

DT-AVR

DT-AVR *Application Note*

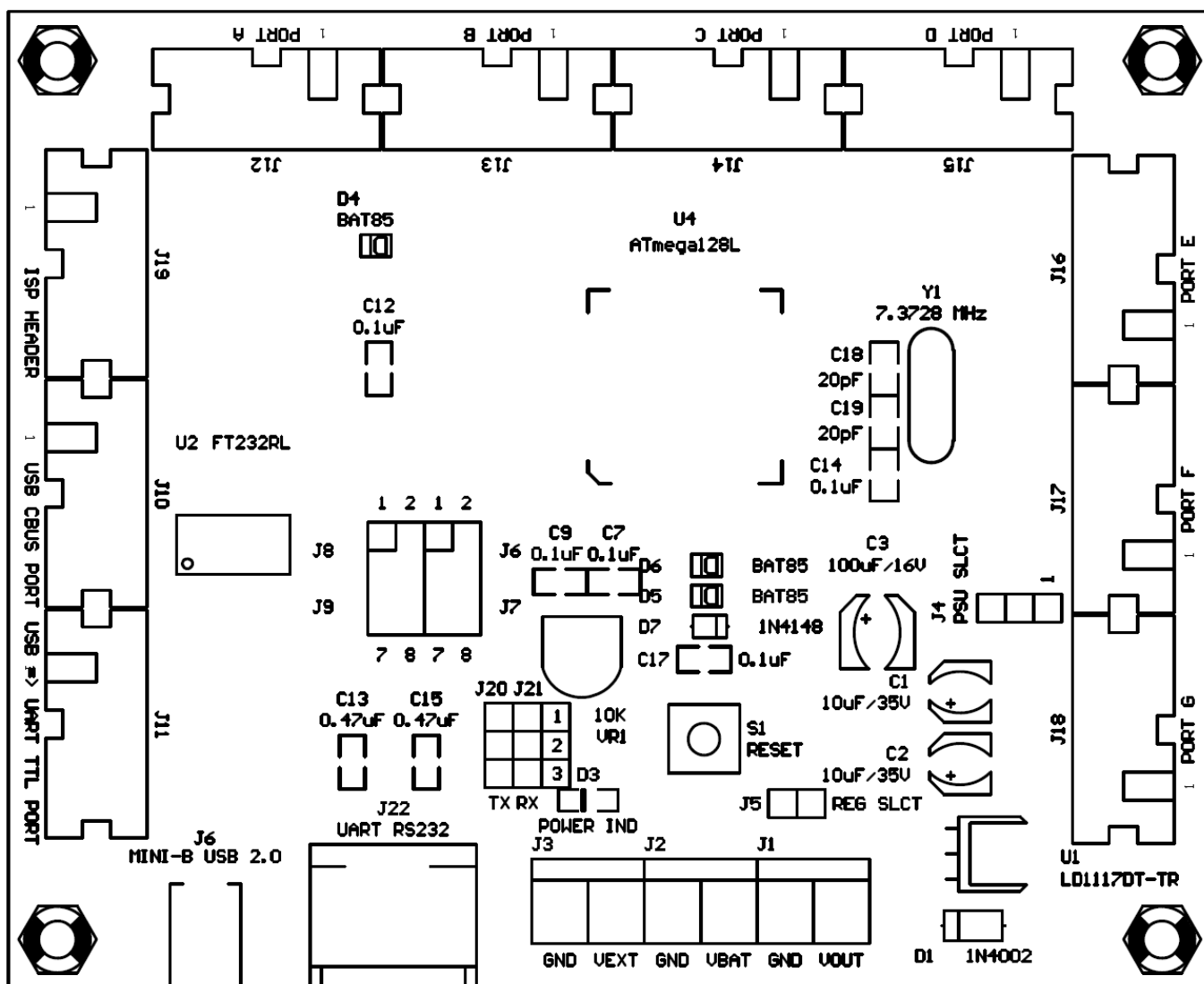
AN167 – How2Use DT-AVR

ATMEGA128L BMS

Oleh : Tim IE

Application Note (AN) ini disusun untuk memberikan penjelasan tentang cara penggunaan DT-AVR ATMEGA128L Bootloader Micro System beserta software pendukungnya.

Tata letak konektor DT-AVR ATMEGA128L BMS adalah sebagai berikut:



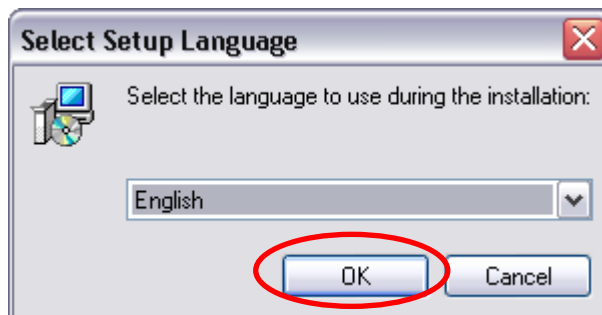
Gambar 1
Tata Letak DT-AVR ATMEGA128L BMS

Persiapan *hardware* DT-AVR ATMEGA128L BMS adalah sebagai berikut:

1. Atur *jumper* J20 dan J21 pada posisi 1-2 agar PE.0 dan PE.1 berfungsi sebagai jalur komunikasi serial.
2. Atur *jumper* J6, J7, J8, dan J9 agar menggunakan jalur USB untuk *bootloader* atau jalur RS-232 untuk *bootloader*.
3. Hubungkan kabel USB ke modul dan PC jika menggunakan jalur USB untuk *bootloader*, atau hubungkan kabel serial ke COM *port* komputer dan konektor RJ45 pada modul jika menggunakan jalur RS232 untuk *bootloader*.
4. Jika terdapat rangkaian atau modul lain yang akan dihubungkan ke DT-AVR ATMEGA128L BMS, disarankan untuk menghubungkan rangkaian tersebut dengan DT-AVR ATMEGA128L BMS terlebih dahulu. Perhatikan koneksi, terutama untuk jalur VCC dan GND jangan sampai terbalik.
5. Atur *jumper* J4 pada posisi 1-2 (sumber tegangan dari VEXT) dan lepas *jumper* pada J5 (5V).
6. Hubungkan catu daya 6 - 12 VDC ke konektor "VEXT" untuk memberi tegangan ke modul.

