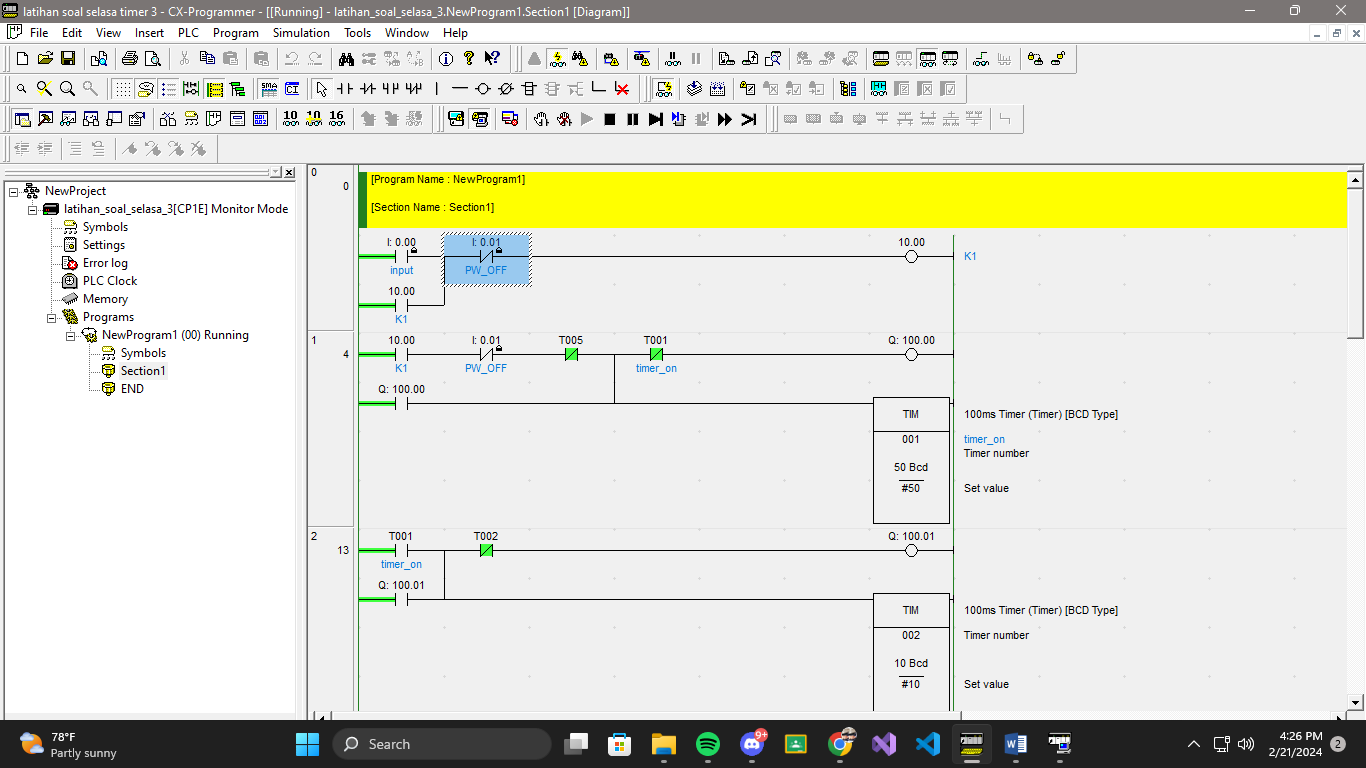
Setelah 1 detik output B akan off, kemudian output C akan on selama 4 detik



 Setelah 4 detik output C akan off, kemudian output D akan on selama 2 detik

Setelah 2 detik output D akan off, kemudian output E akan on selama 3 detik.Setelah E off output akan kembali menyala dari A dan seterusnya hingga button stop ditekan



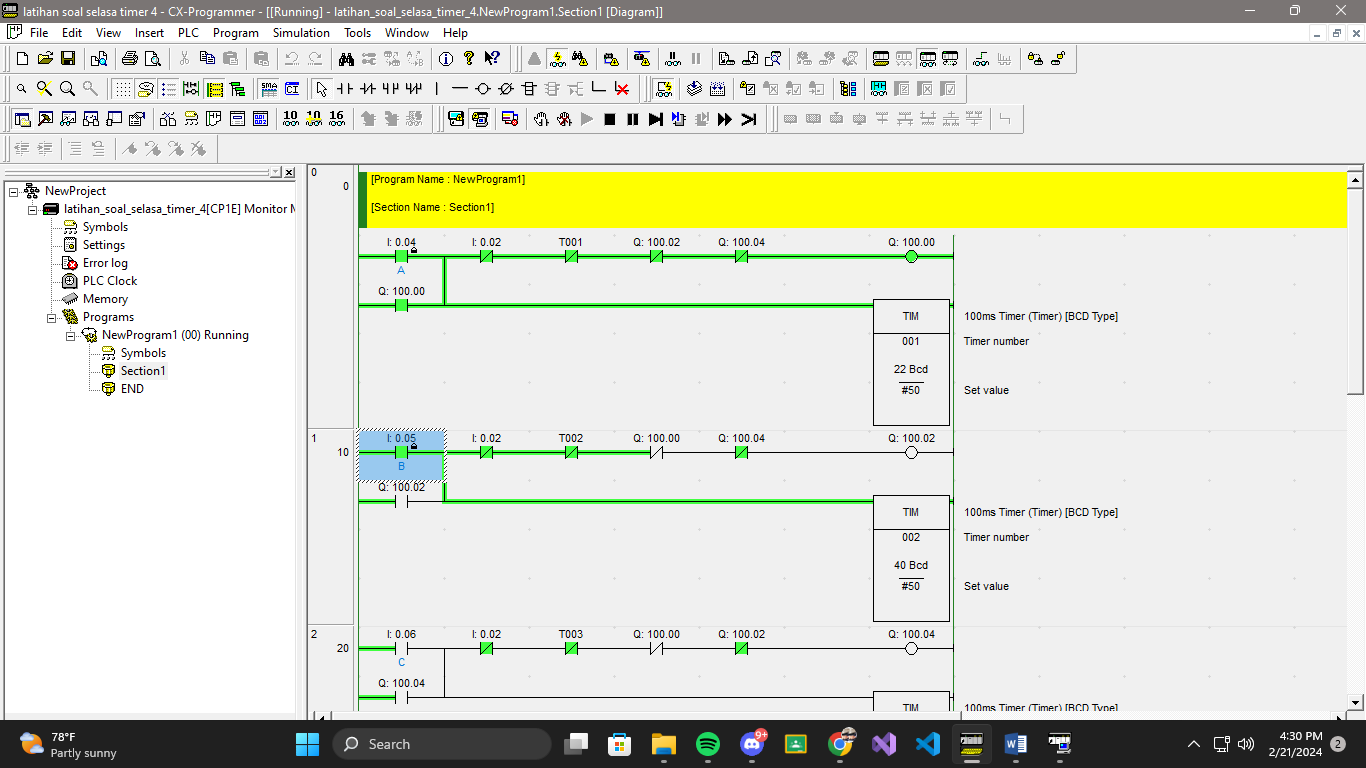
 Kondisi ketika sudah di stop semuanya mati

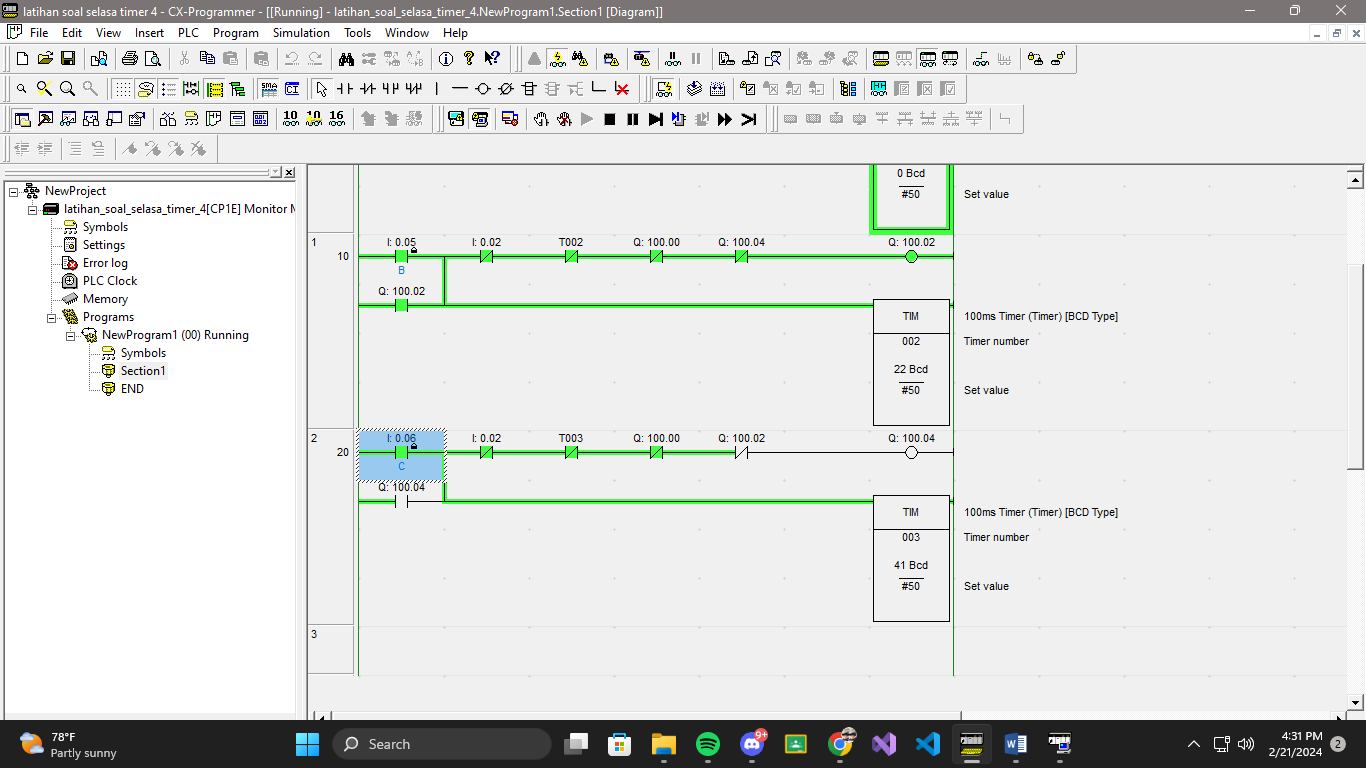
Buatlah program untuk soal berikut:

Terdapat 3 peserta kuis, yaitu A, B, dan C. Di masing-masing meja pesertaterdapat tombol dan lampu. Jika juri membacakan pertanyaan dan peserta yangakan menjawab pertanyaan tersebut harusmenekan tombol di meja peserta sendiri- sendiri. Siapapun yang menekan tombol lebih dulu lampu peserta tersebut akan menyala selama 5 detik dan lampu peserta laintidak bisa menyala ketika salah seorang peserta sudah menyalakan lampunya

**Hasil dan Analisis**

Jika tombol A ditekan maka output A akan menyala tetapi output B dan C tidak dapat menyala begitupun kerja tombol yang lainnya sama seperti A.





# PRAKTIKUM 3 INSTRUKSI DASAR PLC – COUNTER

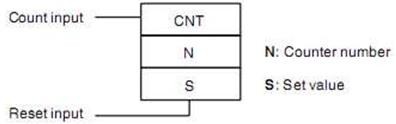
**Tujuan :**

1. Memahami konsep dasar penghitung dalam PLC.
2. Memahami berbagai jenis counter seperti up-counter (penghitung naik), down-counter (penghitung turun)
3. Menggunakan counter dalam logika kontrol untuk memenuhi kebutuhan sistem.

## Teori Dasar

**COUNTER: CNT(XXXX)**

Adalah instruksi yang digunakan untuk meng*handle* pencacah peristiwa. Bentuk fungsi *Counter* adalah sebagai berikut:



Ketika *Counter* mendapat *input*, maka *set value counter* akan berkurang, sampai menjadi nol.Jika sudah nol maka *Counter* tersebut akan aktif.

Nilai *set value counter* diisi dengan bilangan BCD, dimana nilai pencacahnya adalah sebanyakbilangan BCD yang dimasukkan. (BCD : Binary Code Decimal).

Sintak penulisan adalah sebagai berikut:

CNT(spasi) alamat (spasi) #nilaicounter.

Contoh CNT 0000 #3 -> Counter 00 akan aktif setelah diberi masukkan sebanyak tiga kali

CNT : instruksi counter

0000 : initial value (0000 – 9999) -> N #3: counter value -> SV

## Alat Dan Bahan

* PLC Omron CP1E-N30
* Laptop
* Kabel jumper
* Kabel USB

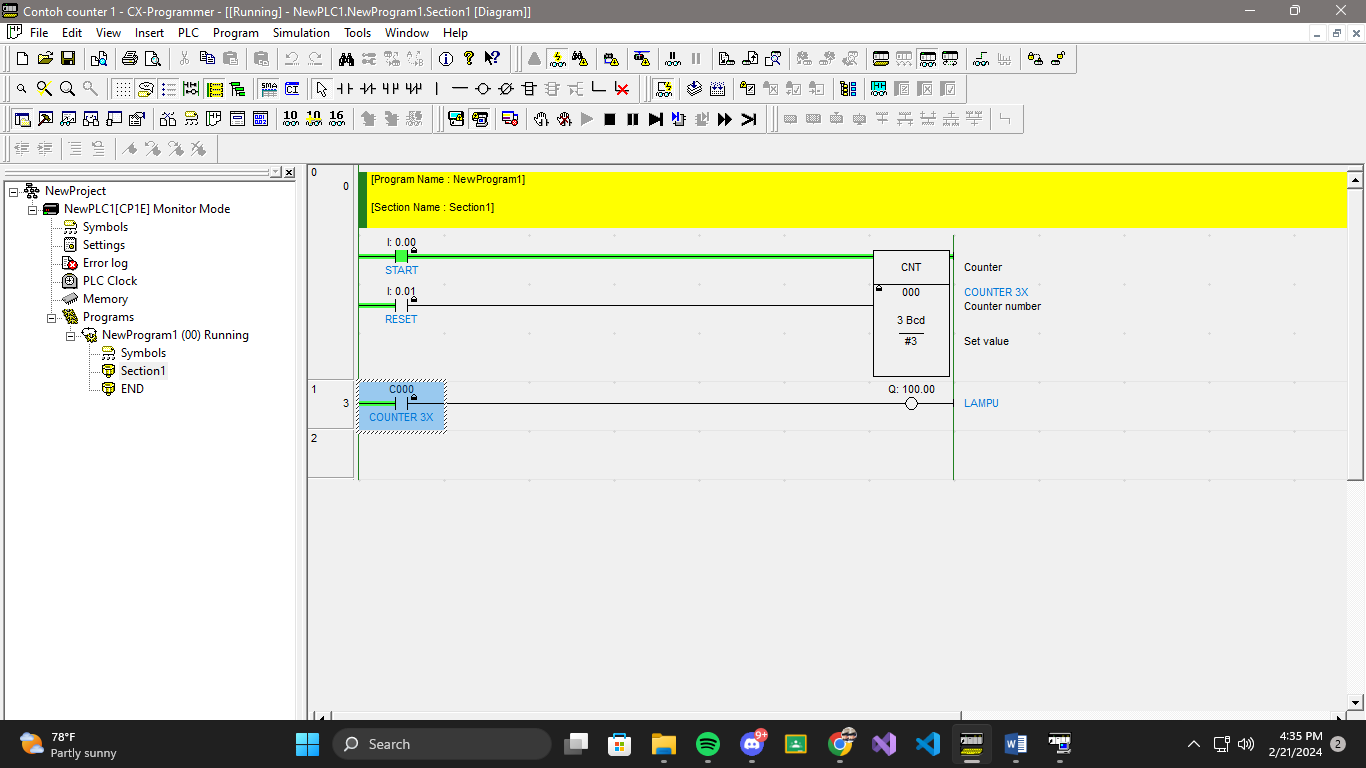
## Petunjuk Praktikum

1. Pertama yaitu download aplikasi CXONE\_v4.40 terlebih dahulu. Ikutilah langkah-langkah penginstalannya.
2. Setelah selesai terinstall,buka lah aplikasinya. Lalu kita pilih file lalu *New...* , isi device name dan device type sesuai PLC yang anda punya. Disini saya menggunakan PLC dengan device type CP1E-N30.
3. Jika sudah maka tekan OK barulah kita bisa membuat rangkaiannya.
4. Pasang PLC pada panel kontrol atau lokasi yang sesuai.
5. Sambungkan PLC ke sumber daya listrik sesuai dengan spesifikasi produsen.
6. Hubungkan laptop dengan PLC menggunakan kabel yang sesuai. Pastikan bahwa koneksi antara laptop dan PLC berfungsi dengan baik.
7. Pasang modul input/output ke slot yang sesuai di PLC.
8. Hubungkan perangkat input dan output di lapangan ke modul input/output.
9. Gunakan fasilitas simulasi jika tersedia untuk menguji program tanpa perlu terhubung ke PLC fisik.
10. Simulasikan kondisi input dan periksa respons output sesuai dengan logika program.
11. Setelah pemrograman dan simulasi selesai, transfer program ke memori PLC menggunakan koneksi laptop. Pastikan transfer program berjalan tanpa kesalahan.
12. Monitor operasi PLC melalui laptop untuk memastikan bahwa program berjalan sesuai dengan harapan.
13. Gunakan fasilitas debugging untuk mengidentifikasi dan memperbaiki kesalahan dalam program.

