

Pengenalan Pemrograman C dan chip seri ATMega.

Achmadi ST MT

Daftar Isi

1	Requirement	3
1.1	Software	3
1.1.1	Code::Blocks	3
1.1.2	GCC AVR	4
1.1.3	Downloader	5
1.1.4	Simul-IDE	6
1.1.5	KiCad	7
1.2	Hardware	8
1.2.1	Downloader	8
1.2.2	Minimum-System	9
2	Bahasa C	11
2.0.1	Kompilasi	11
2.0.2	Struktur Dasar C	12
2.0.3	Komentar	15
2.0.4	Tipe Data	16

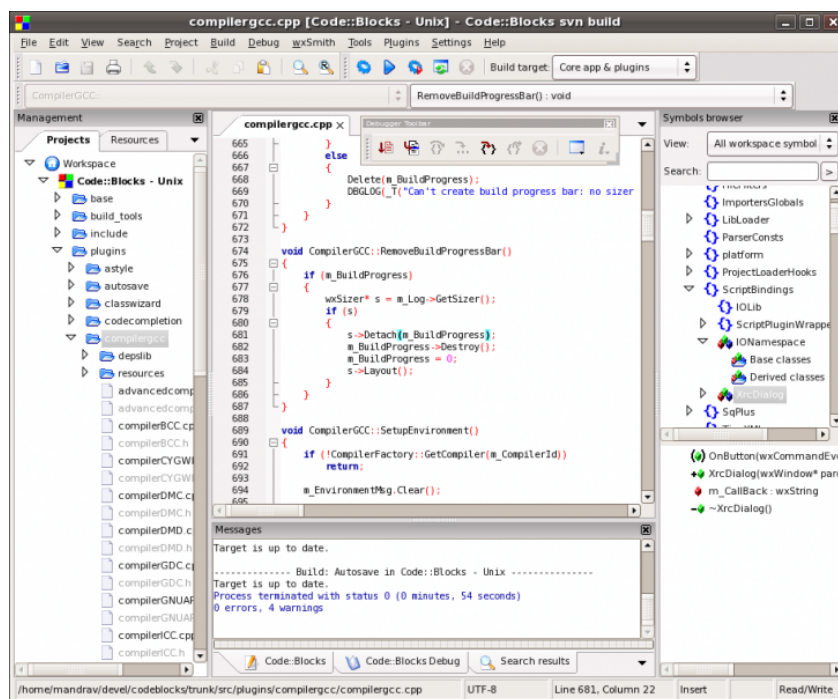
1 Requirement

Dalam pengembangan aplikasi berbasis ATmega, dibutuhkan beberapa hardware dan software. Berikut akan disebutkan dan dijelaskan untuk beberapa sistem operasi:

1.1 Software

1.1.1 Code::Blocks

Code::Blocks adalah suatu program lingkungan pengembangan terpadu bebas, gratis, bersumber terbuka dan lintas platform. Program yang ditulis dalam C++ beserta wxWidgets untuk GUI-nya ini bisa digunakan bersama dengan berbagai macam kompilator, contohnya GCC dan CLang. Peralatannya yang tersedia tergantung dari "plugin" yang ada dipasang. Sekarang ini, Code::Blocks lebih tersedia sebagai perangkat pengembangan dalam bahasa C dan C++, walaupun program ini juga bisa disesuaikan, dan mungkin akan membutuhkan pemasangan tambahan, untuk pengembangan perangkat lunak ARM, AVR, Fortran, GLFW, GLUT, GTK+, MATLAB, OGRE, OpenGL, Qt, atau SDL. Code::Blocks tersedia di sistem operasi Windows, Linux, Mac OS X dan FreeBSD.



Gambar 1: contoh tampilan Code::Blocks

Berikut instalasi:

- Windows.
 - Buka alamat <http://www.codeblocks.org/downloads/26>.
 - Klik file codeblocks-17.12mingw-setup.exe.
 - Tunggu proses download selesai.
 - Jalankan file codeblocks-17.12mingw-setup.exe untuk instalasi.
- Ubuntu/Debian.
 - Buka Terminal (pastikan terhubung internet).
 - Masukkan perintah.

```
sudo apt-get install codeblocks build-essential
```

- Tunggu proses selesai.