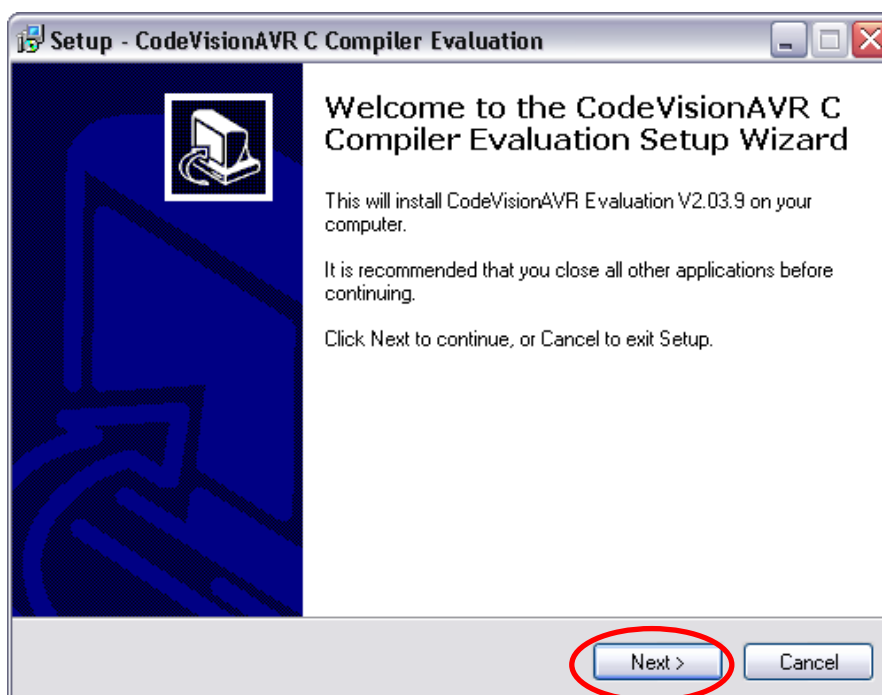
Instalasi CodeVisionAVR evaluasi adalah sebagai berikut:

1. CodeVisionAVR evaluasi terdapat pada CD/DVD program yang disertakan. *File* **setup.exe** ada pada *folder* **CVAVR Evaluation**. Jalankan *file* setup.exe untuk melakukan proses instalasi.
2. Pilih bahasa yang akan digunakan, lalu tekan tombol **OK**.