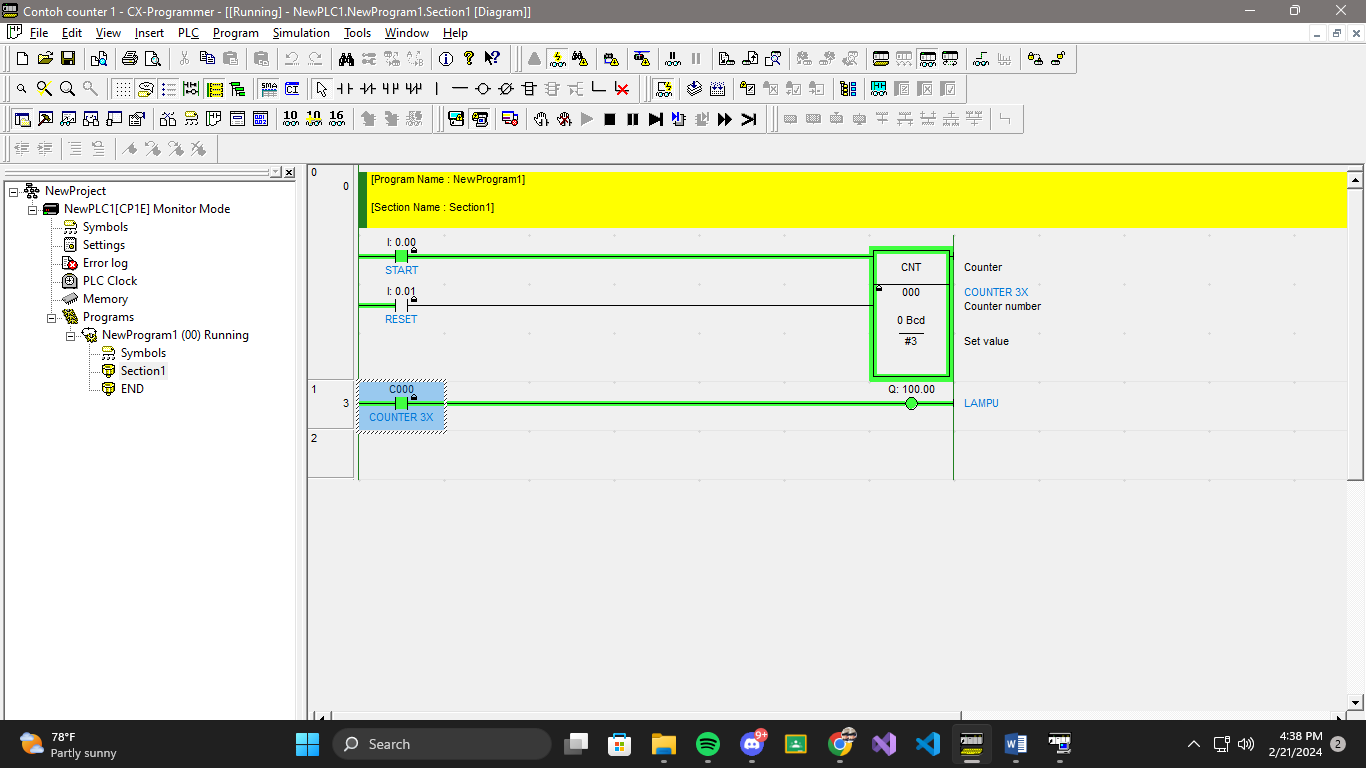
## Praktikum

### Percobaan 1: Counter

**Hasil dan Analisis**

Kondisi jika input start di on kan output belum menyala

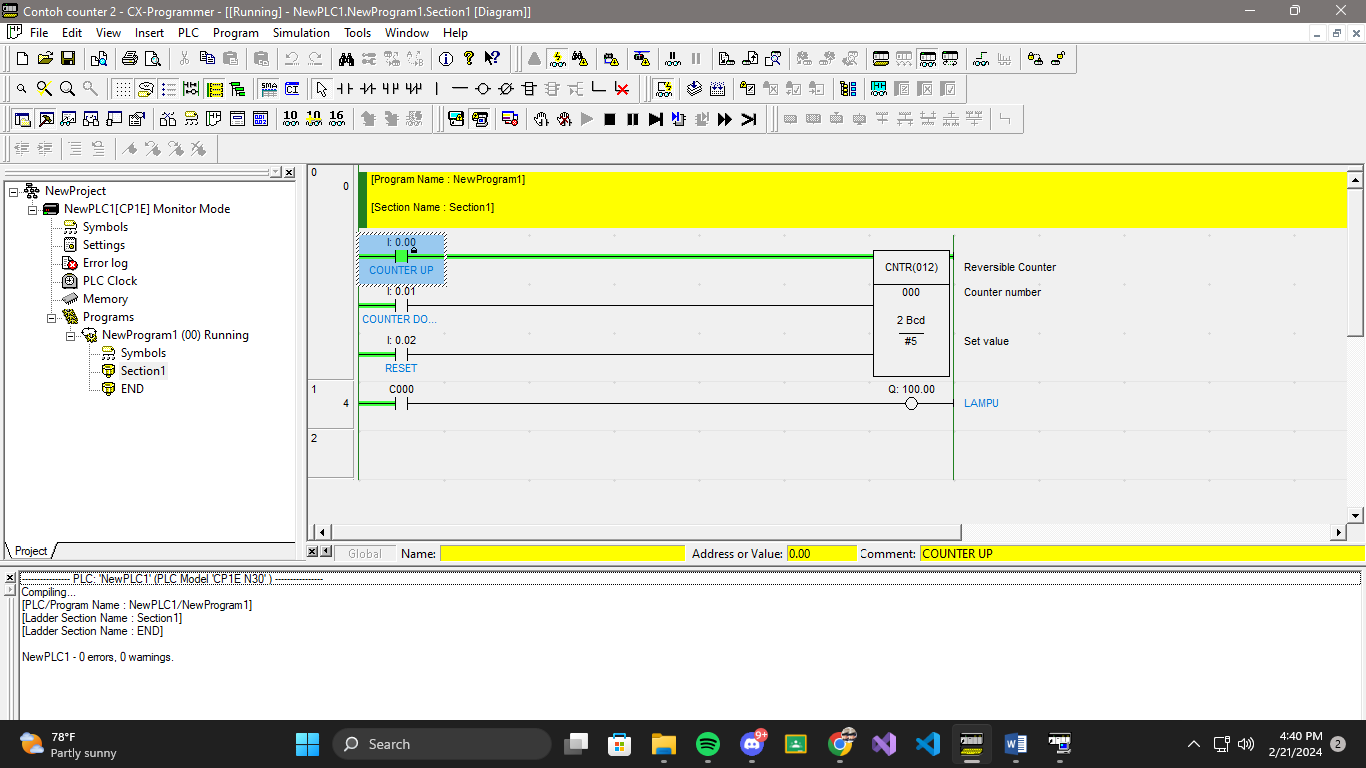
Kondisi saat C000 di on kan maka output akan menyala

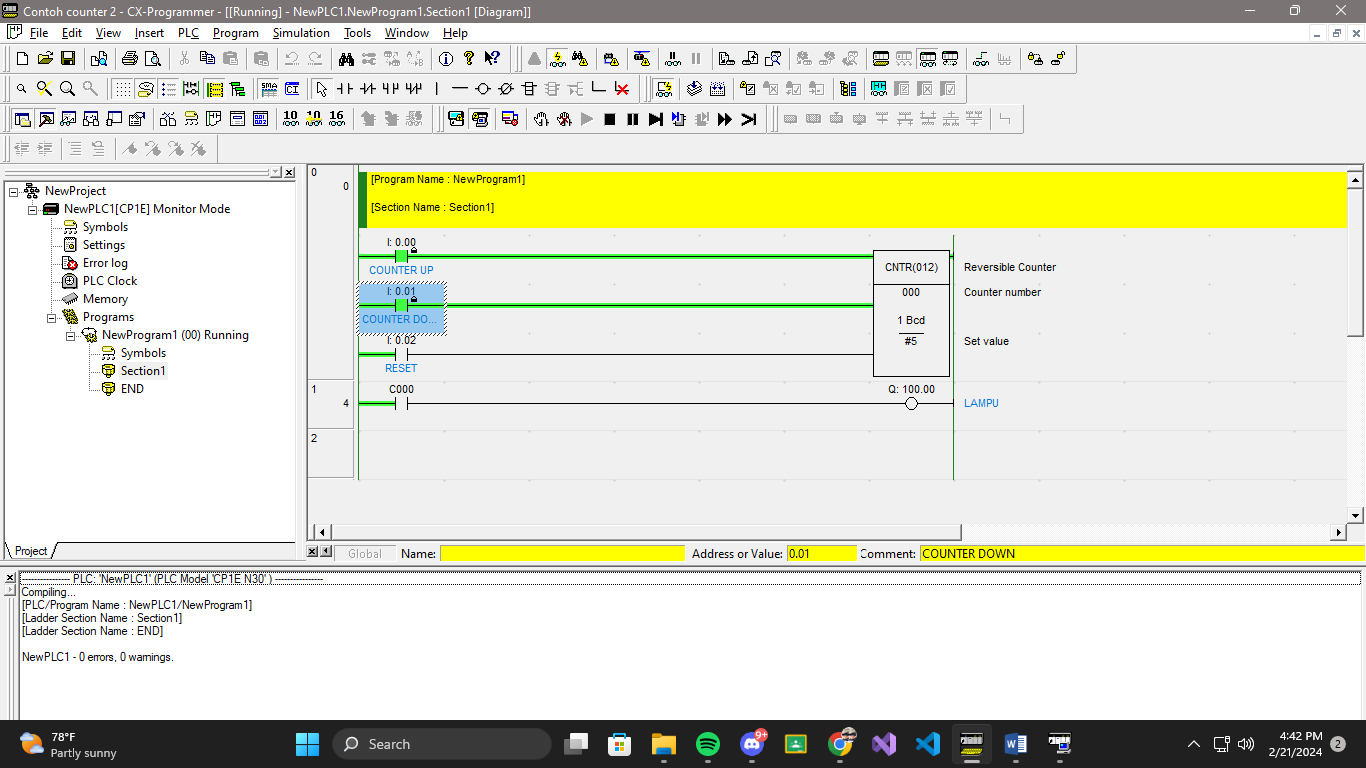


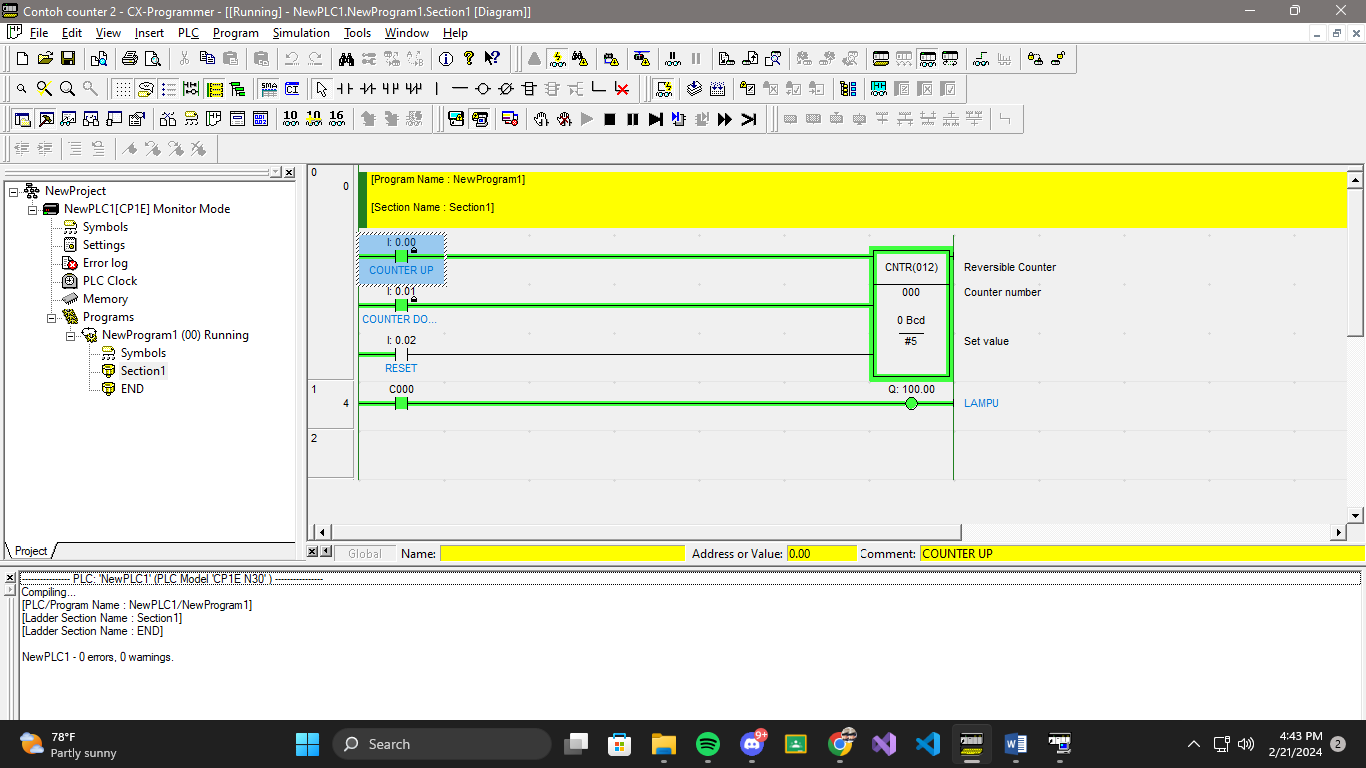
### Percobaan 2: Counter Up/Down

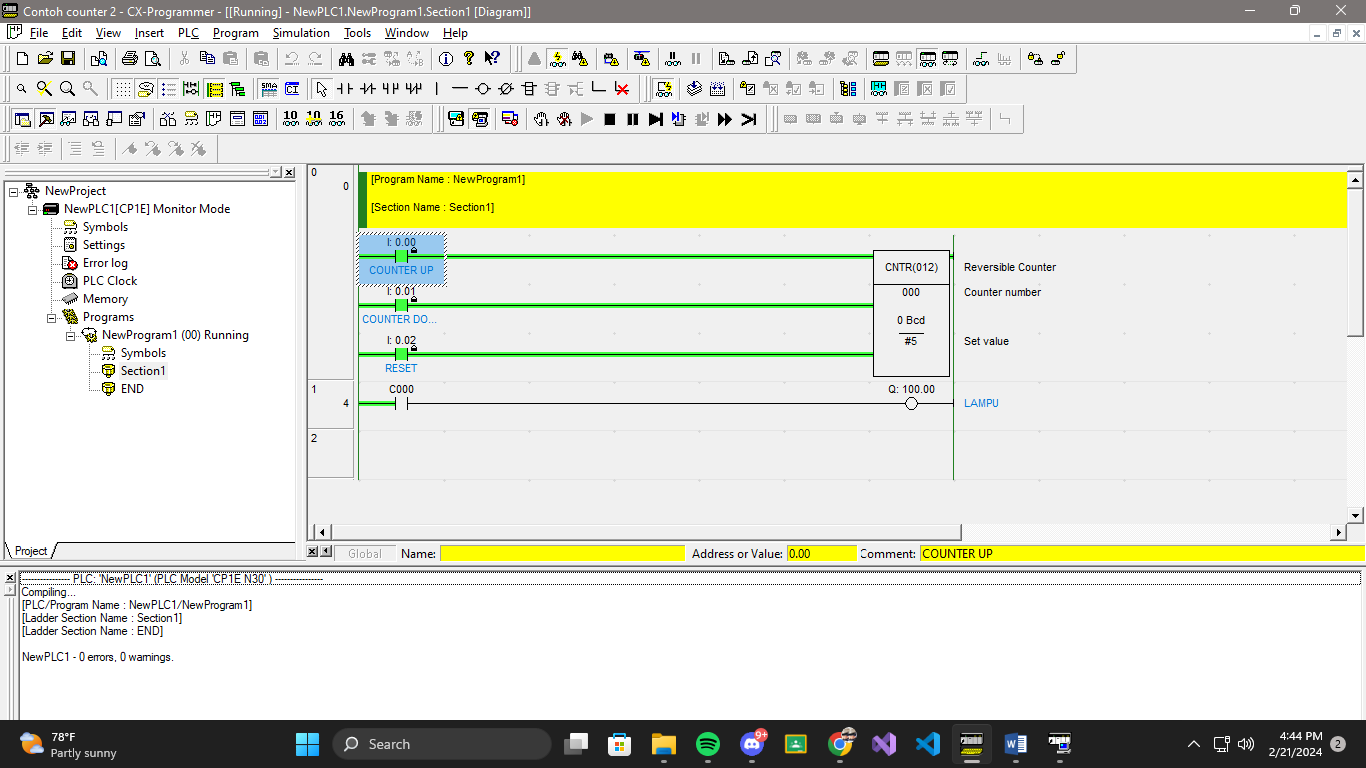
**Hasil dan Analisis**

Jika counter up di tekan maka pv counter akan terisi oleh 1 jika sekali tekan



 Jika Counter down ditekan maka pv counter akan mengurangi 1 jika sekali ditekan

 Jika pv counter sudah mencapai 5 maka output akan menyala

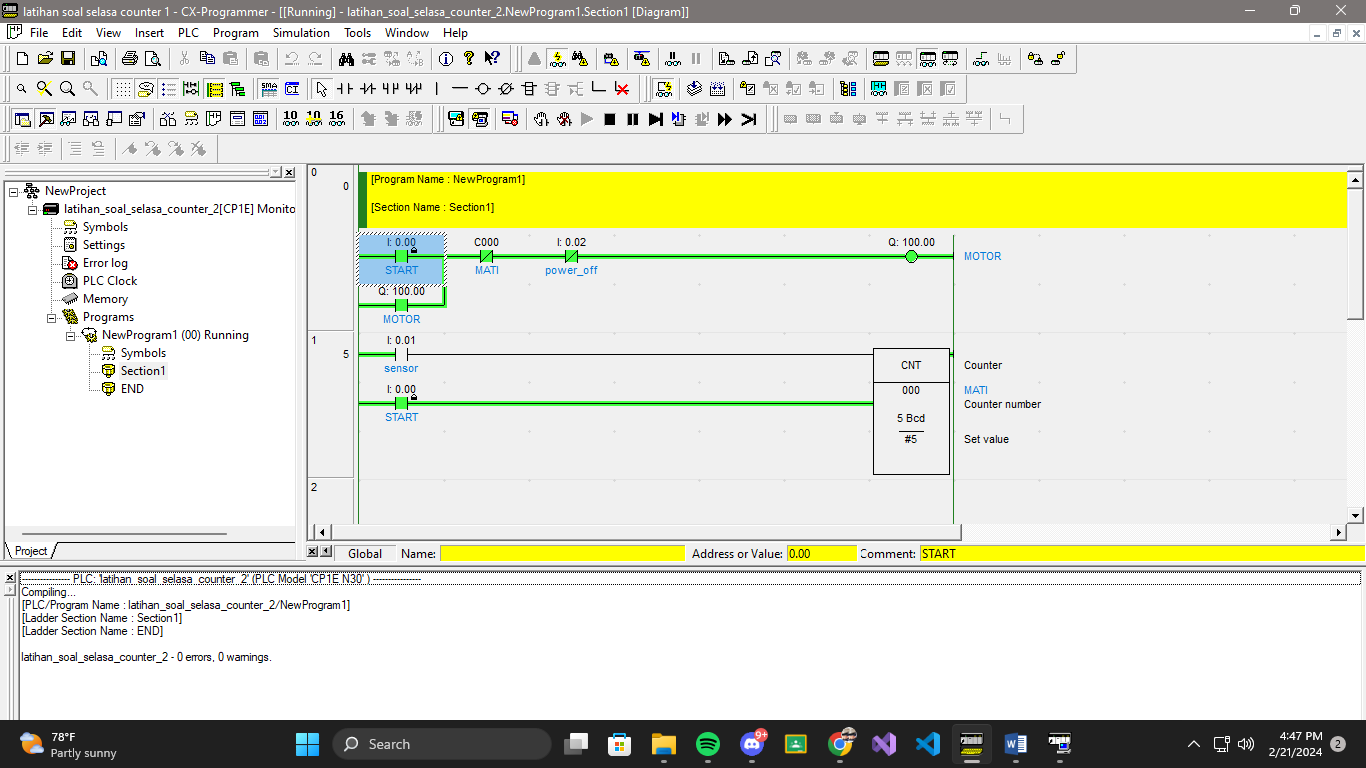
 Jika ingin memulihkan dari awal tekan reset

