

MATERI PERKULIAHAN PRAKTIKUM MIKROPROSESOR



OLEH :

AHMAD SHAUFI

STMIK INDONESIA BANJARMASIN

2018

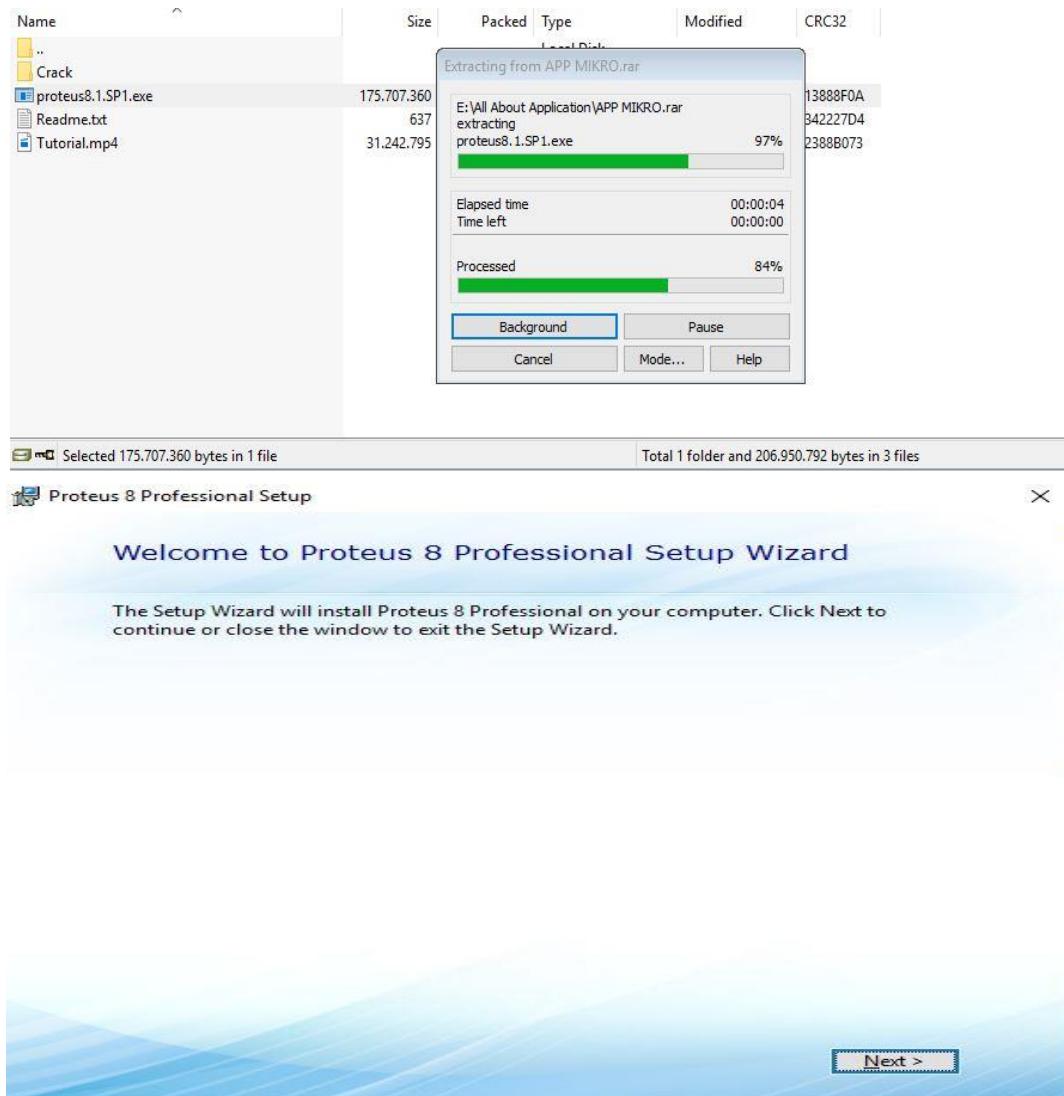
BAB I

INSTALASI PROTEUS 8.1 DAN CODE VISION

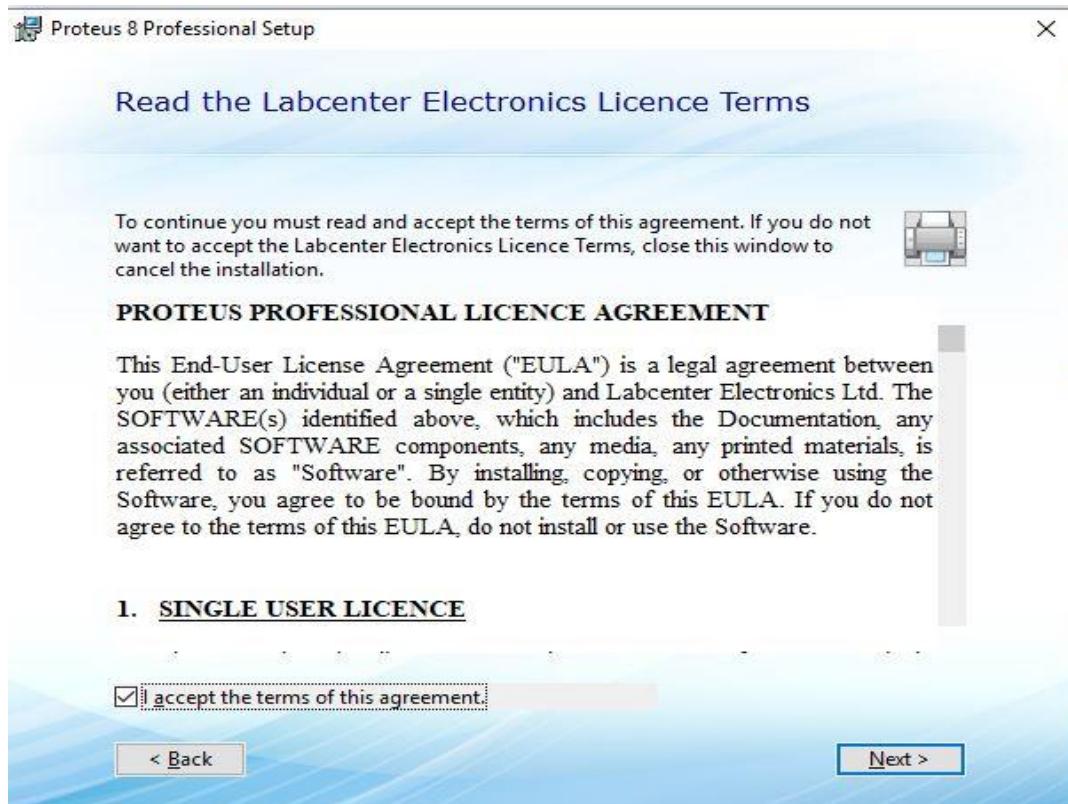
Dalam membuat Desain Rangkaian dan melakukan uji coba terhadap desain rangkaian tersebut, kita bisa menggunakan Aplikasi Proteus yang digunakan untuk mendesain rangkaian digital dan Code Vision digunakan untuk membuat perintah agar rangkaian yang dibuat bisa difungsikan sebagaimana mestinya, adapun tahapan-tahapan pemasangan kedua aplikasi tersebut akan dibahas pada bab ini. Sebelumnya sediakan terlebih dahulu paket aplikasinya yang tersedia bersamaan dengan modul ini.

A. Instalasi Proteus Versi 8.1

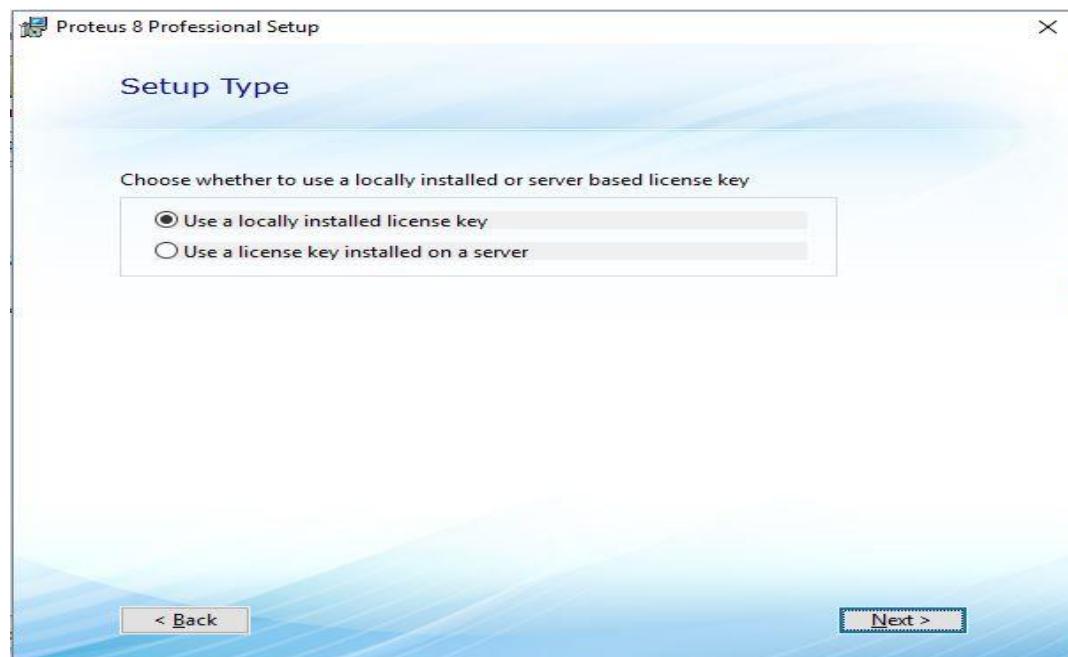
1. Extract terlebih dahulu paket aplikasi yang terdapat pada Zip file, setelah itu jalankan file **“proteus8.1.SP1.exe”** sehingga muncul jendela awal instalasi dan Next pada jendela Welcome to Proteus 8.



2. Pada jendela Proteus Licence Agreement, Centang “I accept the terms of this agreement” dan Next.



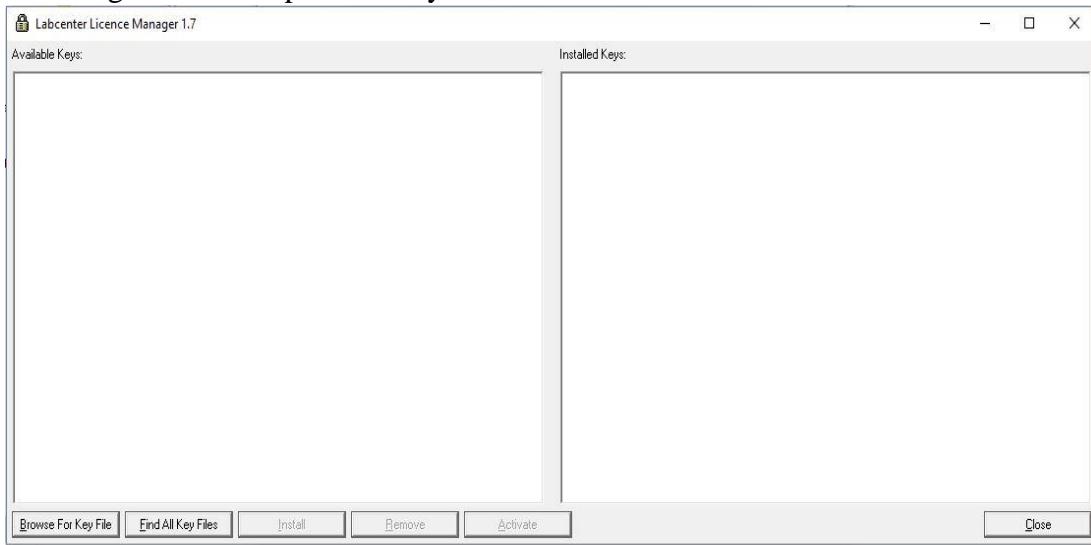
3. Pada bagian Setup Type, pilih “Use a locally”

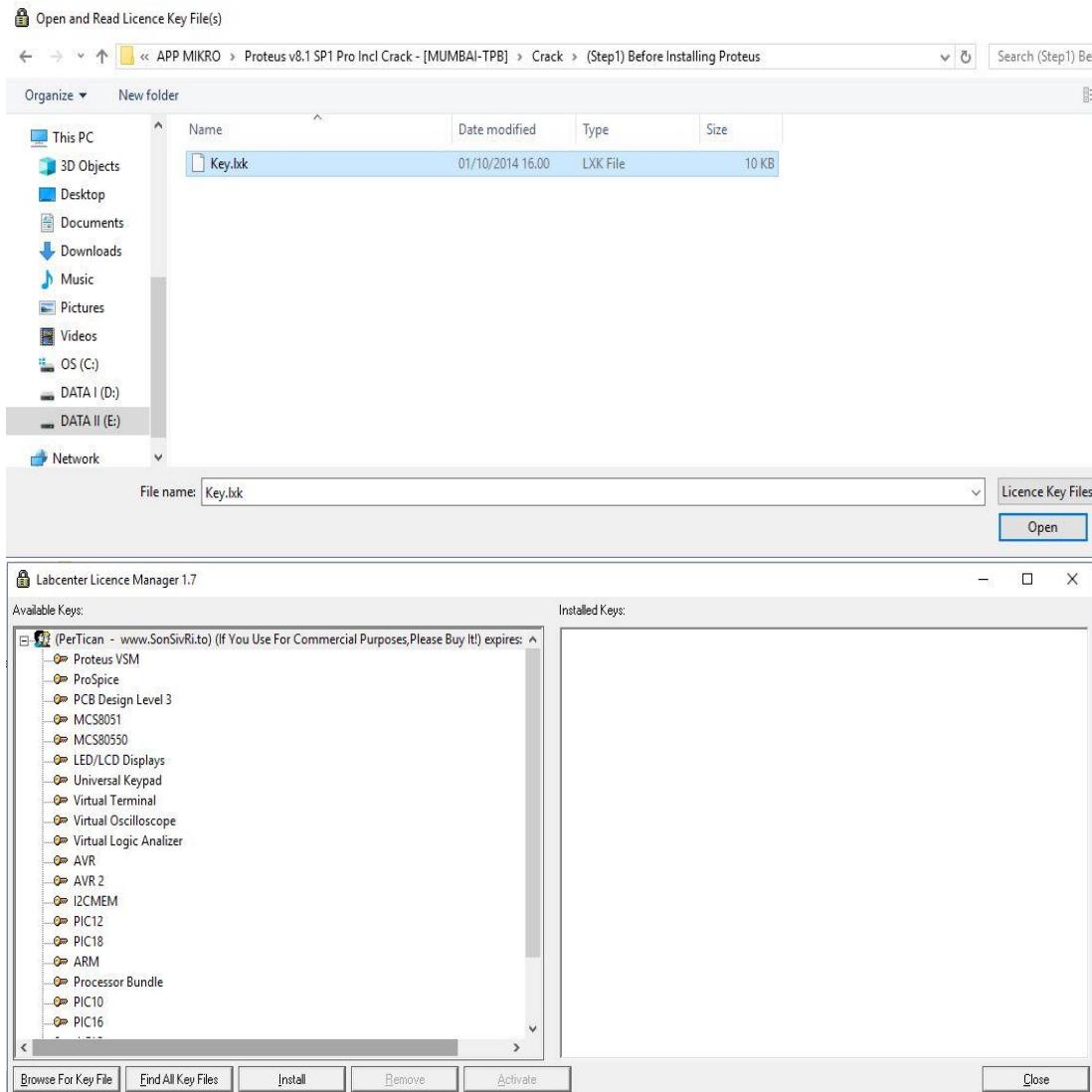


4. Product License Key, Langsung Next saja.

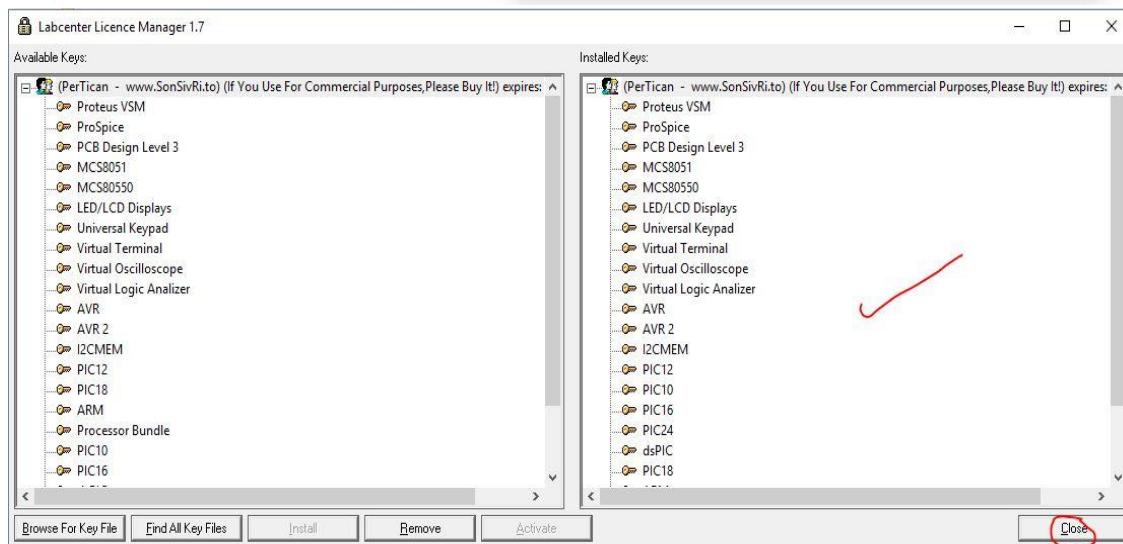
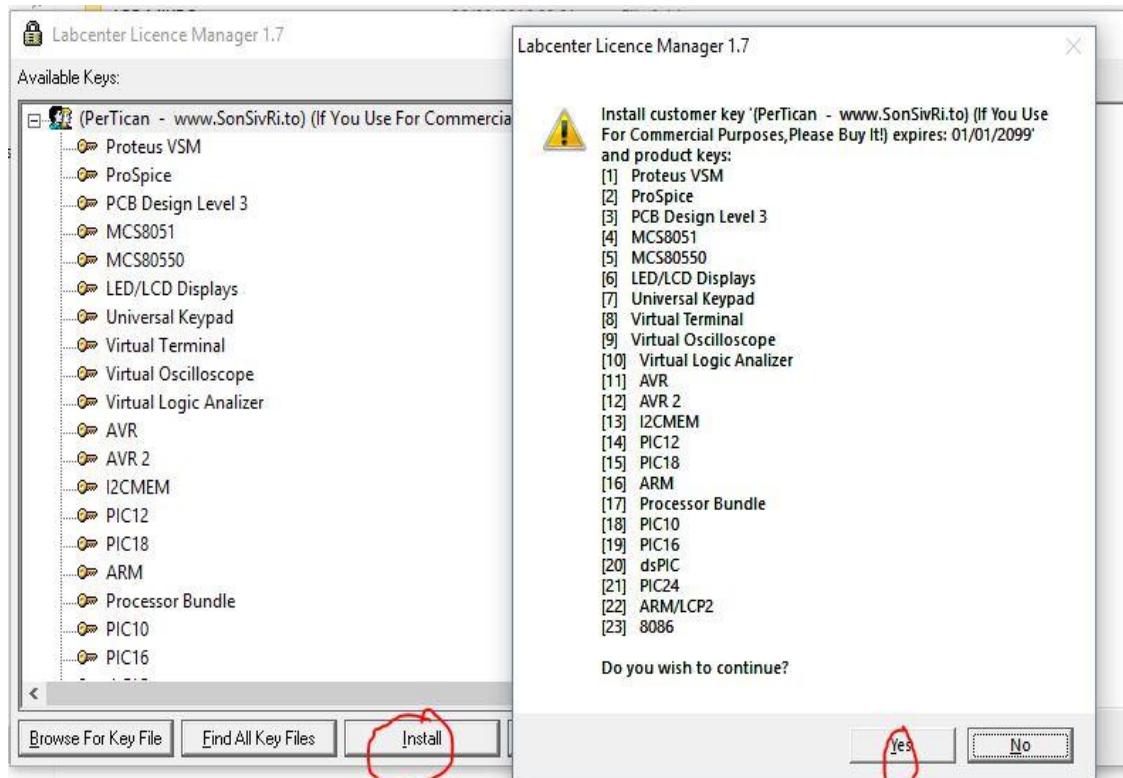


5. Akan muncul Jendela Labcenter Licence Manager 1.7, untuk melakukan install komponen-komponen key pada aplikasi, pilih “Browse For Key File”, arahkan kedalam folder paket aplikasi mikro kita, letak key ada pada folder Proteus v8.1 > Crack > Step 1 before installing Proteus > Open file Key.lxk .

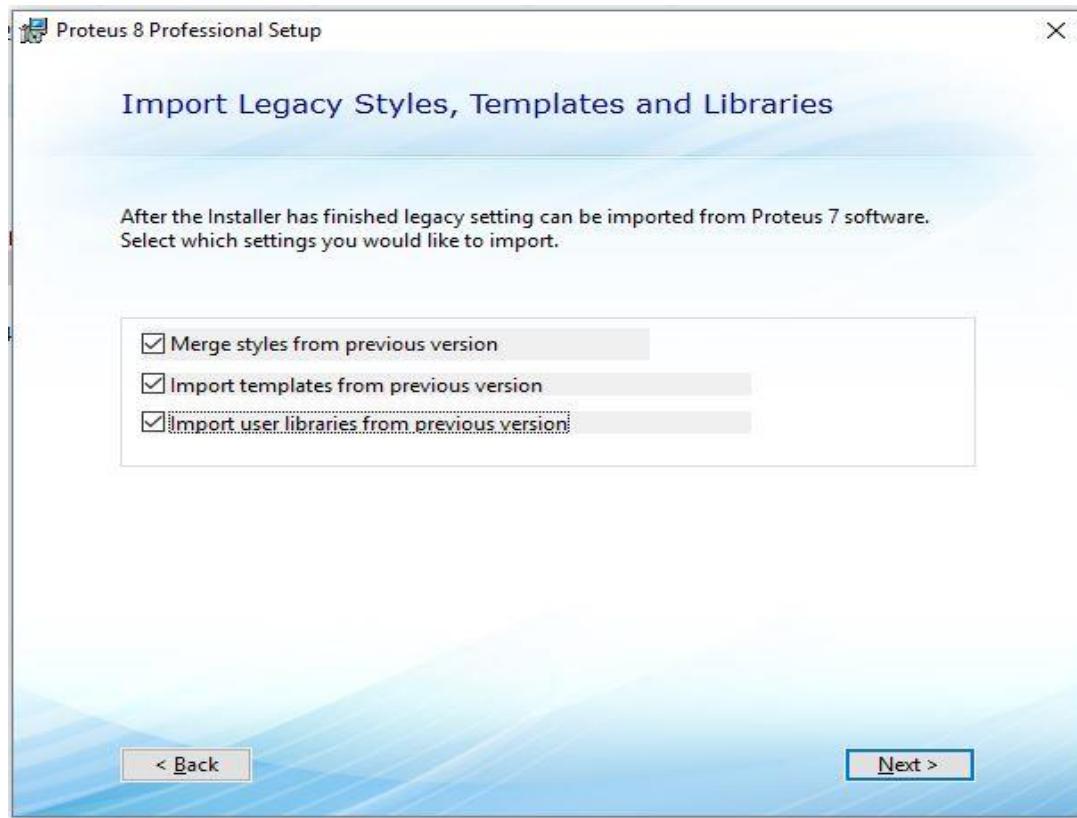




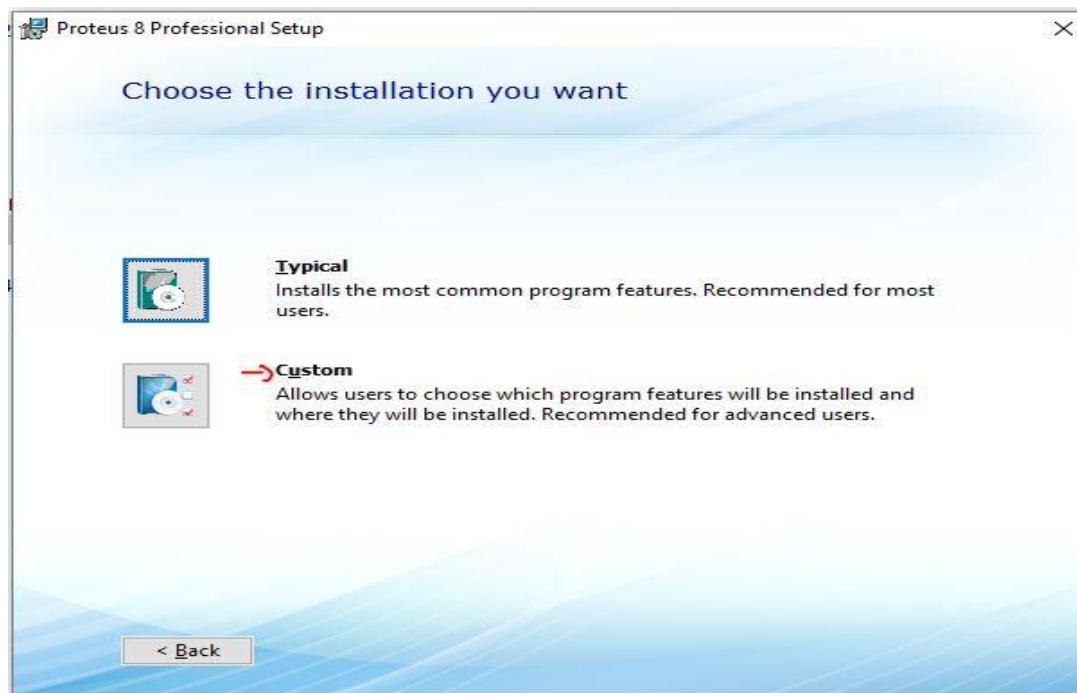
- Setelah semua key berhsiai tampil pada jendela sebelah kiri, klik install dan akan muncul pesan konfirmasi klik yes, jika key berhasil terinstal maka jendela sebelah kanan akan berikan file key yang sudah terinstall, setelah itu Close jendela dan instalasi akan dilanjutkan.