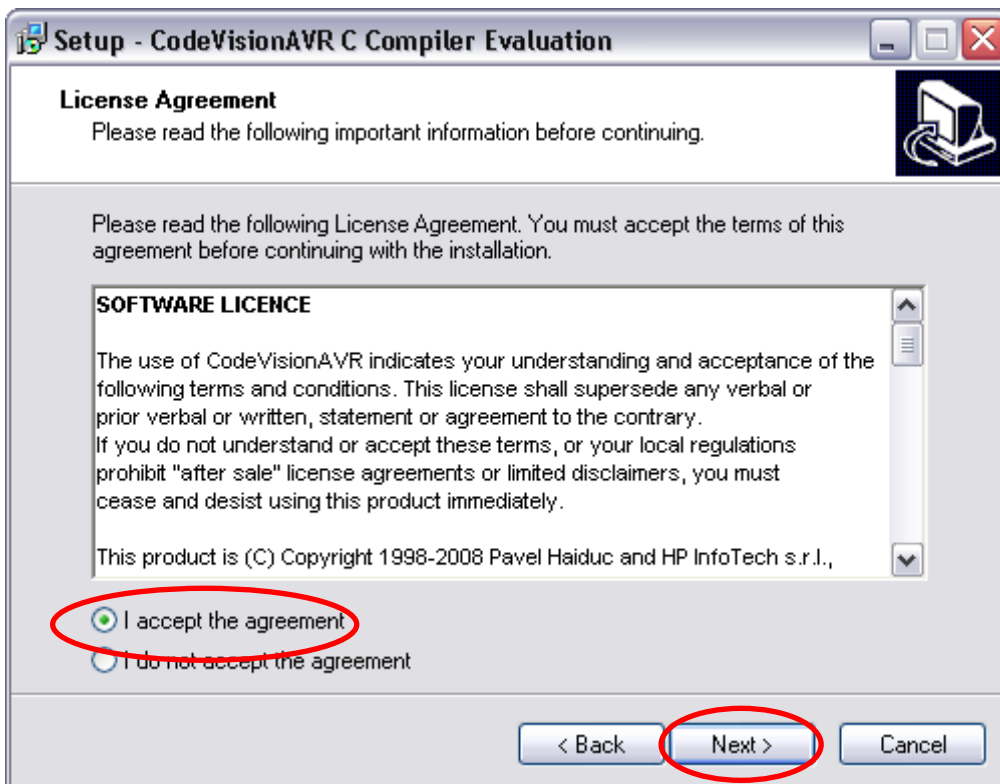
Gambar 2
Pilih Bahasa Instalasi

3. Tampilan awal instalasi CodeVisionAVR. Tekan tombol **Next >** untuk melanjutkan proses instalasi.



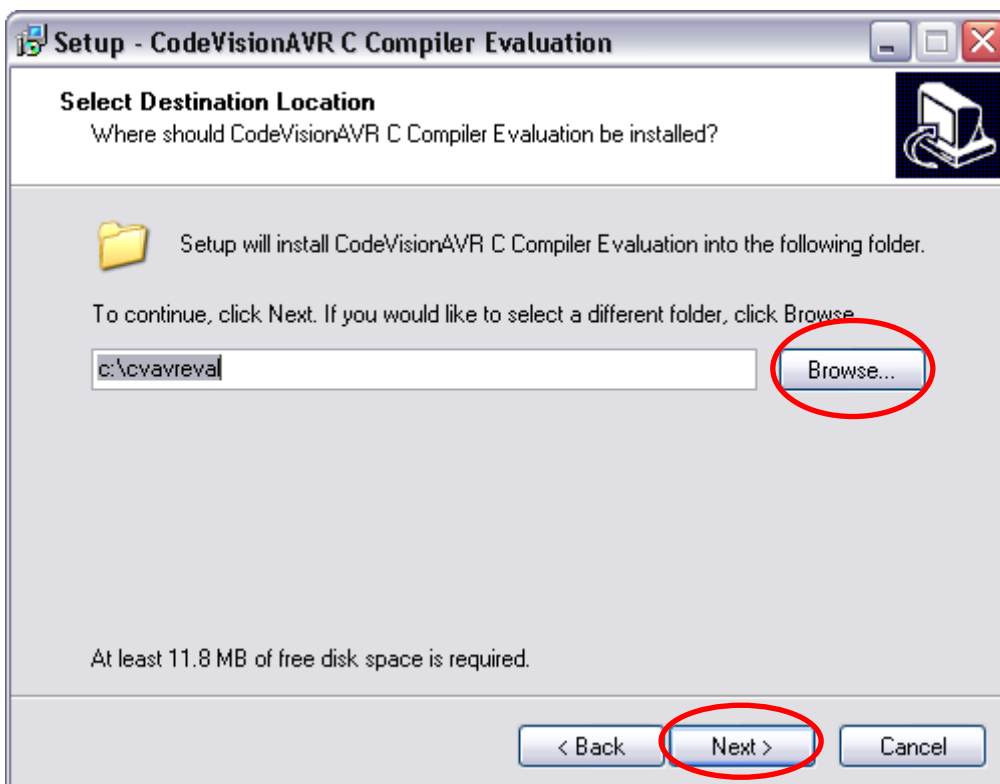
Gambar 3
Awal Instalasi CodeVisionAVR

4. Kemudian masuk pada *License Agreement*. Klik pada "I accept the agreement" lalu tekan tombol **Next >** untuk menyetujui lisensi dan melanjutkan proses instalasi.



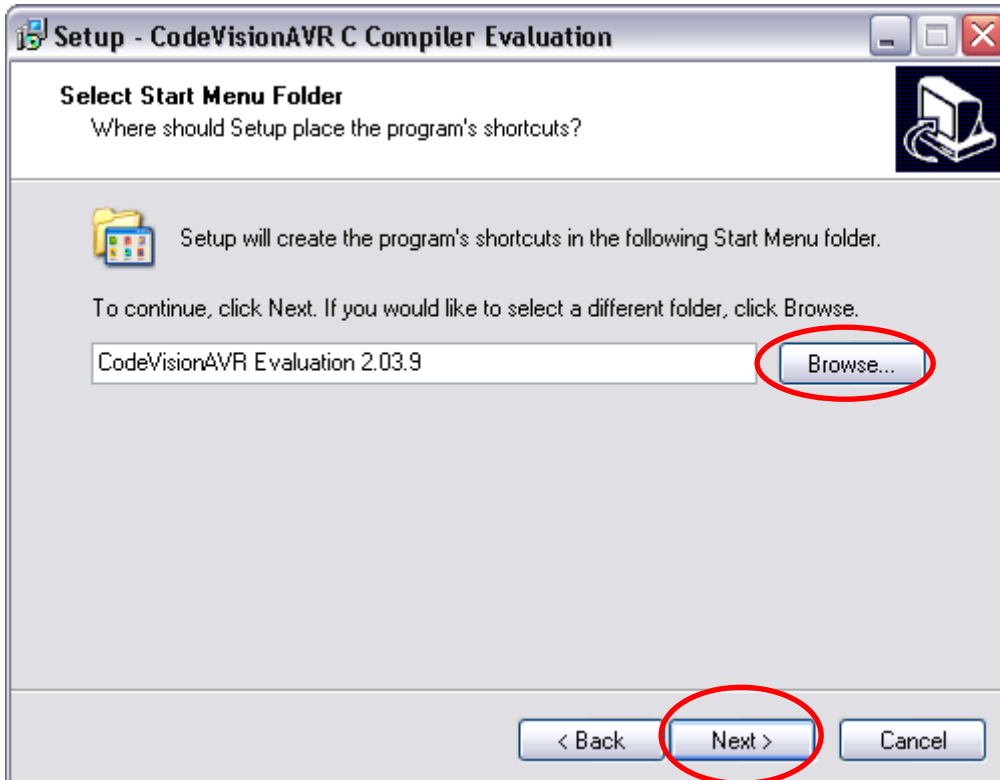
Gambar 4
Persetujuan Lisensi CodeVisionAVR