## Soal Latihan

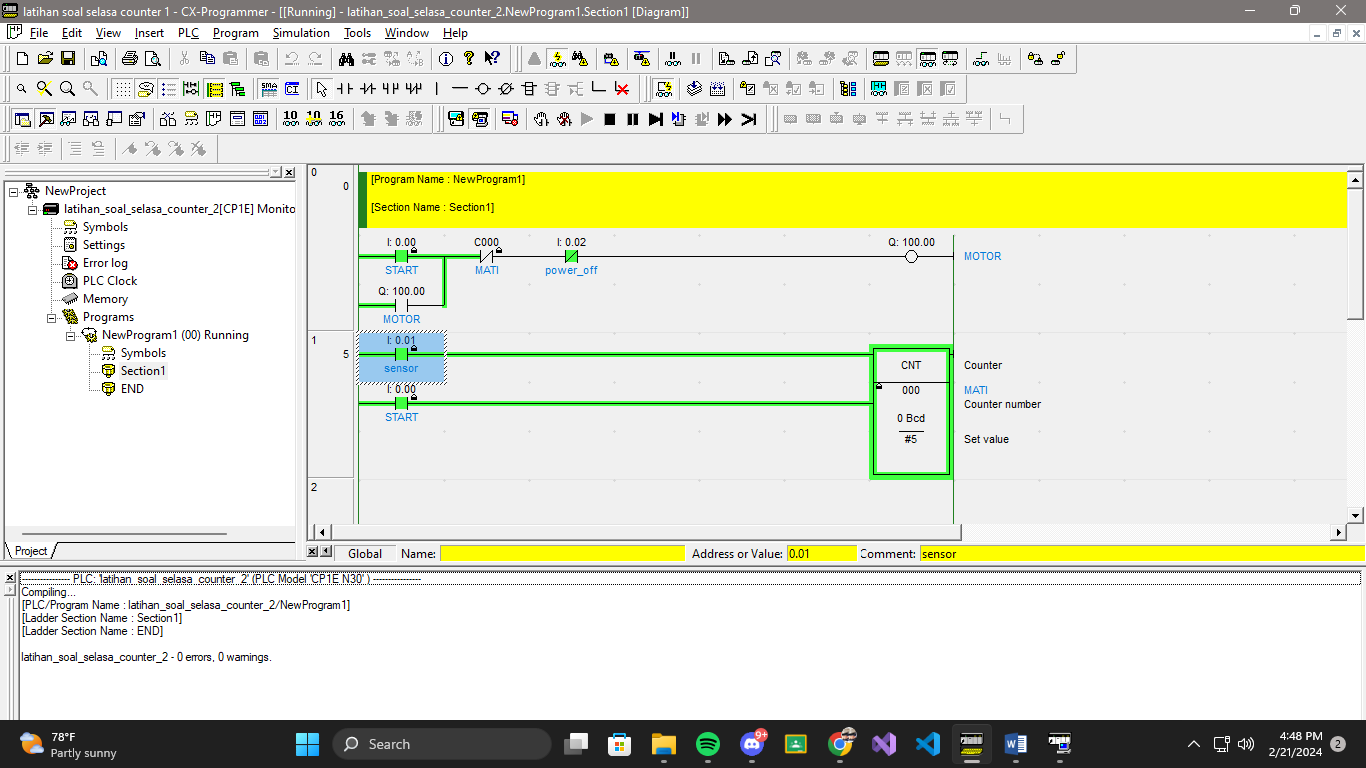
Buatlah program untuk soal berikut:

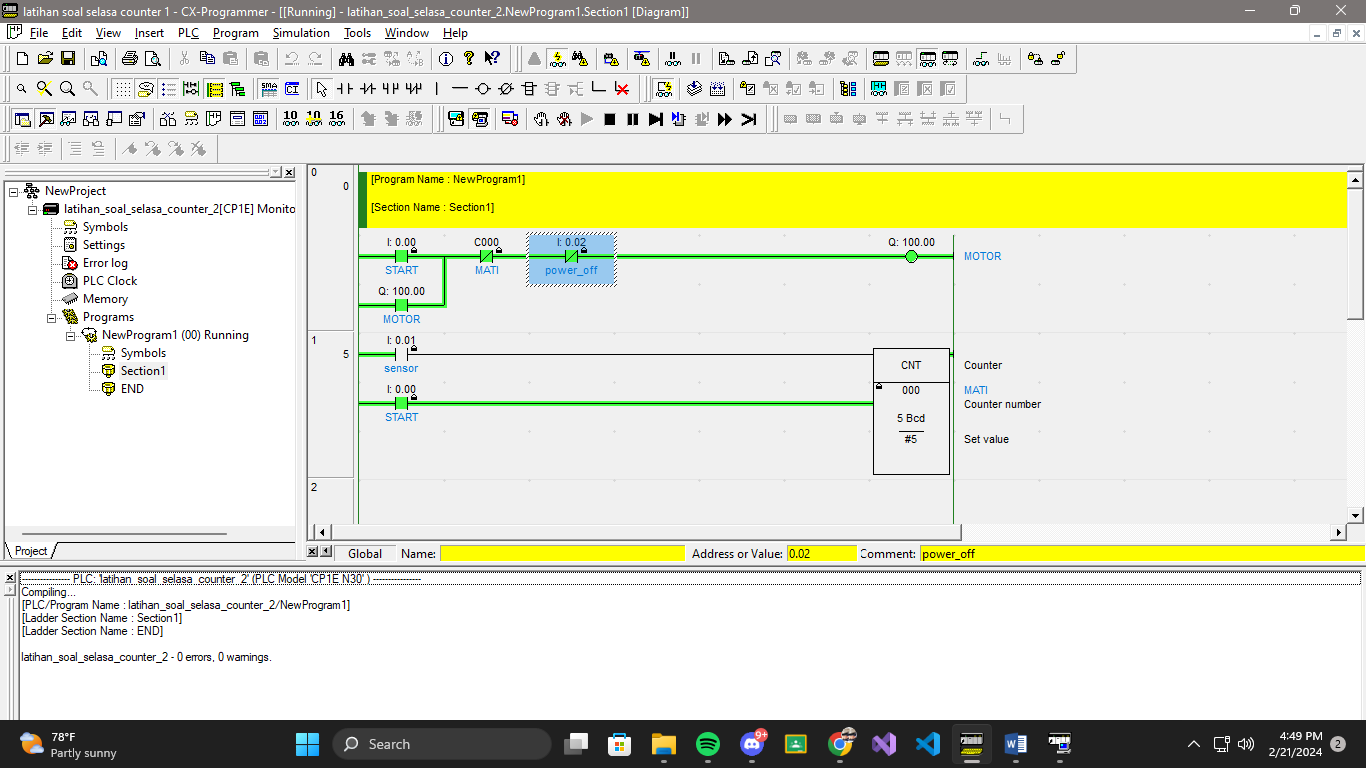
Terdapat suatu konveyor sederhana, jika *Input start* di-*on*-kan maka motor konveyor akan *on*. Jika motor konveyor *on* maka sensor akan mendeteksi produk yang melewatinya. Jika boks sudahterisi sampai 5 produk maka motor konveyor akan *off*. Untuk meng-*on*-kan motor konveyor *Input start* harus di-*on*-kan lagi.

**Hasil dan Analisis**

Jika Input start di-on-kan maka motor konveyor akan on. Jika motor konveyor on maka sensor akan mendeteksi produk yang melewatinya.

Jika boks sudah terisi sampai 5 (menekan sensor sebanyak 5 kali) produk maka motor konveyor akan off.



 Untuk meng-on-kan motor konveyor Input start harus di-on-kan lagi.

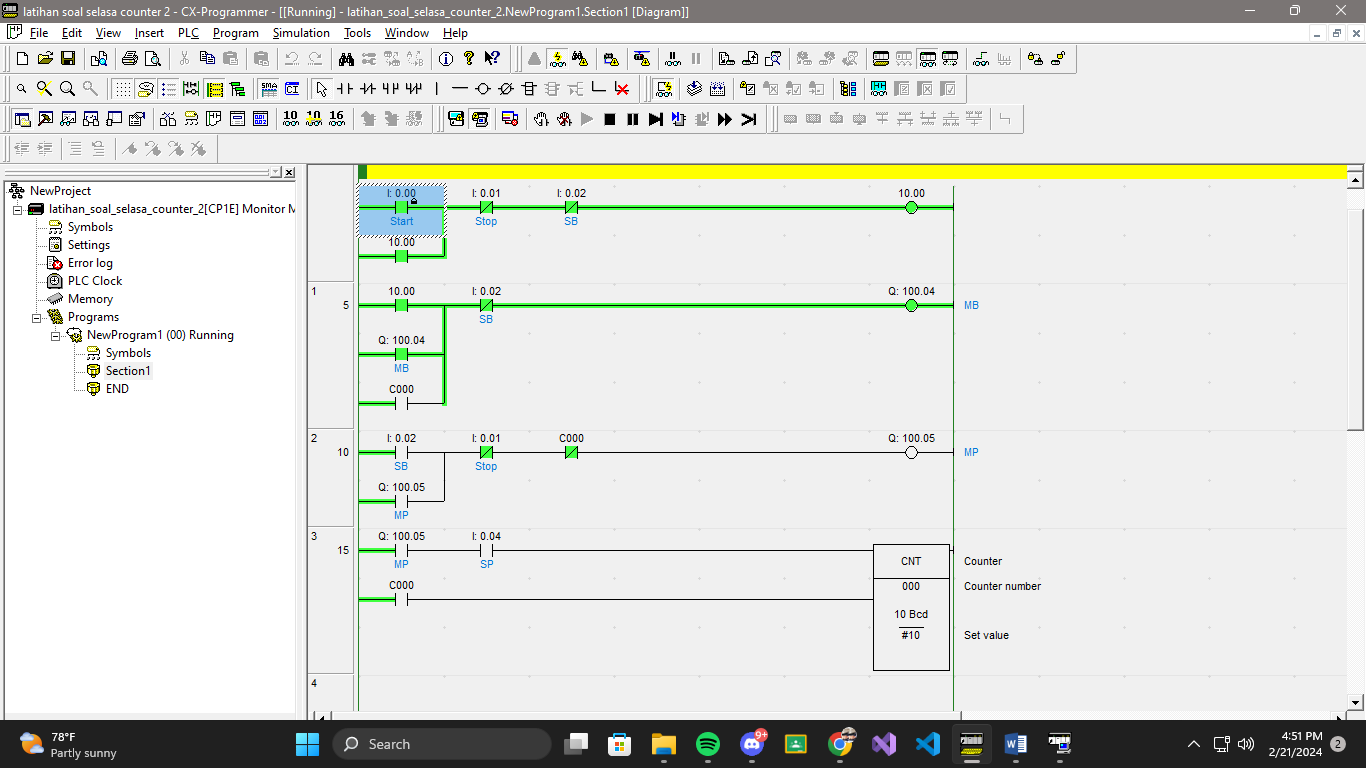
Buatlah program untuk soal berikut:

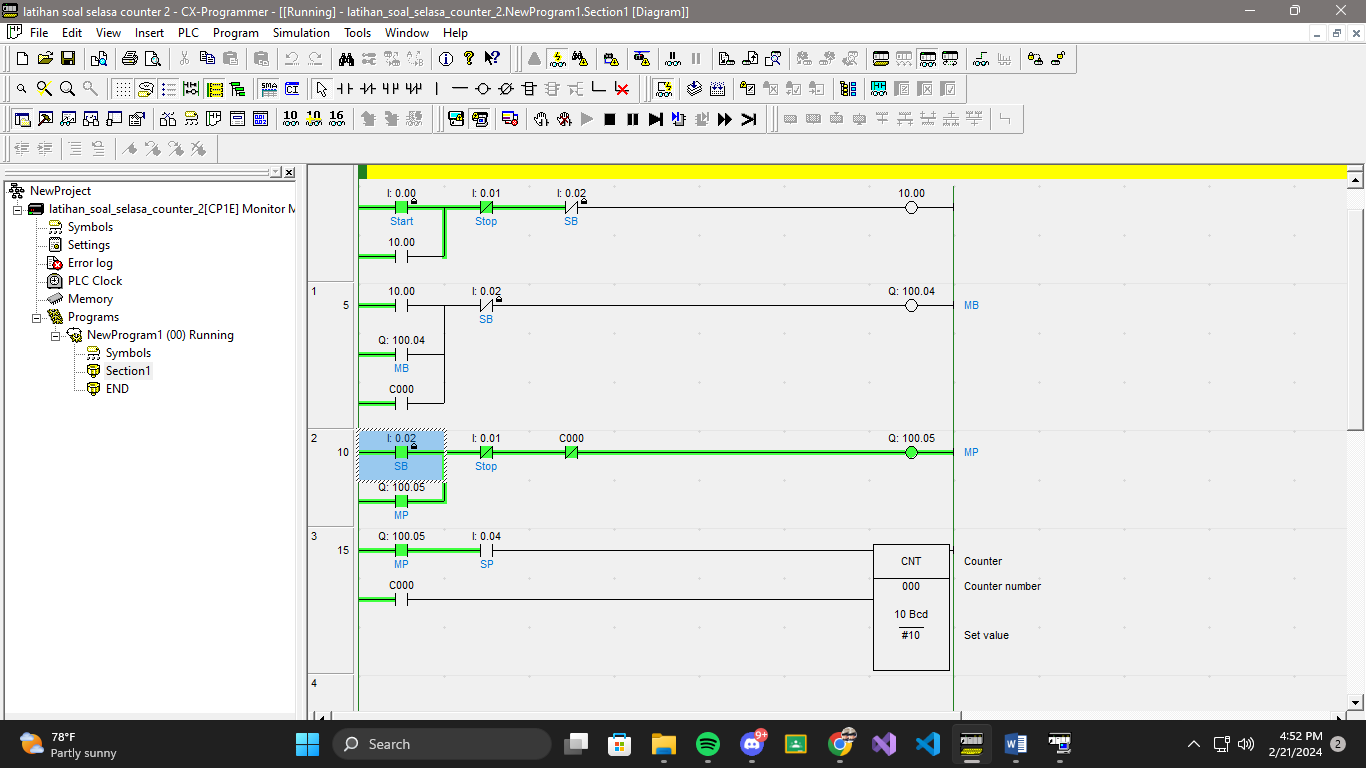
Terdapat dua konveyor yang bekerja, yaitu konveyor boks dan konveyor produk. Jika *Input start* di-*on*-kan maka konveyor boks akan berjalan, dan berhenti ketika SB(Sensor Boks) mendeteksi adanya boks. Begitu konveyor boks berhenti maka konveyor produk akan jalan dan SP (Sensor

Produk) akan mendeteksi produk yang melewatinya. Jika sudah 10 produk yang beradadalam boks, konveyor produk akan berhenti dan konveyor boks akan berjalan lagi tanpa harus meng-*on*-kan *Input start*. Proses akan berlangsung terus sampai *Input stop* diaktifkan

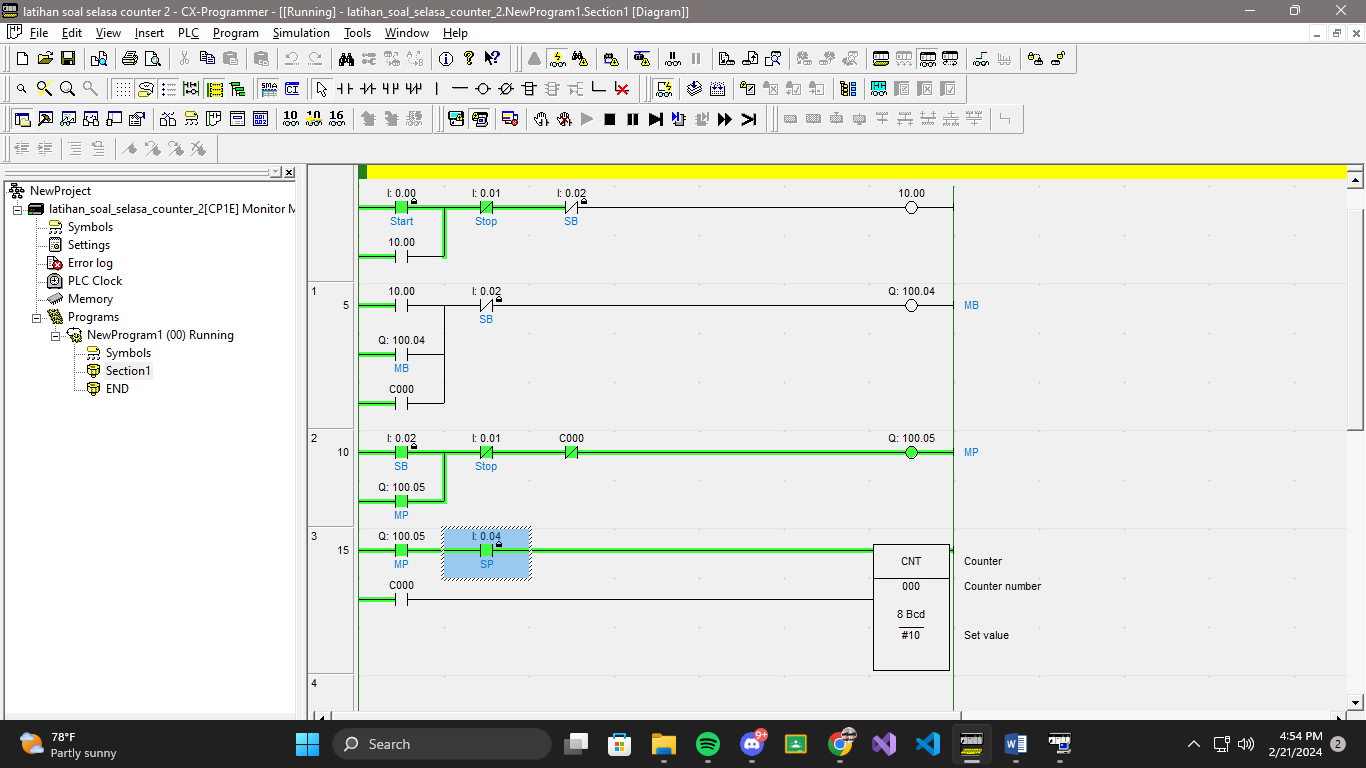
**Hasil dan Analisis**

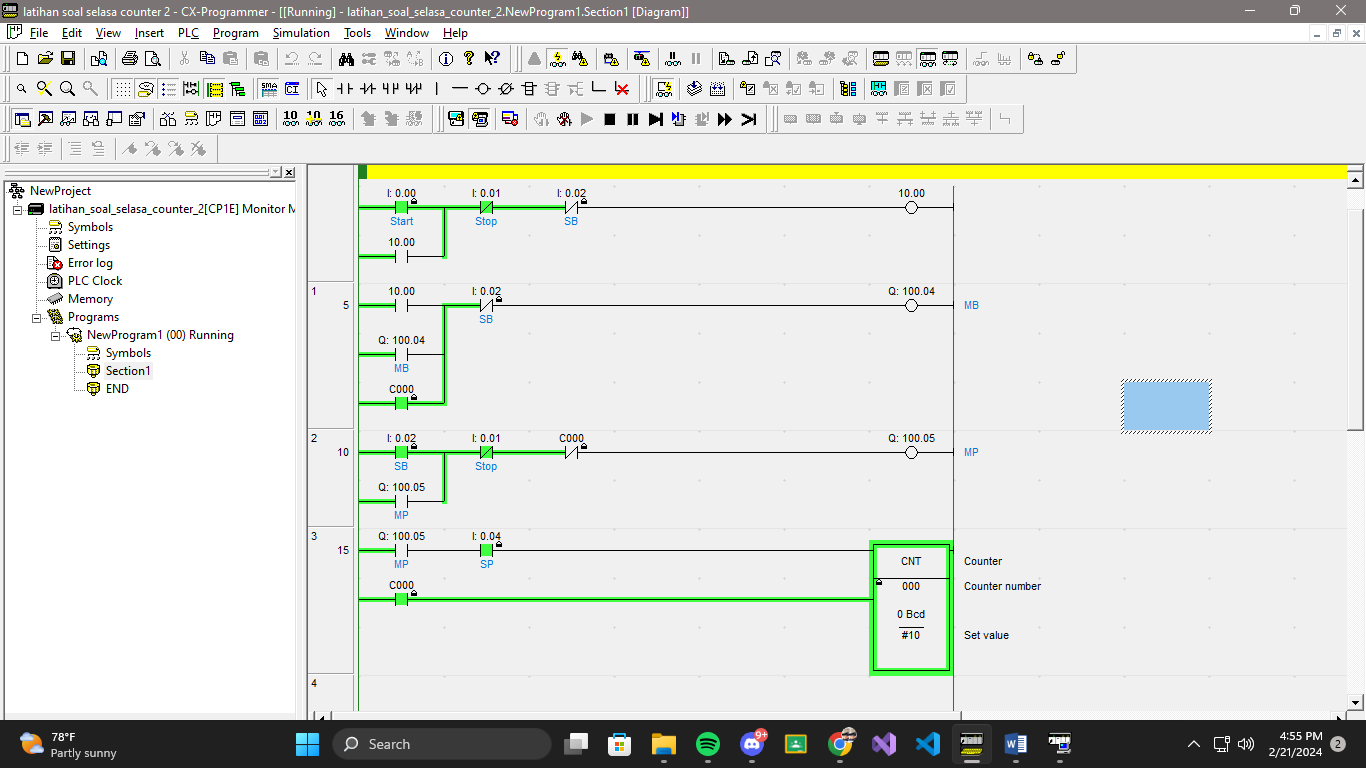
Jika Input start di-on-kan maka konveyor boks akan berjalan, dan berhenti ketika SB(Sensor Boks) mendeteksi adanya boks.