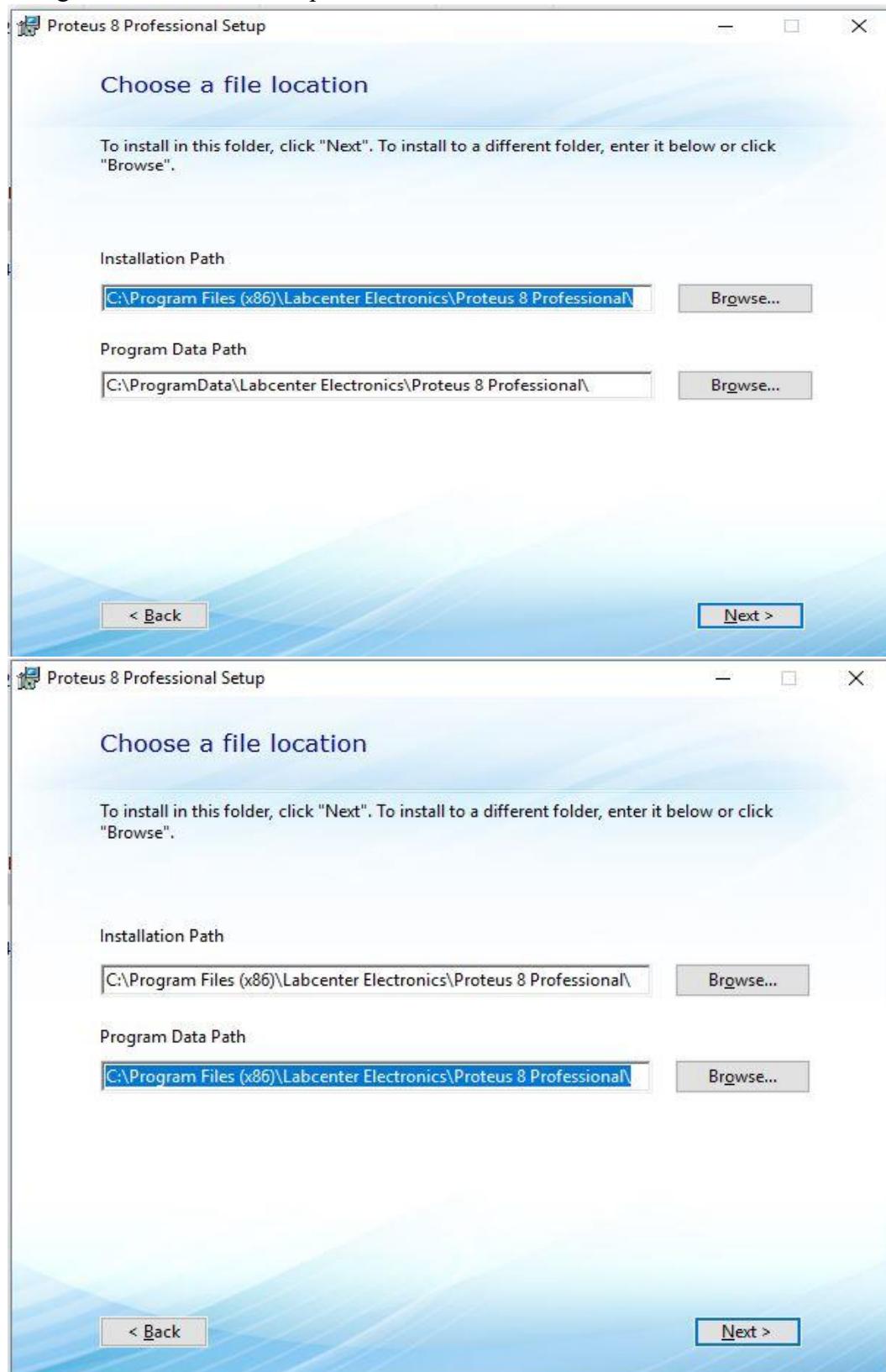
7. Selanjutnya akan muncul jendela Import legacy styles, templates and libraries. Centang semua pilihan yang tersedia, ada 3 pilihan dan next.



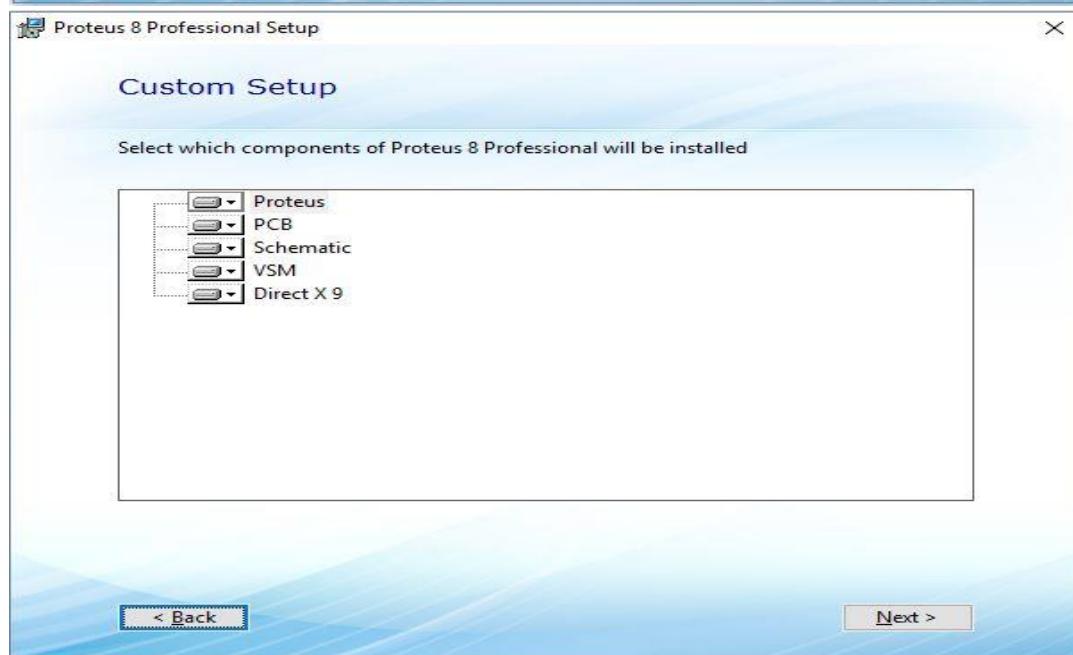
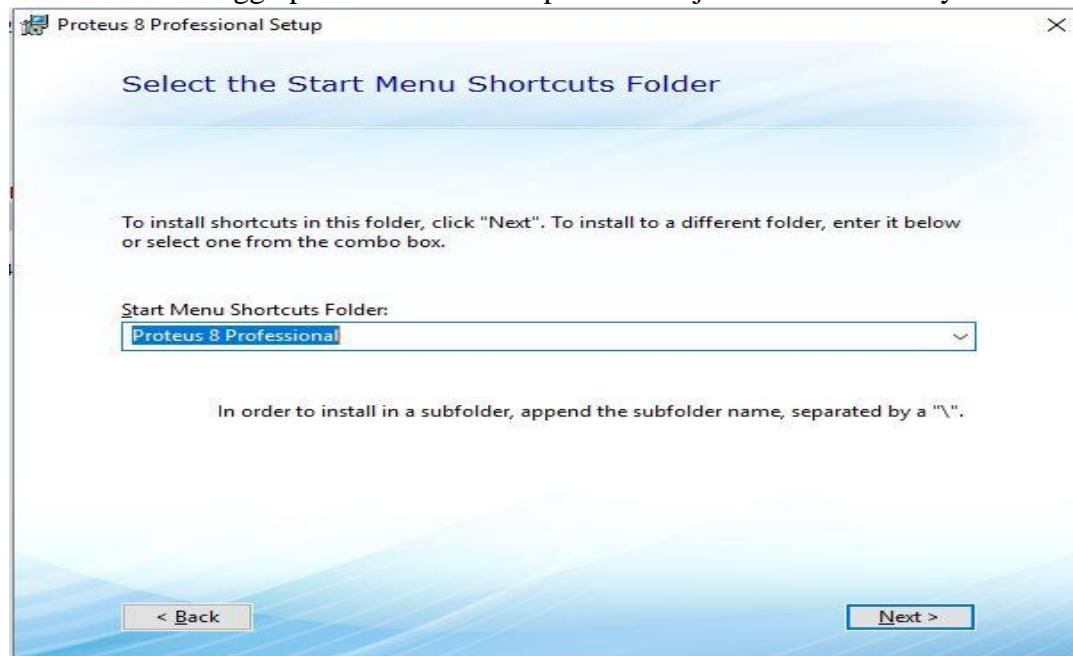
8. Pada bagian Choose the installation you want, pilih Costum.

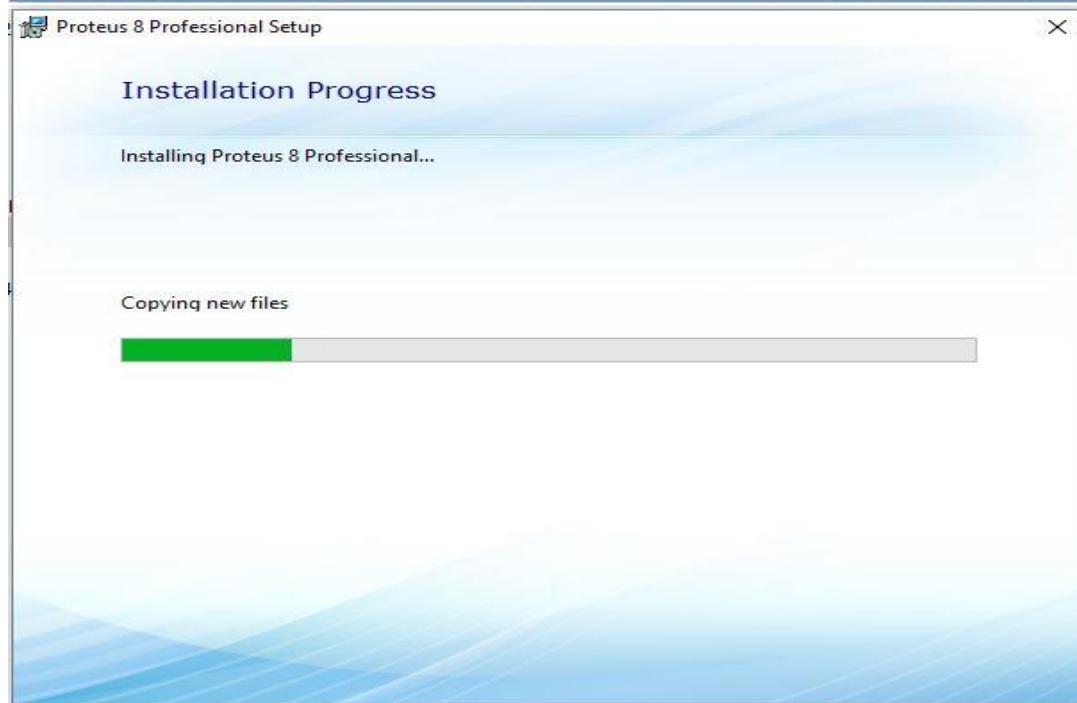
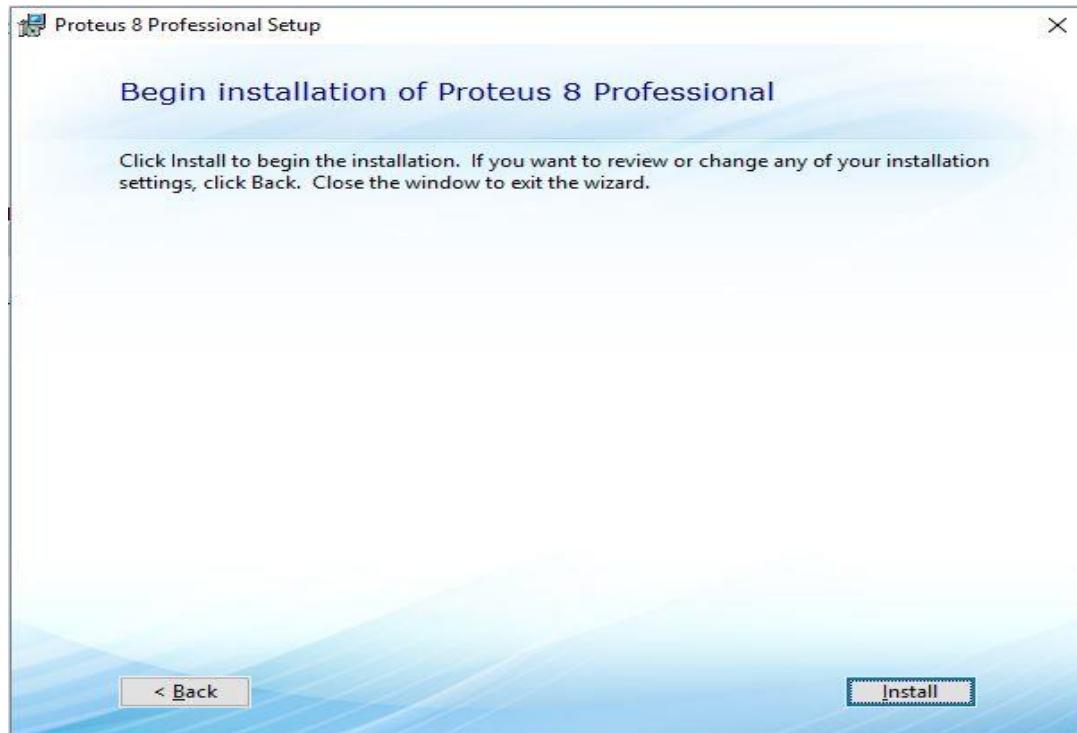


9. Copykan Path yang sama yang terdapat pada “Installation Path” dan Pastekan dibagian “Program Data Path”, tidak perlu dibrowse. Dan Next.

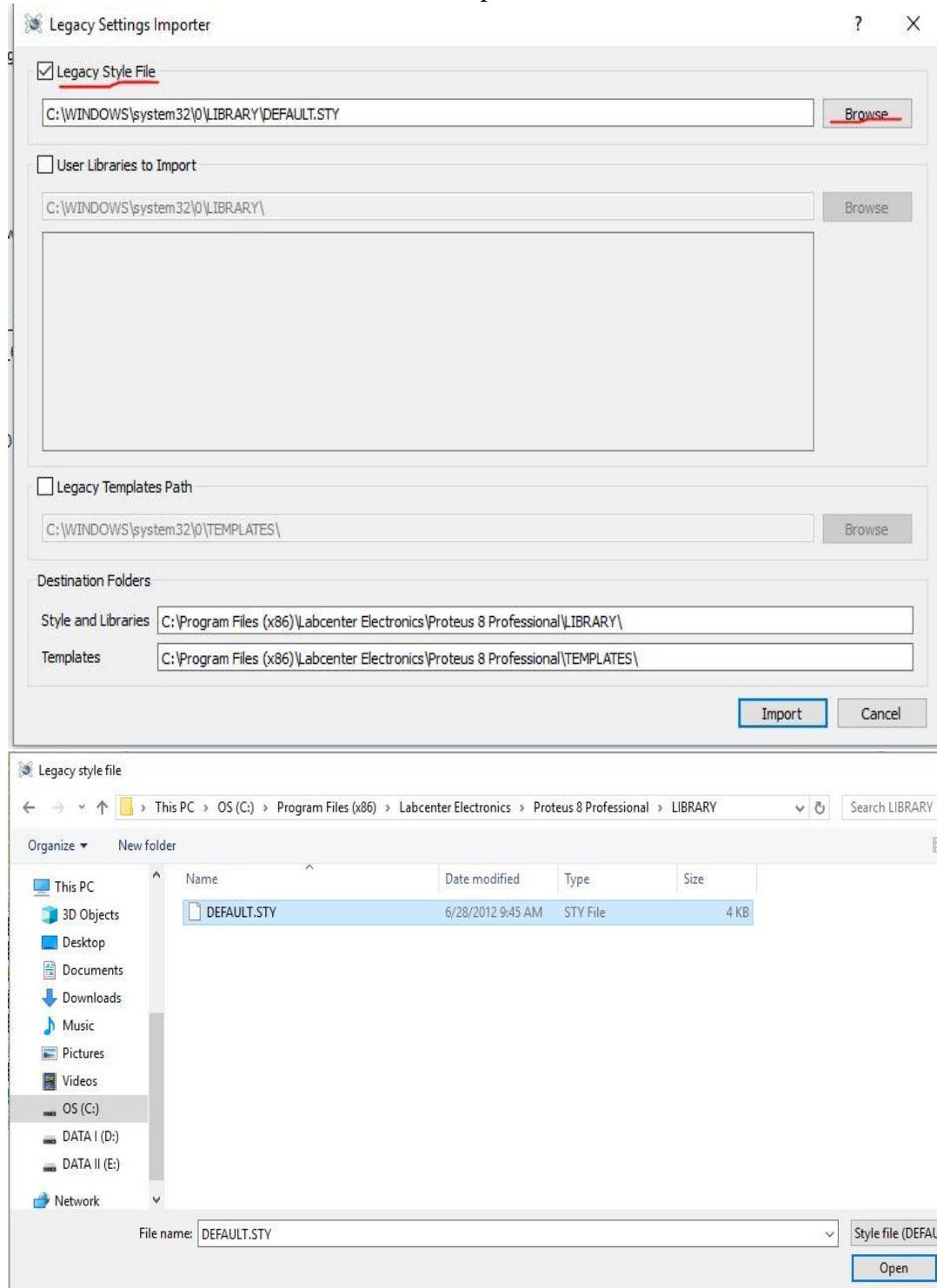


10. Dari Jendela Custom Setup sampai dengan Jendela Begin Installation of Proteus 8 Professional, tidak ada yang perlu diubah, jadi next saja sampai muncul tombol install dan tekan Install. Tunggu proses instalasi sampai muncul jendela baru nantinya.

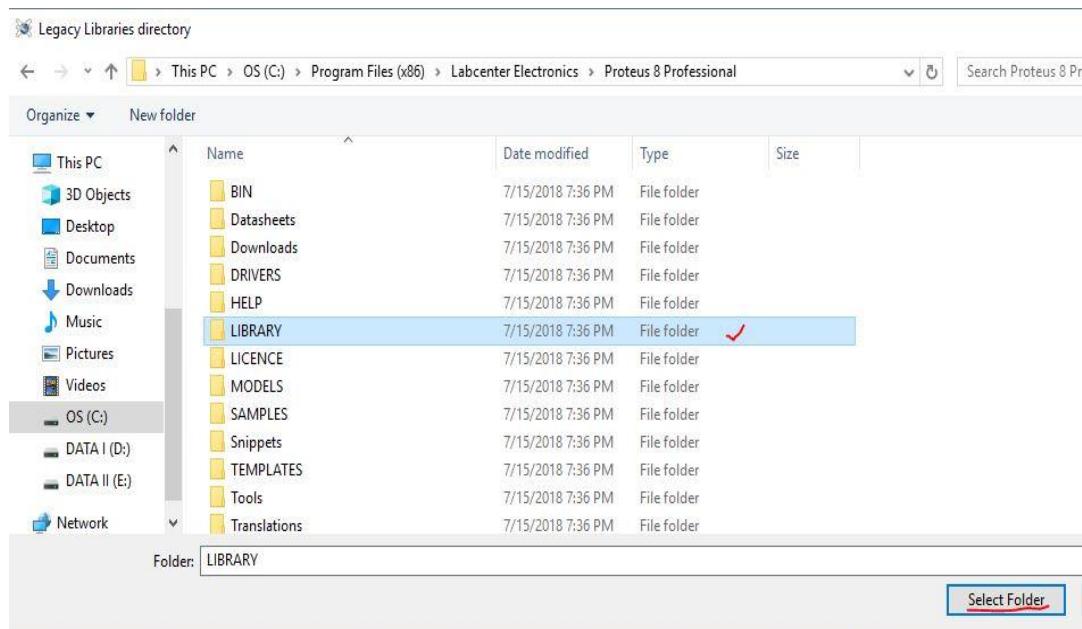
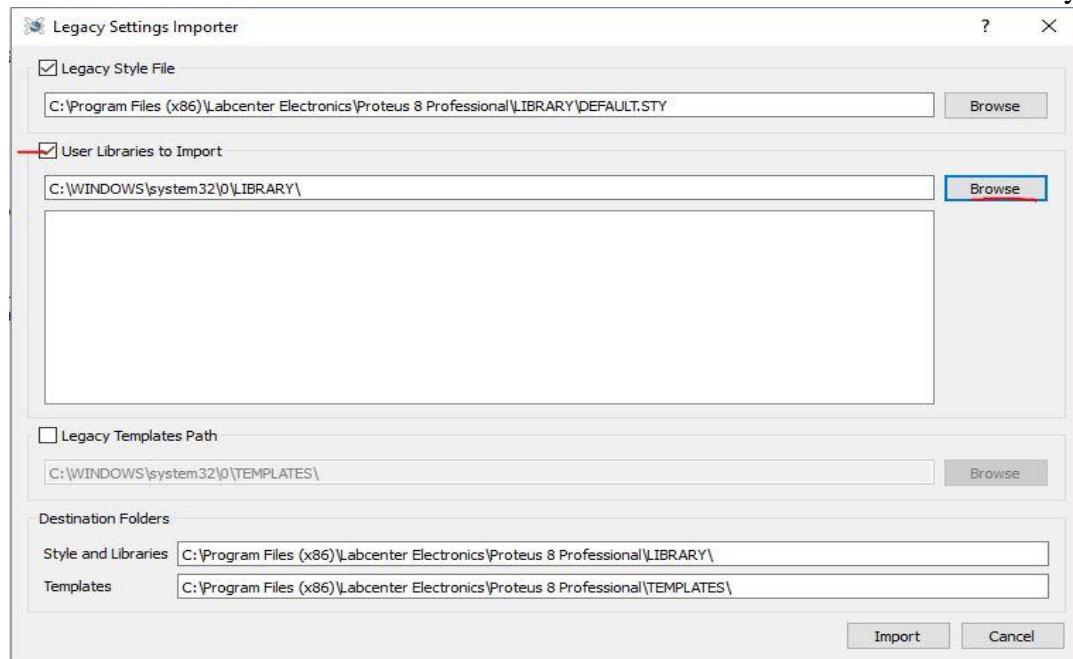


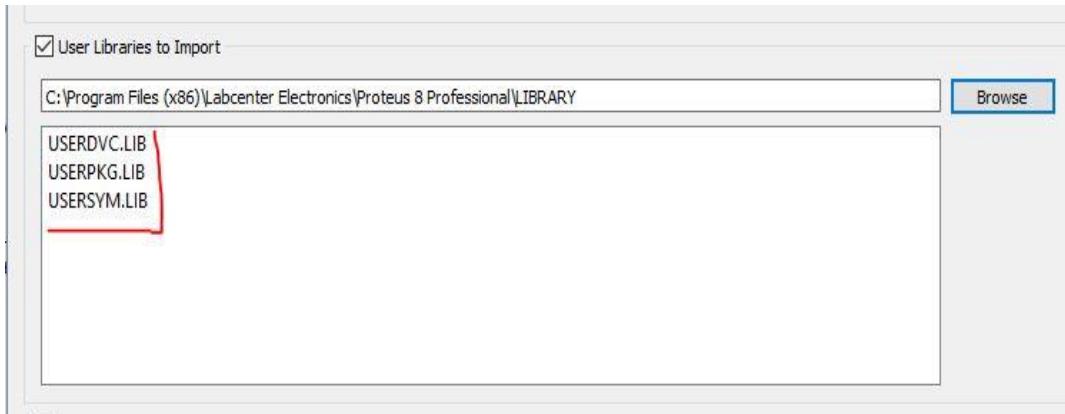


11. Setelah itu akan muncul jendela baru tanpa mengakhiri jendela sebelumnya, yaitu jendela Legacy Settings Importer. Pertama kita centang Legacy Style File dan Browse. Arahkan kedalam Folder Program Files (Jika OS 64 bit, Program Files(x86)) > Labcenter electronics > Proteus 8 Professional > LIBRARY > Open file DEFAULT.STY.