5. Pilih lokasi instalasi CodeVisionAVR, lalu tekan tombol **Next >**.



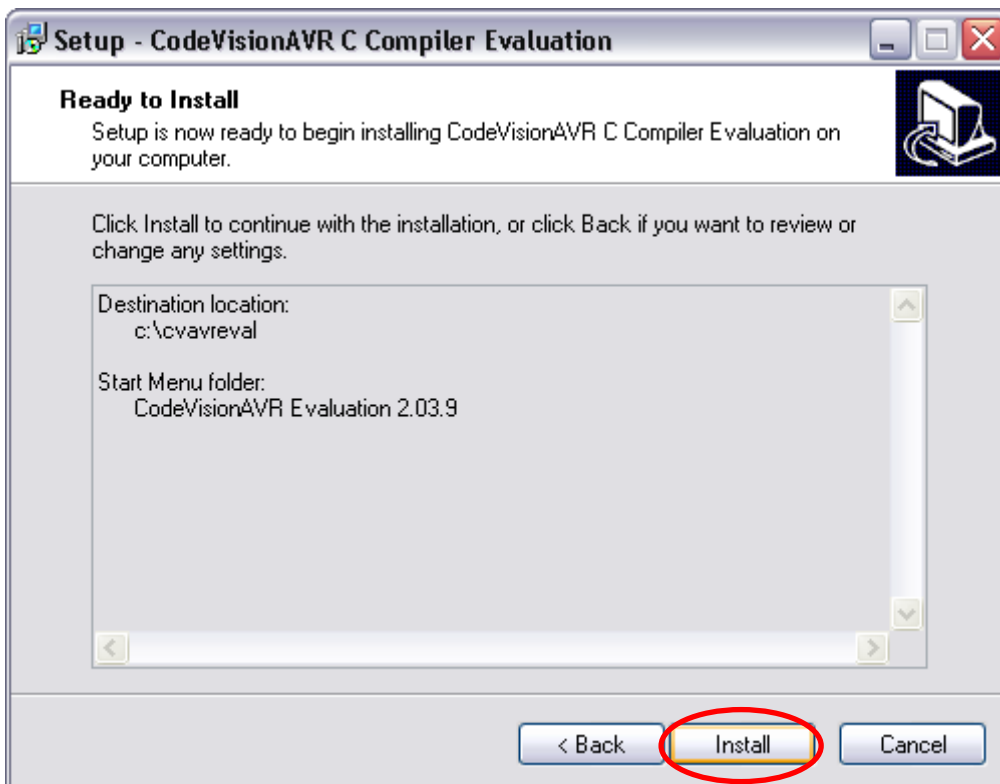
Gambar 5
Pemilihan Lokasi Instalasi

6. Pilih *folder* pada menu Start, lalu tekan tombol **Next >**.



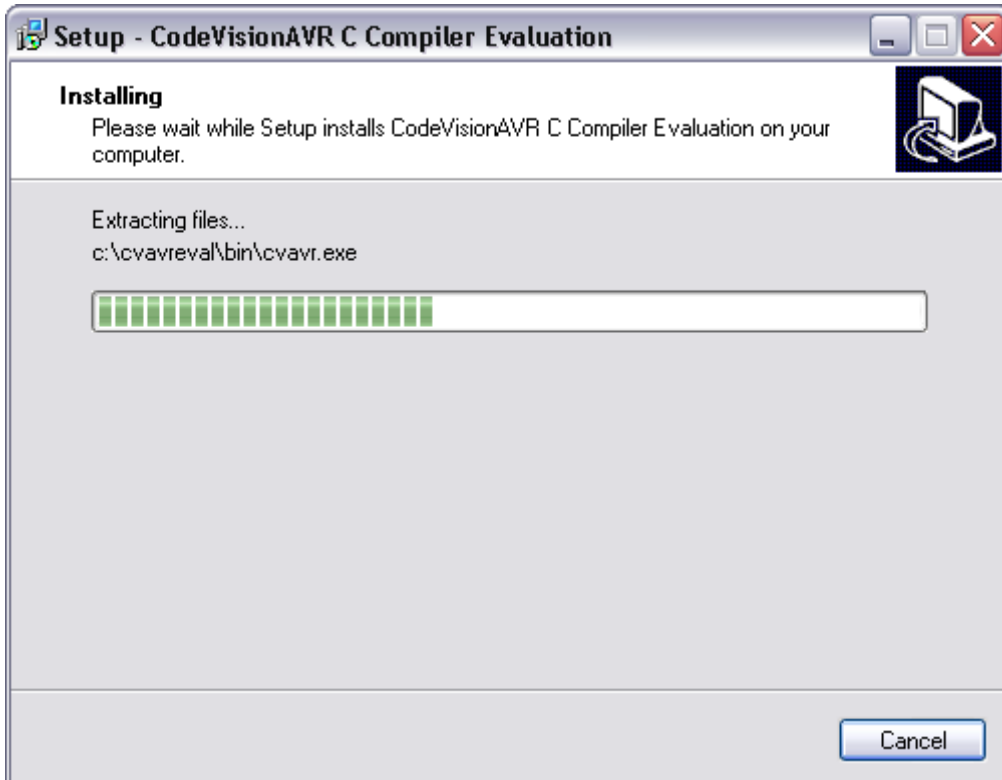
Gambar 6
Tahap Pemilihan Folder Pada Menu Start

7. Persiapan instalasi CodeVisionAVR. Tekan tombol **Install** untuk memulai proses instalasi.



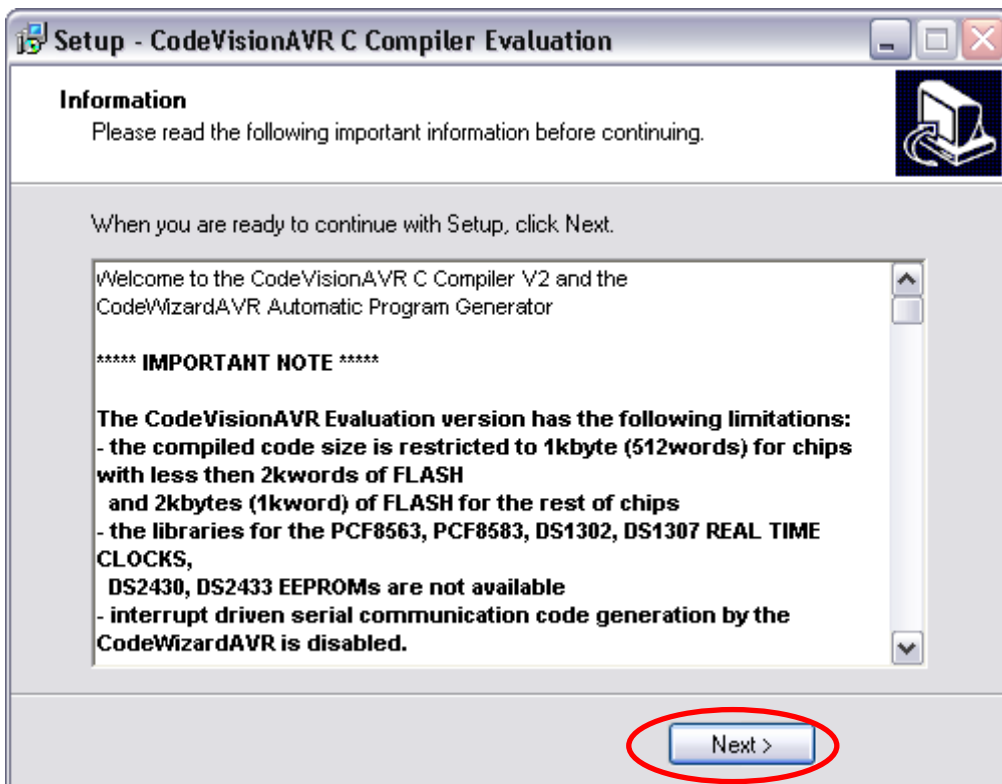
Gambar 7
Tahap Persiapan Instalasi

8. Proses instalasi CodeVisionAVR Evaluation.



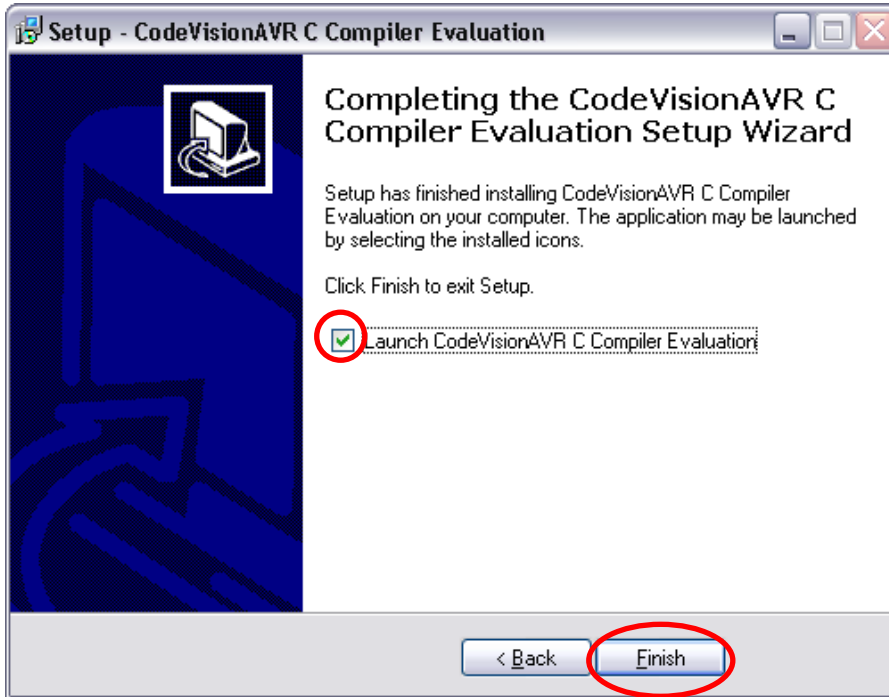
Gambar 8
Proses Instalasi CodeVisionAVR Evaluation