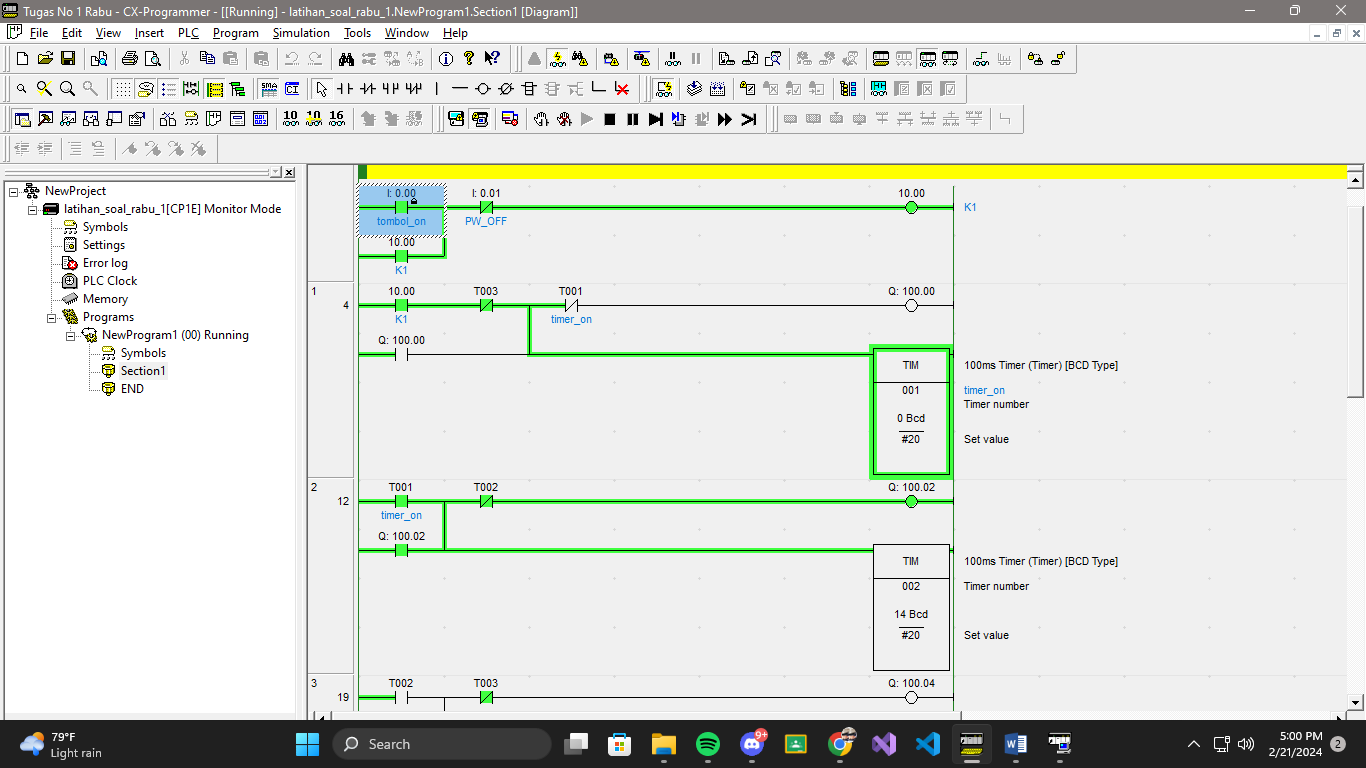
Begitu konveyor boks berhenti maka konveyor produk akan jalan dan SP (Sensor Produk) akan mendeteksi produk yang melewatinya dengan cara menekan SP. Jika sudah 10 produk yang beradadalam boks, konveyor produk akan berhenti. Proses akan berlangsung terus sampai Input stop diaktifkan.



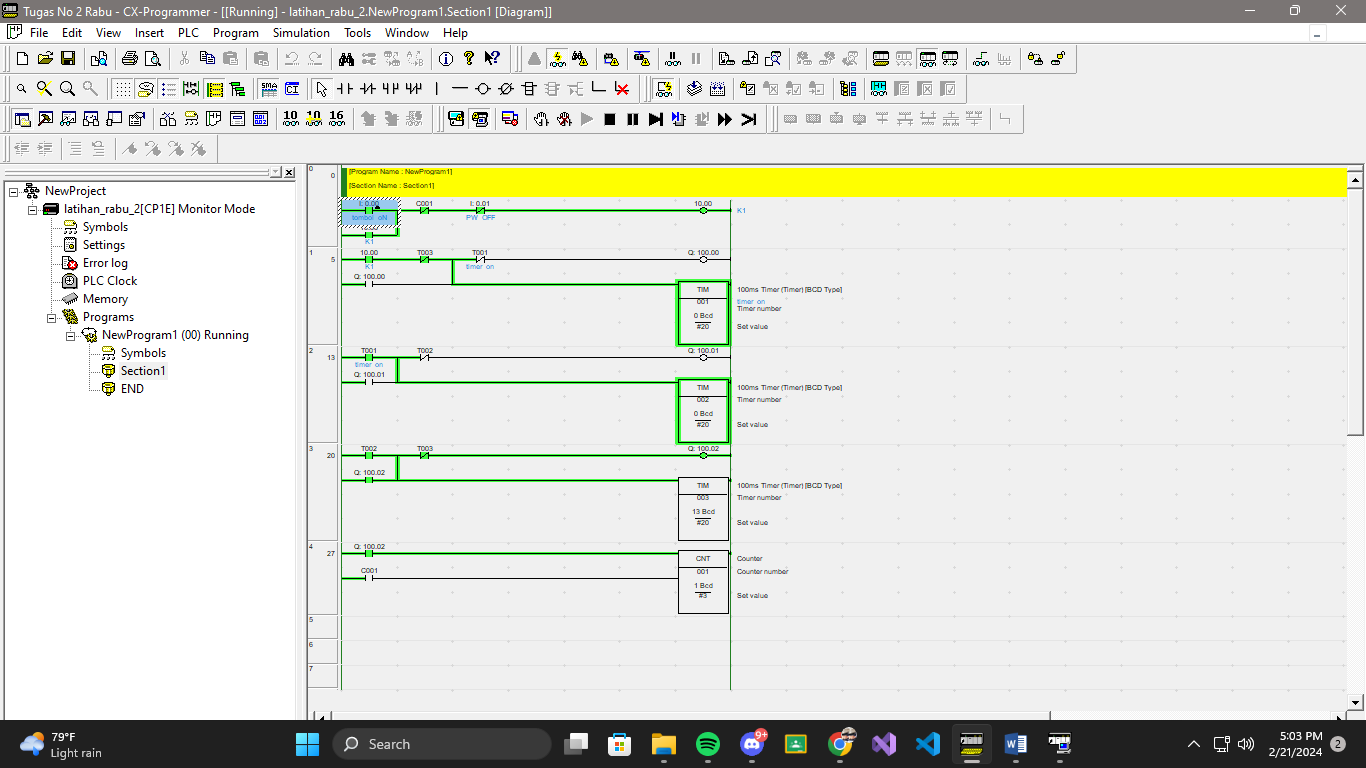


**SOAL TAMBAHAN**

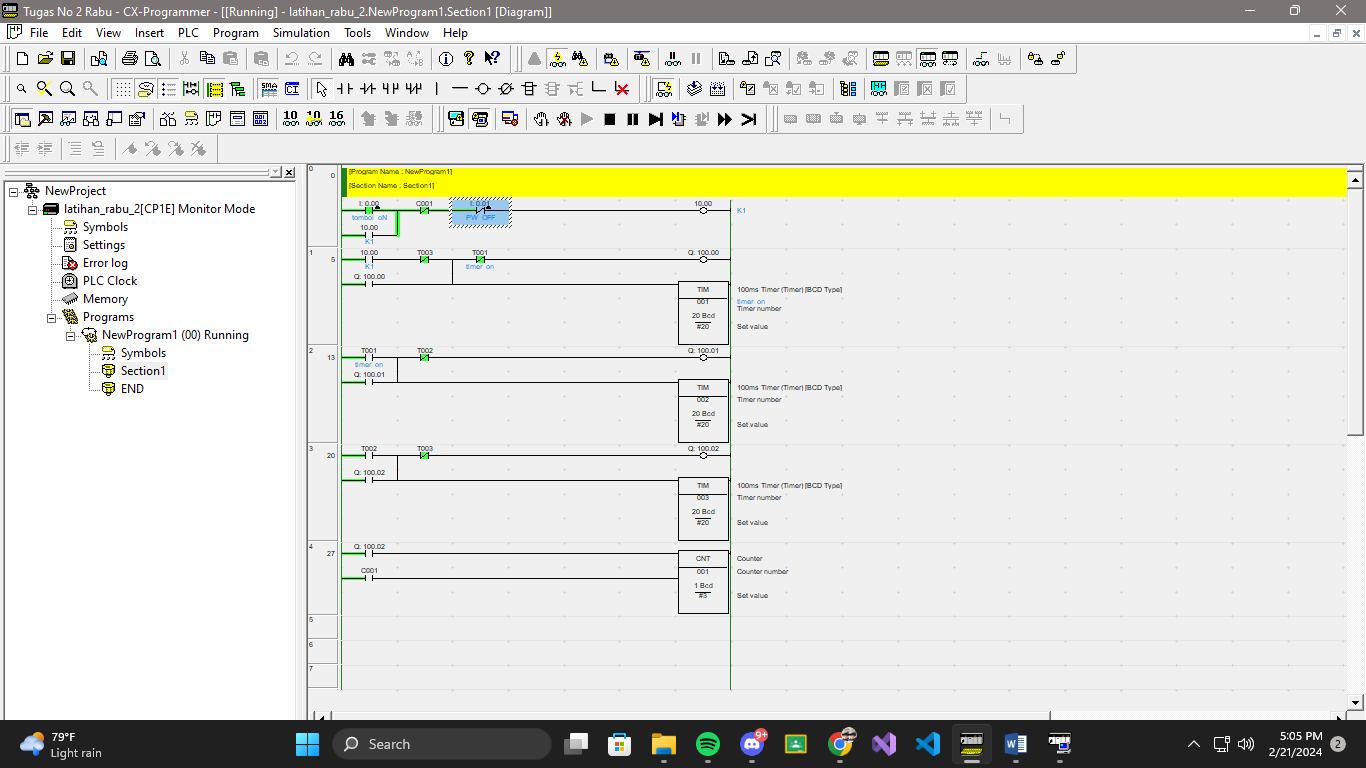
Jika tombol on ditekan maka lampu 1 akan menyala selama 2 detik begitupun dengan lampu 2 dan 3 memiliki siklus yang sama dan terjadi perulangan terus menerus. Jika tombol stop ditekan lampu akan memati tetapi melewati hingga lamou terakhir dahulu lalu mati

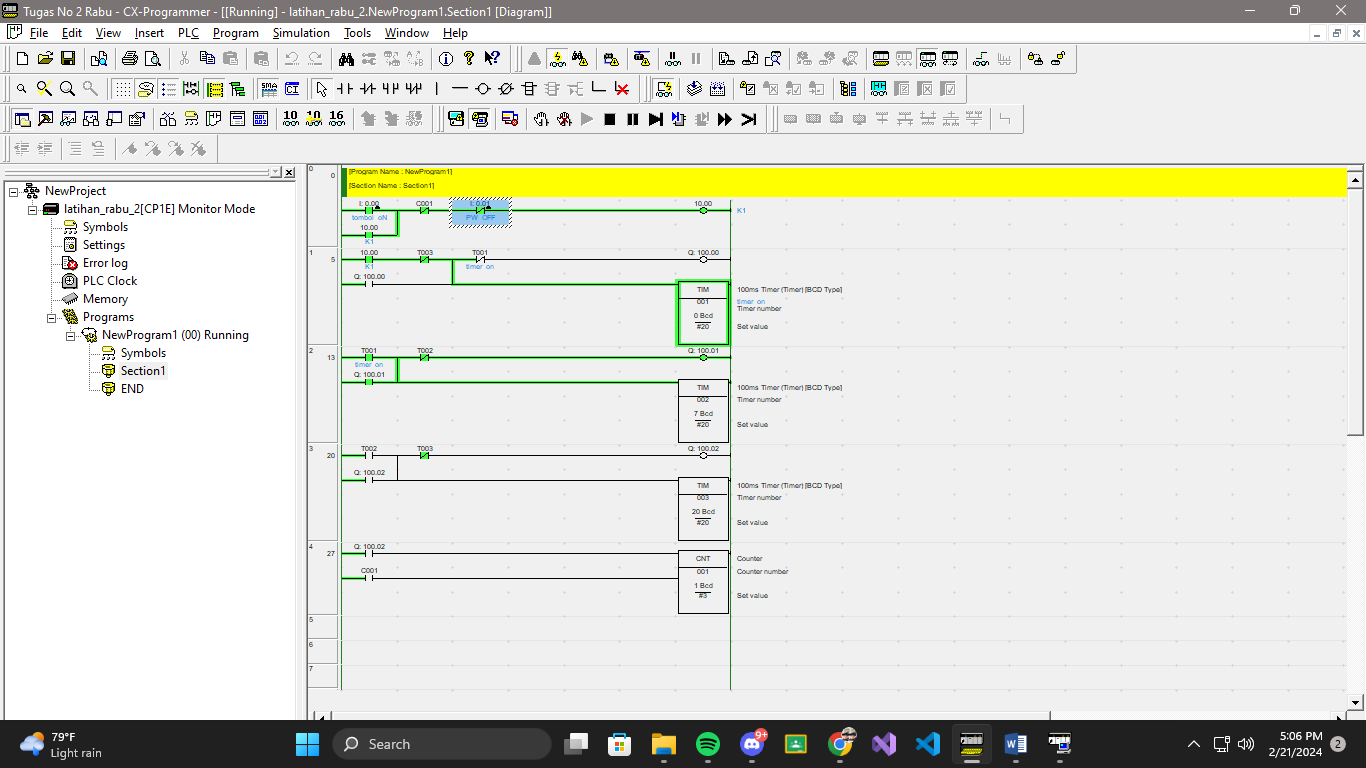


Jika tombol on ditekan maka lampu 1,2 dan 3 akan menyala dengan selang 2 detik. Tetapi hanya memiliki 3 siklus



Jika tombol stop ditekan tetapi siklus belum mencapai 3 maka ketika di start akan menghitung sisa siklus terlebih dahulu





# PRAKTIKUM 4 INSTRUKSI DASAR PLC – DIFU & DIFD

**Tujuan :**

1. Memahami konsep DIFU yang digunakan
2. Mengetahui penggunaan DIFD yang digunakan
3. Mampu memprogram instruksi DIFU dan DIFD menggunakan perangkat lunak pemrograman PLC.
4. Melakukan simulasi program yang melibatkan instruksi DIFU dan DIFD

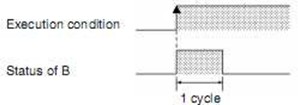
**Sub Pokok Bahasan :** Differentiate Up, Differentiate Down.

## Teori Dasar

**DIFFERENTIATE UP/DOWN**: DIFU(013) dan DIFD(014)

DIFU(013) akan membuat sebuah *bit* menjadi *on* selama satu siklus ketika kondisi

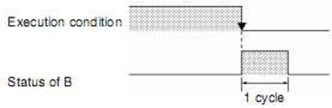
*input* dari *off* menuju *on* (*rising edge*).



DIFU merupakan singkatan dari Differentiate Up, kontrol ini berfungsi untuk meng-ON kan output selama satu scan. Cara kerjanya, apabila input ditekan maka output DIFU akan ON dan kontak DIFU akan hidup hanya sekejap walaupun input ditekan lama. Jadi DIFU ini bekerja pada saat transisi naik dari OFF ke ON.

DIFD(014) akan membuat sebuah *bit* menjadi *on* selama satu siklus ketika kondisi

*input* dari *on* menuju *off* (*falling edge*).



DIFD merupakan singkatan dari Differentiate Down, berfungsi meng-ON kan output selama satu scan. Bedanya, DIFD ini bekerja pada saat transisi turun yaitu perubahan kondisi dari ON ke OFF.

## Alat Dan Bahan

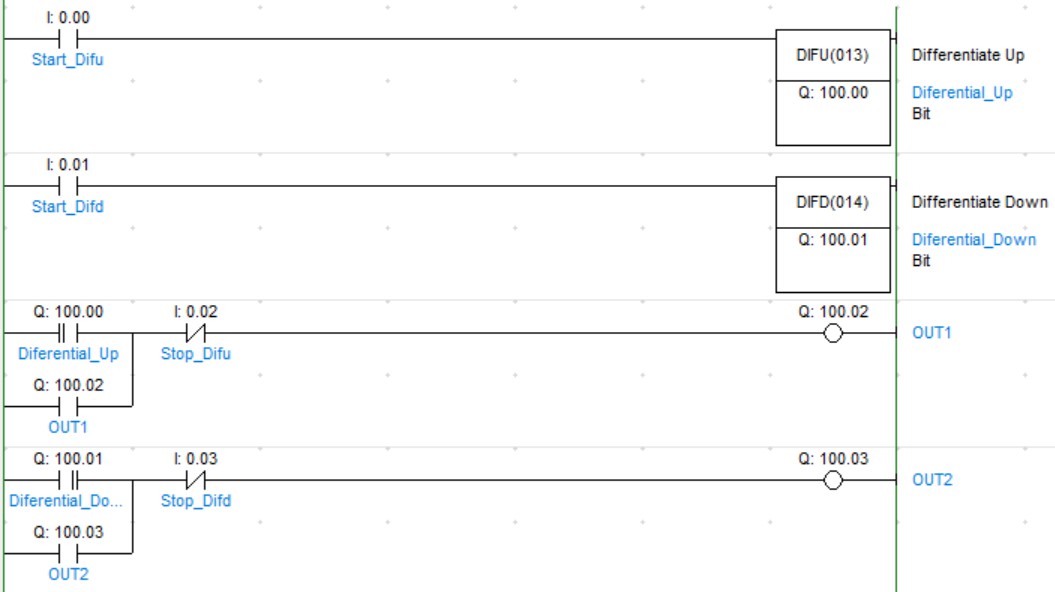
* PLC Omron CP1E-N30
* Laptop
* Kabel jumper
* Kabel USB

## Petunjuk Praktikum

1. Pertama yaitu download aplikasi CXONE\_v4.40 terlebih dahulu. Ikutilah langkah-langkah penginstalannya.
2. Setelah selesai terinstall,buka lah aplikasinya. Lalu kita pilih file lalu *New...* , isi device name dan device type sesuai PLC yang anda punya. Disini saya menggunakan PLC dengan device type CP1E-N30.
3. Jika sudah maka tekan OK barulah kita bisa membuat rangkaiannya.
4. Pasang PLC pada panel kontrol atau lokasi yang sesuai.
5. Sambungkan PLC ke sumber daya listrik sesuai dengan spesifikasi produsen.
6. Hubungkan laptop dengan PLC menggunakan kabel yang sesuai. Pastikan bahwa koneksi antara laptop dan PLC berfungsi dengan baik.
7. Pasang modul input/output ke slot yang sesuai di PLC.
8. Hubungkan perangkat input dan output di lapangan ke modul input/output.
9. Gunakan fasilitas simulasi jika tersedia untuk menguji program tanpa perlu terhubung ke PLC fisik.
10. Simulasikan kondisi input dan periksa respons output sesuai dengan logika program.
11. Setelah pemrograman dan simulasi selesai, transfer program ke memori PLC menggunakan koneksi laptop. Pastikan transfer program berjalan tanpa kesalahan.
12. Monitor operasi PLC melalui laptop untuk memastikan bahwa program berjalan sesuai dengan harapan.
13. Gunakan fasilitas debugging untuk mengidentifikasi dan memperbaiki kesalahan dalam program.