- Fedora.
 - Buka Terminal (pastikan terhubung internet).
 - Masukkan perintah.

```
sudo dnf groupinstall 'Development Tools'
sudo dnf install codeblocks
```

- Tunggu proses selesai.

- Arch-Linux.
 - Buka Terminal (pastikan terhubung internet).
 - Masukkan perintah.

```
sudo pacman -S codeblocks base-devel
```

- Tunggu proses selesai.

1.1.2 GCC AVR

GCC adalah software kompilasi gratis dan bebas untuk chip AVR. GCC (GNU C Compiler) bertugas mengkompilasi kode sumber menjadi file *binary* yang siap dijalankan atau di-*download* ke dalam chip.

Berikut instalasi:

- Windows.
 - Download file instalasi di alamat:
 - * Windows x86 (32bit): <http://blog.zakkemble.net/download/avr-gcc-8.3.0-x86-mingw.zip>
 - * Windows x64 (64bit): <http://blog.zakkemble.net/download/avr-gcc-8.3.0-x64-mingw.zip>
 - Ekstrak file yang sudah di download.
 - Masukkan alamat kompiler dalam *binary path* milik Windows.

- Ubuntu/Debian.
 - Buka Terminal (pastikan terhubung internet).
 - Masukkan perintah.

```
sudo apt-get install gcc-avr binutils-avr avr-libc
```

- Tunggu proses selesai.

- Fedora.
 - Buka Terminal (pastikan terhubung internet).
 - Masukkan perintah.

```
sudo dnf install avr-binutils avr-gcc avr-libc
```

- Tunggu proses selesai.

- Arch-Linux.
 - Buka Terminal (pastikan terhubung internet).
 - Masukkan perintah.

```
sudo pacman -S avr-binutils avr-gcc avr-lib
```

- Tunggu proses selesai.

1.1.3 Downloader

Download adalah software bantu untuk download (memasukkan) binary hasil kompilasi ke dalam chip. Tersedia banyak software tergantung tipe hardware downloader yang dipakai. Untuk kursus ini, digunakan downloader berbasis UASAsp, maka software downloader yang digunakan adalah Khazama untuk Windows dan avrdude (AVR Downloader/UploaDEr) untuk selain Windows.

Berikut instalasi:

- Windows



Gambar 2: contoh tampilan Khazama

- Download file instalasi Khazama 1.6 di alamat <http://khazama.com/project/programmer/KhazamaAVRProgrammer162.rar>.
- Ekstrak file yang di download dan jalankan file installernya.

- Ubuntu/Debian.

- Buka Terminal (pastikan terhubung internet).
- Masukkan perintah.

```
sudo apt-get install avrdude
```

- Tunggu proses selesai.

- Fedora.

- Buka Terminal (pastikan terhubung internet).
- Masukkan perintah.

```
sudo dnf install avrdude
```

- Tunggu proses selesai.

- Arch-Linux.

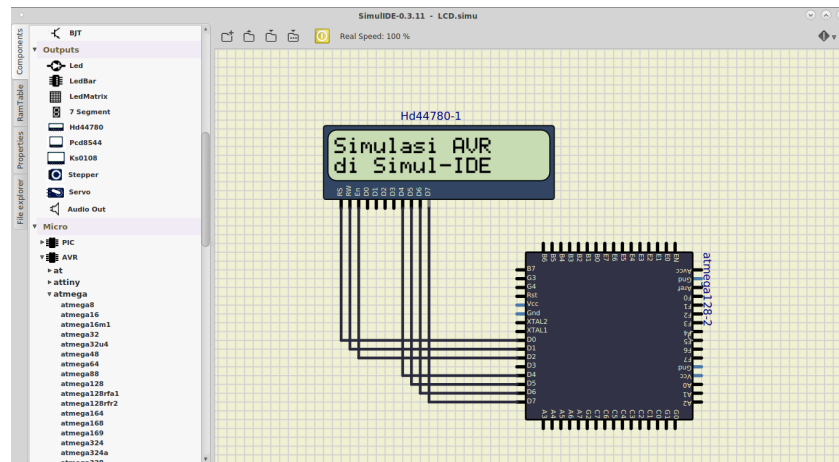
- Buka Terminal (pastikan terhubung internet).
- Masukkan perintah.

```
sudo pacman -S avrdude
```

- Tunggu proses selesai.