9. Kemudian masuk pada tampilan informasi. Tekan tombol **Next >** untuk melanjutkan proses instalasi.



Gambar 9
Tampilan Informasi

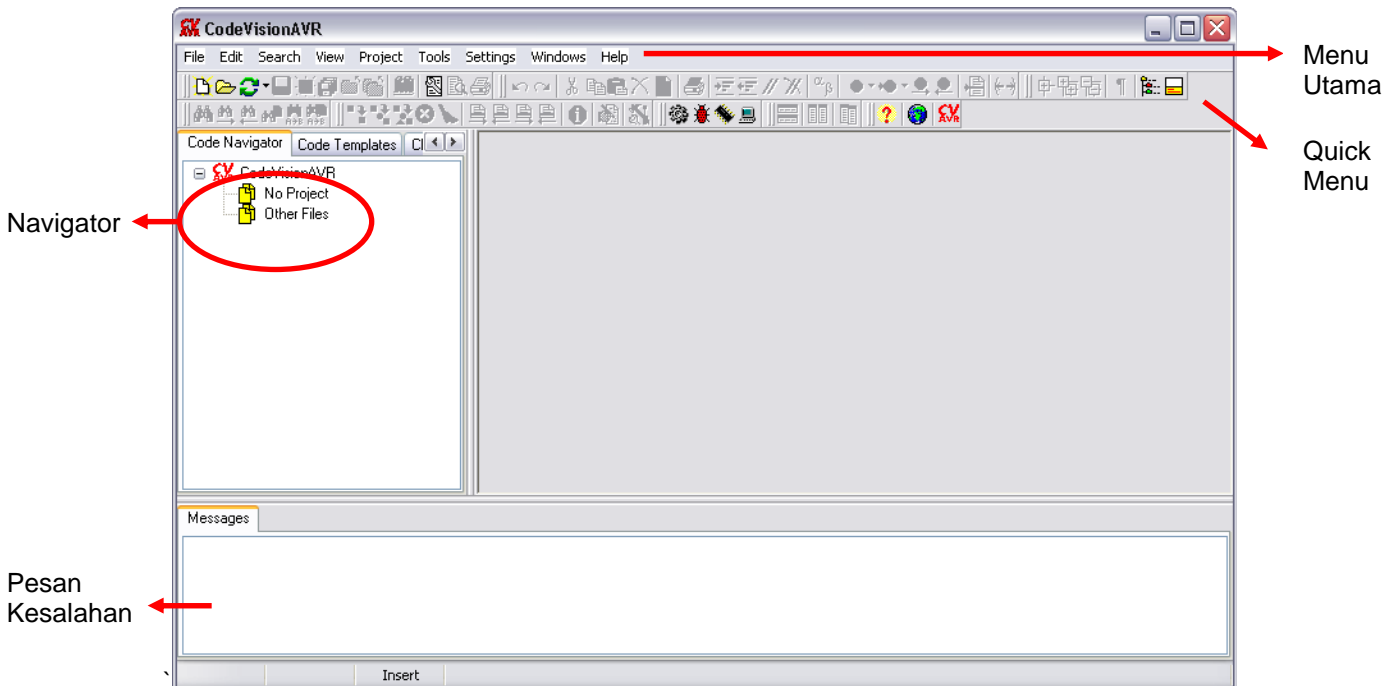
10. Proses instalasi selesai, centang "Launch CodeVisionAVR C Compiler Evaluation" untuk membuka CodeVisionAVR, atau sebaliknya jika tidak ingin membuka CodeVisionAVR setelah proses instalasi selesai. Tekan tombol **Finish** untuk menyelesaikan proses instalasi.



Gambar 10
Proses Instalasi CodeVisionAVR Selesai

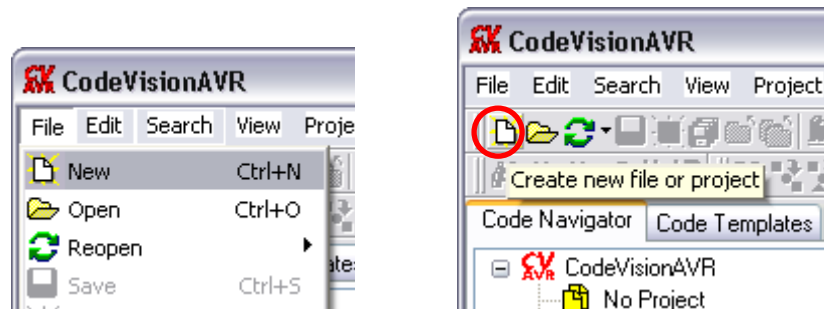
Tahap pembuatan aplikasi sederhana menggunakan CodeVisionAVR Evaluation adalah sebagai berikut:

1. Tampilan dasar CodeVisionAVR Evaluation dan menu yang tersedia.



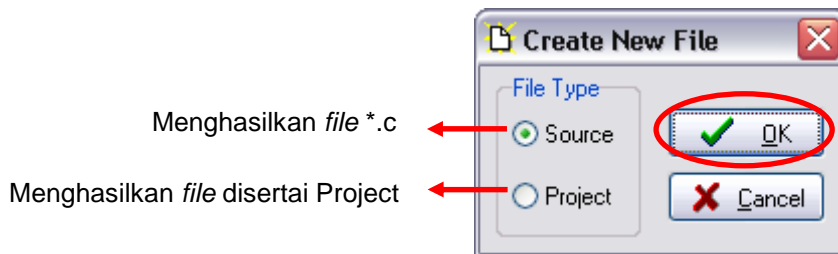
Gambar 11
Tampilan Dasar CodeVisionAVR Evaluation

2. Buat *file* baru melalui menu utama, pilih **File**, lalu pilih **New**. Bisa juga melalui Quick menu **Create new file or project**.