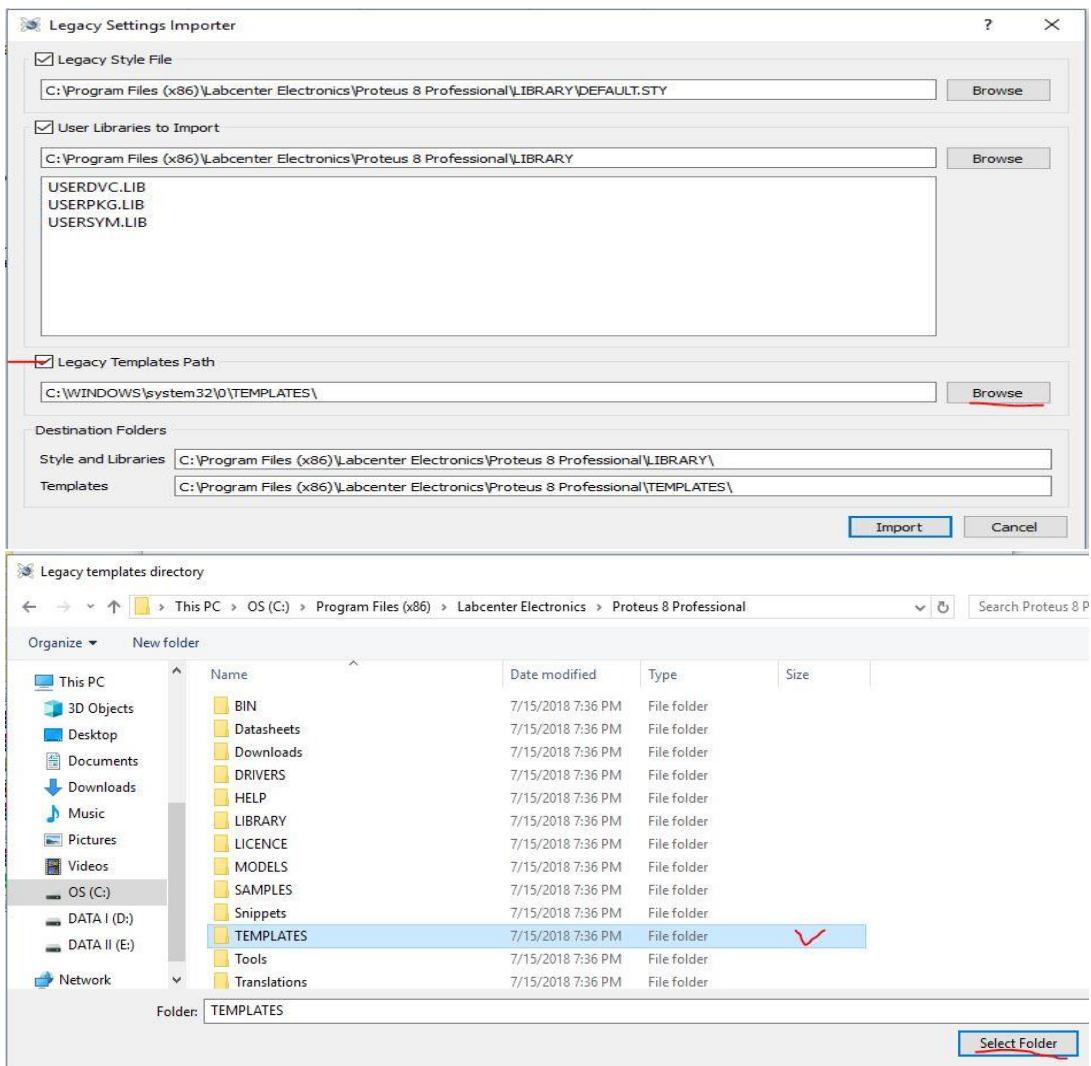


12. Kedua kita centang “User Libraries to Import” dan Browse lagi, arahkan ke Folder Program Files (Jika OS 64 bit, Program Files(x86)) > Labcenter electronics > Proteus 8 Professional > LIBRARY > Select Folder. Setelah itu akan muncul 3 library baru.

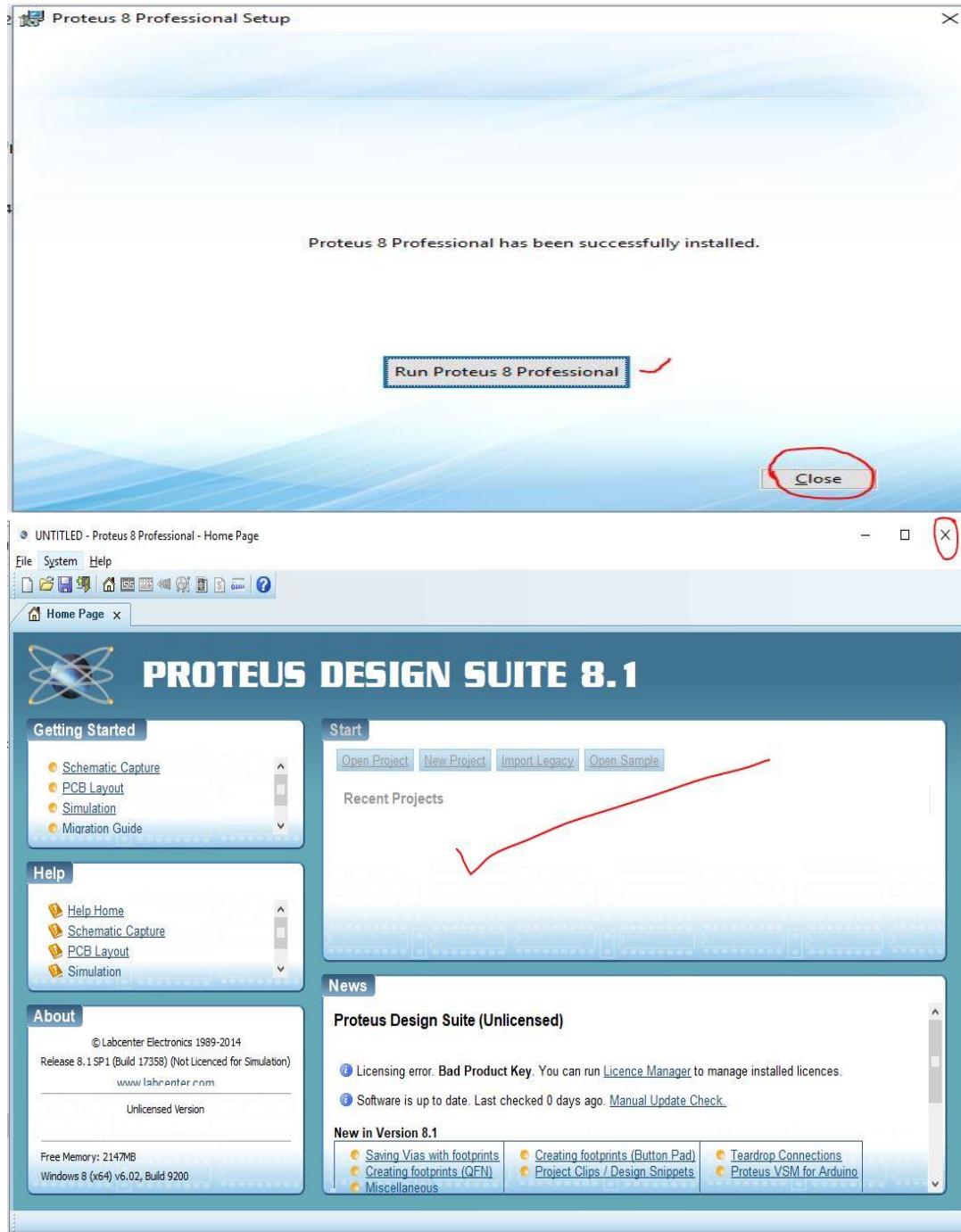




13. Ketiga centang "Legacy Templates Path" dan Browse arahkan ke folder Folder Program Files (Jika OS 64 bit, Program Files(x86)) > Labcenter electronics > Proteus 8 Professional > **TEMPLATES** > Select Folder.

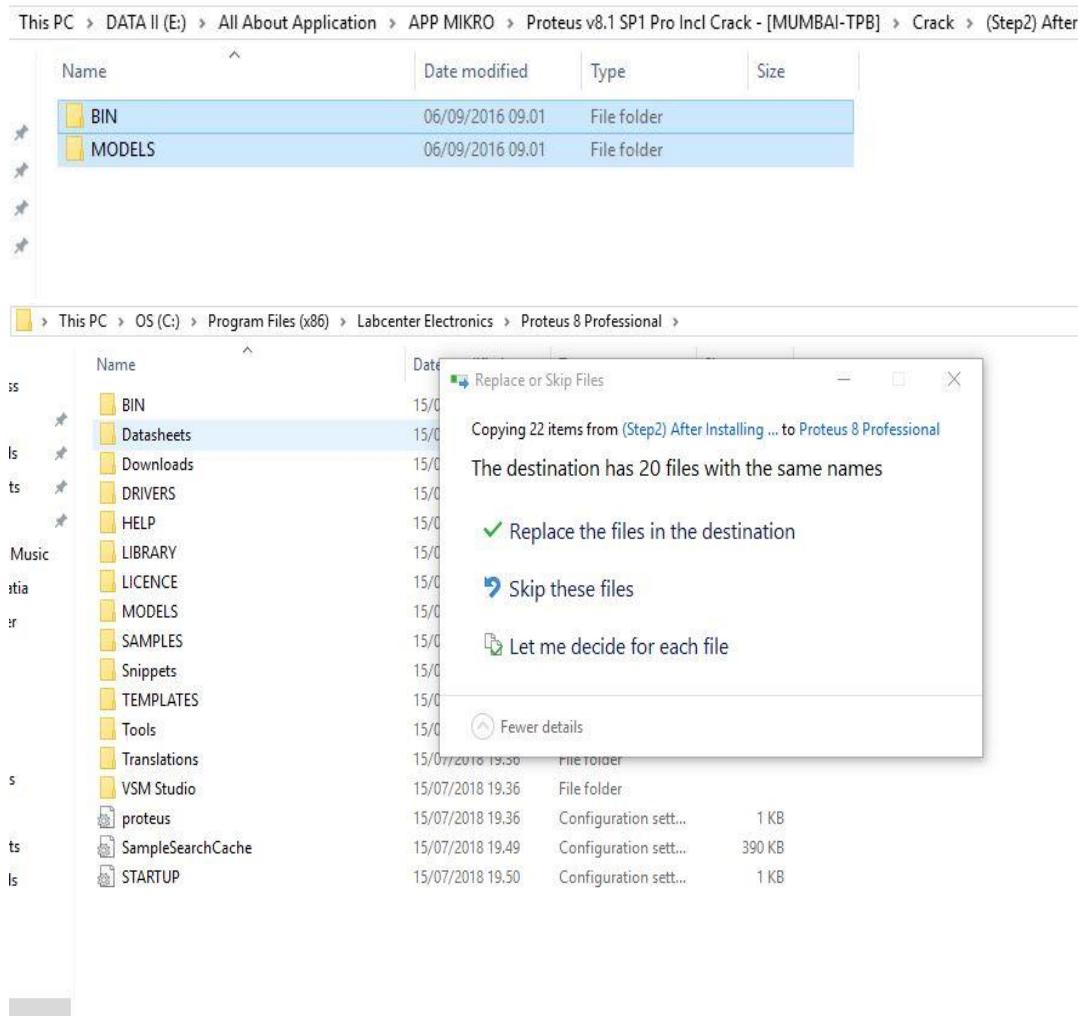


14. Setelah ketiganya dicentang, tinggal klik tombol import akan muncul pesan “Import Error”, abaikan saja langsung tekan OK, Instalasi Selesai perlu dicoba terlebih dahulu (RUN) aplikasinya apakah bisa berjalan, meskipun dalam keadaan belum diberikan lisensi, setelah bisa dijalankan, exit saja terlebih dahulu.



15. Selanjutnya buka folder Paket Aplikasi Mikro sebelumnya, Buka Folder Proteus v8.1 > Crack > Step 2 After Installation > Copy kedua Folder BIN dan MODELS kedalam Folder

Program Files (Jika OS 64 bit, Program Files(x86)) > Labcenter electronics > Proteus 8 Professional > Pastekan Folder tersebut dan akan muncul pesan Replace or Skip > Replace the file in the destination. Setelah berhasil dicopykan kedalam folder tersebut, instalasi selesai dengan full version proteus 8.1, tinggal jalankan aplikasi tersebut.



B. Instalasi CodeVision

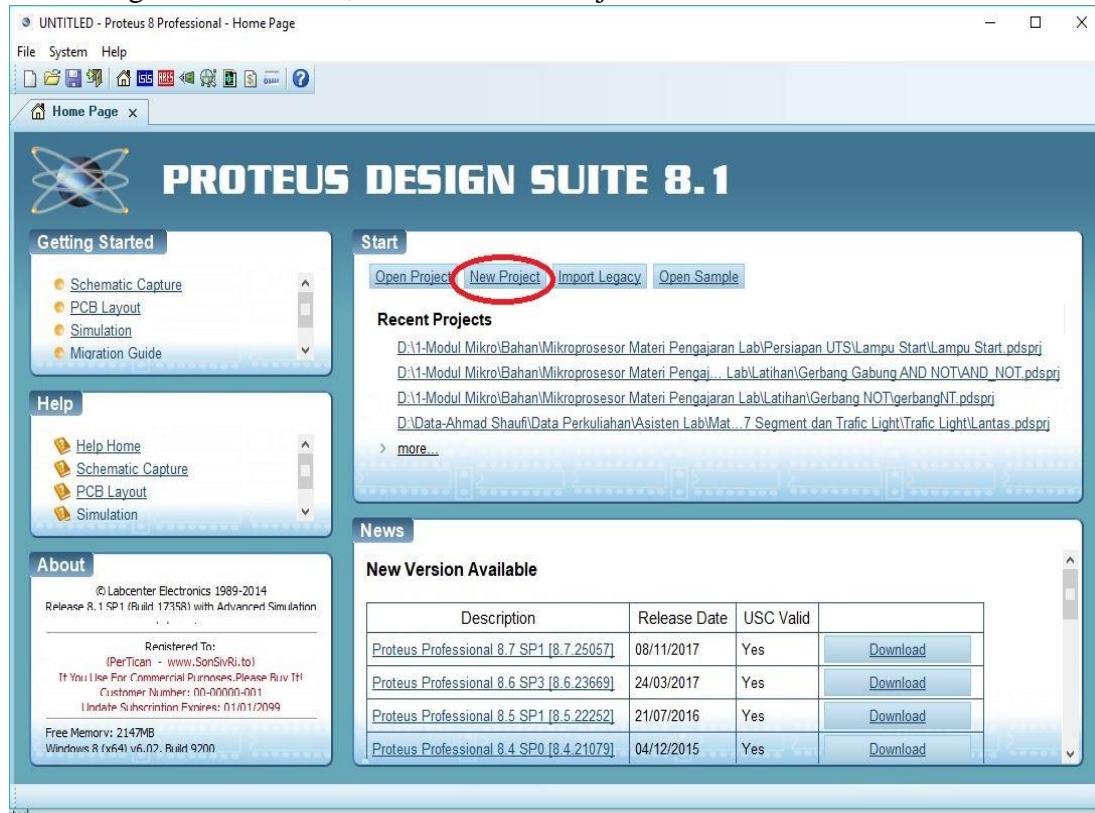
Untuk Instalasi Code Vision sama saja seperti instalasi pada aplikasi-aplikasi biasanya, tidak ada memasukkan path dan sebagainya, jadi tinggal next atau lanjut saja sampai instalasi berhasil.

BAB II

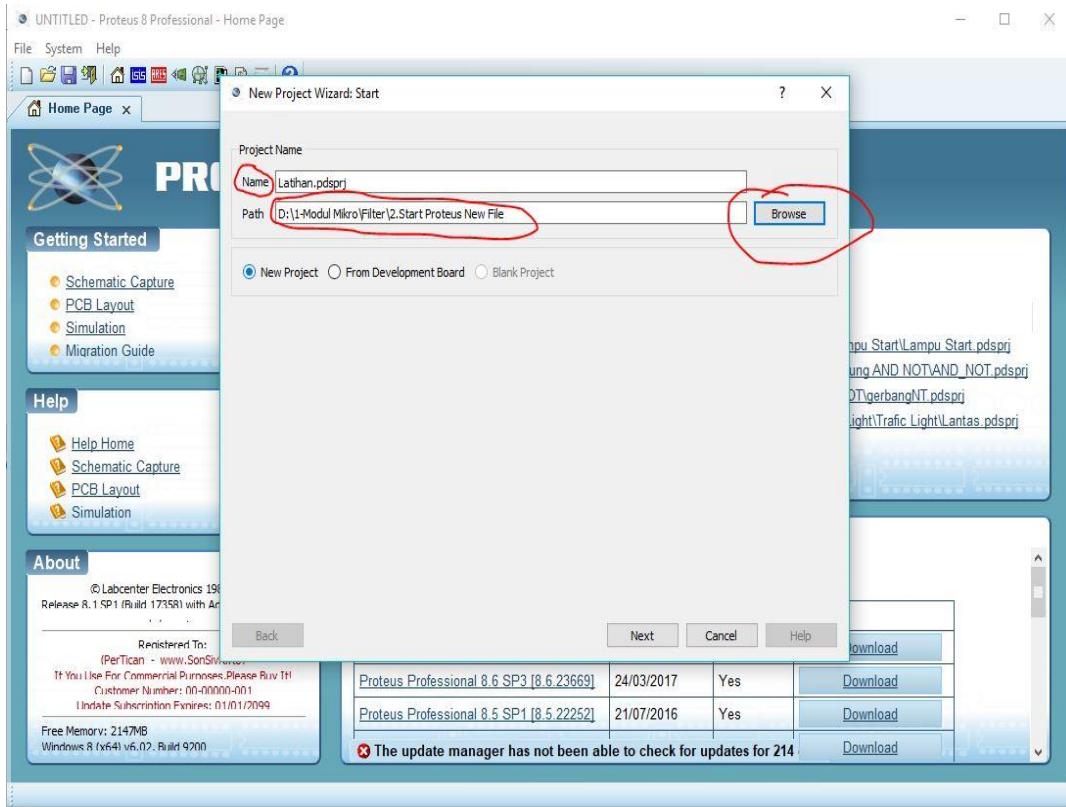
MENGENAL APLIKASI PROTEUS 8.1

Setelah selesai melakukan Pemasangan Aplikasi Proteus 8.1, maka kita bisa memulai membuat berbagai rangkaian yang berhubungan dengan Komponen-komponen Mikroprosessor. Sebelum memulai yang perlu diperhatikan adalah biasakan menyimpan **FILE PROTEUS DIDALAM SATU FOLDER**, dikarenakan File yang dibuat akan terpecah menjadi beberapa file setelah disimpan. Berikut ini adalah langkah-langkah pembuatan File baru di Proteus.

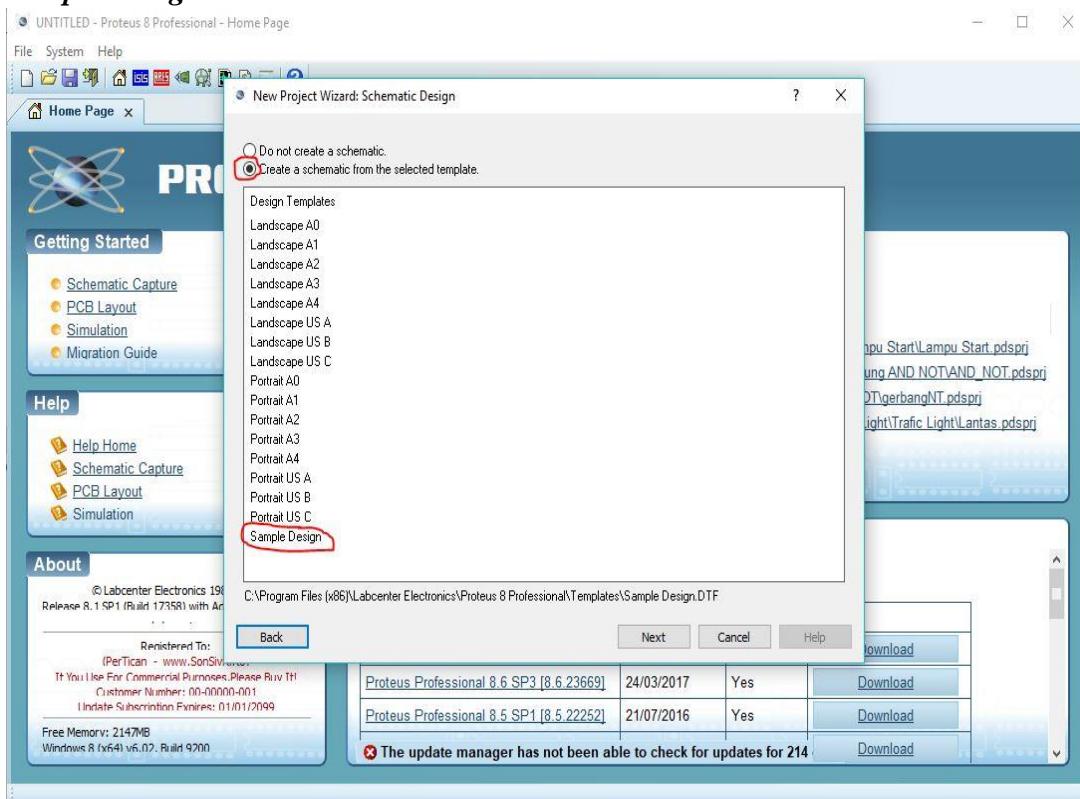
1. Buka Program Proteus 8.1, dan Pilih New Project.