1.1.4 Simul-IDE

Dalam kursus ini, untuk awal pelatihan, digunakan Simulator untuk menjalankan dan menguji program AVR yang dibuat. Simulator yang dipilih disini adalah Simul-IDE. Simul-IDE adalah simulator rangkaian elektronik Digital/Analog sederhana yang dapat berjalan secara *real-time*. Simul-IDE dapat mendukung microcontroller seperti chip seri AVR ataupun PIC.



Gambar 3: contoh tampilan Simul-IDE

Berikut untuk instalasi:

- Windows

- Buka alamat https://sourceforge.net/projects/simulide/files/SimulIDE_0.3.10/SimulIDE_0.3.10_SR1-Win32.zip/download
- Ekstrak file yang sudah di download.
- Jalankan file **run_simulide.bat**.

- Ubuntu/Debian.

- Buka Terminal (pastikan terhubung internet).
- Masukkan perintah.

```
sudo apt-get install simulide simavr
```

- Tunggu proses selesai.
- Buka terminal dimana project AVR berada.
- Masukkan perintah:

```
simulide
```

- Fedora.

- Buka Terminal (pastikan terhubung internet).
- Masukkan perintah.

```
sudo dnf install simulide simavr
```

- Tunggu proses selesai.
- Buka terminal dimana project AVR berada.
- Masukkan perintah:

```
simulide
```

- Arch Linux.

- Buka Terminal (pastikan terhubung internet).
- Masukkan perintah.

```
sudo pacman -S simavr
```

- Tunggu proses selesai.
- Download resep AVR di <https://aur.archlinux.org/cgit/aur.git/snapshot/simulide.tar.gz>
- Ekstrak file tersebut dan buka terminal di alamat tersebut.
- Masukkan perintah:

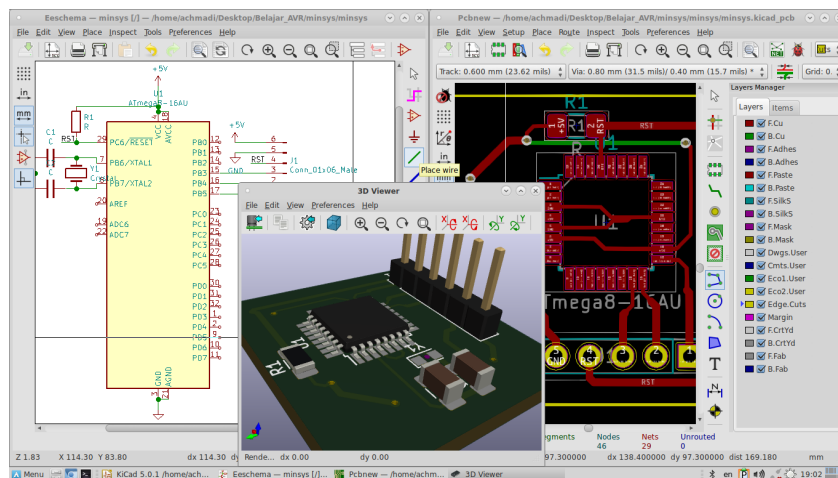
```
makepkg -sir
```

- Tunggu proses instalasi selesai
- Buka terminal dimana project AVR berada.
- Masukkan perintah:

```
simulide
```

1.1.5 KiCad

KiCad adalah software bebas dan gratis untuk membantu merancang circuit elektronik. KiCad menyediakan fitur olah skematik, layout PCB, dan 3D visualisasi.



Gambar 4: contoh tampilan KiCAD

Berikut untuk instalasi:

- Windows

- Buka alamat <http://kicad-pcb.org/download/windows/>
- Pilih dan download file instalasi sesuai laptop/komputer anda.
- Jalankan file instalasi.

- Ubuntu/Debian.

- Buka Terminal (pastikan terhubung internet).
- Masukkan perintah.

```
sudo add-apt-repository ppa:js-reynaud/kicad-5
sudo apt-get update
sudo apt-get install kicad
```

- Tunggu proses selesai.