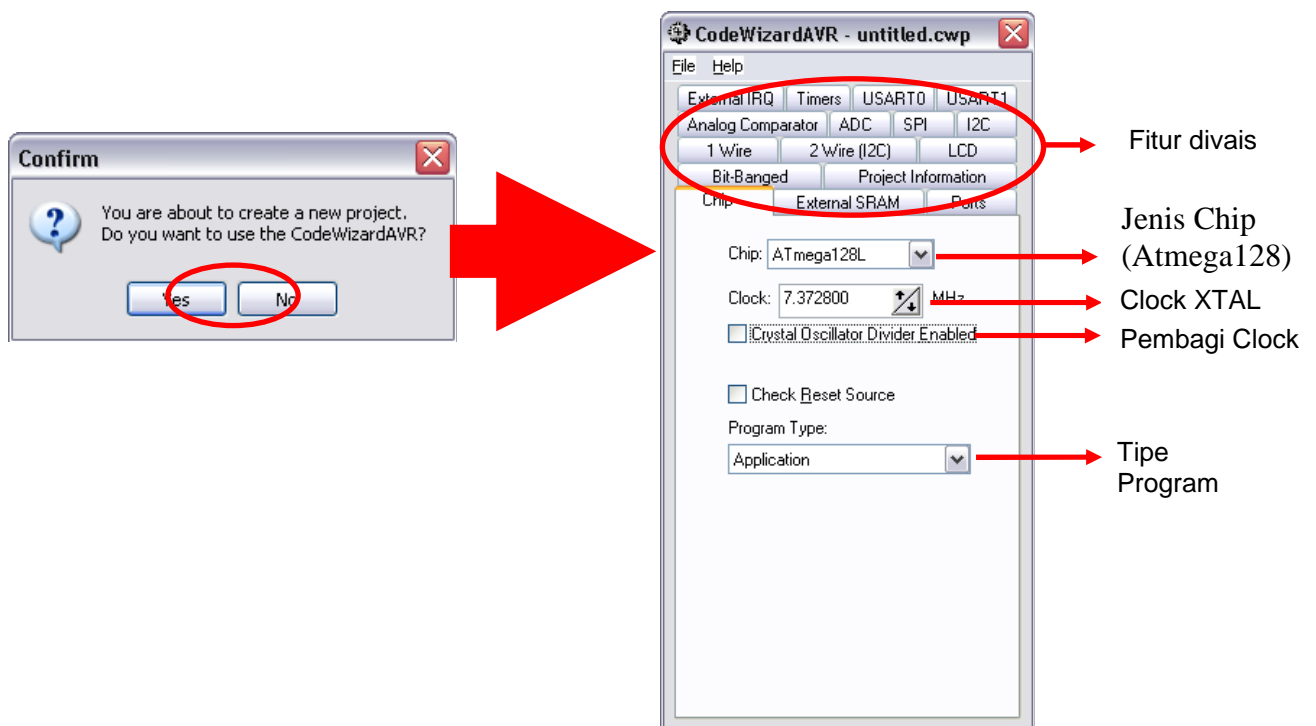
Gambar 12
Membuat *File* Baru

3. Kita akan melihat tampilan **Create New File**. Pada tampilan ini kita bisa membuat *file* disertai **project** atau kita hanya membuat *source file* yang berekstensi ***.c** saja. Kemudian tekan **OK**.



Gambar 13
Tampilan Create New File

4. Untuk mempermudah memulai pembuatan *file*, disarankan memilih opsi **Project**. Karena CodeVisionAVR Evaluation akan membantu melalui **CodeWizardAVR**. Pada **CodeWizardAVR** ini dapat memilih jenis **Chip** dan aktifasi fitur dari **Chip**. Pada saat muncul tampilan konfirmasi tekan tombol **Yes**, sehingga **CodeWizardAVR** muncul.



Gambar 14
Tampilan Konfirmasi dan CodeWizardAVR

5. Pada pembuatan contoh aplikasi ini akan dibahas mengenai program **testing128.c** yang disertakan dalam CD/DVD program. Fitur divais yang digunakan adalah komunikasi serial **USART** yang digunakan untuk melakukan komunikasi secara serial dengan komputer. Berikut penjelasan dari program testing128.c.

```
#include <mega128.h>
#include <delay.h>

// Standard Input/Output functions
#include <stdio.h>
```

Pada program di atas menunjukkan *library* yang digunakan. Pada baris pertama menunjukkan jenis *Chip* yang digunakan. Pada baris kedua merupakan *library delay* yang digunakan untuk menghasilkan waktu *delay*. Yang tercetak biru pada program menunjukkan *statement* dari program. Setiap kalimat yang didahului dengan garis miring rangkap 2 pada program tidak akan dieksekusi. Pada baris keempat menunjukkan *library* Standard Input/Output, yang digunakan untuk komunikasi serial.

6. Berikut penjelasan mengenai fungsi main() dari program testing128.c:

```
void main(void)
{
// Declare your local variables here
unsigned char counter=0,serialCount = 0;

// Input/Output Ports initialization
// Port A initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTA=0x00;
DDRA=0xFF;

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTB=0x00;
DDRB=0xFF;

// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTD=0x00;
DDRD=0xFF;

// Port E initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTE=0x00;
DDRE=0xFF;

// Port F initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTF=0x00;
DDRF=0xFF;

// Port G initialization
// Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State4=0 State3=0 State2=0 State1=0 State0=0
PORTG=0x00;
DDRG=0x1F;

// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: On
```



```

// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud rate: 115200
UCSR0A=0x00;
UCSR0B=0x18;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x03;

while (1)
{
    // Place your code here

    if(UCSR0A>>7 & 1)
    {
        if (getchar()=='A')
        {
            printf("%d -> DT-AVR Boot, Innovative Electronics\r",serialCount);
            serialCount++;
        }
    }

    PORTA = ~(0x01 << counter);
    PORTB = ~(0x01 << counter);
    PORTC = ~(0x01 << counter);
    PORTD = ~(0x01 << counter);
    PORTE = ~(0x01 << counter);
    PORTF = ~(0x01 << counter);
    PORTG = ~(0x01 << counter);
    counter++;
    if (counter>7) counter = 0;

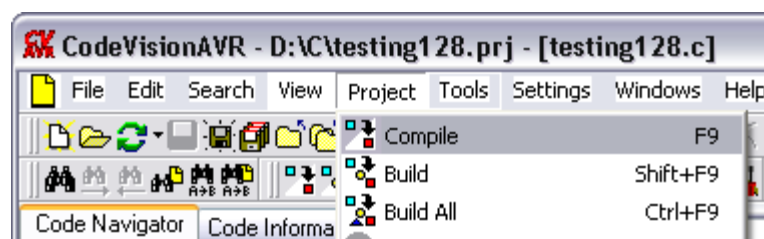
    delay_ms(200);
};
}

```

Pada fungsi **main()**, yang pertama dilakukan adalah deklarasi variabel **counter** dan **serialCount**. Kemudian diikuti dengan inisialisasi I/O dan inisialisasi UART. Selanjutnya program akan melakukan *looping* terus menerus. Pada *looping* tersebut, yang pertama dilakukan adalah memeriksa isi register **UCSR0A** untuk mengetahui apakah ada data serial yang masuk. Jika ada, maka data serial tersebut akan diambil menggunakan fungsi **getchar()**. Jika data serial yang diterima adalah karakter 'A', maka modul akan mengirimkan tulisan "DT-AVR Boot, Innovative Electronics".