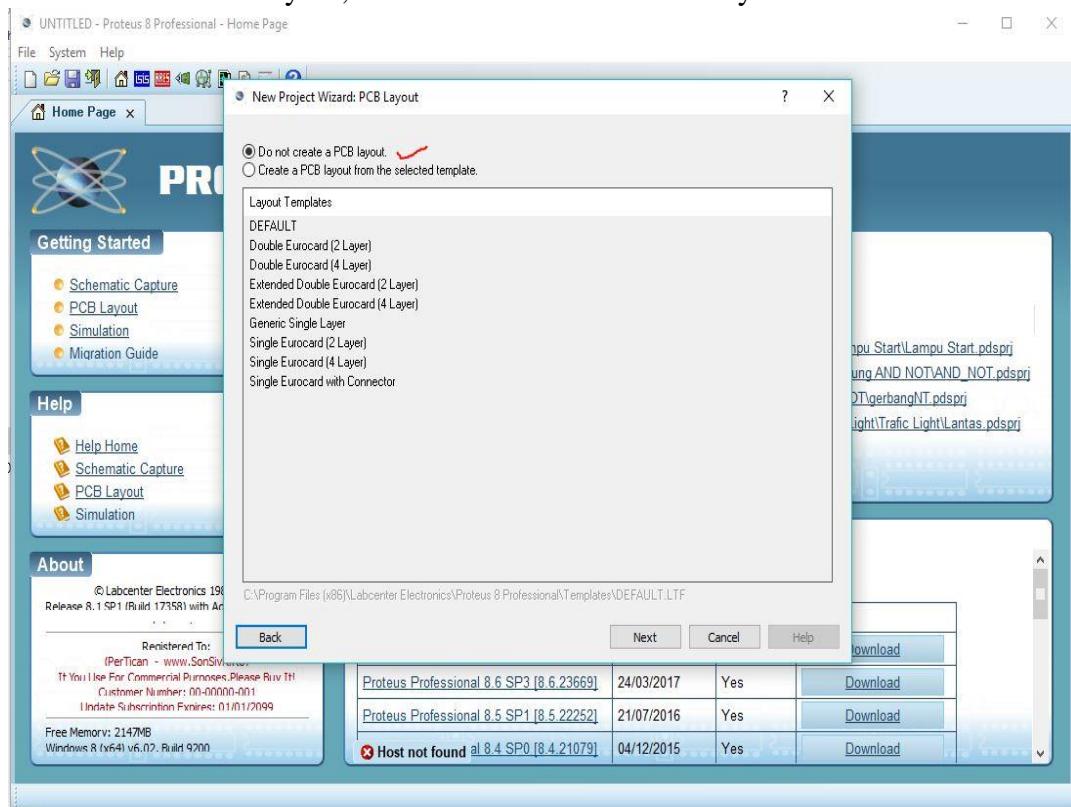
2. Setelah kita pilih New Project, maka akan muncul New Project Wizard. Masukkan Nama File dan Tentukan Letak File tempat kita menyimpan. **Disarankan Buat Folder baru dengan Nama sesuai latihan yang dibuat, dan Simpan Filenya di folder tersebut.** Setelah itu pilih Next.



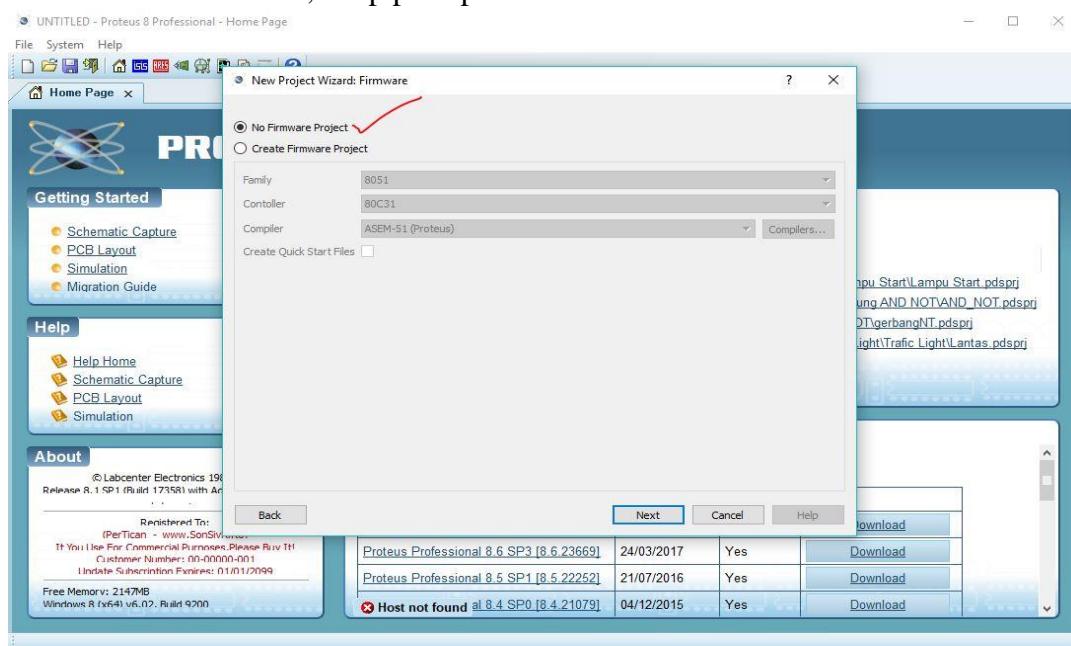
3. Pada Jendela Schematic Design, Pilih **Create a Schematic from the selected template** dan **Sample Design**. Lalu Next.



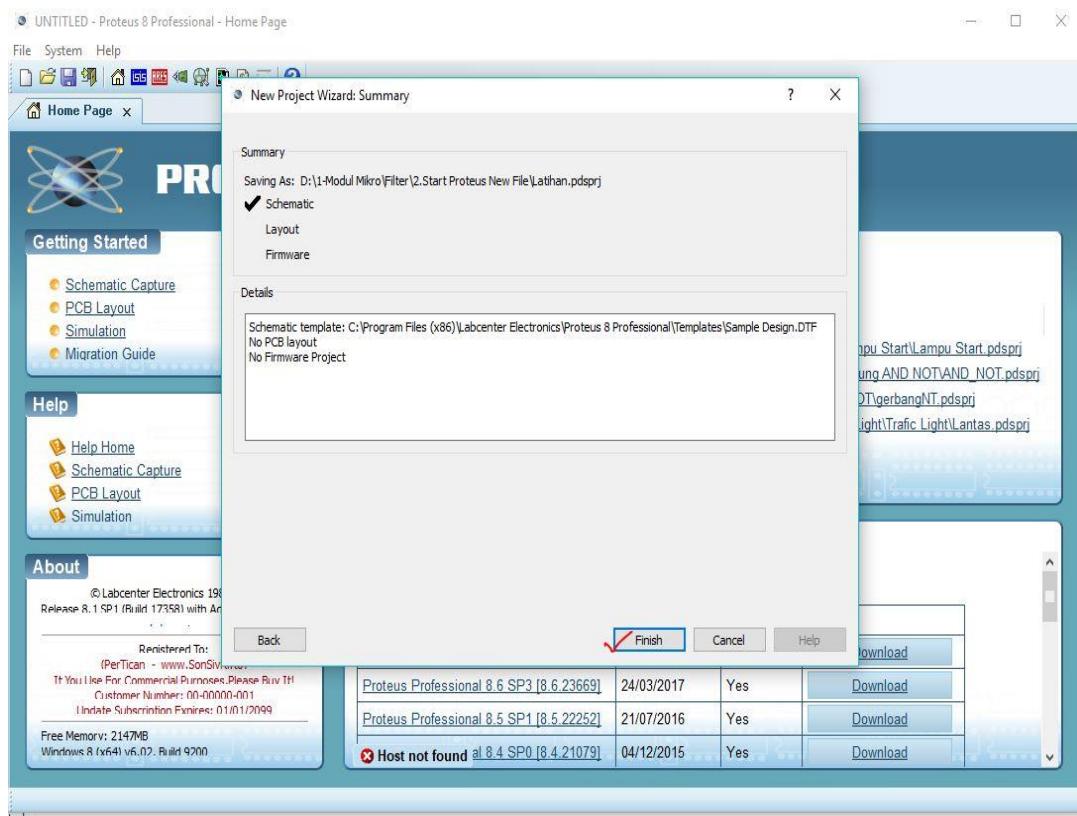
4. Pada Jendela PCB Layout, Pilih Do not Create a PCB Layout



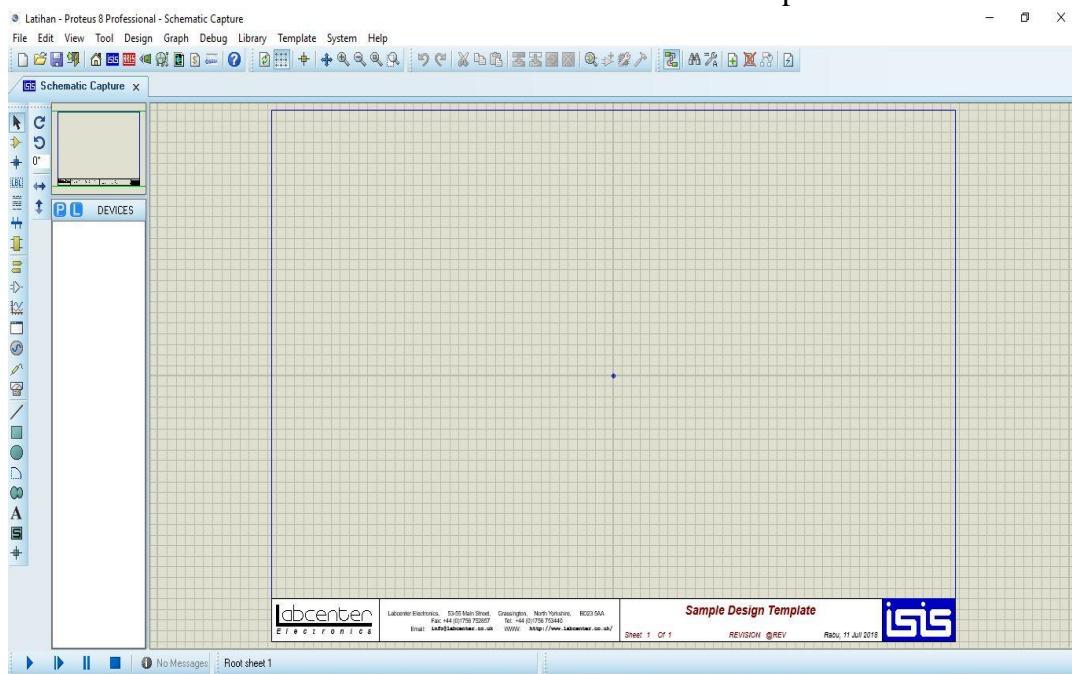
5. Pada Jendela Firmware, tetap pada posisi dan Next.



6. Setelah itu Klik Finish.

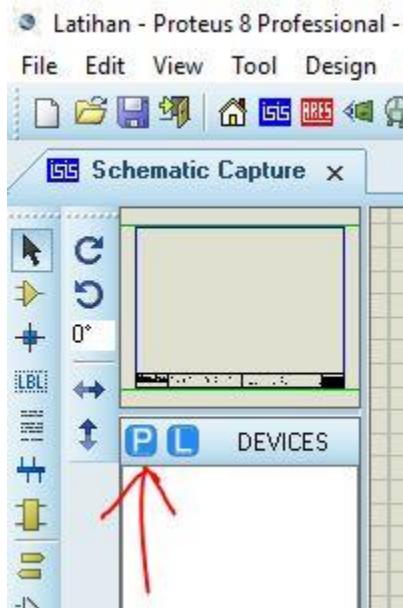


7. Berikut adalah Tampilan Form Kerja pada Proteus, hal yang terpenting pada saat awal belajar adalah bagaimana cara memasukkan Komponen-komponen Mikro yang ada di Proteus. Pada nomor 8 akan dibahas cara memasukkan Komponen mikro ke Form Kerja.

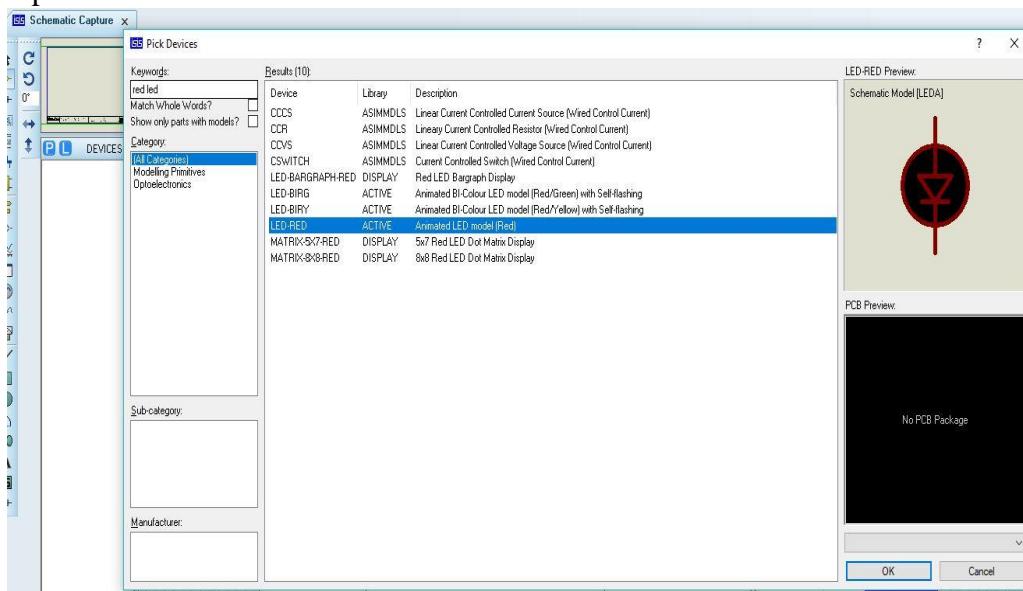


8. Cara memasukkan Komponen- Komponen Mikro kedalam Form, Ikuti langkah-langkah berikut :

- a. Klik Logo huruf (P) disamping Kiri Layar, disana terdapat seluruh Komponen yang biasanya digunakan Pada Rangkaian-rangkaian Digital yang sering kita temui.



- b. Setelah itu akan muncul jendela yang menampilkan Semua Komponen berdasarkan nama-nama yang sudah ditentukan. Silahkan cari nama komponen yang kita perlukan untuk rangkaian yang kita buat pada bagian **Keywords**, setelah kita tuliskan nama komponen tersebut akan muncul dibagian **Results**, Pilihlah Komponen yang kita inginkan dengan catatan pada **Kolom Library** dalam **Kondisi ACTIVE**. Setelah dipilih klik dua kali kiri atau OK.



- c. Maka Komponen yang kita pilih akan muncul didaftar Devices, untuk memasukkan kedalam Form kerja tinggal Klik Satu kali, lalu arahkan Kursor kedalam Form Kerja dan Klik Kiri satu kali. Contohnya seperti dibawah ini :



- d. Silahkan Coba Cari dan Masukkan Komponen baru lagi dengan nama OR, XOR, NOT, NAND.

BAB III

MENDESAIN RANGKAIAN DIGITAL DENGAN PROTEUS 8.1

- **Fitur Running Program dan Cek Error**

Sebelum masuk ke pembuatan Gerbang Logika AND, OR, NOR. Ada satu Fitur yang akan selalu kita gunakan untuk melakukan pengujian terhadap Hasil Rangkaian yang kita buat, Yaitu Fitur Running Program seperti Gambar dibawah ini :



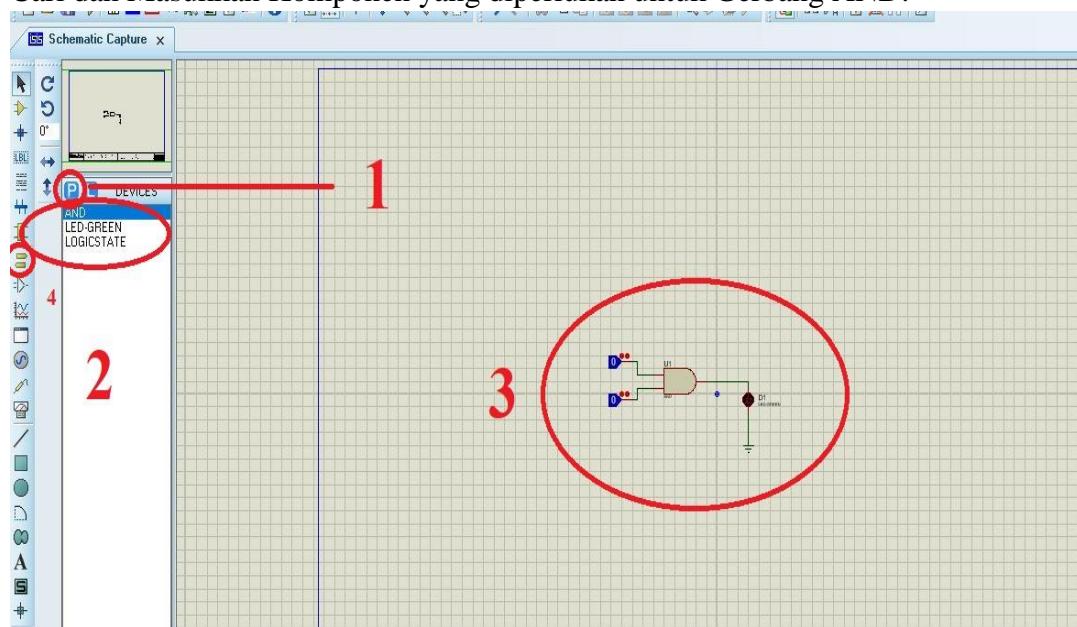
Jika terdapat kesalahan dalam pembuatan rangkaian maka akan muncul pesan error pada Fitur Running Program ini.

1. Gerbang Logika AND

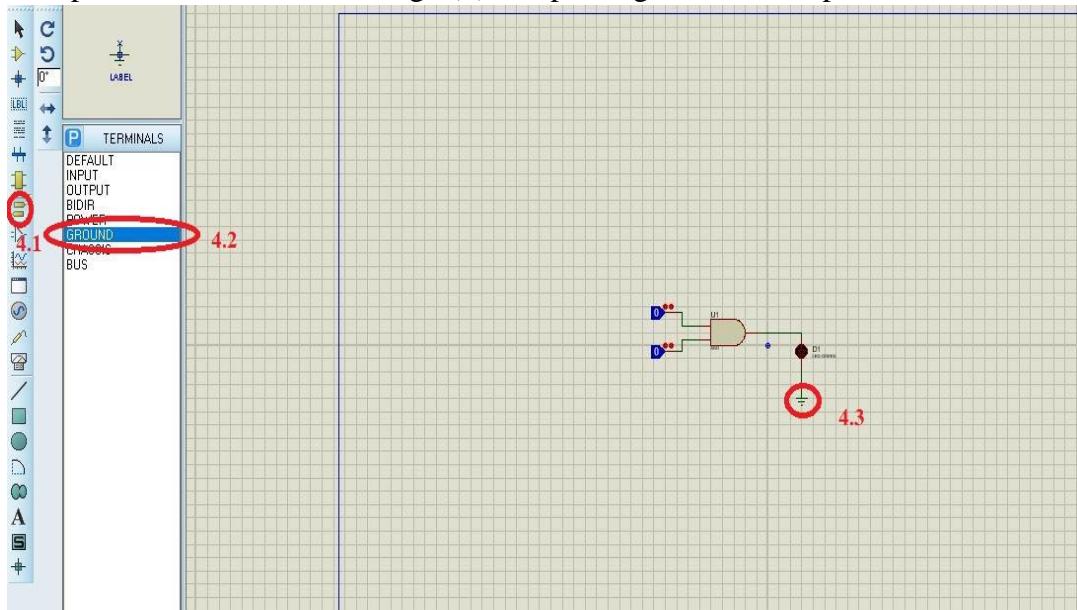
- a. Hal Pertama yang selalu dilakukan dalam membuat Rangkaian Digital adalah mencari dan memasukkan Komponen yang ingin digunakan dalam Rangkaian, Sama Halnya dengan Gerbang Logika AND. AND yang mendasarkan cukup menggunakan 3 Komponen Wajib, yaitu AND, LOGICSTATE,LED-GREEN. Ditambah 1 Alat untuk Daya yang disebut Ground, **dan Ground ini wajib dimasukkan setiap membuat Rangkaian Apapun.**

****Keterangan : AND adalah Gerbang AND, LOGICSTATE digunakan sebagai Input yang masuk kedalam gerbang yang menggunakan Angka Binary (1 dan 0), dan LED-GREEN digunakan sebagai OUTPUT (Sebenarnya Output tidak selalu menggunakan Lampu warna Hijau, bisa saja menggunakan warna lain dalam bahasa inggris seperti Yellow, Red, Blue).**

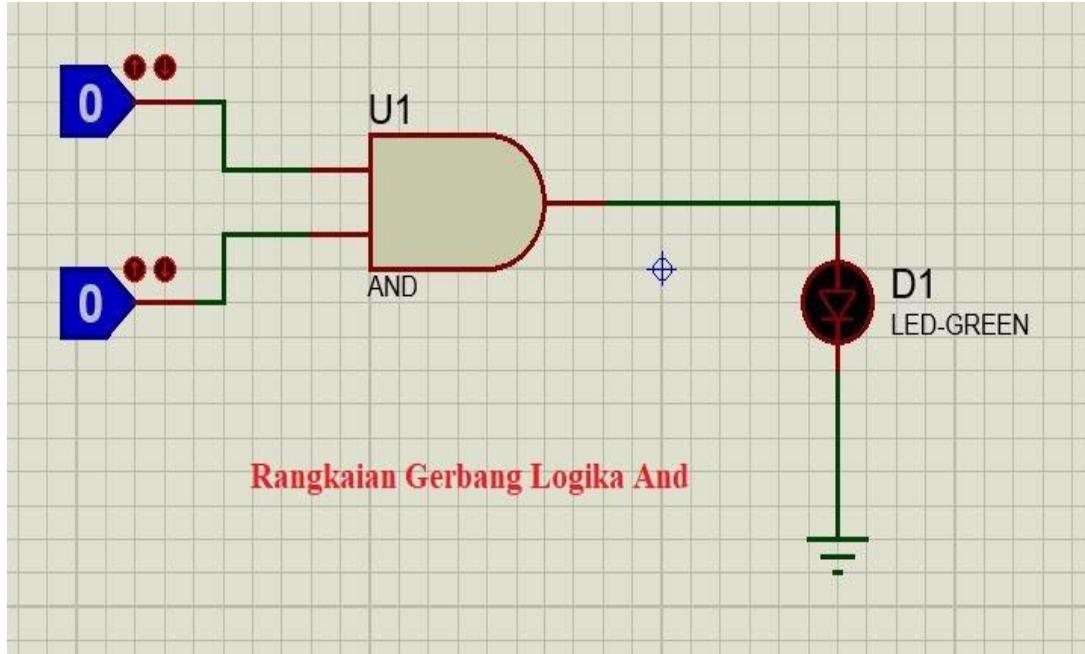
Cari dan Masukkan Komponen yang diperlukan untuk Gerbang AND.



- b. Setelah 3 Komponen AND, Selanjutnya masukkan Ground yang mana Letak Komponen ini tidak dicari di logo (P), tetapi dilogo Nomor 4 seperti dibawah ini.



- c. Setelah Semua Komponen ditemukan, buatlah Rangkaian seperti dibawah ini. Untuk membuat Kabel Penghubung cukup dengan mengklik ujung masing-masing komponen lalu hubungkan ke ujung komponen lainnya.