- Fedora.

- Buka Terminal (pastikan terhubung internet).
- Masukkan perintah.

```
sudo dnf install kicad kicad-packages3d
```

- Tunggu proses selesai.

- Arch Linux.

- Buka Terminal (pastikan terhubung internet).
- Masukkan perintah.

```
sudo pacman -S kicad kicad-library kicad-library-3d
```

- Tunggu proses selesai.

1.2 Hardware

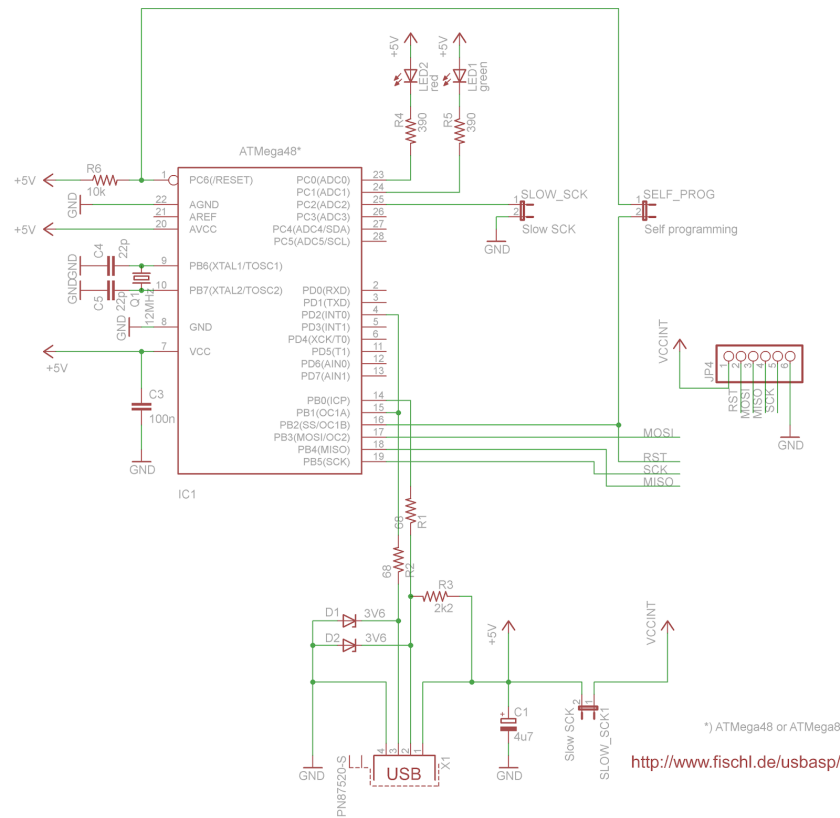
Untuk hardware, kebutuhan pokok dalam membangun aplikasi ATMega hanyalah 2 saja, yaitu: Downloader dan Minimum-System

1.2.1 Downloader

Downloader adalah hardware yang digunakan untuk memasukkan program dalam chip. Tipe downloader yang digunakan disini adalah USBasp dimana baik skematik maupun firmware yang digunakan bersifat open-source. USBasp secara mendasar hanya membutuhkan:

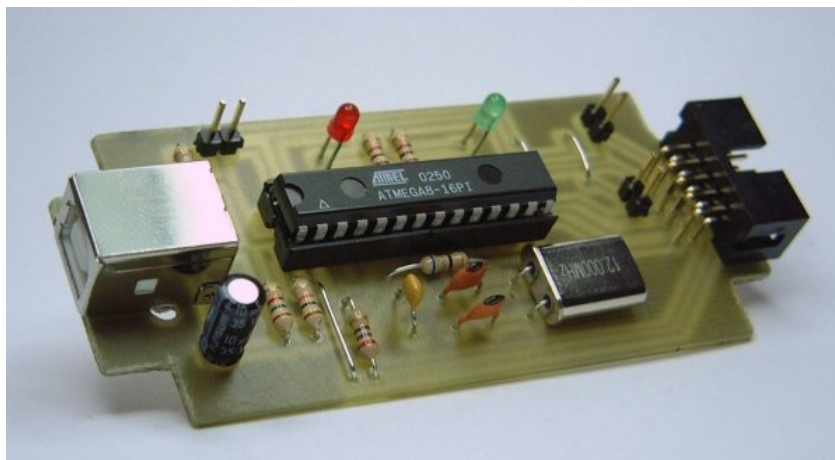
- ATMega48, ATMega8, atau ATMega88
- Clock 12MHz atau lebih
- 2 PIN untuk jalur VCC dan GND
- 4 PIN untuk jalur SPI (SS, MOSI, MISO, dan SCK)
- 4 PIN untuk jalur USB (VCC, GND, D+, dan D-)