## Praktikum

### Percobaan 1: DIFU & DIFD



**Hasil dan Analisis**

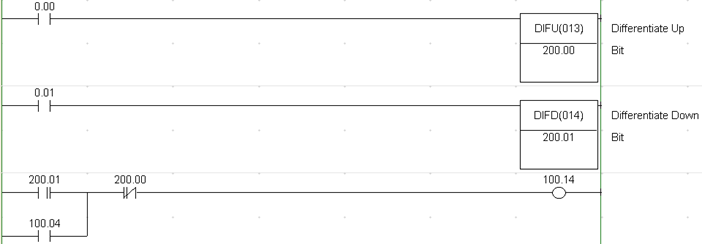
* Saat input 0.00 dinyalakan maka output 100.00 akan menyala dan menyalakan kontak 100.00 dan menyalakan Output1 atau 100.02.
* Output2 kita dapat menyalakannya dengan mengaktifkan input 0.01, tetapi kontak 100.01 belum menyala sebelum output 0.01 dimatikan. Setelah 0.01 dinonaktifkan barulah kontak 100.01 akan aktif dan mengaktifkan output 100.03

A screenshot of a computer

Description automatically generated

## Soal Latihan

Buatlah program untuk soal berikut:

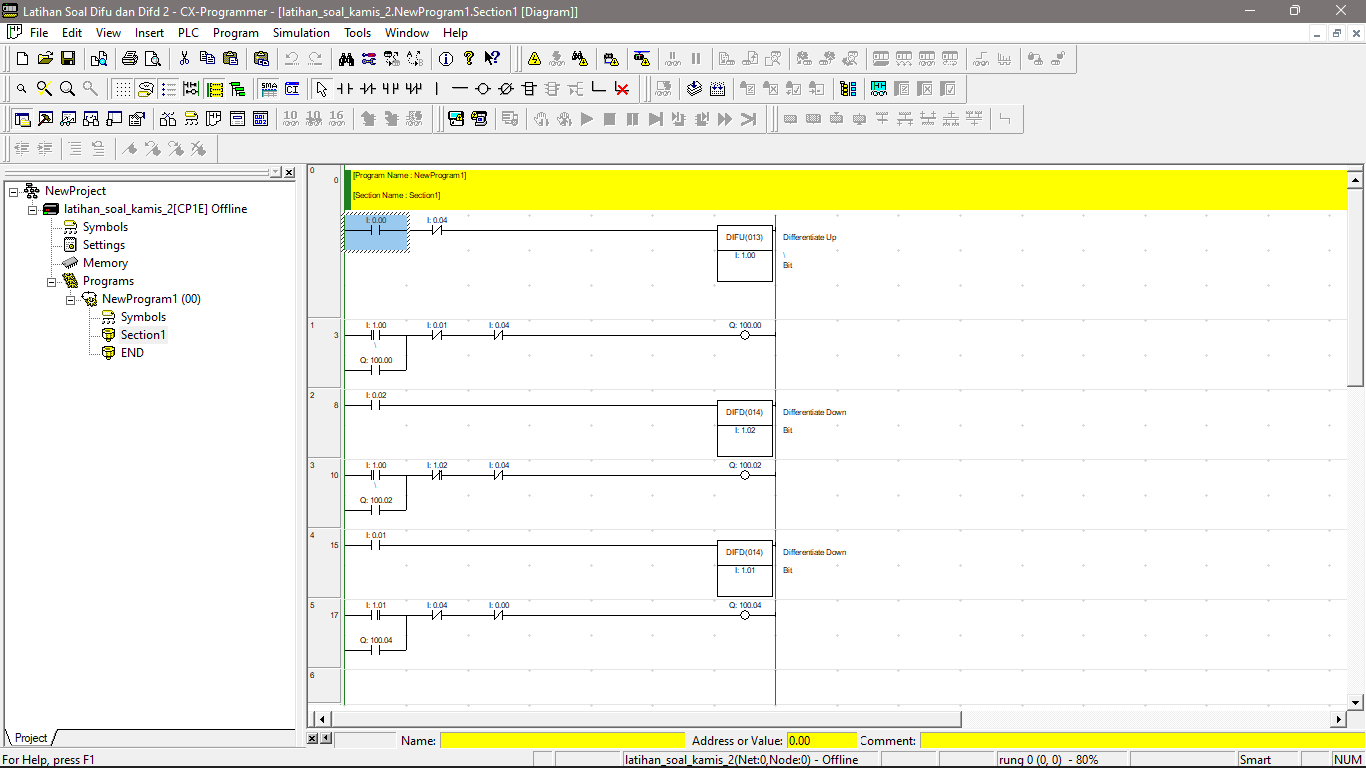


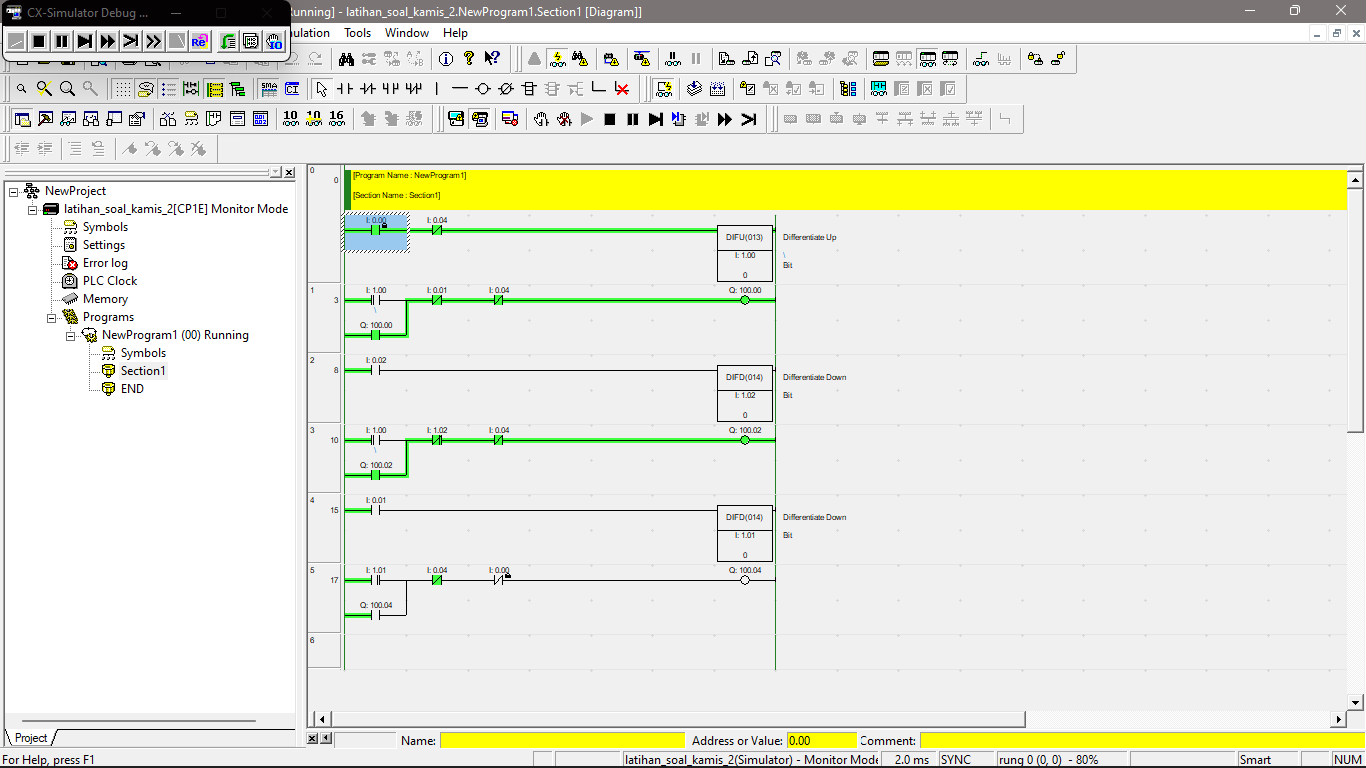
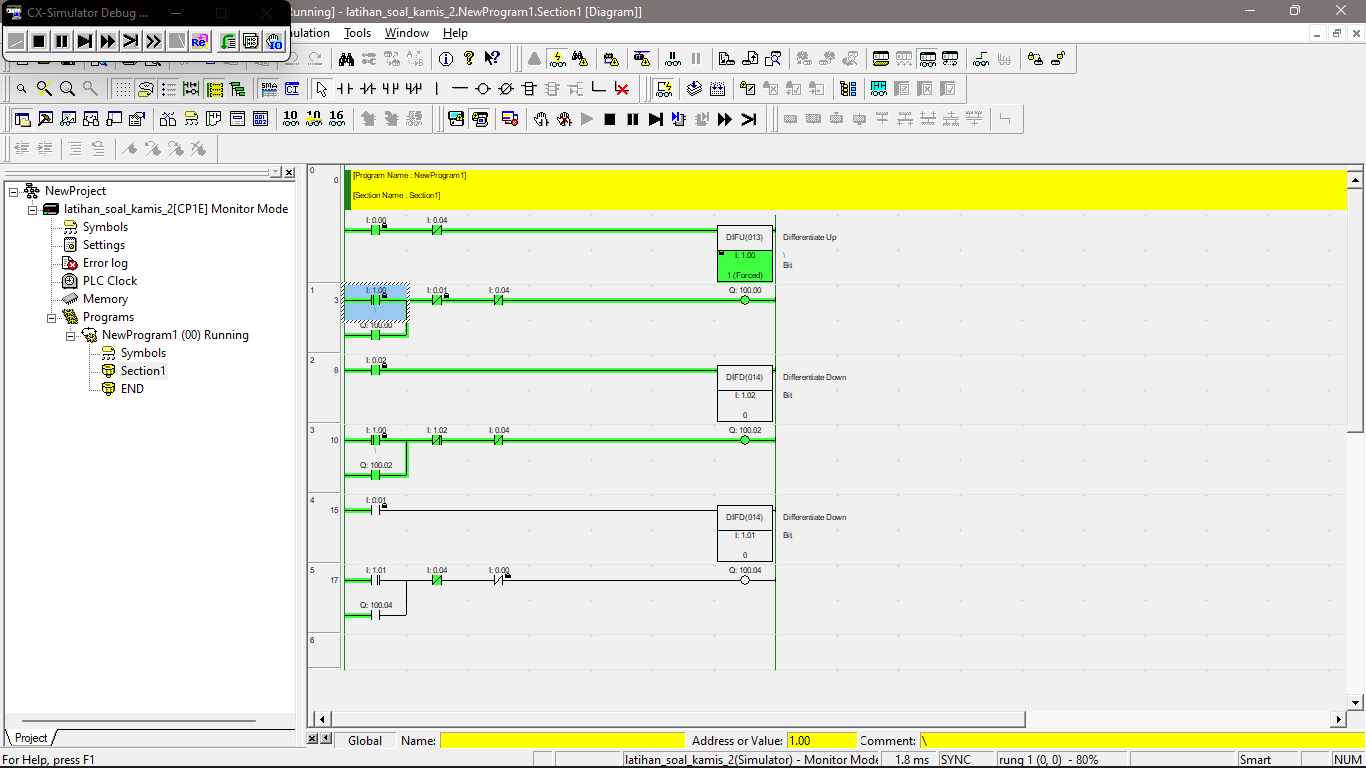
**Hasil dan Analisis**

A screenshot of a video

Description automatically generated

* Saat input 0.00 diaktifkan maka tidak ada perubahan apapun. Tetapi saat output menyala, input 0.00 dapat mematikan output yang sedang aktif.
* Saat input 0.01 diaktifkan maka tidak ada perubahan terhadap output, tetapi saat input 0.01 dimatikan maka akan menyalakan ouput 100.04.
* Differentiate Up (UP):
* Ketika input berubah dari off ke on: Instruksi Differentiate Up akan merespon dengan menghasilkan sinyal output (on) pada saat terjadinya perubahan dari off ke on.
* Ketika input tetap on atau berubah dari on ke off: Output akan tetap mati (off).
* Differentiate Down (DOWN):
* Ketika input berubah dari on ke off: Instruksi Differentiate Down akan merespon dengan menghasilkan sinyal output (on) pada saat terjadinya perubahan dari on ke off.
* Ketika input tetap off atau berubah dari off ke on: Output akan tetap mati (off).

*Input* dari sensor ultrasonik (0.00) digunakan untuk mendeteksi kendaraan yang akan masuk, yang akan memerintahkan motor listrik (1.00) untuk membuka pintu. Sebuah sensor proksimiti (0.02) akan memerintahkan motor berhenti bekerja membuka pintu jika pintu sudah sampai atas. Sebuah sensor inframerah (0.01) akan memberi sinyal ke PLC jika kendaraan sudah melewati pintu, dan memerintahkan motor listrik (1.01) untuk menutup pintu, dan sebuah *limit switch* (0.04) akan tersentuh yang menyebabkan motor listrik berhenti bekerja.



# PRAKTIKUM 5 INSTRUKSI DASAR PLC – SET & RESET

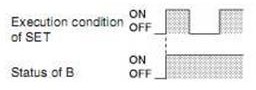
**Tujuan :**

1. Memahami konsep pengaturan (SET) dan mengatur ulang (RESET) keadaan logika dalam program PLC.
2. Mampu memprogram instruksi SET dan RESET menggunakan perangkat lunak pemrograman PLC.
3. Melakukan simulasi program yang melibatkan instruksi SET dan RESET

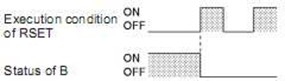
## Teori Dasar

**SET** dan **RESET: SET** dan **RSET**

SET akan membuat sebuah *bit* menjadi *on* selamanya jika kondisi eksekusinya *on*, dan tetap *on* walaupun *input*nya *off*.



RSET akan membuat sebuah *bit* menjadi *off* jika kondisi eksekusinya *on*.



## Alat Dan Bahan

* PLC Omron CP1E-N30
* Laptop
* Kabel jumper
* Kabel USB

## Petunjuk Praktikum

1. Pertama yaitu download aplikasi CXONE\_v4.40 terlebih dahulu. Ikutilah langkah-langkah penginstalannya.
2. Setelah selesai terinstall,buka lah aplikasinya. Lalu kita pilih file lalu *New...* , isi device name dan device type sesuai PLC yang anda punya. Disini saya menggunakan PLC dengan device type CP1E-N30.
3. Jika sudah maka tekan OK barulah kita bisa membuat rangkaiannya.
4. Pasang PLC pada panel kontrol atau lokasi yang sesuai.
5. Sambungkan PLC ke sumber daya listrik sesuai dengan spesifikasi produsen.
6. Hubungkan laptop dengan PLC menggunakan kabel yang sesuai. Pastikan bahwa koneksi antara laptop dan PLC berfungsi dengan baik.
7. Pasang modul input/output ke slot yang sesuai di PLC.
8. Hubungkan perangkat input dan output di lapangan ke modul input/output.
9. Gunakan fasilitas simulasi jika tersedia untuk menguji program tanpa perlu terhubung ke PLC fisik.
10. Simulasikan kondisi input dan periksa respons output sesuai dengan logika program.
11. Setelah pemrograman dan simulasi selesai, transfer program ke memori PLC menggunakan koneksi laptop. Pastikan transfer program berjalan tanpa kesalahan.
12. Monitor operasi PLC melalui laptop untuk memastikan bahwa program berjalan sesuai dengan harapan.
13. Gunakan fasilitas debugging untuk mengidentifikasi dan memperbaiki kesalahan dalam program.

## Praktikum

### Percobaan 1: SET & RESET

**Hasil dan Analisis**

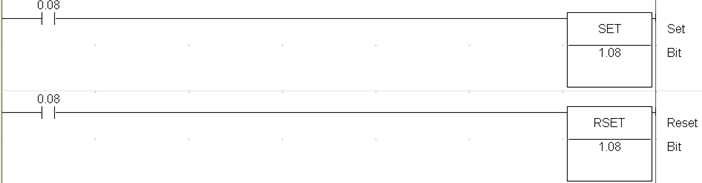
* Saat pushbutton 0.00 diaktifkan kemudian dinonaktifkan maka output 100.00 akan tetap menyala.
* Input pushbutton 0.01 diaktifkan kemudian dinonaktifkan?

Maka output yang tadi nya menyala akan mati oleh intruksi RESET.

* Jika kedua nya menyala maka output akan tetap mati.

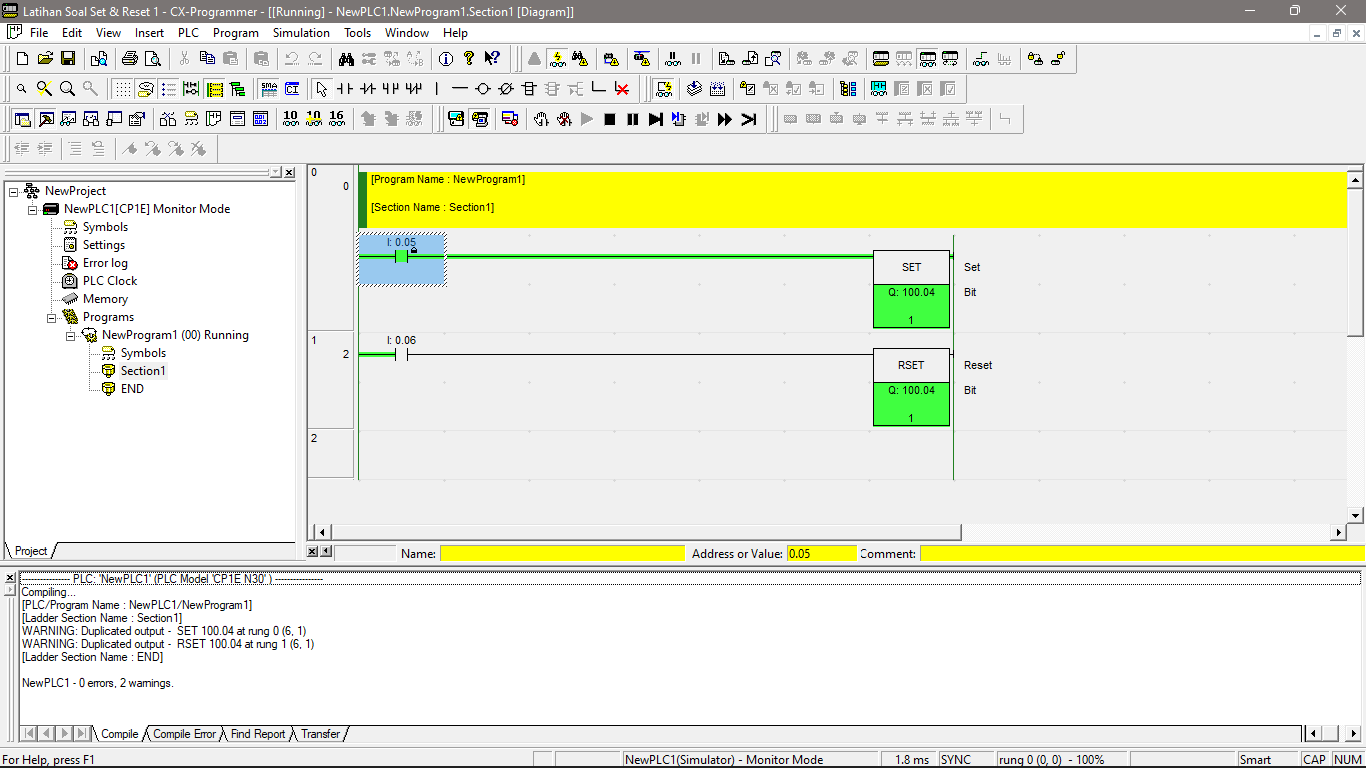
## Soal Latihan

Buatlah program untuk soal berikut:

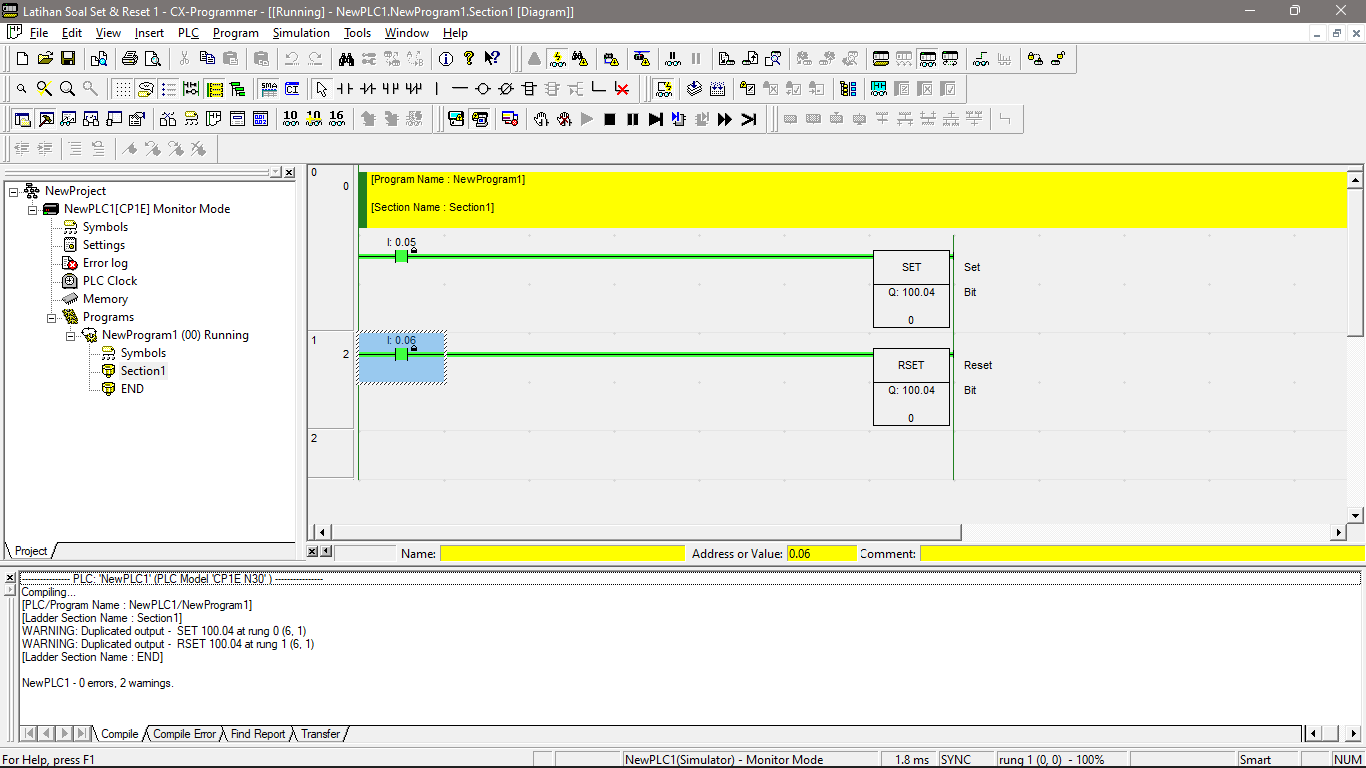


**Hasil dan Analisis**

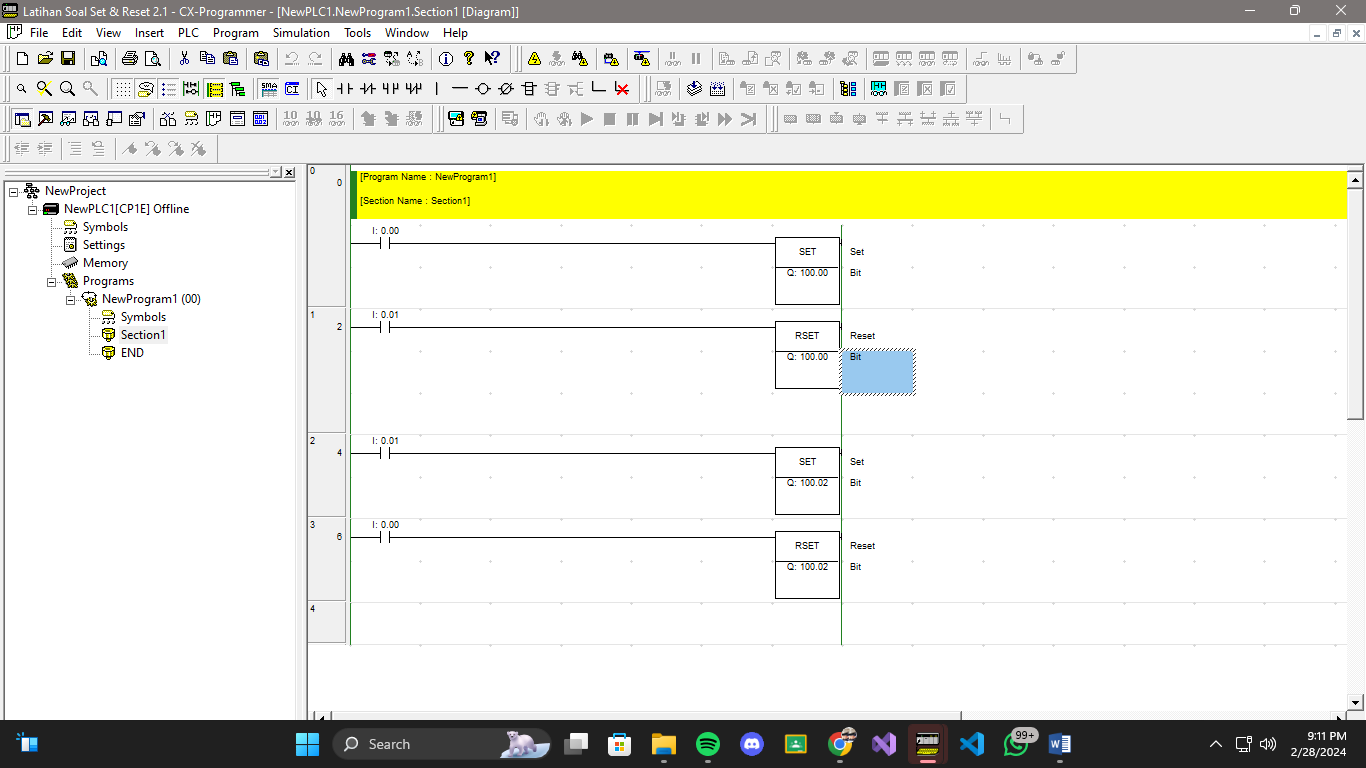
Jika input 0.05 ditekan maka output 100.04 akan menyala



Input 0.06 berguna untuk mematikan output 100.04

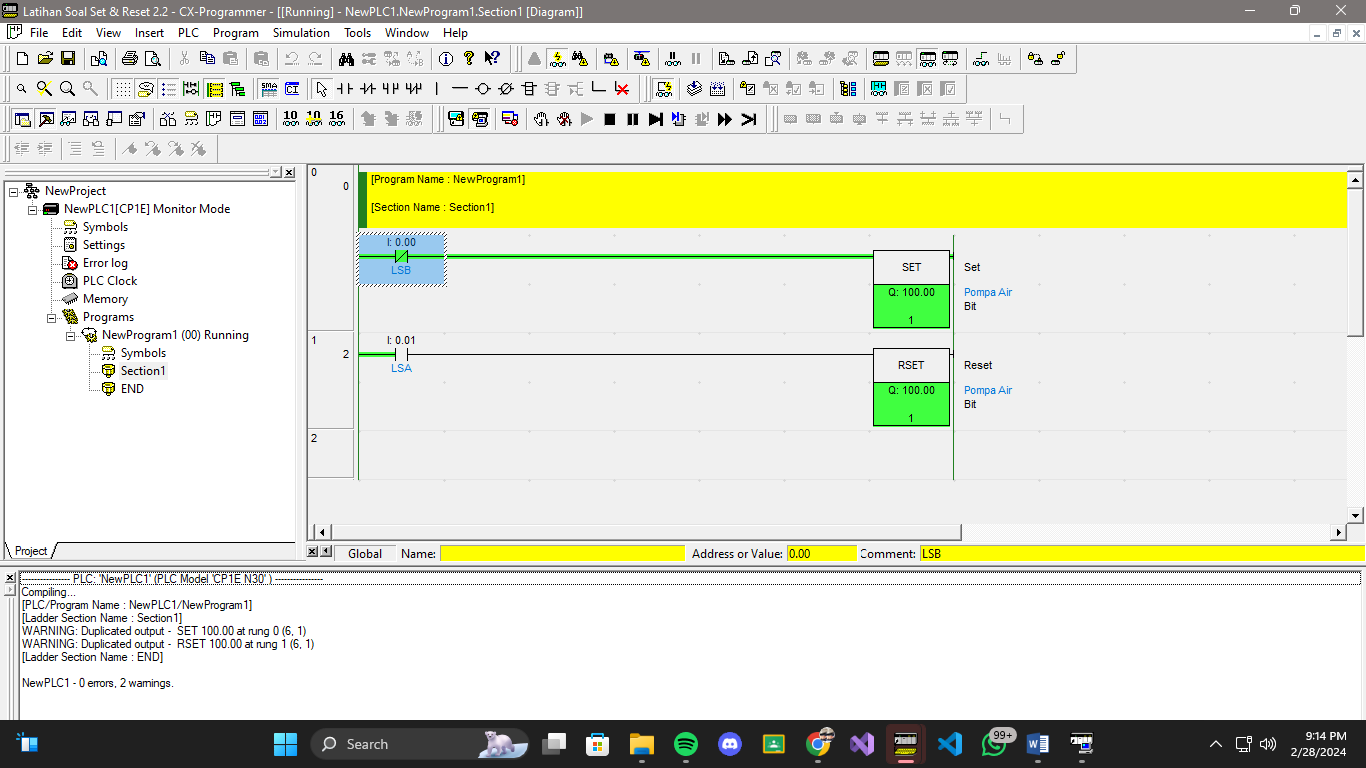


Buat program pada soal nomor 1, 2, 3, dan 4 pada praktikum input dan output menggunakaninstruksi SET dan RSET.



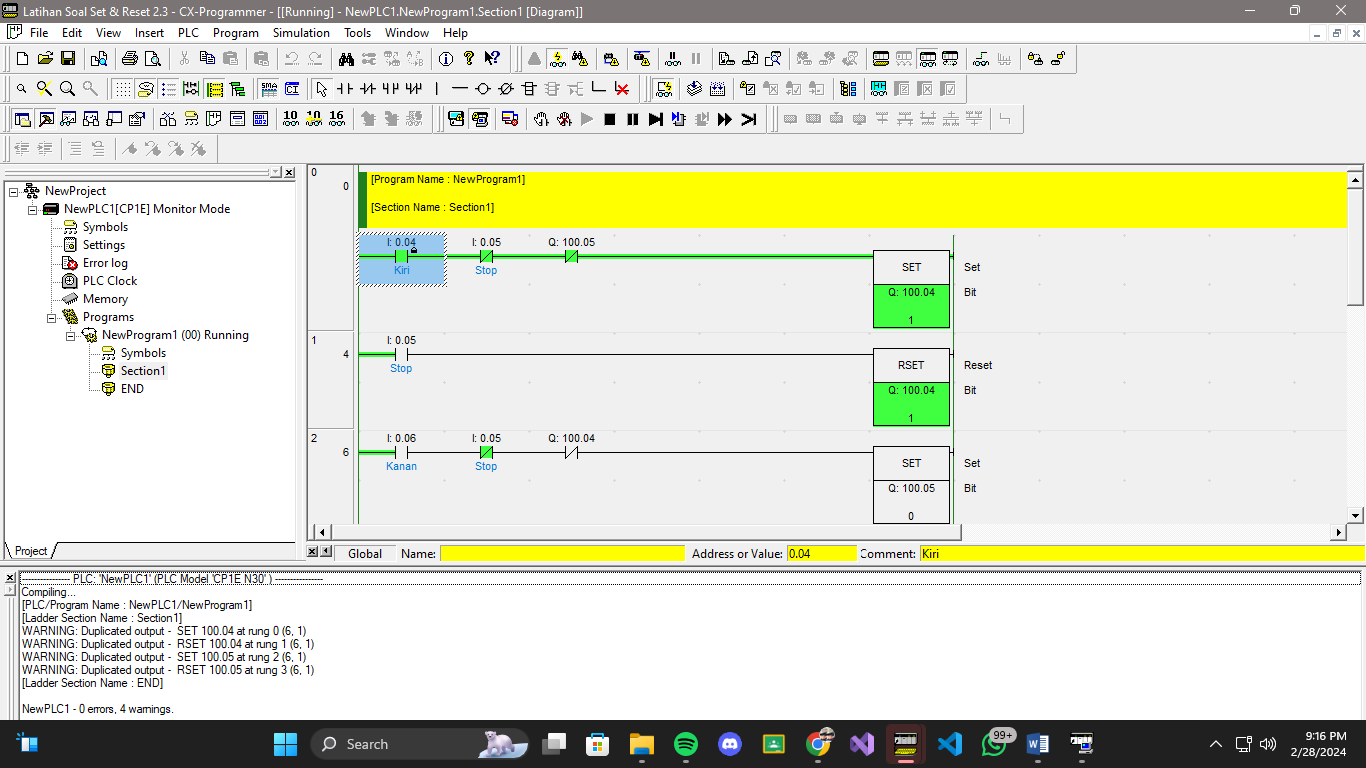
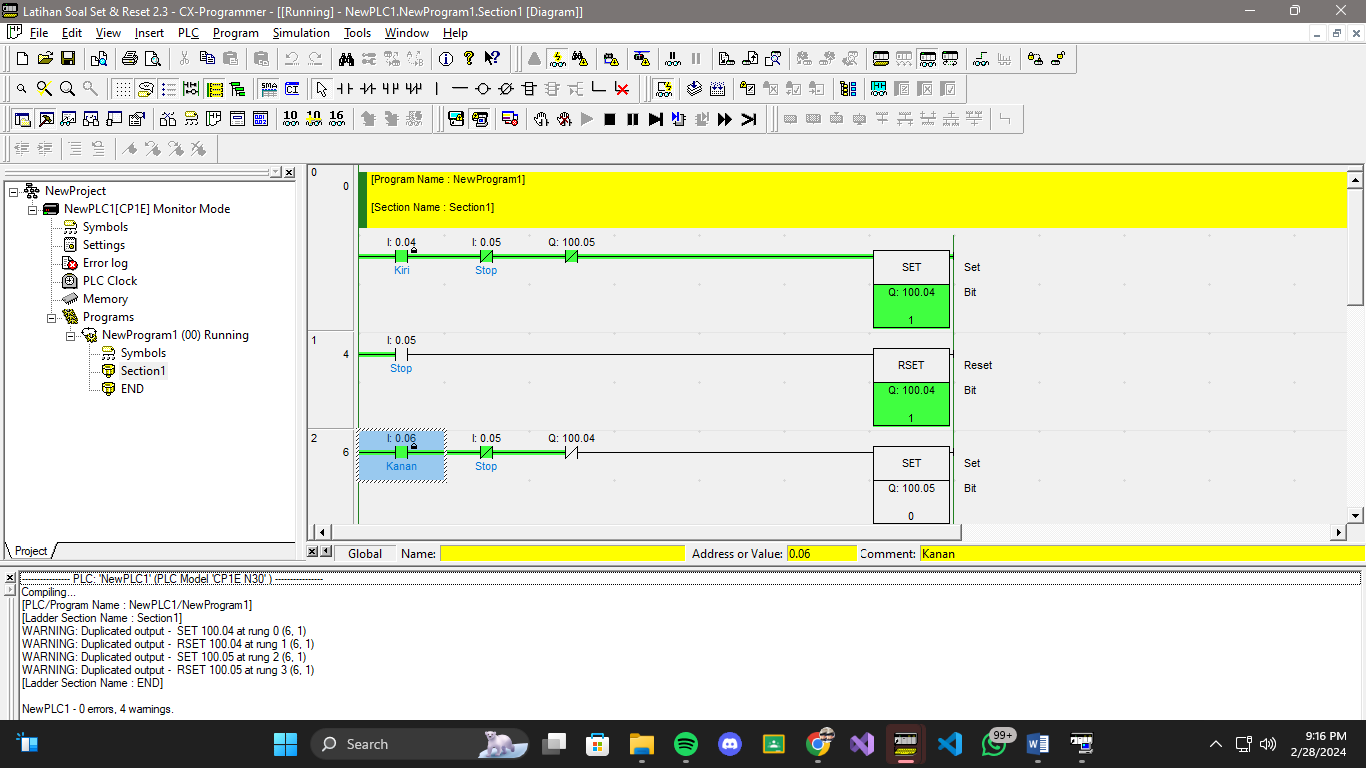
**Hasil dan Analisis**

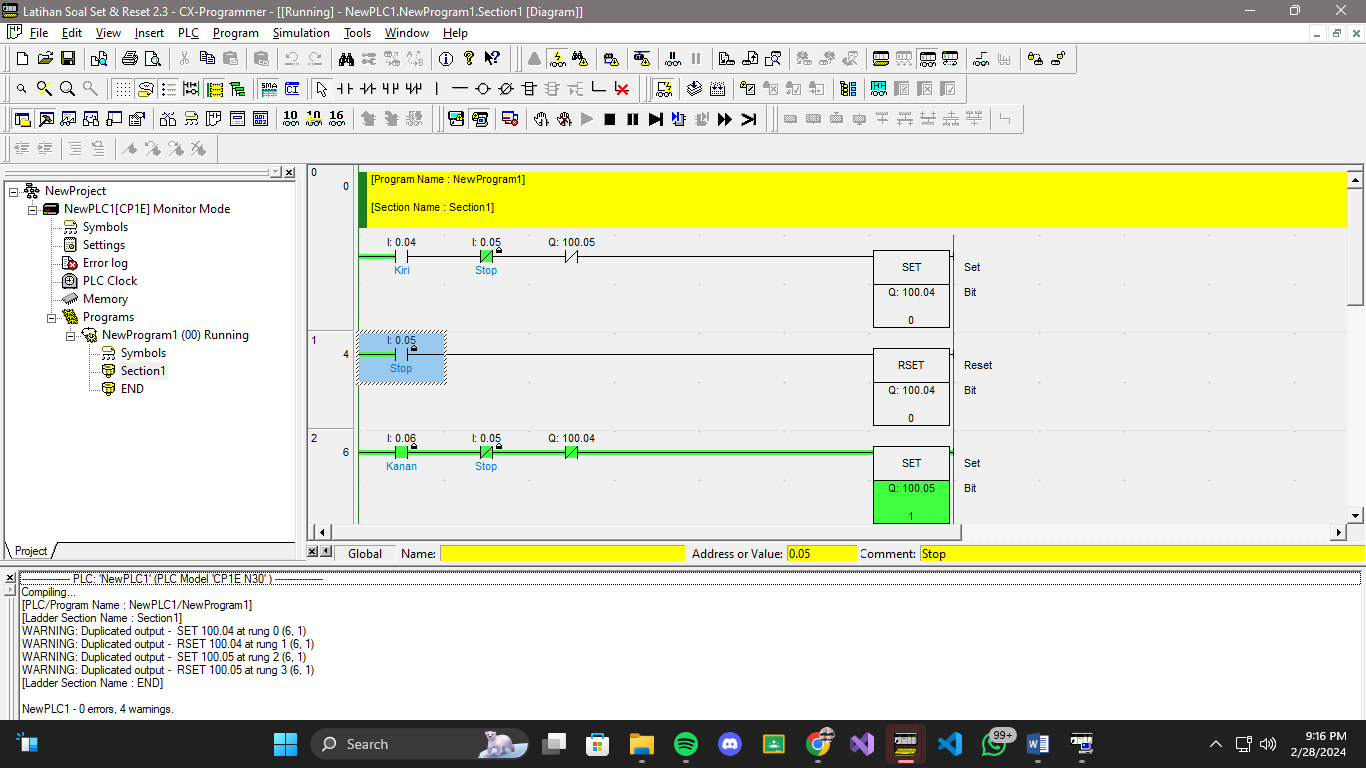
Jika Input\_1 di-on-kan maka output\_1 akan on, demikian sebaliknya jika Input\_2 di-on-kan maka output\_2 akan on. Kedua output tidak boleh on bersamaan.