- d. Setelah dirangkai, maka kita bisa lakukan Pengujian dengan Menyalakan Lampu Hijau yang terdapat diujung Rangkaian yang terhubung dengan Ground dengan Mengklik Fitur Running Program dibagian bawah kiri Jendela. Adapun Rumus Persamaan dari

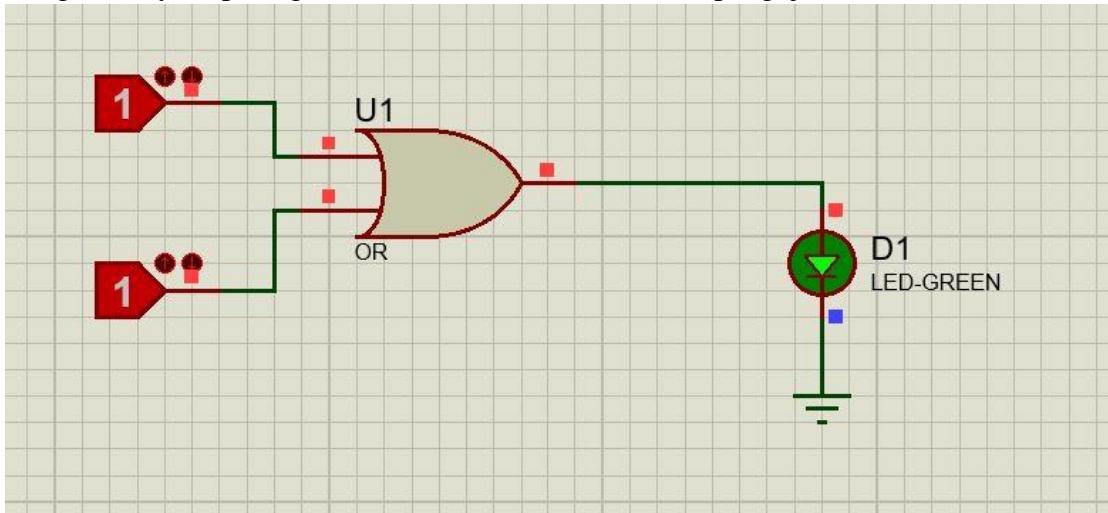
Gerbang AND adalah $Y = (A * B)$, pengujian bisa dilakukan Mengacu kedalam Tabel Kebenaran AND dibawah ini :

TABEL KEBENARAN LOGIKA AND

INPUT/LOGICSTATE		OUTPUT/LED
A	B	HASIL
1	1	1
0	1	0
1	0	0
0	0	0

2. Gerbang Logika OR

- a. Komponen yang diperlukan untuk membuat Gerbang logika OR ada 3, yaitu Gerbang OR, LOGICSTATE, LED-RED. Jangan lupa tambahkan Ground diujung lampu untuk Daya. Adapun Rumus Persamaan dari Gerbang OR adalah $Y = (A + B)$, Silahkan buat Rangkaianya seperti gambar dibawah ini dan lakukan pengujian :

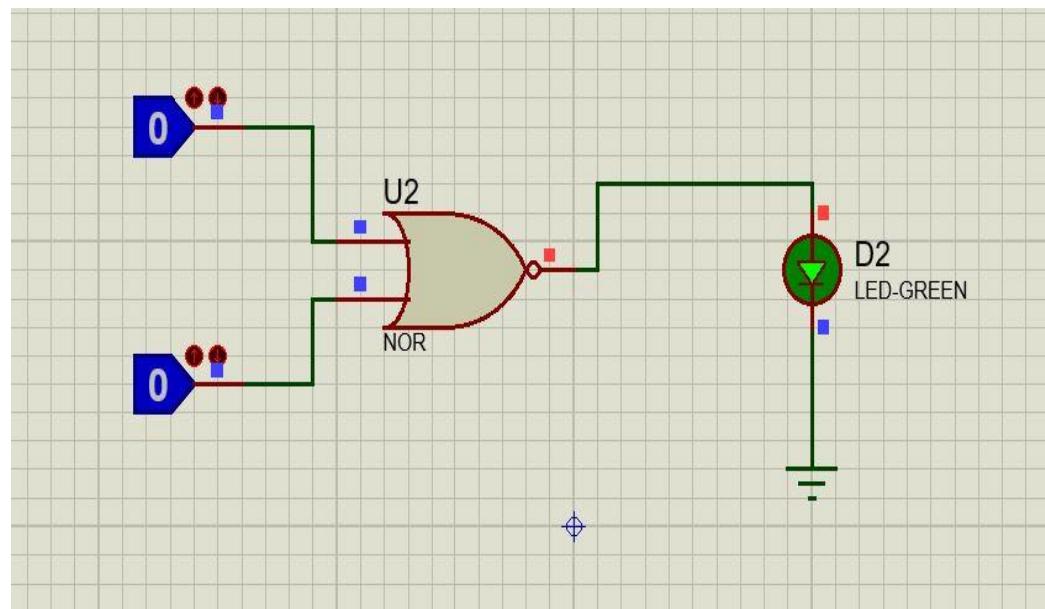


TABEL KEBENARAN LOGIKA OR

INPUT/LOGICSTATE		OUTPUT/LED
A	B	HASIL
1	1	1
0	1	1
1	0	1
0	0	0

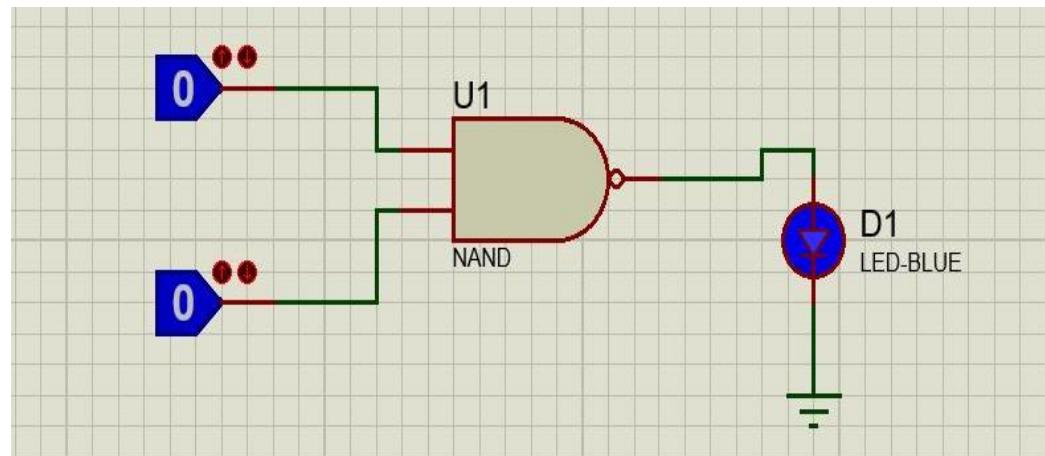
3. Gerbang Logika NOR(NOT OR) / LATIHAN MANDIRI

- Silahkan Analisa, Buat Rangkaian, Tabel Kebenaran Gerbang Logika NOR dan Lakukan Pengujian berdasarkan Tabel Kebenaran yang dibuat !!!!!!



4. Gerbang Logika NAND(NOT AND) / LATIHAN MANDIRI

- Silahkan Analisa, Buat Rangkaian, Persamaan serta Tabel Kebenaran Gerbang Logika NAND dan Lakukan Pengujian berdasarkan Tabel Kebenaran yang dibuat !!!!!!



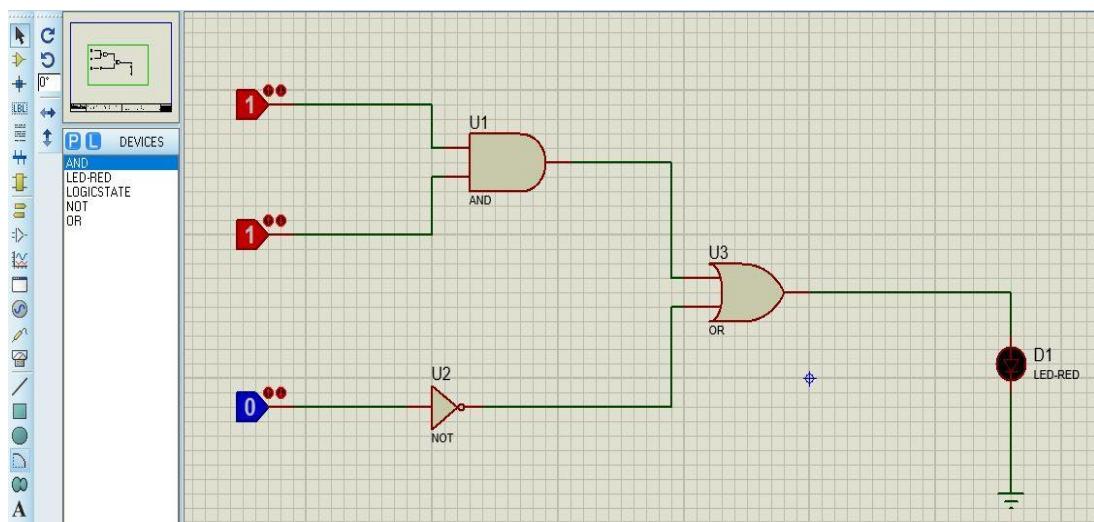
BAB IV

RANGKAIAN KOMBINASI ANTAR GERBANG

Pada Bab sebelumnya kita sudah mengenal beberapa gerbang dasar yang sering kita gunakan pada saat pembelajaran, adapun dalam sebuah rangkaian agar bisa menghasilkan Output yang kita inginkan maka kita harus bisa membuat Rumus Persamaan, kalau didalam gerbang logika AND persamaannya adalah $Y = A * B$, maka setelah didapat persamaan baru bisa kita melihat hasil yang kita inginkan, bagaimana dengan Rangkaian yang kita buat Sendiri dengan mengkombinasikan beberapa gerbang dasar ? Akan dibahas didalam Bab ini.

Berikut beberapa contoh membuat Rangkaian Kombinasi, Rumus Persamaan dan Tabel Kebenaran rangkaian tersebut.

1. Contoh pertama dengan menggunakan gerbang AND, OR, NOT, dengan 3 Input Variabel.



Setelah membuat rangkaian seperti pada gambar diatas, maka langkah selanjutnya adalah membuat Rumus Persamaannya agar bisa menentukan Input yang bisa menghasilkan Output lampu menyala. Rumus Persamaannya adalah :

$$Y = (A * B) + \sim C$$

Bagaimana bisa mendapatkan Persamaan tersebut, Perhatikan Langkah berikut :

- Tentukan Nama Input Variabel yang terdapat pada rangkaian misalkan **A, B C dari LOGICSTATE paling atas**.
- Setelah itu tentukan darimana kita memulai mencari rumus-perumusnya, misalkan kita awali dari paling kiri atas dengan Gerbang AND sebagai penentuan rumus pertama. Maka kita bisa mendapatkan rumus pertama yaitu ($A * B$).

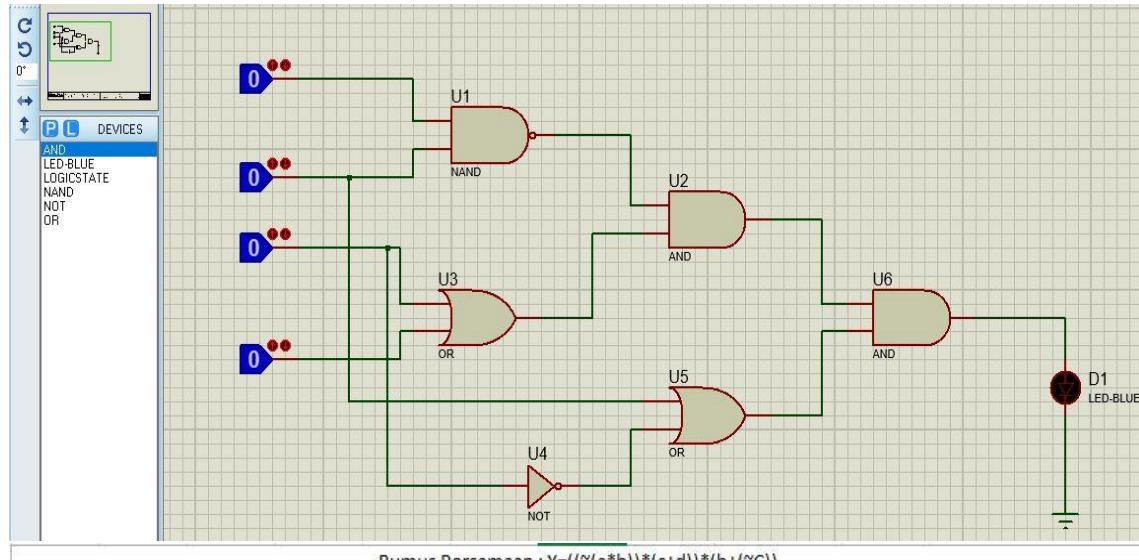
- c. Setelah dapat rumus pertama, lanjut dengan Variabel selanjutnya yaitu C, karena input C bertemu dengan Gerbang NOT, artinya gerbang NOT adalah Nilai sebelumnya dilakukan Pembalikan Binary, jika nilai awal $C = 0$ setelah masuk gerbang NOT maka nilai C menjadi $C = 1$, jika nilai awal $C = 1$ setelah masuk gerbang NOT maka nilai C menjadi $C = 0$. Maka kita dapat rumus kedua yaitu ($\sim C$) atau Negasi C.
- d. Langkah terakhir adalah dengan menggabung rumus yang didapat sebelumnya dengan gerbang yang tersisa terakhir yaitu gerbang OR (+), maka rumus persamaan terakhirnya adalah $Y = (A * B) + \sim C$.

Setelah kita menentukan rumus persamaan, maka lanjut kita tentukan tabel kebenarannya, sebelum kita membuat Tabelnya, tentukan terlebih dahulu jumlah baris dari Angka binary yang kita perlukan dengan melihat jumlah Input pada Rangkaian. Cara menentukan jumlah angka binary pada Tabel kebenaran dengan input yang dibuat adalah Dengan Menghitung Jumlah Input dipangkatkan 2, Contohnya jika 2 Input (A,B) maka 2^2 maka ada 4 baris angka binary nantinya (**perhatikan tabel kebenaran AND, OR bab 3**). Karena kita mempunyai 3 Input maka (2^3) yaitu 8 baris. Maka Tabel Kebenarannya Seperti yang dibawah ini :

<u>A</u>	<u>B</u>	<u>C</u>	<u>A*B</u>	<u>$\sim C$</u>	<u>$Y=(A*B)+\sim C$</u>
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

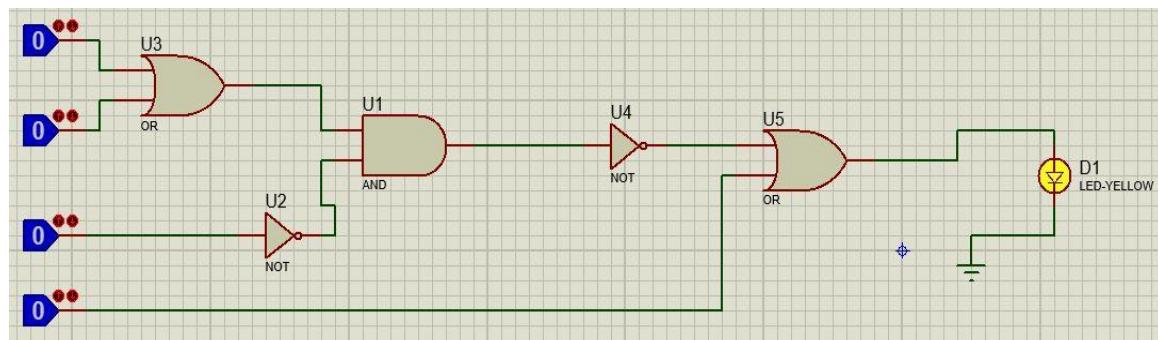
Jumlah Angka Binary Per-Kolom ada 8 Baris, setelah kita sudah selesai membuat Tabel Kebenaran, maka tinggal melakukan pengujian dengan Input A,B,C dan Output Hasil pada Kolom $Y = (A*B) + \sim C$.

- 2. Contoh kedua, silahkan Perhatikan dan Pahami Rangkaian, Rumus Persamaan dan Tabel Kebenaran dibawah ini, Setelah itu buat rangkaian dan lakukan pengujian menggunakan Aplikasi Proteus.**



A	B	C	D	$a*b$	$\sim a*b$	$c+d$	$\sim c$	$((\sim a*b)*(c+d))$	$(b+\sim c)$	$Y=((\sim a*b)*(c+d))*(b+\sim c)$
0	0	0	0	0	1	0	1	0	1	0
0	0	0	1	0	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	0	0
0	0	1	1	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0
0	1	0	1	0	1	1	1	1	1	1
0	1	1	0	0	1	1	0	1	1	1
0	1	1	1	0	1	1	0	1	1	1
1	0	0	0	0	1	0	1	0	1	0
1	0	0	1	0	1	1	1	1	1	1
1	0	1	0	0	1	1	0	1	0	0
1	0	1	1	0	1	1	0	1	0	0
1	1	0	0	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1	0	1	0
1	1	1	0	1	0	1	0	0	1	0
1	1	1	1	1	0	1	0	0	1	0

- 3. Kerjakan Rangkaian, Rumus Persamaan dan Tabel Kebenaran berikut ini :**



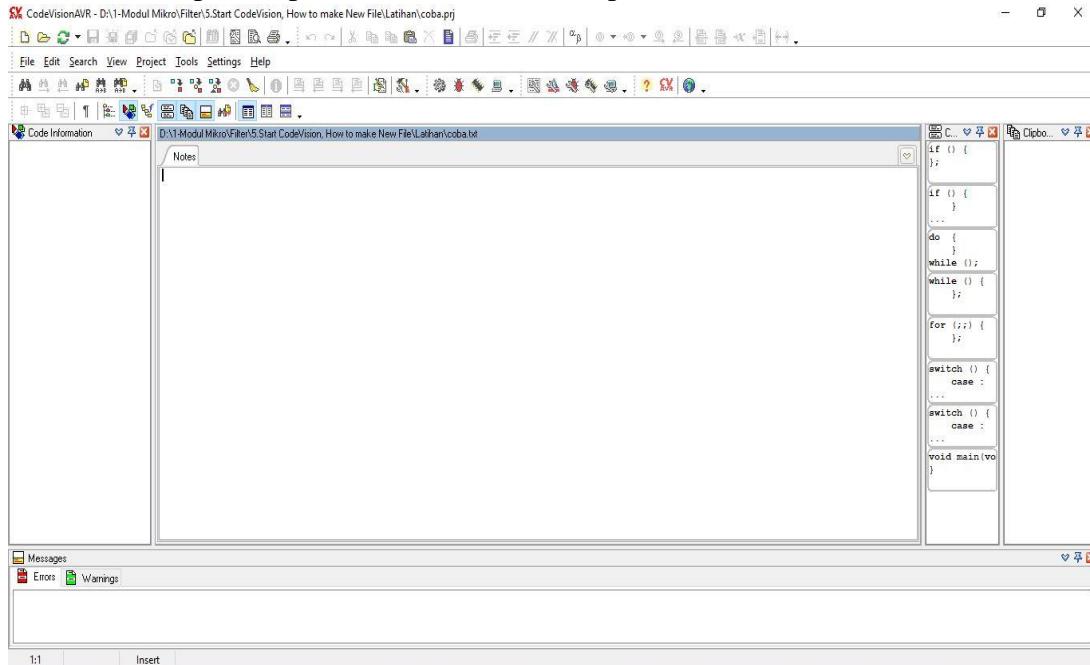
BAB V

PENGENALAN APLIKASI CODEVISION

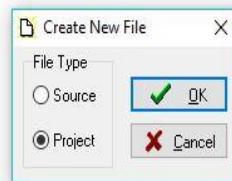
CodeVision adalah salah satu aplikasi berbasis bahasa program C++ untuk membuat perintah terhadap rangkaian yang dari segi Komponen sudah mulai rumit dan tidak memungkinkan lagi menggunakan/memasukkan satu-persatu Bilangan Binary secara manual. Salah satu kegunaan CodeVision adalah memfungsikan Chip-chip mikro yang mana didalam chip tersebut terdapat banyak gerbang, rangkaian dan semacamnya yang sudah disisipkan kedalam satu Chip.

Dalam membuat File baru dicodevision, ada beberapa hal yang harus diperhatikan antara lain seperti **Komponen Chip Rangkaian yang digunakan, Jumlah Port yang digunakan, Komponen Output yang digunakan, dan sebagainya.** Adapun cara membuat file baru adalah seperti dibawah ini.

1. Pada awal tampilan Aplikasi CodeVision, kita pilih File > New File



2. Setelah New File, pilih File Type “Project”, klik OK.

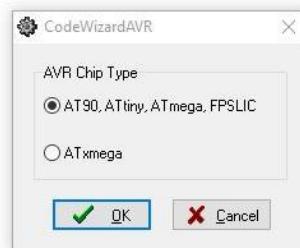


3. Muncul pesan Konfirmasi, Pilih Yes.

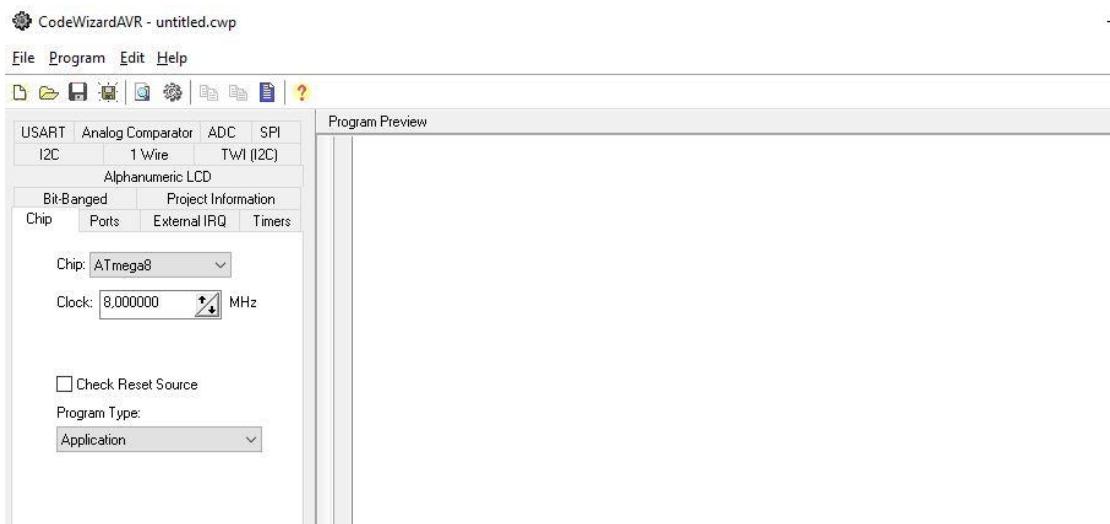


4. Selanjutnya muncul AVR Chip Type, sesuaikan dengan chip yang kita gunakan dirangkaian yang kita buat diproteus, misalkan kita menggunakan ATmega 8, maka cukup Pilih Chip Type AT90, ATtiny, ATmega, FPLIC.