Project USBasp dapat anda temukan di alamat <https://www.fischl.de/usbasp/>.



Gambar 5: contoh skematik USBasp

Sedangkan untuk hardware dapat anda beli di pasaran dengan harga terjangkau atau anda dapat merakit sendiri.



Gambar 6: contoh USBasp murah meriah

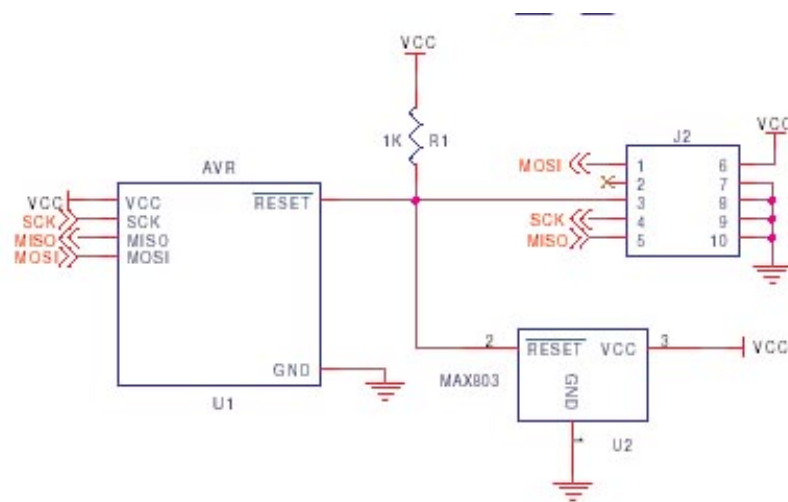
Advance: Selain menggunakan USBasp fisik, anda dapat juga menanamkan *virtual* downloader USBasp kedalam bagian *bootloader* ATmega. Keuntungannya anda tidak lagi perlu USBasp fisik terpisah dengan Minimum-System, namun kekurangannya adalah alamat aplikasi berpindah ke alamat bootloader sehingga terkadang memory tidak sampai alamat aplikasi. Project ini dikenal dengan USBaspLoader dan dapat anda temukan di <https://www.obdev.at/products/vusb/usbasploader.html>.

1.2.2 Minimum-System

Minimum-System didefinisikan sebagai rangkaian dasar agar chip ATmega dapat berjalan. Pada dasarnya ATmega hanya membutuhkan:

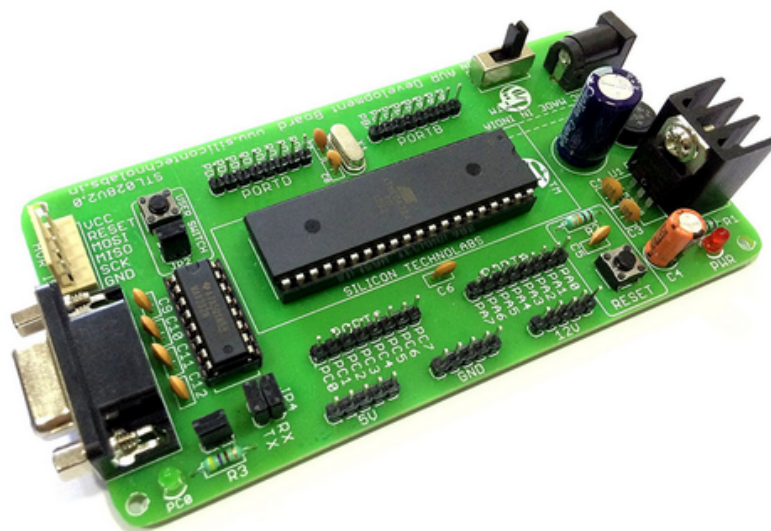
- Power VCC (5v) dan GND (0v). Power ini bisa didapat dari tegangan USB atau output regulator 7805/LM2569s.