**Hasil dan Analisis**

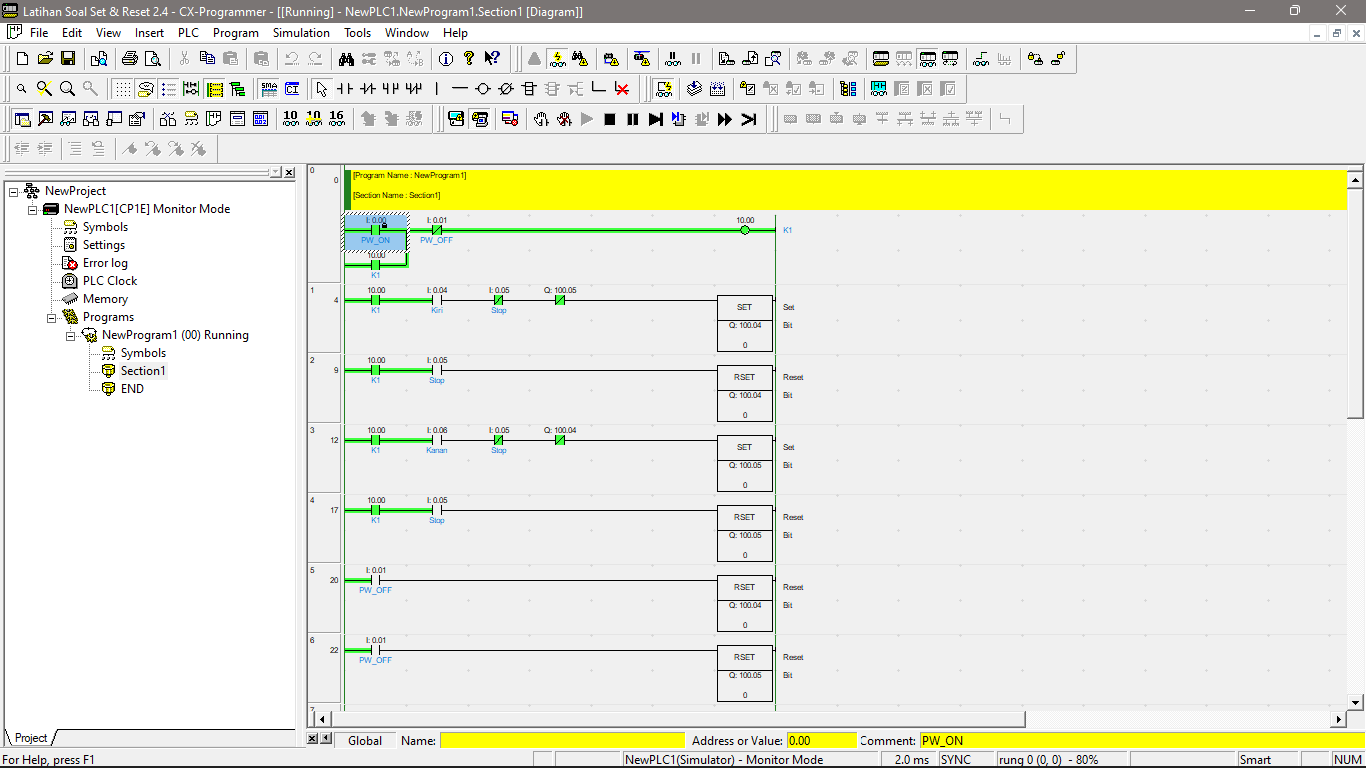
Jika tangki kosong maka pompa akan bekerja memompa air dari sumur ke dalam tangki. Jika LSB (*Level Switch* Bawah) terkena bahkan terendam air, pompa masih bekerja. Pompa akan mati jika LSA (*Level Switch* Atas) tersentuh air. Karena air digunakan terus, maka level air akan turun. Karena level air turun maka ketika LSB tidak terendam airpompa akan bekerja kembali.

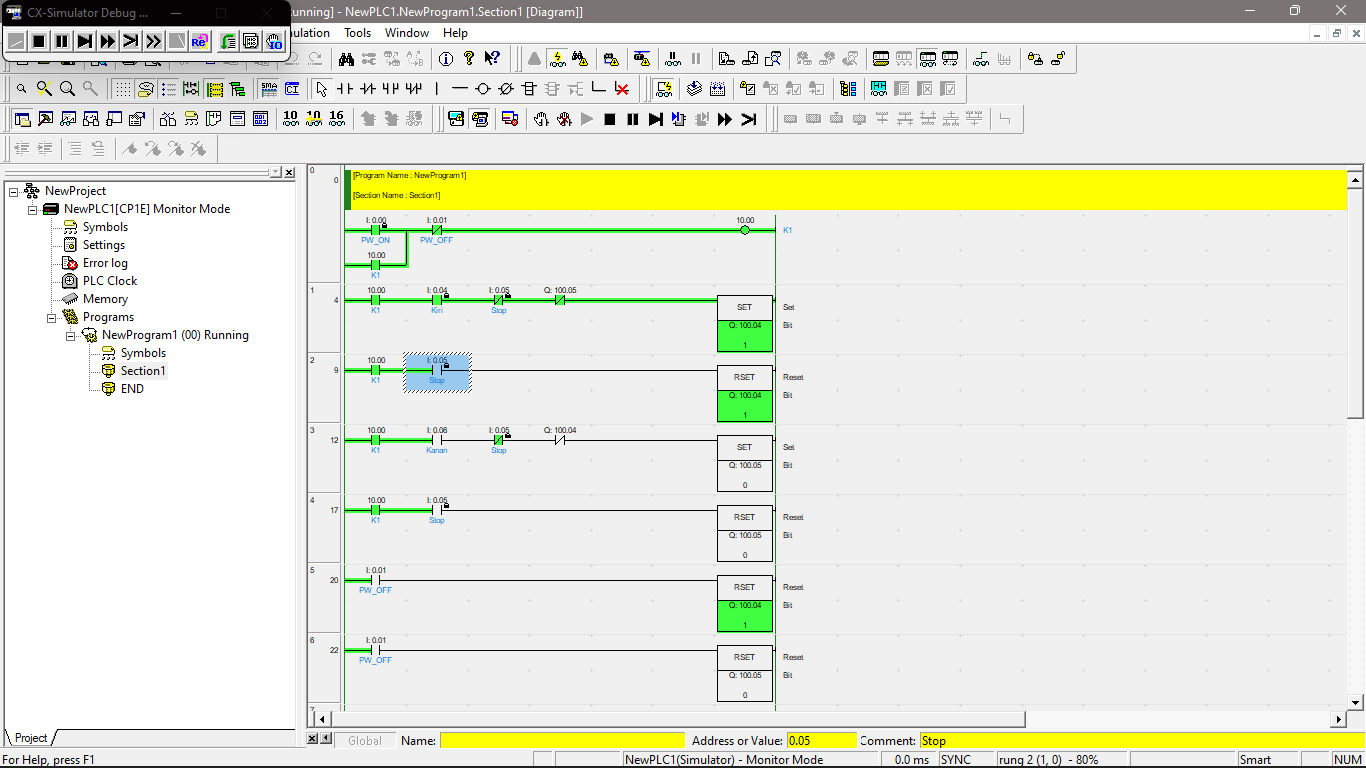
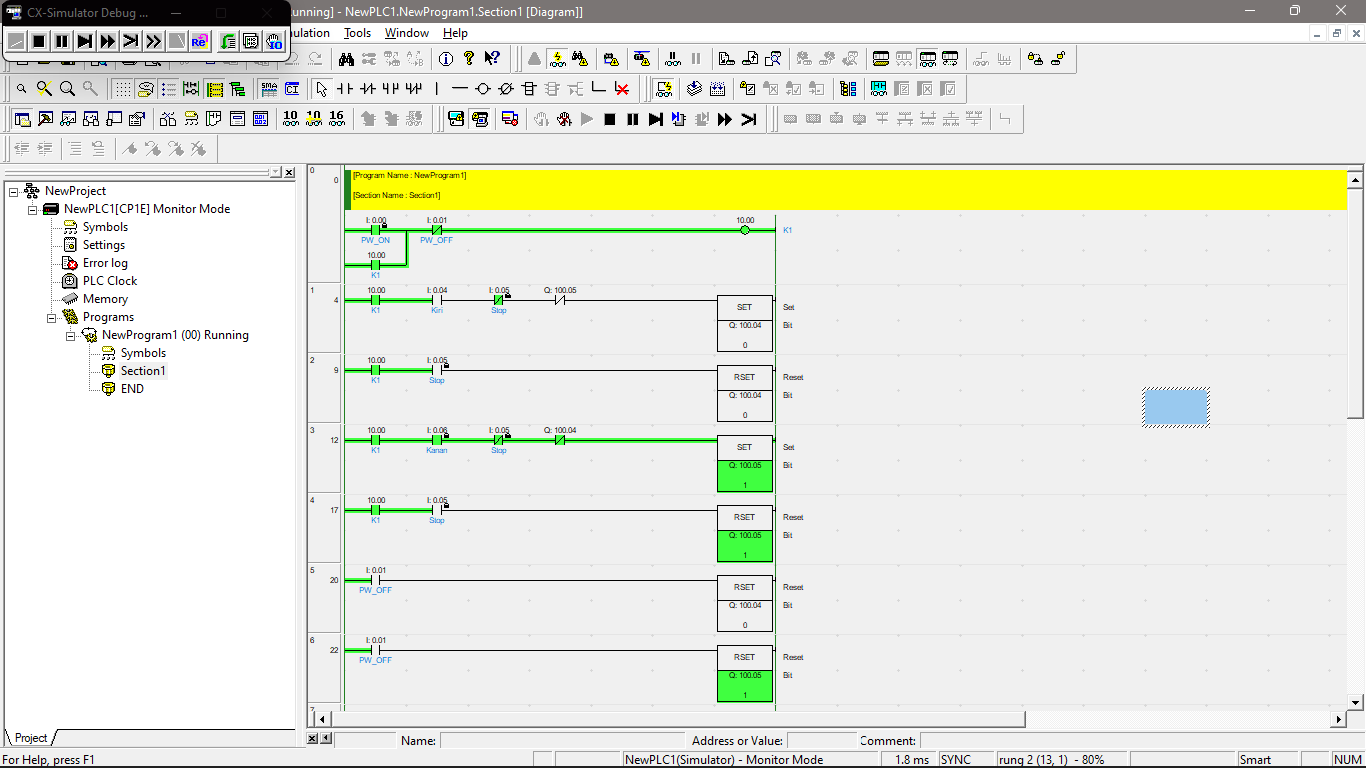




**Hasil dan Analisis**

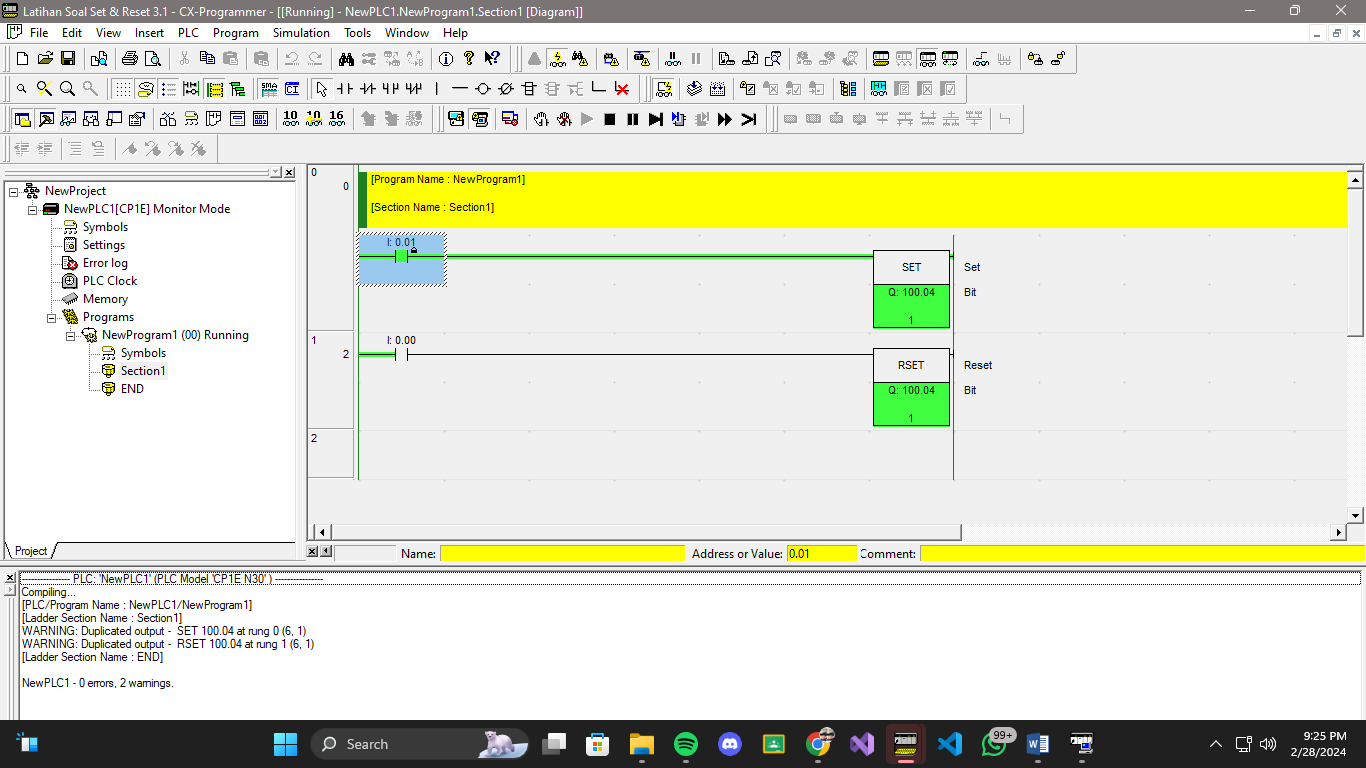
Ketika *Input* kiri di- *on*-kan maka MC\_kiri (*Magnetic Contactor*\_kiri) akan*on*. Saat MC\_kiri *on*, *input* kanan di-*on*-kan tidak akan mengakibatkan MC\_kanan *on*. Untuk meng-*on*-kan MC\_kanan, tombol *stop* harus di- *on*-kan lebih dulu. Demikian juga sebaliknya





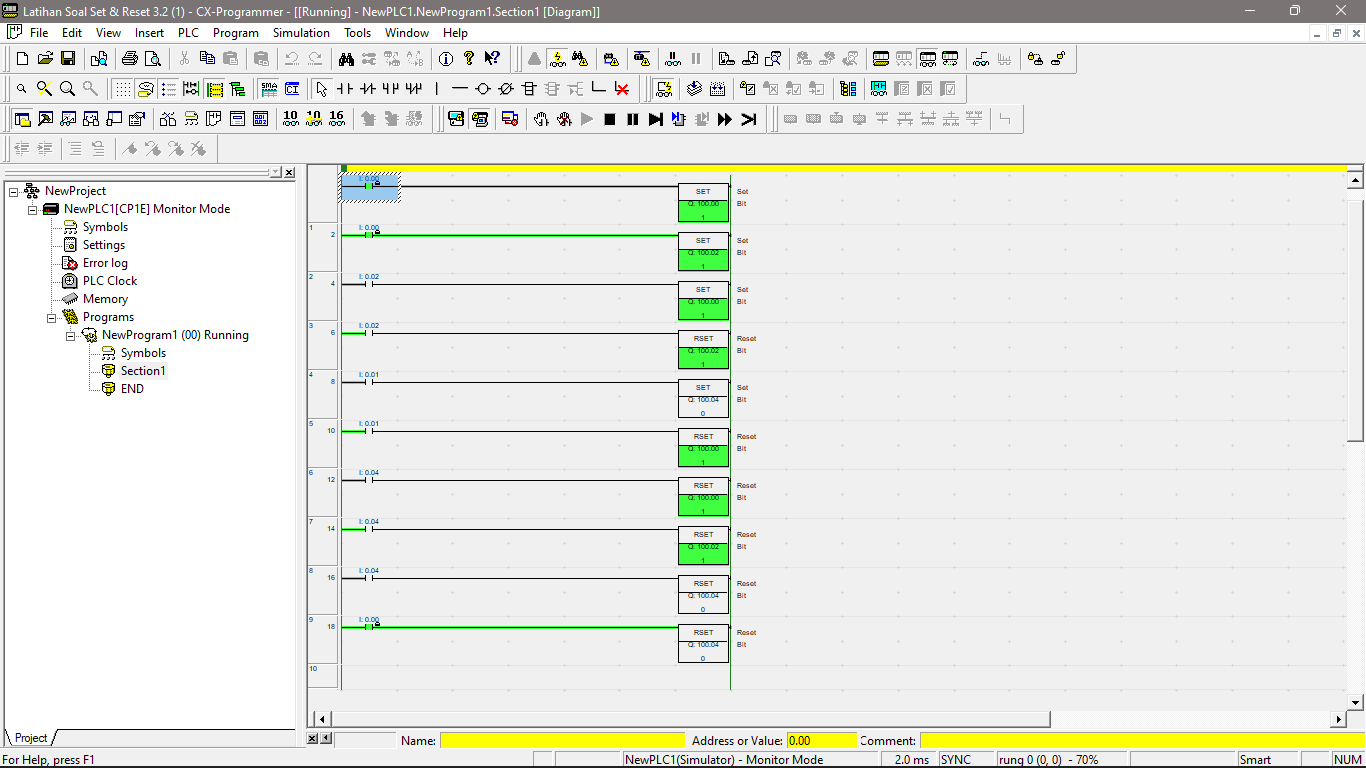
**Hasil dan Analisis**

Ketika *Input* kiri di- *on*-kan maka MC\_kiri (*Magnetic Contactor*\_kiri) akan*on*. Saat MC\_kiri *on*, *input* kanan di-*on*-kan tidak akan mengakibatkan MC\_kanan *on*. Untuk meng-*on*-kan MC\_kanan, tombol *stop* harus di- *on*-kan lebih dulu. Demikian juga sebaliknya*Input* PW\_*ON* (*Power*\_*ON*)dan PW\_*OFF* (*Power*\_*OFF*) *Input PW\_ON* berfungsisebagai saklar *ON*. Jika saklar ini belum di-*on*-kan maka mesin *milling* ini tidak bisa dioperasikan

Buat program pada soal nomor 1, dan 2 pada praktikum DIFU(013) dan DIFD(014)menggunakan instruksi SET dan RSET.

**Hasil dan Analisis**

Jika input 0.01 ditekan maka output 100.04 akan menyala. Dan jika input 0.00 ditekan maka output 100.04 akan mati



**Hasil dan Analisis**

*Input* dari sensor ultrasonik (0.00) digunakan untuk mendeteksi kendaraan yang akan masuk, yang akan memerintahkan motor listrik (1.00) untuk membuka pintu. Sebuah sensor proksimiti (0.02) akan memerintahkan motor berhenti bekerja membuka pintu jika pintu sudah sampai atas. Sebuah sensor inframerah (0.01) akan memberi sinyal ke PLC jika kendaraan sudah melewati pintu, dan memerintahkan motor listrik (1.01) untuk menutup pintu, dan sebuah *limit switch* (0.04) akan tersentuh yang menyebabkan motor listrik berhenti bekerja.