...Lalu Materi Mengajar Mikroprosesor Untuk Materi Node 6 LCD V1.txt

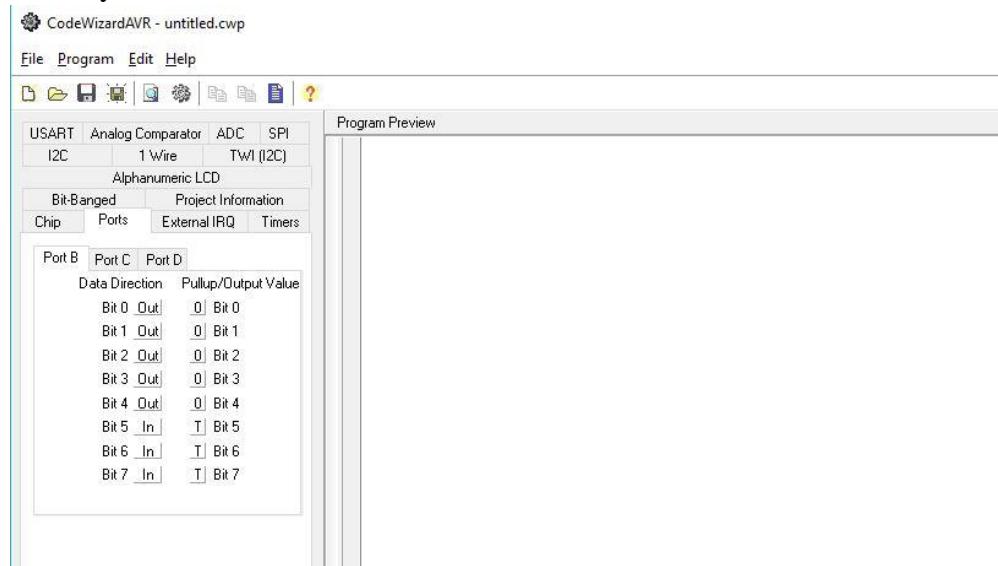


5. Selanjutnya akan muncul Pengaturan terhadap Chip yang kita gunakan seperti Jenis Chip, Port, dan Sebagainya. Sebagai Contoh dibagian Tab Chip kita atur dengan Chip ATmega8, Clock 8.000000 Mhz

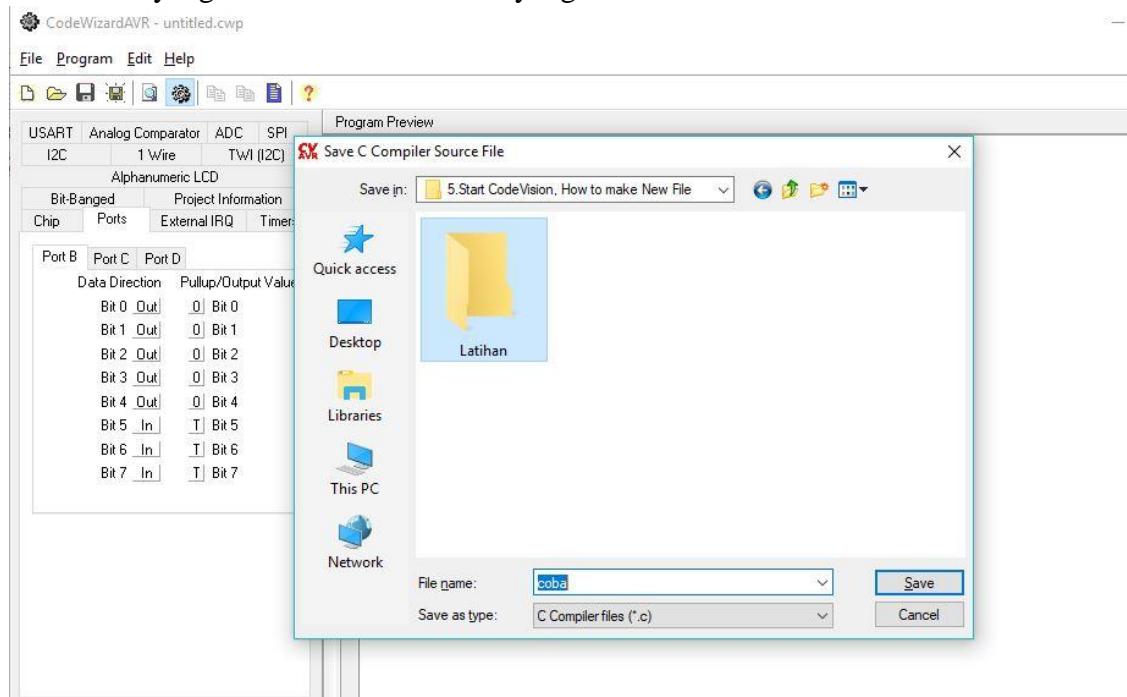


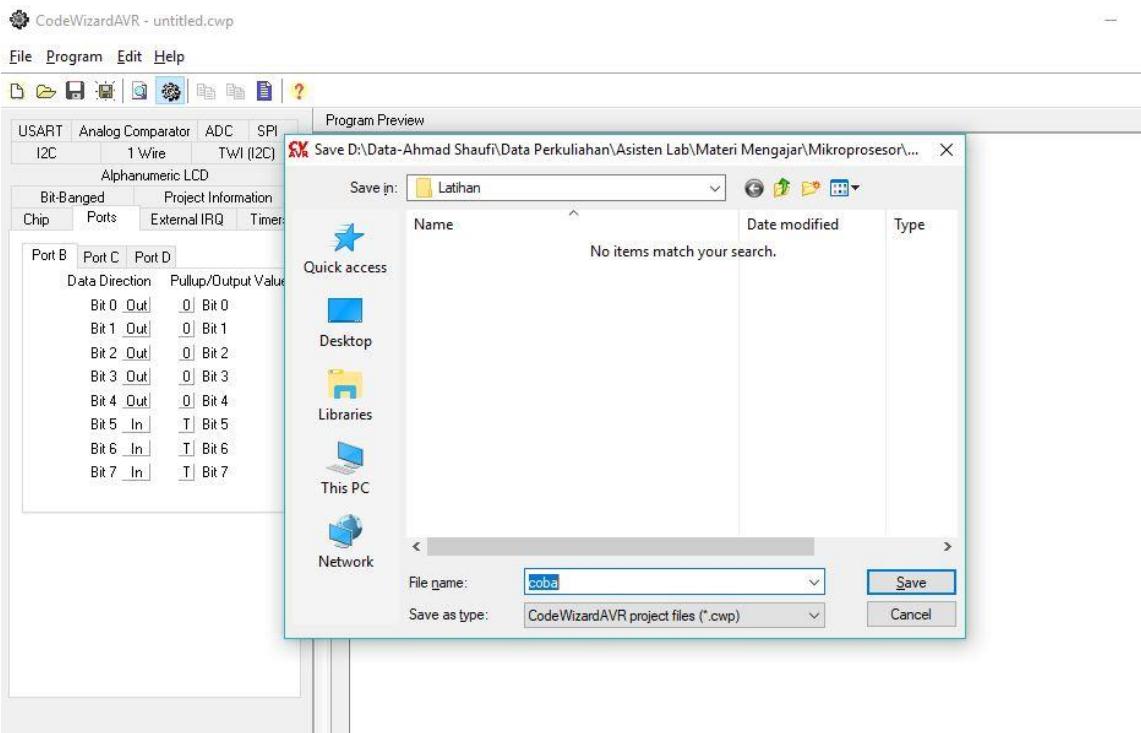
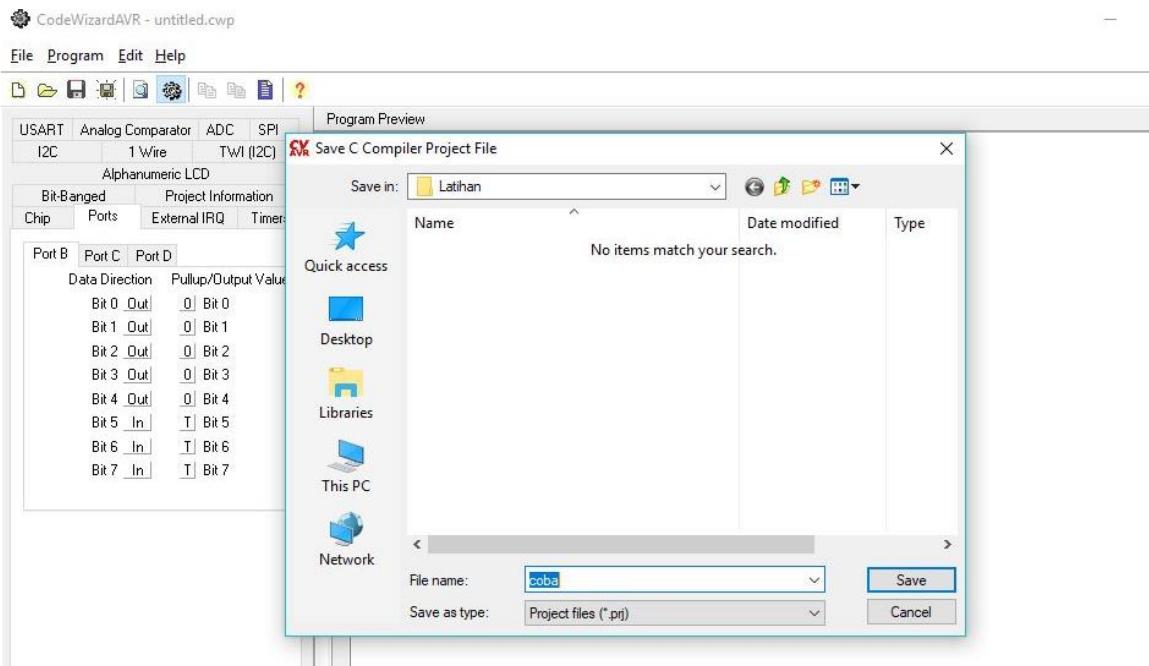
6. Lanjut kebagian Tab Port, Kita sesuaikan dengan Port yang kita gunakan pada rangkaian yang kita buat didalam Proteus, contohnya jika ATmega8 mempunyai 3 Port yaitu B, C, D, lalu yang kita gunakan adalah Port B, maka cukup Port B yang kita atur, yang lain diabaikan. Atur sesuai kebutuhan sebagai contoh OUT berarti kita ingin mengeluarkan

hasil dari Chip, IN memasukkan data dari luar ke dalam Chip, 0 untuk Value/Nilai Mati, 1 untuk Nilai Menyala, dll. Untuk mengganti cukup kita Klik 1 kali dibagian yang ingin kita ganti, misalnya dari IN ke OUT.



7. Ketika semuanya sudah kita atur, dilanjutkan dengan proses Generate Program dan Save. Perlu diperhatikan dalam menyimpan file disarankan membuat folder baru, dan proses penyimpanannya **sebanyak 3 kali dengan Nama File Wajib sama**. Untuk melakukan Generate dan simpan ada pada pilihan Program > Generate, Save and Exit. Simpan dalam satu folder yang baru dibuat dan Nama yang sama.





8. Berikut adalah Tampilan File yang sudah digenerate, save and exit, untuk kebutuhan Delay waktu perlu kita tambahkan Header baru yaitu #include <delay.h> dengan satuan waktu MS atau millisecond.

The screenshot shows a CodeVision IDE window with the title bar "D:\1-Modul Mikro\Filter\5.Start CodeVision, How to make New File\Latihan\coba.c". The main area displays a C code file with the following content:

```
14
15
16 Chip type : ATmega8
17 Program type : Application
18 AVR Core Clock frequency: 8,000000 MHz
19 Memory model : Small
20 External RAM size : 0
21 Data Stack size : 256
22 ****
23
24 #include <mega8.h>
25 #include <delay.h>
26
27 // Declare your global variables here
28
29 void main(void)
30 {
31 // Declare your local variables here
32
33 // Input/Output Ports initialization
34 // Port B initialization
35 // Func7=In Func6=In Func5=In Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
36 // State7=T State6=T State5=T State4=0 State3=0 State2=0 State1=0 State0=0
37 PORTB=0x00;
38 DDRB=0x1F;
39
```

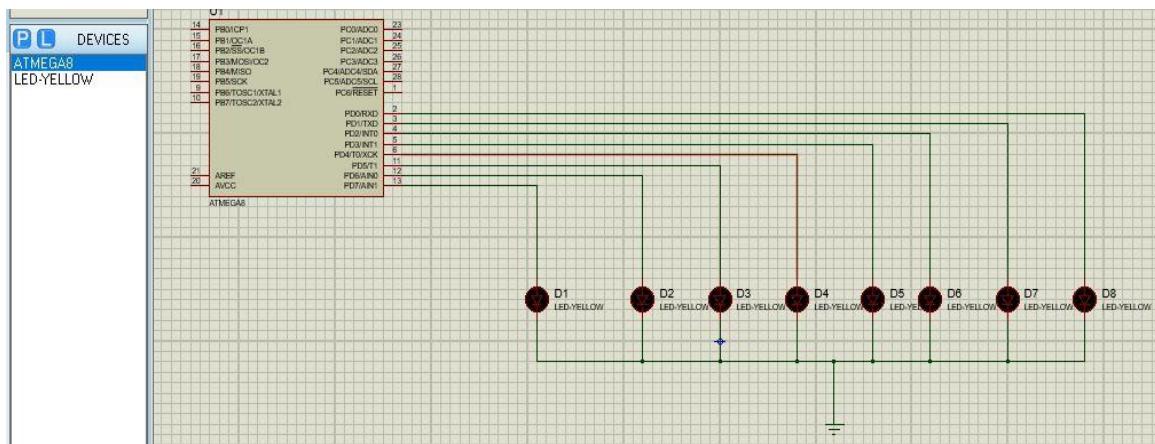
BAB VI

PENGGUNAAN CHIP ATMEGA DENGAN CODEVISION

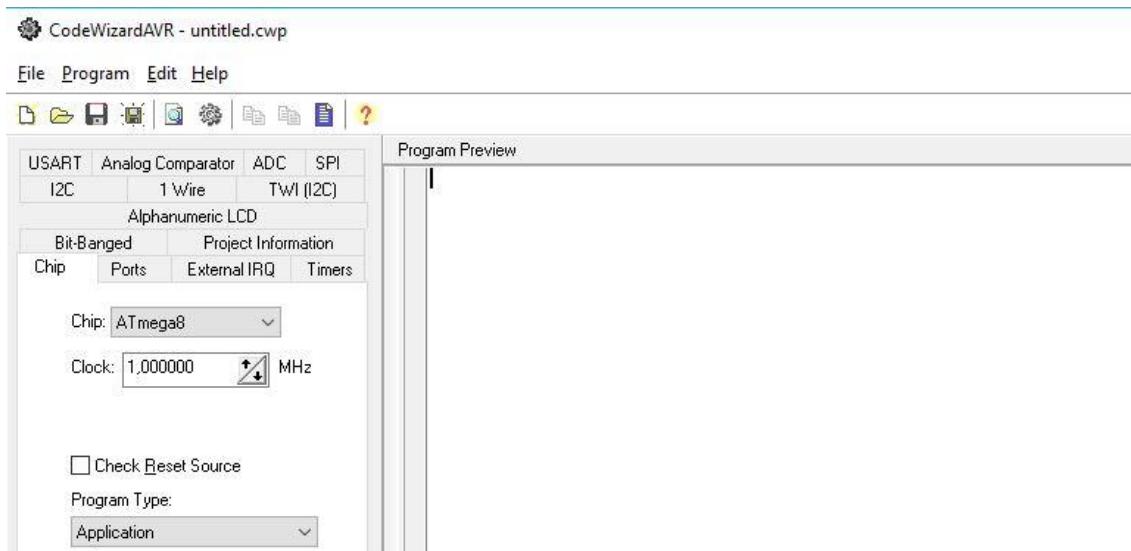
A. RUNNING LED

Disini kita akan mencoba membuat Lampu Berjalan dengan menggunakan Chip ATmega8 yang mana untuk memfungsikannya kita menggunakan perintah yang kita masukkan kedalam CodeVision, ikuti langkah-langkah berikut untuk membuat Lampu Berjalan :

1. Buat Rangkaian terlebih dahulu di Proteus dengan menggunakan Komponen ATmega8, LED-Yellow, Ground, yang mana Lampu dihubungkan ke Port D di ATmega8 seperti dibawah ini.

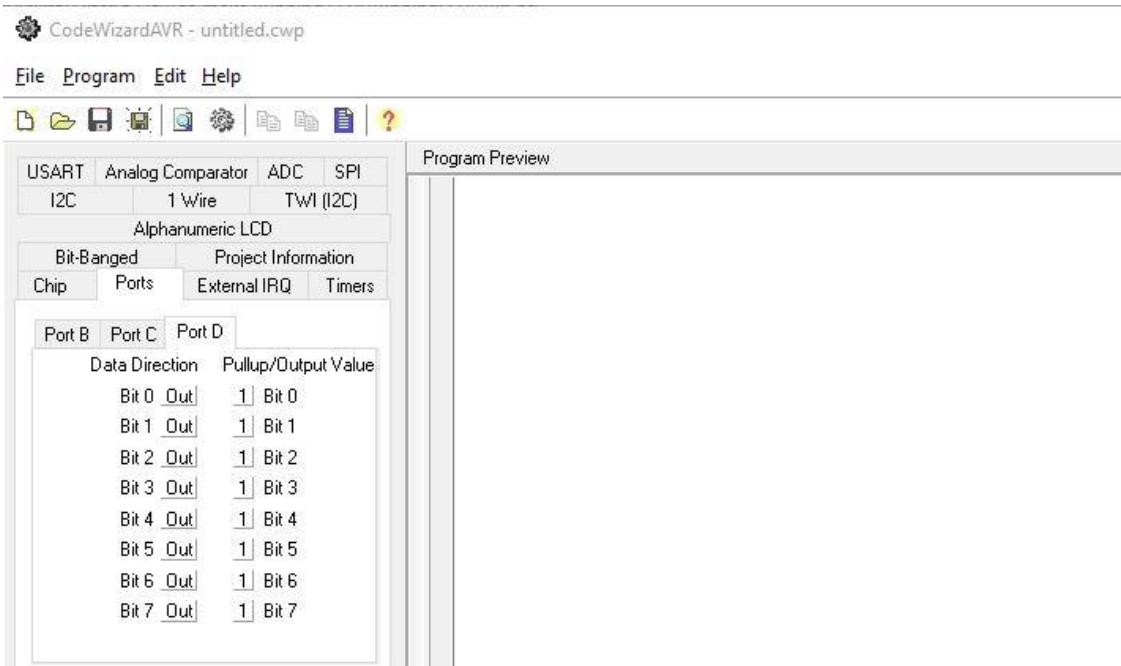


2. Setelah rangkaian selesai dibuat, sekarang buat file baru di CodeVision sama seperti yang dijelaskan pada bab sebelumnya, sampai pada Tahap seperti dibawah ini.

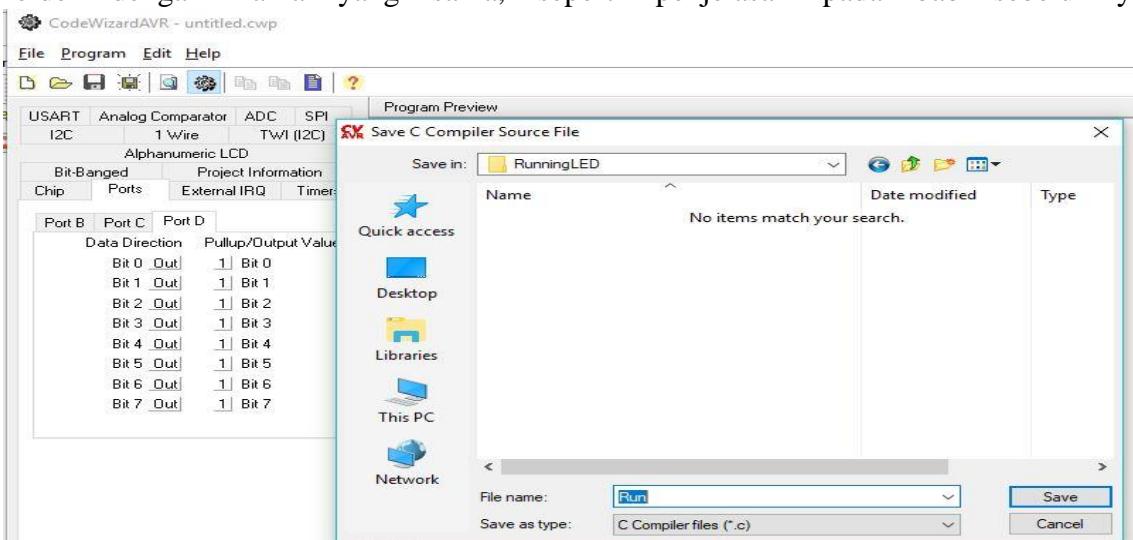


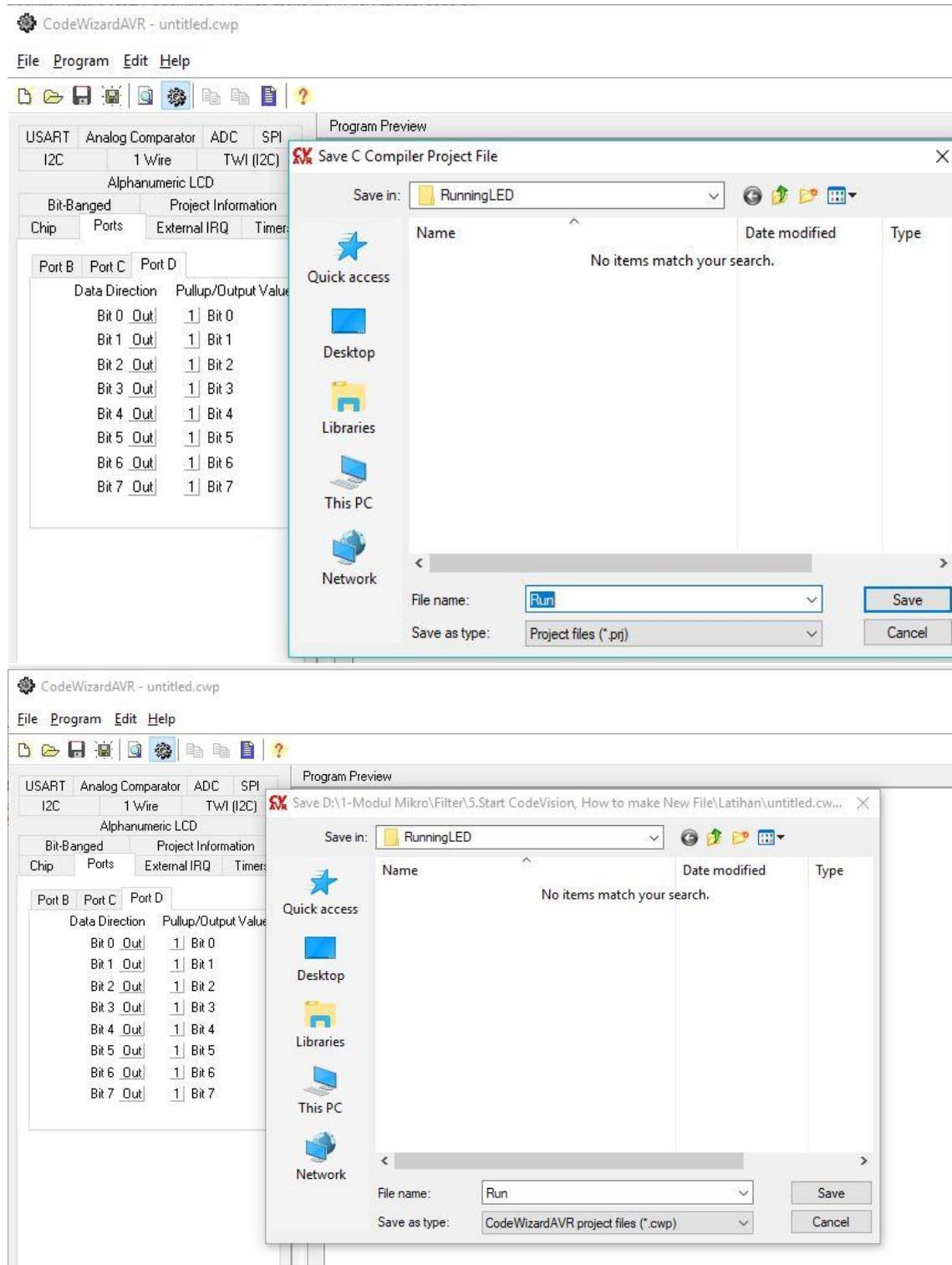
Pada Tab Chip kita gunakan ATmega8 sesuai dengan rangkaian proteus kita, dengan Clock 1.000000 MHz.

- Setelah itu pada Tab Ports, kita pilih Tab Port D dengan Data Direction Bit “OUT” dan Pullup/Output Value adalah “1” artinya kita akan mengeluarkan Perintah Menyala dari ATmega8.



- Setelah itu lakukan penyimpanan (Generate, Save and Exit) Sebanyak 3 kali dalam satu folder dengan nama yang sama, seperti penjelasan pada bab sebelumnya.