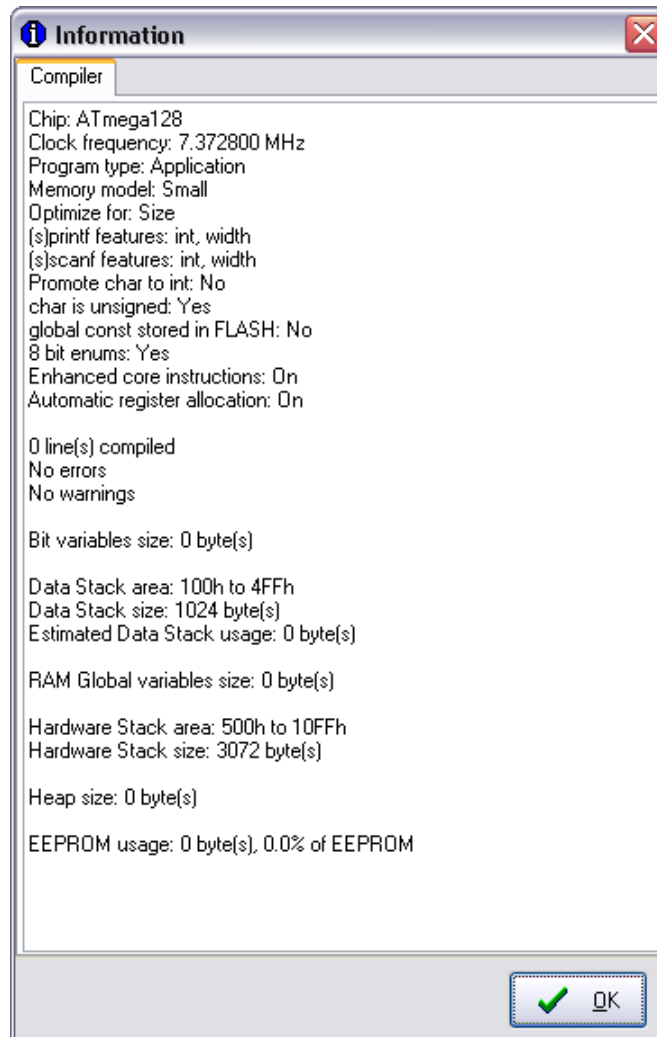
Pada Port A, Port B, Port C, Port D, Port E, Port F, dan Port G akan selalu dikeluarkan logika *low* pada 1 pin dalam setiap *port* secara bergantian, kecuali pin PE.0 dan PE.1 yang difungsikan untuk komunikasi serial.

- Langkah selanjutnya adalah melakukan *compiling* terhadap program yang telah dibuat. Proses ini dilakukan untuk memeriksa kesalahan penulisan program. Proses ini terdapat pada menu **Project - Compile** atau bisa juga dengan menekan tombol **F9** pada *keyboard*.



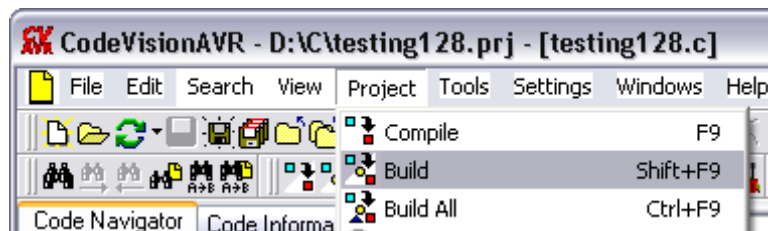
Gambar 15
Menu Compile

8. Jika *listing* program yang dibuat tidak mengandung kesalahan (termasuk kesalahan *syntax* atau penulisan), maka akan tampil *form* sebagai berikut.



Gambar 16
Hasil Proses Compile

9. Setelah proses *compiling* maka langkah selanjutnya adalah membentuk *file* HEX (*.hex). Proses ini bias dilakukan jika proses *compiling* berhasil dilakukan. Pembentukan *file* HEX ini melalui menu **Project - Build**. Atau bisa juga dilakukan dengan menekan tombol **Shift + F9** pada keyboard.