# PRAKTIKUM 6 INSTRUKSI DASAR PLC – PERBANDINGAN, PERPINDAHAN, DAN PERGESERAN DATA

**Tujuan :**

1. Memahami konsep perbandingan dan bagaimana instruksi perbandingan digunakan untuk membandingkan nilai atau keadaan dalam program PLC.
2. Melakukan simulasi program yang melibatkan instruksi perbandingan, perpindahan, dan pergeseran data untuk memastikan bahwa operasi data berjalan sesuai dengan yang diharapkan.

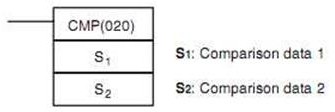
## Teori Dasar

**Instruksi Perbandingan Data**

Adalah instruksi yang akan membandingkan dua data, baik data yang tersimpan dalam suatu memori atau data berupa konstanta dalam bentuk BCD, dan hasil perbandingan tersebut akan mengaktifkan sebuah *bit* pada *Auxiliary Area*.

**COMPARE: CMP(020)**

Bentuk instruksi CMP(020) sebagai berikut:



## Alat Dan Bahan

* PLC Omron CP1E-N30
* Laptop
* Kabel jumper
* Kabel USB

## Petunjuk Praktikum

1. Pertama yaitu download aplikasi CXONE\_v4.40 terlebih dahulu. Ikutilah langkah-langkah penginstalannya.
2. Setelah selesai terinstall,buka lah aplikasinya. Lalu kita pilih file lalu *New...* , isi device name dan device type sesuai PLC yang anda punya. Disini saya menggunakan PLC dengan device type CP1E-N30.
3. Jika sudah maka tekan OK barulah kita bisa membuat rangkaiannya.
4. Pasang PLC pada panel kontrol atau lokasi yang sesuai.
5. Sambungkan PLC ke sumber daya listrik sesuai dengan spesifikasi produsen.
6. Hubungkan laptop dengan PLC menggunakan kabel yang sesuai. Pastikan bahwa koneksi antara laptop dan PLC berfungsi dengan baik.
7. Pasang modul input/output ke slot yang sesuai di PLC.
8. Hubungkan perangkat input dan output di lapangan ke modul input/output.
9. Gunakan fasilitas simulasi jika tersedia untuk menguji program tanpa perlu terhubung ke PLC fisik.
10. Simulasikan kondisi input dan periksa respons output sesuai dengan logika program.
11. Setelah pemrograman dan simulasi selesai, transfer program ke memori PLC menggunakan koneksi laptop. Pastikan transfer program berjalan tanpa kesalahan.
12. Monitor operasi PLC melalui laptop untuk memastikan bahwa program berjalan sesuai dengan harapan.
13. Gunakan fasilitas debugging untuk mengidentifikasi dan memperbaiki kesalahan dalam program.

## Praktikum

### Percobaan 1: Compare

Contoh penggunaan instruksi CMP(20) pada suatu program:

A screenshot of a calendar

Description automatically generated

**Hasil dan Analisis**

A screenshot of a computer

Description automatically generated

Saat dinyalakan maka data1 dan data2 akan dibandingkan, jika hasilnya lebih dari atau lebih dari sama dengan maka akan mengaktifkan output 1.00. Jika kurang dari maka akan mengaktifkan output 1.01.

### Percobaan 2: Mov

Contoh instruksi MOV(020) pada PLC CJ1M CPU11:

A grid of black lines

Description automatically generated with medium confidence

**Hasil dan Analisis**

A screenshot of a computer

Description automatically generated

*Rangkaian diatas saya menggunakan CP1E tetapi contohnya menggunakan CJ1M. Jadi versi yang saya pakai dengan rangkaian yang di contoh berbeda.* Di saat saya jalankan, hanya sedikit terjadi perubahan. Ketika saya ganti value di destination #1111 maka Hex Source word D0 yang berada dibawah pun ikut berubah menyesuaikan seperti yang diatas.

### Percobaan 3: SET

Contoh instruksi SFT(010) pada PLC CJ1M CPU11:

A screenshot of a computer

Description automatically generated

**Hasil dan Analisis**

A screenshot of a phone

Description automatically generated

Saat saya memulai program nya, value yang berada dalam SFT belum mulai bertambah. Saat input 0.02 ditekan maka value terus bertambah dan input P\_1s akan menyala setiap 1 detik untuk menambahkan value SFT. Value akan bertambah terus sampai kita menekan input 0.03, saat ditekan value mulai dari 0 dan jika kita ingin memulainya kembali kita cukup menekan lagi input 0.02.

### Percobaan 4: ++B

Contoh instruksi ++B(594) pada PLC CJ1M CPU11:

A black and white image of a person

Description automatically generated

**Hasil dan Analisis**

A screenshot of a computer

Description automatically generated

Saat input 0.04 dinyalakan maka Bcd yang asalnya 1111 akan berubah menjadi 1112 dikarenakan increment ++ yaitu menjumlahkan 1. Setelah dimatikan input nya lalu dinyalakan lagi maka akan dijumlah lagi menjadi 1113 dan seterusnya.

### Percobaan 5: --B

A black line on a white background

Description automatically generated

**Hasil dan Analisis**

A black line on a white background

Description automatically generated

Untuk Decrement kebalikan dari increment yaitu mengurangi 1. Saat input 0.05 diaktifkan maka value yang asalnya 1115 Bcd menjadi 1114 Bcd. Jika input dimatikan lalu dinyalakan lagi maka value akan berkurang lagi 1 menjadi 1113 dan seterusnya.

## Soal Latihan

Buatlah program untuk soal berikut:

A screenshot of a computer

Description automatically generated

**Hasil dan Analisis**

A screenshot of a computer

Description automatically generated

* Amati apa yang terjadi jika input 0.01 diaktifkan

Saat input 0.01 diaktifkan maka value yang asalnya 0000 akan berubah menjadi 0010.

* Amati apa yang terjadi jika input 0.02 diaktifkan

Saat input 0.02 diaktifkan maka value yang asalnya berubah 0010 akan kembali lagi ke awal yaitu menjadi 0000.

Buatlah program berikut, kemudian jalankan

A screenshot of a computer

Description automatically generated

**Hasil dan Analisis**

A screenshot of a computer program

Description automatically generated

* Amati apa yang terjadi jika input 0.01 diaktifkan

Saat input 0.01 diaktifkan maka output dengan alamat 1.01 akan aktif dikarenakan data 1 lebih besar nilainya dari data 2.

* Amati apa yang terjadi jika input 0.02 diaktifkan

Saat input 0.02 diaktifkan maka output 1.02 akan aktif tetapi arus mengalir ke output 1.00.

A screenshot of a computer

Description automatically generated

Buatlah program berikut, kemudian jalankan.

A screenshot of a computer

Description automatically generated

**Hasil dan Analisis**

A black line on a white background

Description automatically generated

* Amati apa yang terjadi jika input 0.00 diaktifkan

Saat input 0.00 diaktifkan maka value yang asalnya 0 Bcd akan bertambah satu menjadi 1 Bcd dikarenakan menggunakan intruksi increment tanpa ada arus yang mengalir.

* Amati apa yang terjadi jika input 0.01 diaktifkan

Saat input 0.01 diaktifkan maka value yang asalnya 1 Bcd akan berkurang satu menjadi 0 Bcd dikarenakan menggunakan intruksi decrement tanpa ada arus yang mengalir.

Buatlah program berikut, kemudian jalankan.

A white rectangular object with black lines

Description automatically generated

**Hasil dan Analisis**

A black line on a white background

Description automatically generated

* Amati apa yang terjadi jika input 0.00 diaktifkan

Saat input 0.00 diaktifkan maka value yang asalnya 0 Bcd akan bertambah satu menjadi 1 Bcd dikarenakan menggunakan intruksi increment

* Amati apa yang terjadi jika input 0.01 diaktifkan

Saat input 0.01 diaktifkan maka value yang asalnya 1 Bcd akan berkurang satu menjadi 0 Bcd dikarenakan menggunakan intruksi decrement

Buatlah program berikut, kemudian jalankan.

A white background with black lines

Description automatically generated

**Hasil dan Analisis**

A green line on a white background

Description automatically generated

* Amati apa yang terjadi jika input 0.00 diaktifkan

Saat input 0.00 diaktifkan maka value yang asalnya 0 Bcd akan bertambah menjadi bilangan acak. Saat bilangan menyentuh 9999 maka nilai akan kembali menjadi 0 lalu bilangan akan naik lagi seperti sebelumnya

* Amati apa yang terjadi jika input 0.01 diaktifkan

Saat input 0.01 diaktifkan maka value yang asalnya bilangan acak akan berubah menjadi seperti yang di awal. Kita dapat memulainya kembali dengan menekan input 0.00

* Apa kesimpulan yang bisa anda ambil dari soal nomor 1, 2, dan 3?

Kesimpulannya yaitu nomor 1 menggunakan intruksi MOV yang digunakan untuk memindahkan data, nomor 2 menggunakan intruksi MOV dan Intruksi CMP. Intruksi CMP digunakan untuk membandingkan sebuah nilai. Nomor 3 menggunakan intruksi increment dan decrement untuk menambahkan satu atau mengurangi satu dari sebuah nilai

# PRAKTIKUM 7 INTRUKSI DASAR PLC-ARITMATIKA

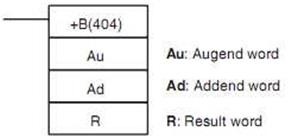
**Tujuan :**

1. Memahami konsep dasar operasi aritmatika, termasuk penambahan, pengurangan, perkalian, dan pembagian dalam konteks PLC.
2. Mampu menggunakan perangkat lunak pemrograman PLC untuk memprogram instruksi aritmatika.
3. Melakukan simulasi program yang melibatkan instruksi aritmatika untuk memastikan bahwa perhitungan berjalan sesuai dengan yang diharapkan.

## Teori Dasar

**BCD ADD WITHOUT CARRY: +B(404)**

Instruksi ini akan melakukan proses penjumlahan data dalam bentuk BCD. Data yang ada ataukonstanta yang dibuat di Au (*Augend*) dan Ad (*Addend*) dan diletakkan hasil penjumlahannya pada R (*Result*). Bentuk instruksi ini adalah sebagai berikut:



Bentuk operasi yang terjadi pada operasi penjumlahan adalah sebagai berikut:



**BCD SUBTRACT WITHOUT CARRY: -(414)**

Instruksi ini akan melakukan proses pengurangan data dalam bentuk BCD. Data yang ada ataukonstanta yang dibuat di Mi (*Minuend*) dikurangi dengan yang ada di Su (*Subtrahend*) dan hasilnya diletakkan pada R (*Result*). Bentuk instruksi ini adalah sebagai berikut:

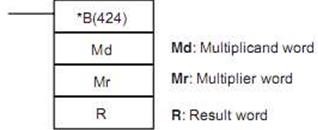


Bentuk operasi yang terjadi pada operasi pengurangan adalah sebagai berikut:



**BCD MULTIPLY: \*B(424)**

Instruksi ini akan melakukan proses perkalian data dalam bentuk BCD. Data yang ada atau konstanta yang dibuat di Md (*Multiplicand*) dikalikan dengan data yang ada di Mr (*Multiplier*) dan hasilnya diletakkan pada R (*Result*). Bentuk instruksi ini adalah sebagai berikut:



Bentuk operasi yang terjadi pada operasi perkalian adalah sebagai berikut:



## Alat Dan Bahan

* PLC Omron CP1E-N30
* Laptop
* Kabel jumper
* Kabel USB

## Petunjuk Praktikum

1. Pertama yaitu download aplikasi CXONE\_v4.40 terlebih dahulu. Ikutilah langkah-langkah penginstalannya.
2. Setelah selesai terinstall,buka lah aplikasinya. Lalu kita pilih file lalu *New...* , isi device name dan device type sesuai PLC yang anda punya. Disini saya menggunakan PLC dengan device type CP1E-N30.
3. Jika sudah maka tekan OK barulah kita bisa membuat rangkaiannya.
4. Pasang PLC pada panel kontrol atau lokasi yang sesuai.
5. Sambungkan PLC ke sumber daya listrik sesuai dengan spesifikasi produsen.
6. Hubungkan laptop dengan PLC menggunakan kabel yang sesuai. Pastikan bahwa koneksi antara laptop dan PLC berfungsi dengan baik.
7. Pasang modul input/output ke slot yang sesuai di PLC.
8. Hubungkan perangkat input dan output di lapangan ke modul input/output.
9. Gunakan fasilitas simulasi jika tersedia untuk menguji program tanpa perlu terhubung ke PLC fisik.
10. Simulasikan kondisi input dan periksa respons output sesuai dengan logika program.
11. Setelah pemrograman dan simulasi selesai, transfer program ke memori PLC menggunakan koneksi laptop. Pastikan transfer program berjalan tanpa kesalahan.
12. Monitor operasi PLC melalui laptop untuk memastikan bahwa program berjalan sesuai dengan harapan.
13. Gunakan fasilitas debugging untuk mengidentifikasi dan memperbaiki kesalahan dalam program.

## Praktikum

### Percobaan 1: +B

Contoh instruksi +B(404) pada PLC CJ1M CPU11:

A white background with black lines

Description automatically generated

**Hasil dan Analisis**

A white background with black lines

Description automatically generated

Sebagai contoh kita dapat memanggil intruksi ADD WITHOUT CARRY dengan kode +B(404), maka akan otomatis memunculkan intruksinya. Ini hanya sebagai contoh saja jadi hanya sedikit perubahan yang terjadi

### Percobaan 2: -B

Contoh instruksi –B(414) pada PLC CJ1M CPU11

A black line on a white background

Description automatically generated

**Hasil dan Analisis**

A white background with black lines

Description automatically generated

Sebagai contoh kita dapat memanggil intruksi SUBTRACT WITHOUT CARRY dengan kode -B(414), maka akan otomatis memunculkan intruksinya. Ini hanya sebagai contoh saja jadi hanya sedikit perubahan yang terjadi

### Percobaan 3: \*B

Contoh instruksi \*B(424) pada PLC CJ1M CPU11: A white background with black dots

Description automatically generated

**Hasil dan Analisis**

A white rectangular object with a black line

Description automatically generated

Sebagai contoh kita dapat memanggil intruksi MULTIPLY dengan kode \*B(424), maka akan otomatis memunculkan intruksinya. Ini hanya sebagai contoh saja jadi hanya sedikit perubahan yang terjadi

## Soal Latihan

Buatlah program untuk soal berikut:

A screenshot of a graph

Description automatically generated

**Hasil dan Analisis**

A screenshot of a computer

Description automatically generated

* Amati apa yang terjadi jika input 0.00 diaktifkan

Saat input 0.00 diinputkan maka value akan terus bertambah sampai 9999. Setelah diatas 9999 maka value akan kembali dari awal dan bertambah lagi seperti sebelumnya.

* Amati apa yang terjadi jika input 0.01 diaktifkan

Saat input 0.01 diaktifkan maka value yang awalnya 0 akan bertambah satu menjadi 1.

* Amati apa yang terjadi jika input 0.02 diaktifkan

Saat input 0.02 diaktifkan maka value yang awalnya 0 akan bertambah satu menjadi 1.

* Apa yang bisa anda simpulkan?

Dapat disimpulkan bahwa dengan intruksi ADD kita dapat menambahkan suatu data/nilai/value. Kita juga dapat menggunakan intruksi ADD WITHOUT CARRY menggunakan input diff-up

A screenshot of a computer

Description automatically generatedBuatlah program untuk soal berikut:

**Hasil dan Analisis**A screenshot of a computer

Description automatically generated

* Amati apa yang terjadi jika input 0.00 diaktifkan

Saat input 0.00 diaktifkan maka seketika nilai berubah menjadi 1 lalu berubah lagi seketika menjadi 0 secara terus menerus

* Amati apa yang terjadi jika input 0.01 diaktifkan

Saat input 0.01 diaktifkan maka nilai akan berubah menjadi 1, saat diinputkan lagi yang kedua kalinya maka akan berubah menjadi 0 lagi.

* Amati apa yang terjadi jika input 0.02 diaktifkan

Saat input 0.02 diaktifkan maka nilai akan berubah menjadi 1, saat diinputkan lagi maka akan berubah menjadi 0 lagi.

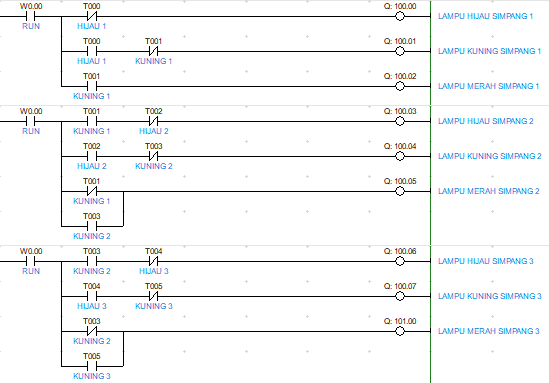
* Apa yang bisa anda simpulkan?

Kesimpulannya yaitu sekarang kita tau bahwa SUBSTRACT WITHOUT CARRY mirip penggunaanya seperti pada ADD WITHOUT CARRY hanya saja untuk substract adalah kebalikannya dari add.

1. Buat program untuk pengaturan lampu lalu lintas di sebuah perempatan menggunakan instruksi-instruksi yang sudah dipelajari. Lama waktu untuk lampu merah adalah 10 detik, lampu kuning 2 detik, dan lampu hijau 10 detik untuk masing-masing jalan.A screenshot of a computer program

   Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Hasil dan Analisis**

A diagram of a computer

Description automatically generated with medium confidence

Disini saya membuat 4 lampu lalu lintas karena memiliki 4 simpang artinya terdiri dari 16 total lampu. Untuk waktunya saya atur untuk lampu hijau yaitu 30 detik dan lampu kuning yaitu 2 detik sedangkan lampu merah sama seperti lampu hijau yaitu 30 detik.

Cara kerjanya yaitu Saat program dimulai maka timer akan berjalan. Timer yang sedang berjalan berarti lampu itulah yang sedang menyala. Jadi dalam kasus ini tidak ada lampu hijau yang menyala secara bersamaan karena tiap simpang hanya 1 lampu hijau yang menyala dan untuk yang lainnya lampu berwarna merah yang menyala.

Sebelum merah ke hijau atau hijau ke merah lampu pindah terlebih dahulu ke lampu kuning selama 2 detik. Siklus ini akan terus menerus berjalan sampai input 0.00 diberhentikan

# PENUTUP

**Kesimpulan**

Laporan ini merinci pengalaman dan pembelajaran selama praktek PLC, yang melibatkan pemahaman, penerapan, dan pemrograman PLC. Selama praktek, pemahaman tentang konsep dasar PLC, seperti instruksi logika, instruksi aritmatika, instruksi perbandingan, dan instruksi pemindahan data, semakin meningkat. Penerapan konsep-konsep ini dalam situasi nyata membantu mengembangkan keterampilan dalam merancang dan mengimplementasikan kontrol otomatis.

Proses pemrograman PLC, yang mencakup instruksi dasar seperti SET dan RESET, TIMER, COUNTER, dan instruksi lainnya, menjadi bagian integral dari laporan ini. Pemahaman tentang prinsip-prinsip dasar pemrograman dan logika kontrol memainkan peran penting dalam menyelesaikan tugas-tugas yang melibatkan pengaturan waktu, penghitungan, dan pengendalian keadaan logika.

**Saran**

Lebih giat untuk mempelajari segala hal yang mencakup PLC. Lebih teliti jika sedang mengerjakan latihan latihan

# DAFTAR PUSTAKA

Loyalty To Growth,”Mengenal PLC: Jenis, Fungsi, Komponen, dan Prinsip Kerjanya”,< <https://laskarotomasi.com/plc/>> [28 Februari 2024]

UGM,”Pengantar PLC-Menara Ilmu”,17 September 2019, < <https://plc.mipa.ugm.ac.id/pengantar-plc/>> [28 Februari 2024]

Modul Praktik PLC POLITEKNIK MANUFAKTUR BANDUNG