5. Setelah disimpan, maka masukkan header #include <delay.h>.

```

15
16 Chip type : ATmega8
17 Program type : Application
18 AVR Core Clock frequency: 1,000000 MHz
19 Memory model : Small
20 External RAM size : 0
21 Data Stack size : 256
22 ****
23
24 #include <mega8.h>
25 #include <delay.h>
26 // Declare your global variables here

```

- Setelah itu kita pastikan pada bagian Port yang kita gunakan (Port D) sudah dalam kondisi ON/Menyala dalam bilangan Hexadecimal, Tarik Kebawah pada bagian Port D, jika dalam kondisi semua menyala maka Perintah tersebut berisikan PORTD = 0xFF; dan DDRD=0xFF; jika belum maka bisa diganti/tulis secara manual seperti ini.

```

33 // Port B initialization
34 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
35 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
36 PORTB=0x00;
37 DDRB=0x00;
38
39 // Port C initialization
40 // Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
41 // State6=T State5=T State4=T State3=T State2=T State1=T State0=T
42 PORTC=0x00;
43 DDRC=0x00;
44
45 // Port D initialization
46 // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=O
47 // State7=1 State6=1 State5=1 State4=1 State3=1 State2=1 State1=1 State0=1
48 PORTD=0xFF;
49 DDRD=0xFF;
50
51 // Timer/Counter 0 initialization
52 // Clock source: System Clock
53 // Clock value: Timer 0 Stopped
54 TCCR0=0x00;
55 TCNT0=0x00;
56

```

- Selanjutnya Scrool kepalng bawah pada bagian perulangan While, didalam perulangan tersebut terdapat kalimat “//Place your code here”, Sintaks perintah bisa kita masukkan dibawah tulisan tersebut.

```

114 |     SPCR=0x00;
115 |
116 |     // TWI initialization
117 |     // TWI disabled
118 |     TWCR=0x00;
119 |
120 |     while (1)
121 |     {
122 |         // Place your code here
123 |         PORTD=0b10000000;
124 |         delay_ms(500);
125 |         PORTD=0b01000000;
126 |         delay_ms(500);
127 |         PORTD=0b00100000;
128 |         delay_ms(500);
129 |         PORTD=0b00010000;
130 |         delay_ms(500);
131 |         PORTD=0b00001000;
132 |         delay_ms(500);
133 |         PORTD=0b00000100;
134 |         delay_ms(500);
135 |         PORTD=0b00000010;
136 |         delay_ms(500);
137 |         PORTD=0b00000001;
138 |         delay_ms(500);
139 |
140 |

```

```

PORTD=0b10000000;
delay_ms(500);
PORTD=0b01000000;
delay_ms(500);
PORTD=0b00100000;
delay_ms(500);
PORTD=0b00010000;
delay_ms(500);
PORTD=0b00001000;
delay_ms(500);
PORTD=0b00000100;
delay_ms(500);
PORTD=0b00000010;
delay_ms(500);
PORTD=0b00000001;
delay_ms(500);

```

Berikut adalah penjelasan perintah yang kita gunakan diatas :

a. **PORTD=0b10000000;**

PORTD adalah Tempat Port yang kita hubungkan ke Lampu.

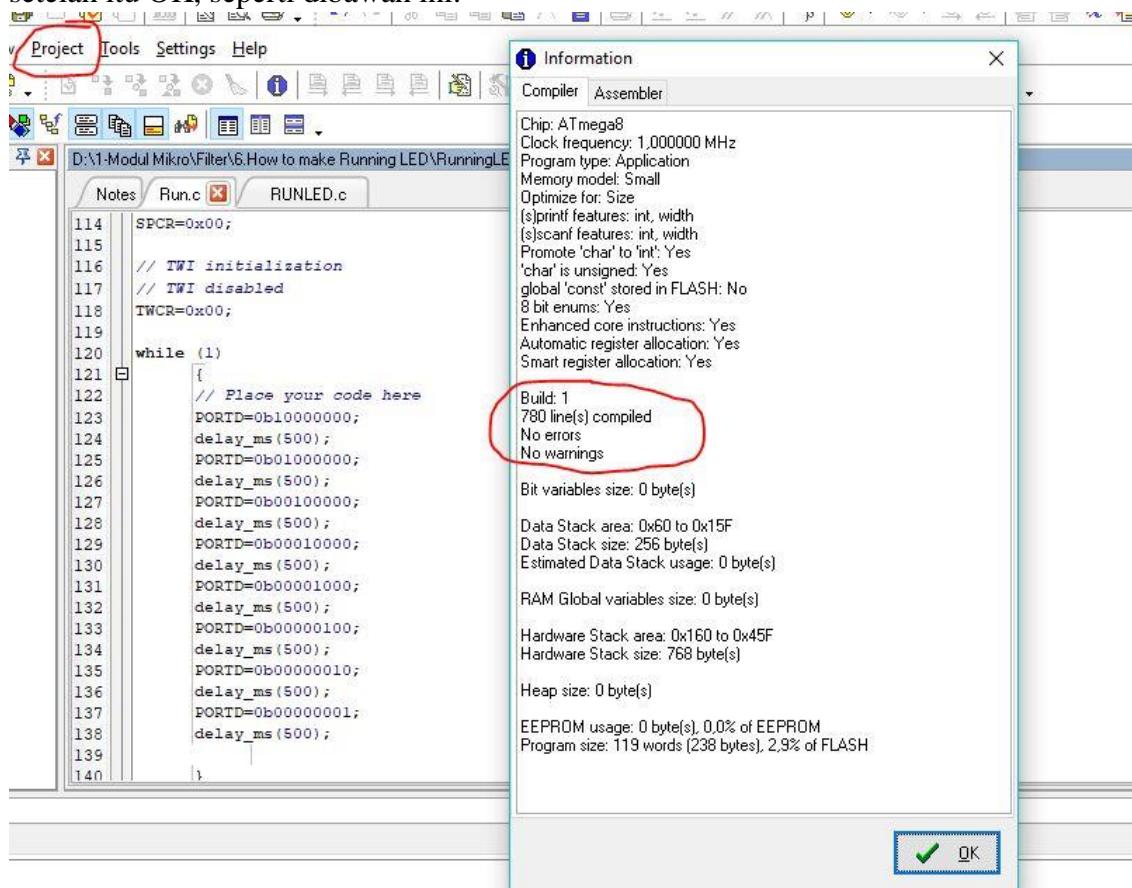
0b Artinya kita akan menggunakan perintah Bilangan Binary.

10000000 adalah Angka Binary yang kita gunakan ada 8 Digit angka, adapun angka 1 ditempatkan didepan artinya Lampu Pertama saja yang menyala, 7 lampu lainnya mati. Misalkan **00000100** Artinya Lampu ke 6 yang menyala, lampu yang lainnya keadaan mati.

b. **delay_ms(500);**

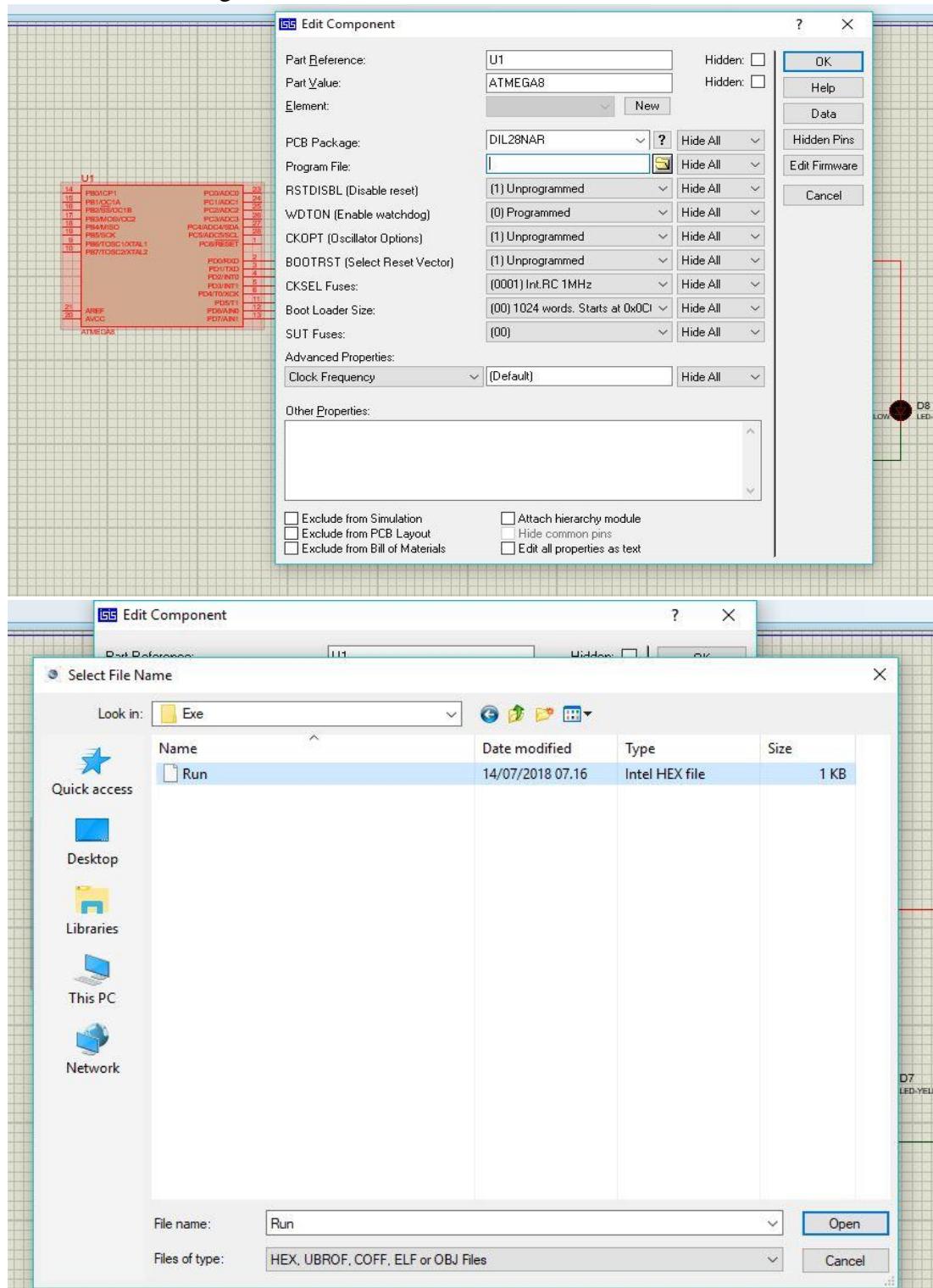
delay_ms(Satuan_Waktu_Dalam_Millisecond) adalah berapa lama jeda lampu menyala antar satu lampu ke lampu yang lain. Tidak selalu kita atur 500, bisa saja 1000 atau lebih tergantung keinginan berapa lama mengatur Waktunya.

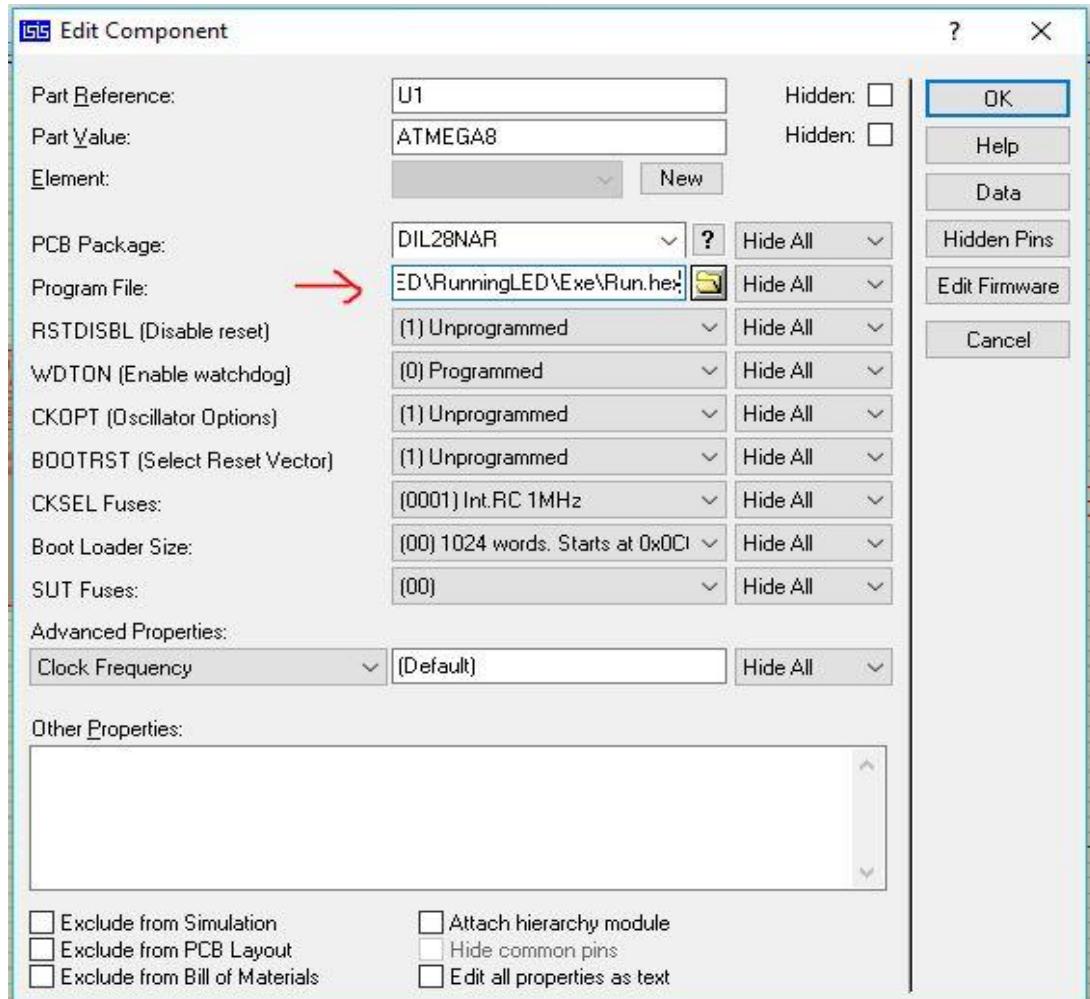
- Setelah selesai kita masukkan Sintaks perintahnya, selanjutnya perintah tersebut kita Build All agar menjadi File format HEX/Intel HEX file, dengan cara pilih Project pada Toolbar > Build All, atau tekan Ctrl + F9, dan pastikan tidak ada Pesan Error setelah di Build, setelah itu OK, seperti dibawah ini.



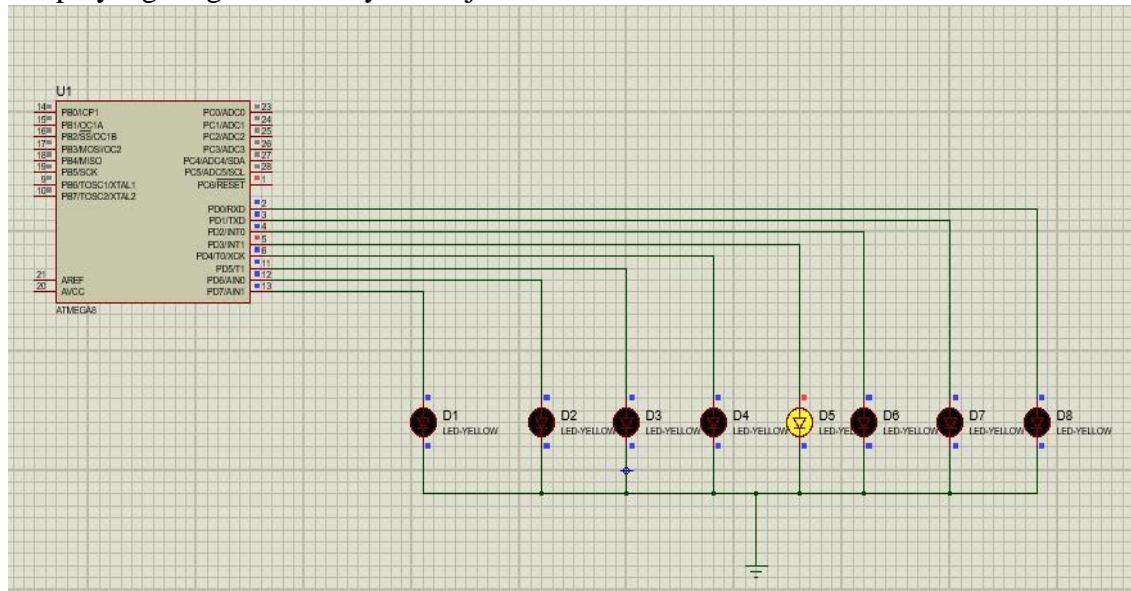
- Setelah Berhasil di Build All, tinggal kita masukkan file tersebut kedalam Chip ATmega8 yang ada pada rangkaian diproteus. Untuk memasukkannya buka rangkaian diproteus kita > Klik kiri 2 kali pada ATmega8 > akan muncul jendela baru, pada bagian Program File kita Browse Folder > Masukkan File HEX nya, Letaknya ada pada (**Folder Simpan File**

CodeVision – Folder Exe – NamaFile.hex/Type : Intel HEX file) > Open dan OK > Setelah itu Run Program.





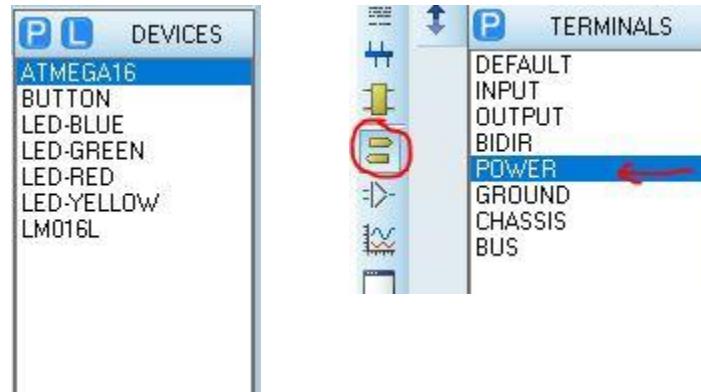
Jika tidak ada pesan Error maka setelah kita Run Rangkaian tersebut akan menghasilkan lampu yang bergantian menyala/berjalan.



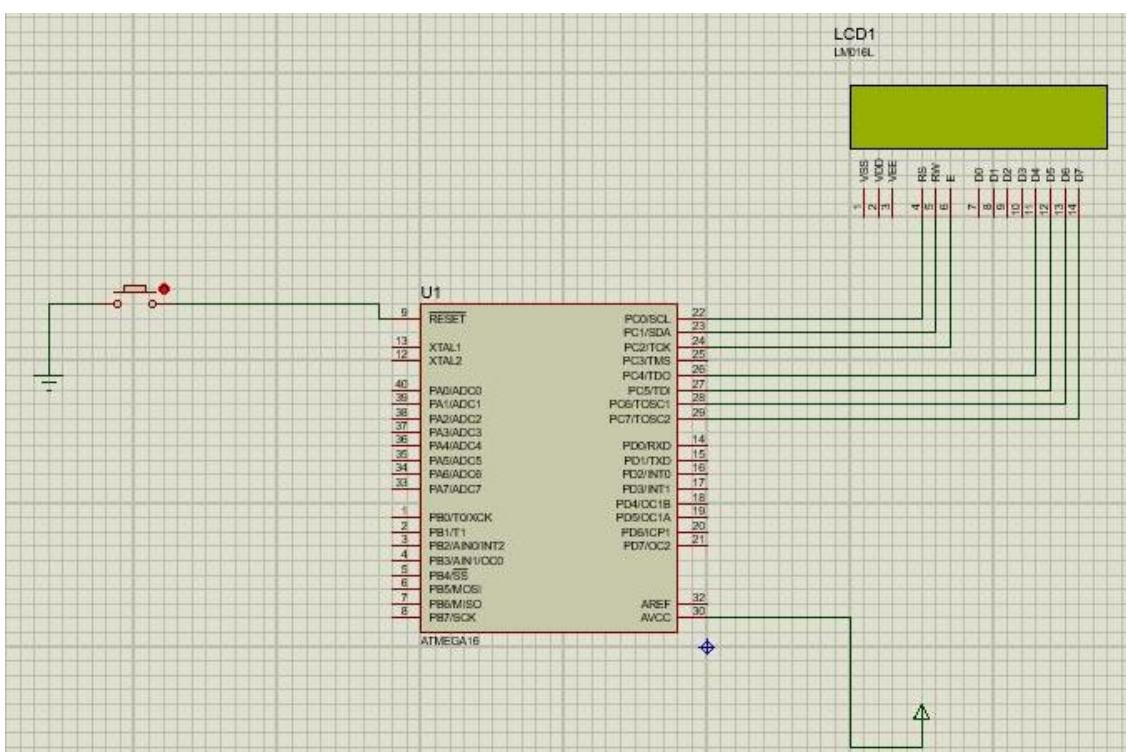
B. TEXT IN LCD

Teks di LCD biasanya kita temui dipusat-pusat perbelanjaan ataupun pada saat kita kekasir pada saat ingin membayar belanjaan kita yang digunakan untuk menampilkan total belanja kepada kita. Adapun tahapan pembuatannya seperti dibawah ini :

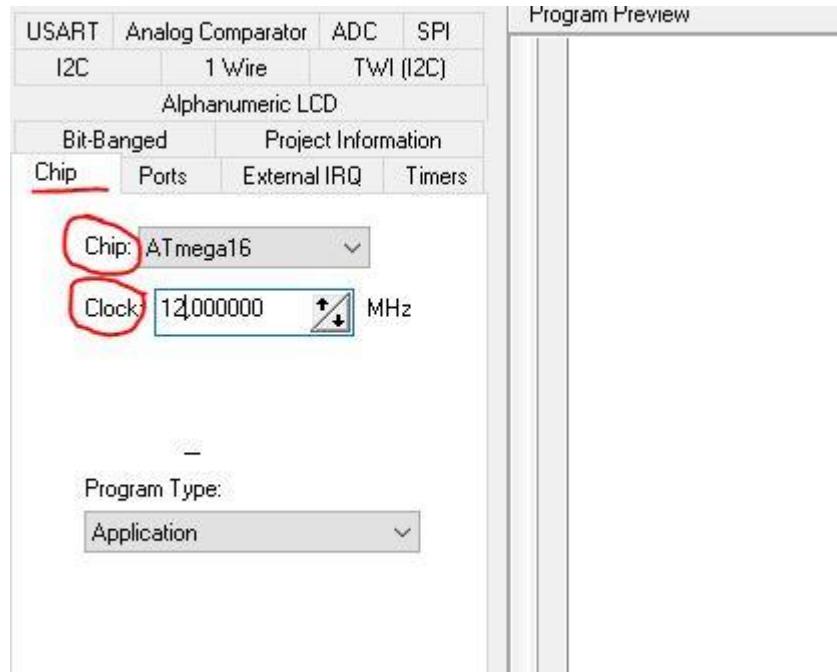
1. Komponen yang diperlukan untuk membuat rangkaianya adalah ATmega16, Button, LM016L, catatan kita tidak menggunakan Ground lagi, tetapi menggunakan Power.



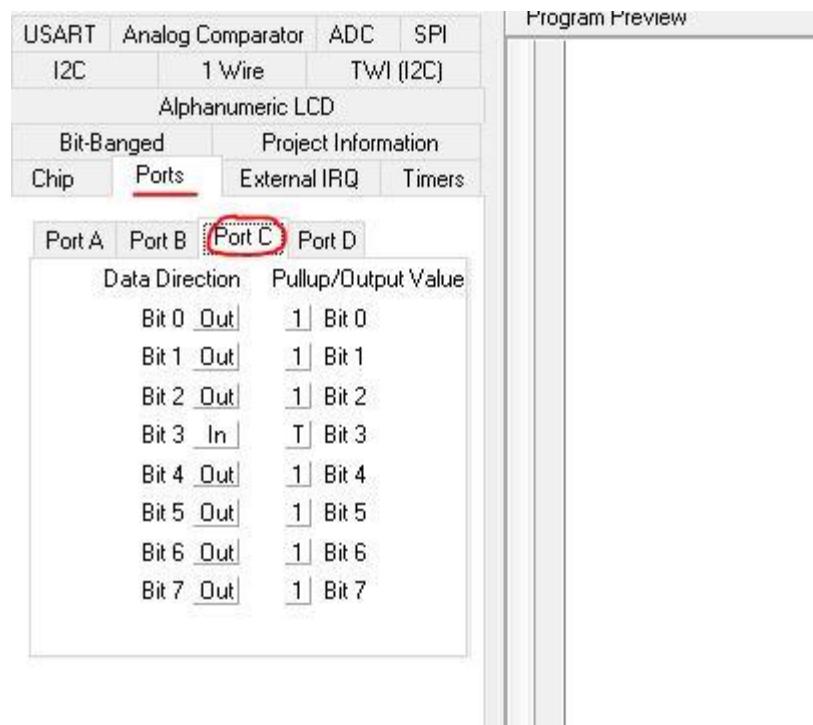
2. Setelah Komponennya dicari dan dimasukkan, buat rangkaian yang terhubung dengan PORT C di ATmega16, catatan untuk Sambungan PC3 tidak dipakai, seperti dibawah ini.