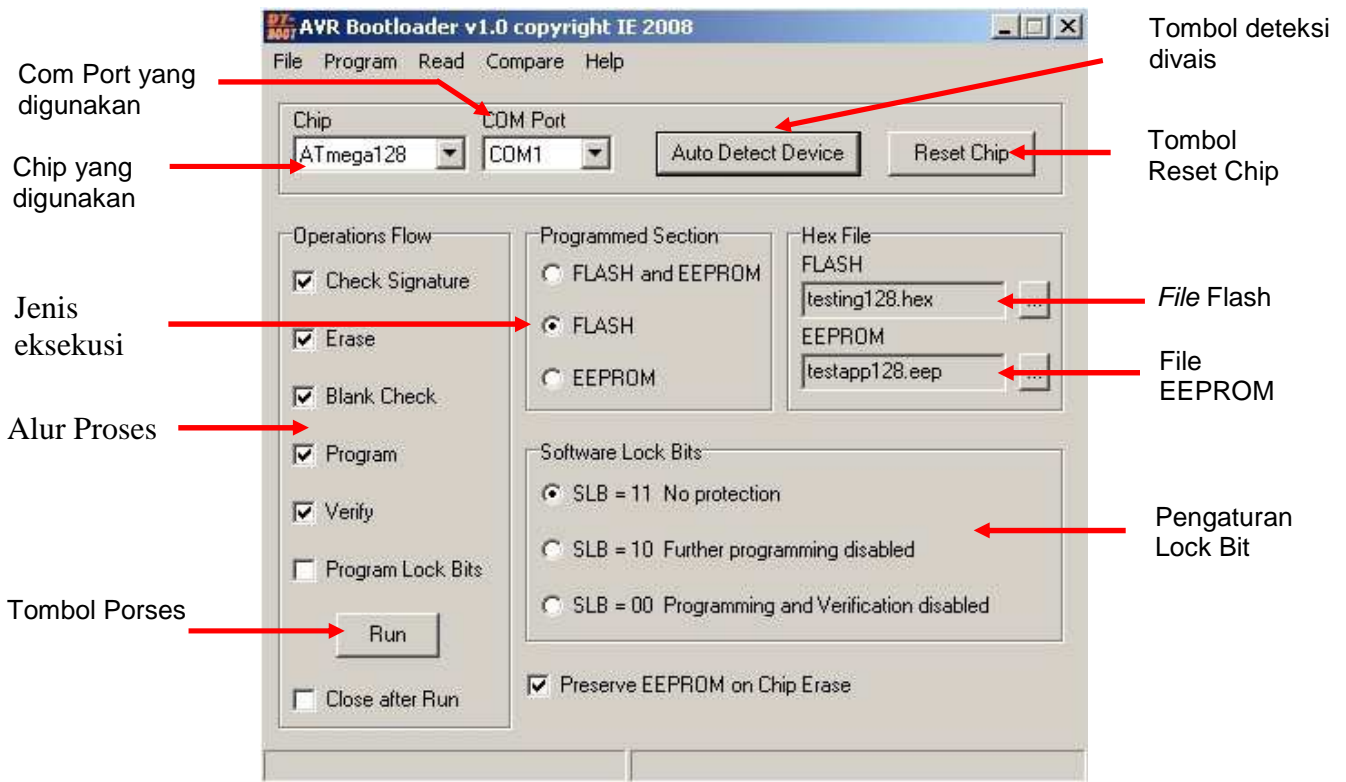


Gambar 17
Proses Membuat File HEX

10. Jika pada langkah di atas berhasil maka akan dihasilkan *file* HEX (*.hex). *File* HEX ini diletakkan pada *folder* yang sama dengan *project* (atau terletak dalam *folder* EXE di dalamnya) dengan nama sama dengan nama *project*. Dalam contoh ini, *file* bernama **testing128.hex**.

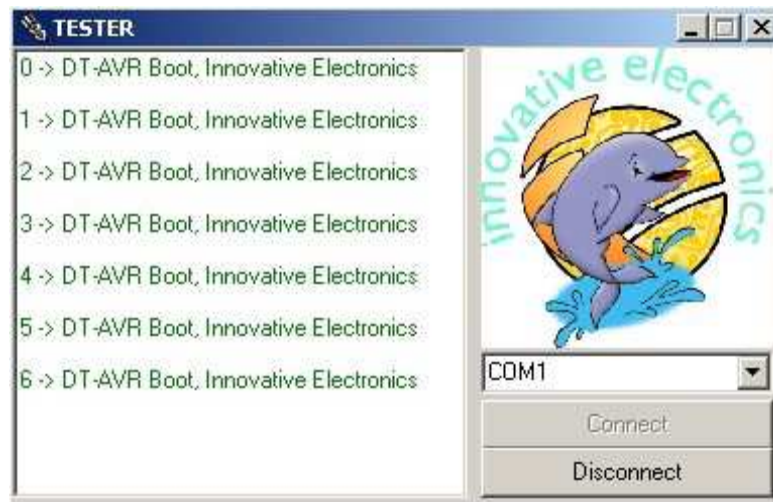
AVR Bootloader v1.0 adalah sebuah perangkat lunak dari Innovative Electronics yang mendukung pemrograman mikrokontroler secara *bootloader* pada DT-AVR ATMEGA128L BMS. AVR Bootloader v1.0 terdapat dalam CD/DVD yang disertakan bersama DT-AVR ATMEGA128L BMS.

1. Tampilan **AVR Botloader v1.0** serta penjelasan fungsi program.



Gambar 18
Tampilan AVR Bootloader v.1.0

2. Sebelum menjalankan program AVR Bootloader v1.0, DT-AVR ATMEGA128L BMS harus dihubungkan ke *port* USB atau serial komputer dan dinyalakan. Pada DT-AVR ATMEGA128 BMS yang menjadi divais programmer adalah mikrokontroler ATMEGA128 itu sendiri. Jalankan program AVR Bootloader v1.0, lalu tekan tombol **Auto Detect Device**. Program ini akan mendeteksi adanya modul yang dihubungkan ke komputer. Jika AVR Bootloader menampilkan pesan tidak adanya modul yang terdeteksi, tekan tombol *reset* pada modul dan jalankan kembali AVR Bootloader atau pilih IC yang digunakan secara manual.
3. Bagian **Hex File** berfungsi untuk memilih *file* yang akan diprogramkan ke mikrokontroler. Perlu diketahui bahwa AVR Bootloader v1.0 membedakan *file* yang akan diprogramkan ke Flash Memory dan EEPROM. Tekan tombol "..."**(Browse)** untuk memilih *file* yang diinginkan. Pilih *file* HEX untuk memprogram Flash Memory. *File* HEX (testing128.hex) dapat pula dibuka melalui menu **File > Load FLASH**.
4. Pada bagian **Programmed Section**, pilih **Flash** untuk memrogram, memeriksa hasil pemrograman dan membaca Flash Memory mikrokontroler.
5. Pilih menu pada **Operations Flow** untuk memilih beberapa opsi proses yang akan dijalankan, atau dapat juga melalui menu **Program** kemudian pilih proses yang akan dilakukan. Untuk memprogram file **testing128.hex** pada contoh ini, **centang** semua **Operation Flow (Check Signature, Erase, Blank Check, Program Verify)** serta centang **Programmed Section** pada menu **Flash**. Kemudian tekan tombol **Run** untuk memprogram mikrokontroler.
6. Untuk mencoba hasil pemrograman file **testing128.hex**, jalankan program **TESTER128.exe** yang disertakan pada CD/DVD.
7. Pilih **COMPort** yang akan digunakan melalui menu combobox lalu tekan tombol "**Connect**".
8. Bila komunikasi serial berjalan dengan sukses maka akan tampil tulisan "DT-AVR Boot, Innovative Electronics" secara terus menerus dengan interval waktu sekitar 1 detik sekali.



Gambar 19
Tampilan Program TESTER128.exe

9. Untuk melihat output yang dihasilkan pada Port A, Port B, Port C, Port D, Port E (kecuali PE.0 dan PE.1), Port F, dan Port G dapat menggunakan osiloskop, voltmeter, atau dihubungkan langsung dengan rangkaian LED atau DT-I/O LED LOGIC TESTER sehingga tampak nyala LED yang bergantian.

Selamat berinovasi!

AVR Bootloader is copyright by Innovative Electronics.
CodeVisionAVR is copyright by Pevel Haiduc, HP Info Tech s.r.l.