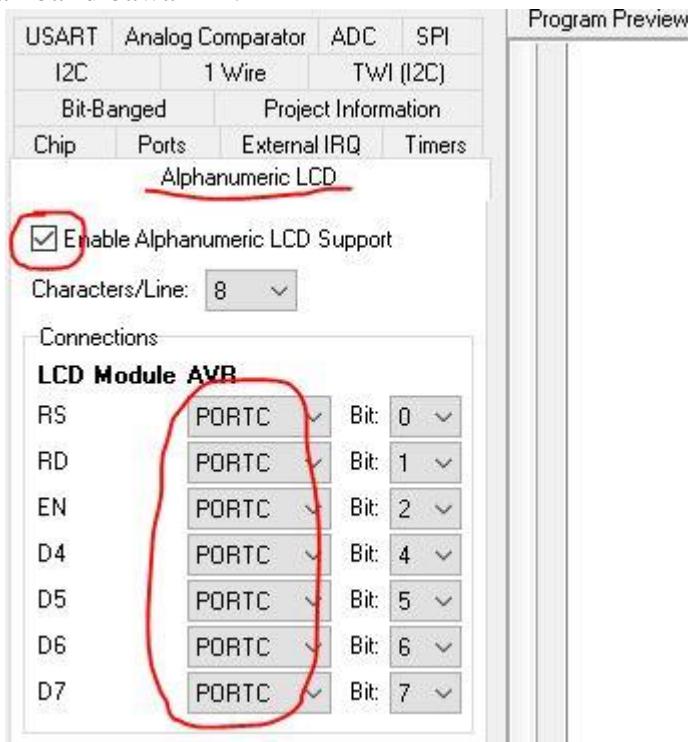
3. Selanjutnya masuk kedalam CodeVision, sama seperti yang sebelumnya dalam pembuatan file baru, yang membedakannya hanya pada saat setting komponen yang digunakan saja. Pada Tab Chip, kita atur menggunakan ATmega16 dengan Clock 12.000000 MHz.



4. Pada Tab Ports, kita pilih PORT C dengan Data Direction OUT Semua dan Output Value “1”, kecuali Bit 3 Posisi IN.



5. Selanjutnya beralih menuju Tab Alphanumeric LCD, akan muncul hanya pilihan “Enable Alphanumeric LCD Support” lalu kita centang dulu perintah tersebut. Selanjutnya muncul Characters/Line yang fungsinya adalah mengatur jumlah **maksimal** karakter yang akan ditampilkan di LCD, setelah itu dibagian Connectionnya kita ubah ke PORTC semua dan Bit tetap seperti gambar dibawah ini.



6. Lakukan Generate, Save and Exit seperti biasa yaitu 3 Kali simpan. Setelah itu masuk ke Sintaks tambahkan #include <delay.h> , #include <alcd.h> (jika belum otomatis muncul.

```
*****  
|  
| #include <megal6.h>  
| #include <delay.h>  
| // Alphanumeric LCD Module functions  
| #include <alcd.h> |  
  
| // Declare your global variables here  
  
| void main(void)  
| {  
| // Declare your local variables here
```

7. Kebagian bawah, bagian PORTC kita cek apakah sudah sesuai dengan gambar dibawah ini atau dibiarkan saja apa yang otomatis disetting oleh sistem karena sebenarnya sintaks tersebut sudah kita atur pada saat kita melakukan setting dibagian awal CodeVision.

```
// Port C initialization  
// Func7=Out Func6=Out Func5=Out Func  
// State7=1 State6=1 State5=1 State4=  
PORTC=0xF7;  
DDRC=0xF7;
```

8. Sekarang kebagian paling bawah yaitu tempat kita memasukkan perintah untuk tampilan Outputnya/Teksnya.

```
while (1)  
{  
// Place your code here  
lcd_clear();  
lcd_gotoxy(1,0);  
lcd_putsf("STMIK");  
lcd_gotoxy(1,1);  
lcd_putsf("BJM");  
delay_ms(300);  
lcd_clear();  
lcd_gotoxy(1,0);  
lcd_putsf("MIKRO");  
lcd_gotoxy(1,1);  
lcd_putsf("PROCESS");  
delay_ms(300); |  
}
```

```
// Place your code here  
lcd_clear();  
lcd_gotoxy(1,0);  
lcd_putsf("STMIK");
```

```
lcd_gotoxy(1,1);
lcd_putsf("BJM");
delay_ms(300);
lcd_clear();
lcd_gotoxy(1,0);
lcd_putsf("MIKRO");
lcd_gotoxy(1,1);
lcd_putsf("PROCESS");
delay_ms(300);
```

Berikut adalah penjelasan perintah yang kita gunakan diatas :

a. **lcd_clear()**

Berfungsi untuk menghapus/mengosongkan layar lcd.

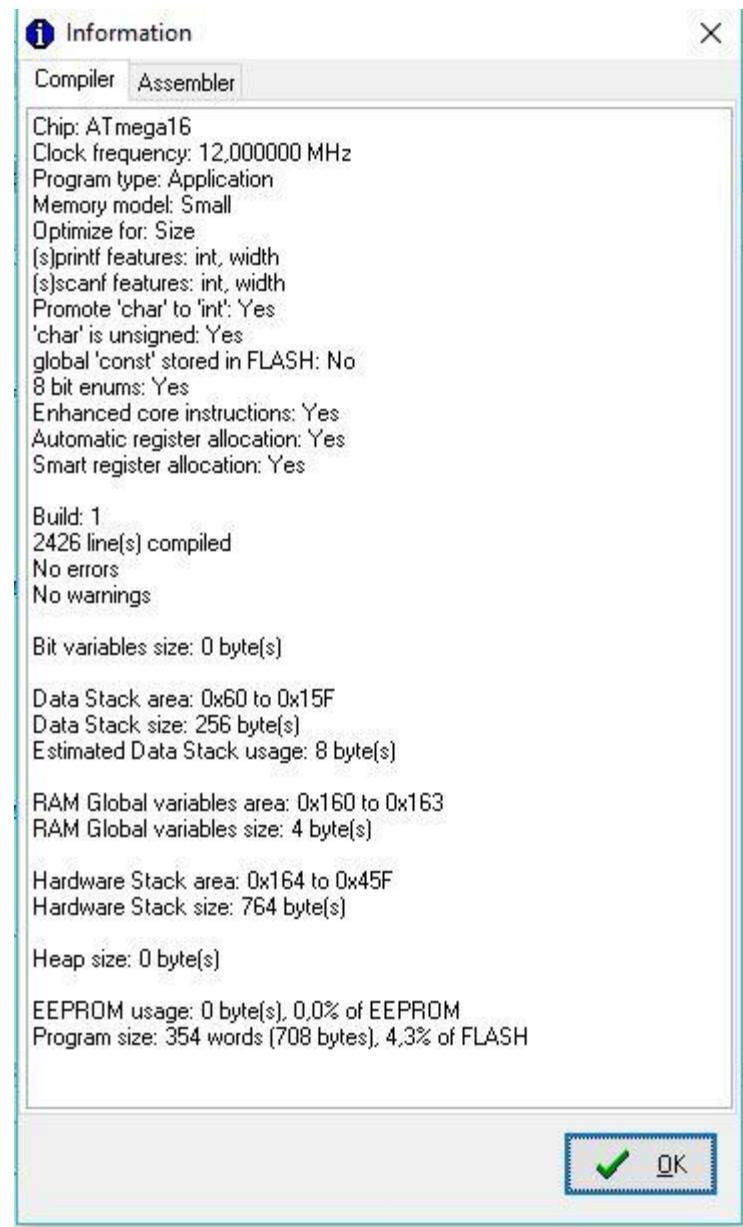
b. **lcd_gotoxy(X,Y)**

Berfungsi untuk pengaturan posisi karakter pertama/spasi/baris, bisa diambil paham (X = SPASI, Y=BARIS).

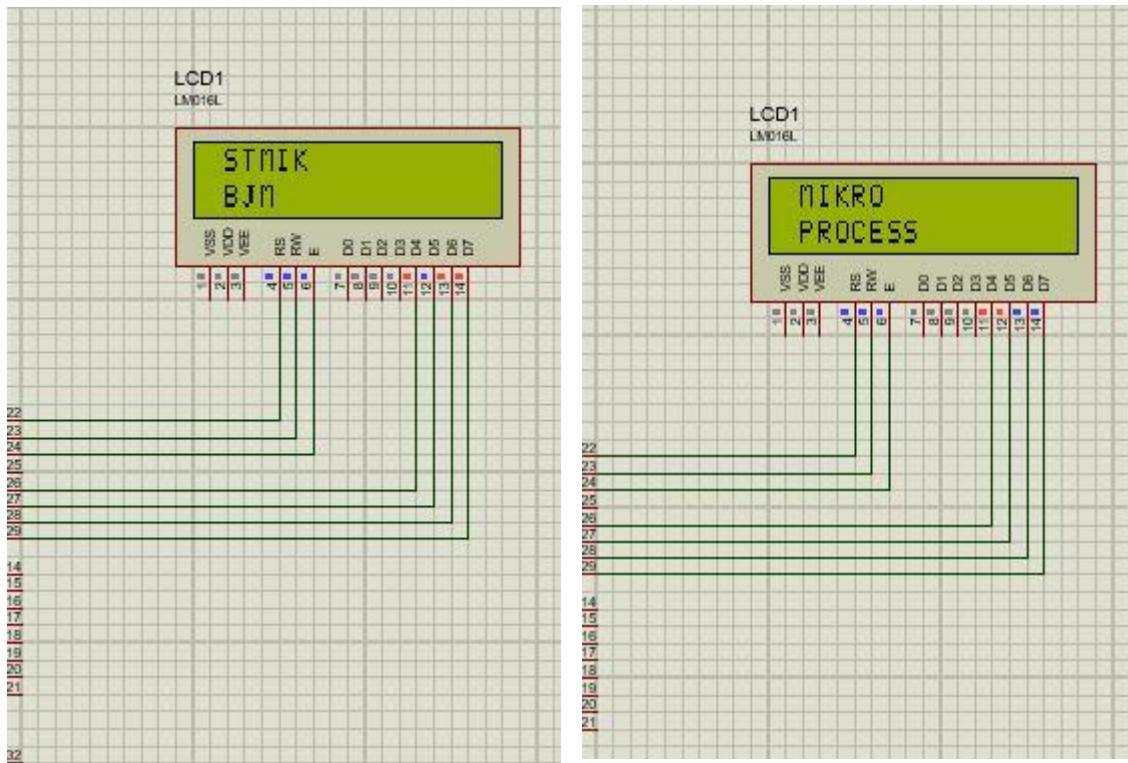
c. **lcd_putsf("Teks")**

Berfungsi untuk memasukkan Teks yang ingin ditampilkan, ingat jumlah karakter yang ditampilkan hanya bisa sesuai nilai maksimal yang kita atur di Alphanumeric LCD sebelumnya saja.

9. Setelah itu lakukan Build All dan pastikan tidak ada error dalam pesan yang muncul.



10. Setelah berhasil dibuild all, masukkan file.hex yang kita buat tadi kedalam rangkaian yang ada diproteus dengan cara yang sama seperti pada contoh sebelumnya (Running LED). Hasilnya seperti dibawah ini, silahkan dicoba tambahkan/ganti tulisan yang ditampilkan.

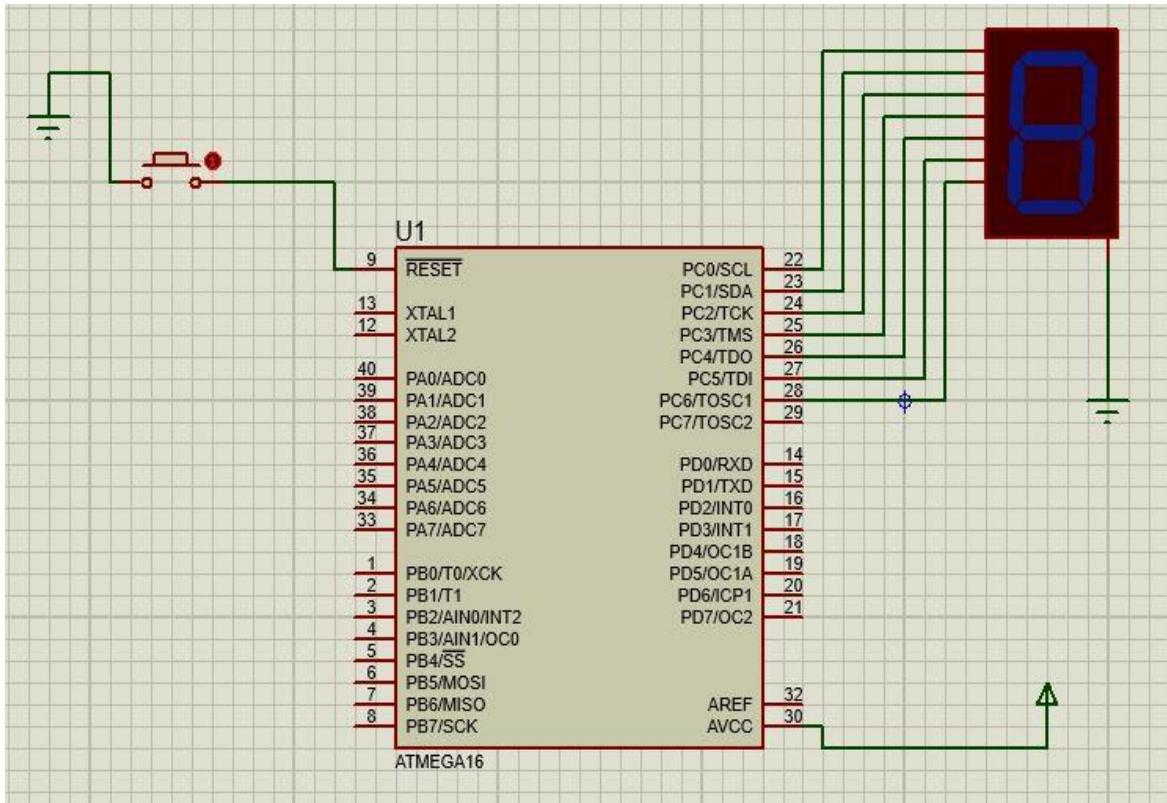


C. 7 SEGMENT

7 Segment atau Seven Segment adalah sebuah papan elektronika yang umumnya digunakan untuk menampilkan angka dari 0 – 9, didalam keseharian sering kita temui pada waktu mundur dilampu lalulintas atau pada papan score pertandingan. Dalam pembuatan Code Visionnya nanti hampir mirip dengan Setting Running LED. Berikut adalah tahapan-tahapan pembuatannya.

1. Komponen yang diperlukan untuk membuat rangkaianya adalah 7SEG-COM-CAT-BLUE, ATmega16, Button, Power. Setelah itu buat rangkaianya seperti dibawah ini.





2. Setelah itu buat File.hex di codevision seperti biasa, dengan ATmega16 Clock 12.000000, Port yang digunakan adalah PORTC dengan Data Direction “OUT” semua dan Output Value “1” semua.

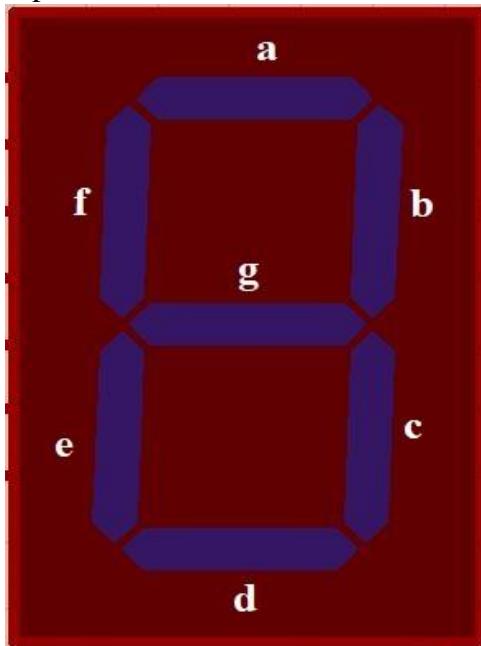
USART	Analog Comparator	ADC	SPI
I2C	1 Wire	TWI (I2C)	
Alphanumeric LCD			
Bit-Banged		Project Information	
Chip	Ports	External IRQ	Timers
Chip: ATmega16 Clock: 12,000000 MHz			
<input type="checkbox"/> Check Reset Source Program Type: Application			

USART	Analog Comparator	ADC	SPI
I2C	1 Wire	TWI (I2C)	
Alphanumeric LCD			
Bit-Banged		Project Information	
Chip	Ports	External IRQ	Timers
Port A Port B Port C Port D Data Direction Pullup/Output Value Bit 0 Out 1 Bit 0 Bit 1 Out 1 Bit 1 Bit 2 Out 1 Bit 2 Bit 3 Out 1 Bit 3 Bit 4 Out 1 Bit 4 Bit 5 Out 1 Bit 5 Bit 6 Out 1 Bit 6 Bit 7 Out 1 Bit 7			

3. Selanjutnya Generate, Save and Exit. Kebagian Sintaks seperti biasa tambahkan header #include <delay.h> dan cek pada dibagian PORTC=0xFF; DDRD=0xFF;

```
// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=1 State6=1 State5=1 State4=1 State3=1 State2=1 State1=1 State0=1
PORTC=0xFF;
DDRD=0xFF;
```

4. Selanjutnya kebagian perulangan, sebelum masuk penulisan perintah harus paham posisi-posisi digit binary yang ingin dinyalakan/membentuk pola pada 7 Segment. Gambarannya seperti dibawah ini.



TABEL POSISI BINARY

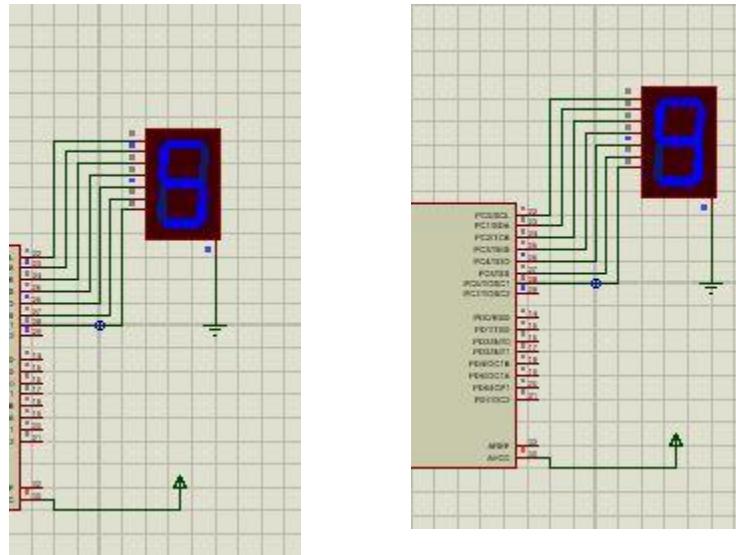
ANGKA	G	F	E	D	C	B	A
0	0	1	1	1	1	1	1
1	0	0	0	0	1	1	0
2	1	0	1	1	0	1	1
3	1	0	0	1	1	1	1
4	1	1	0	0	1	1	0
5	1	1	0	1	1	0	1
6	1	1	1	1	1	0	1
7	0	0	0	0	1	1	1
8	1	1	1	1	1	1	1
9	1	1	0	1	1	1	1

Setelah mengenal posisi-posisi 7 komponen yang terdapat pada Seven Segment, tinggal kita bentuk pola yang kita inginkan, diambil contoh pola angka seperti pada tabel posisi diatas, perintahnya sebagai berikut :

```
// Place your code here
PORTC=0b011111;
delay_ms(100);
PORTC=0b0000110;
delay_ms(100);
PORTC=0b1011011;
delay_ms(100);
PORTC=0b1001111;
delay_ms(100);
PORTC=0b1100110;
delay_ms(100);
PORTC=0b1101101;
delay_ms(100);
PORTC=0b1111101;
delay_ms(100);
PORTC=0b0000111;
delay_ms(100);
PORTC=0b1111111;
delay_ms(100);
PORTC=0b1101111;
delay_ms(100);
```

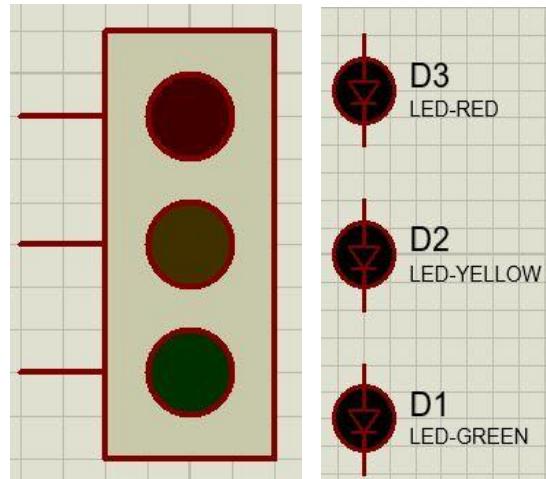
Setelah dimasukkan perintah seperti biasa build all dan tidak ada pesan error.

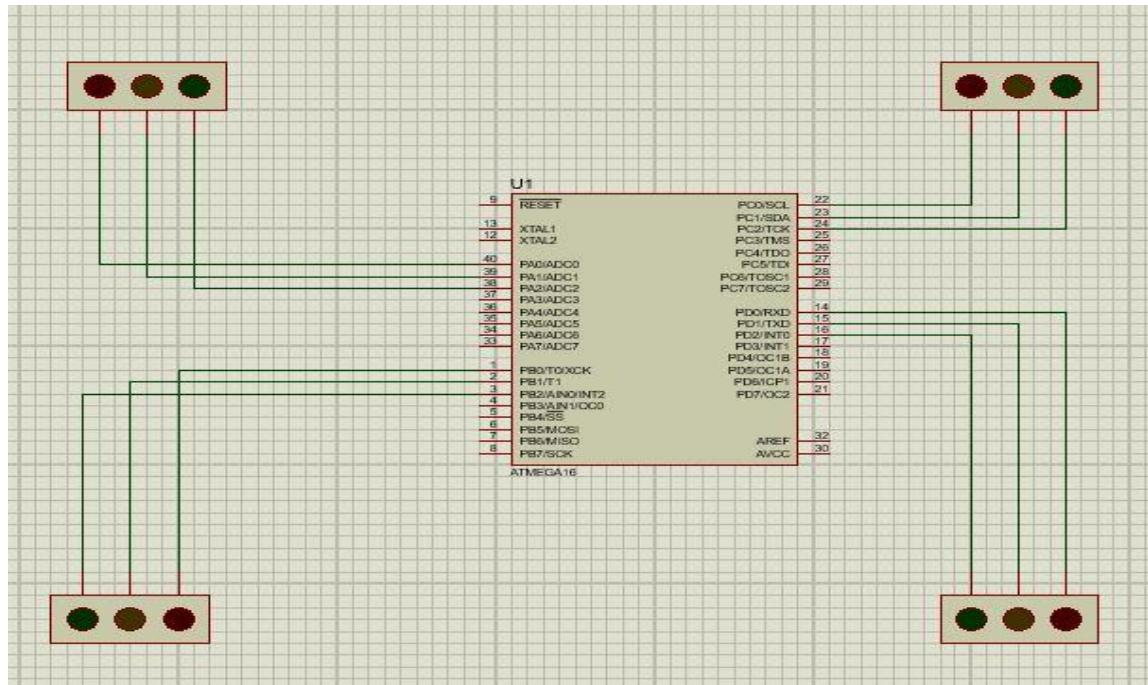
5. Setelah berhasil, File.hex dimasukkan kedalam chip ATmega16 dengan cara yang sama seperti sebelumnya. Jikalau berhasil maka akan muncul tampilan angka 0 – 9 pada papan Seven Segmentnya.



TUGAS AKHIR

- Buatlah sebuah simulasi Trafic Light, bisa menggunakan Lampu Trafic Light atau LED Red, Yellow, Green. Buat simulasi seperti diperempatan jalan seperti yang sering kita temui, dimana lampu hijau secara bergantian menyala dikeempat jalan.





2. Buatlah Rangkaian, Rumus Persamaan dan Tabel Kebenaran Jika Terdapat Input (A,B,C,D) dari Rangkaian Gabungan dibawah ini ?