- 5 PIN untuk download via SPI (GND, RST, MOSI, MISO, dan SCK) jika ingin didownload *on-board*.
- Crystal (XTALL) atau RTC jika sumber Clock ATmega di set eksternal.



Gambar 7: contoh diagram dasar Minimum-System

Berdasarkan rangkaian dasar ini, anda dapat menambahkan beragam komponen lain sesuai kebutuhan anda. Jika anda tidak waktu untuk membangun Minimum-System sendiri, anda dapat membelinya di pasaran dengan beragam harga dan spesifikasi.



Gambar 8: contoh fisik Minimum-System yang dilengkapi RS-232 dan regulator 5v

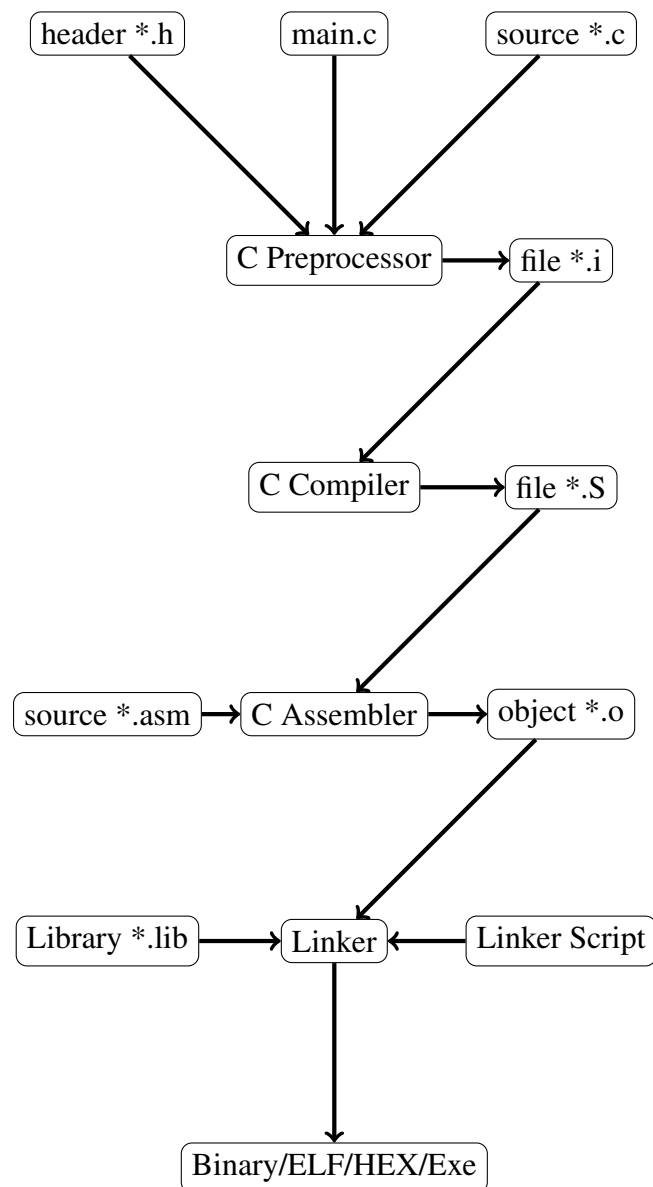
2 Bahasa C

Bahasa pemrograman C merupakan salah satu bahasa pemrograman komputer. Dibuat pada tahun 1972 oleh Dennis Ritchie untuk Sistem Operasi Unix di Bell Telephone Laboratories.

Meskipun C dibuat untuk memprogram sistem dan jaringan komputer namun bahasa ini juga sering digunakan dalam mengembangkan software aplikasi. C juga banyak dipakai oleh berbagai jenis platform sistem operasi dan arsitektur komputer, bahkan terdapat beberapa compiler yang sangat populer telah tersedia. C secara luar biasa memengaruhi bahasa populer lainnya, terutama C++ yang merupakan ekstensi dari C.

2.0.1 Kompilasi

Kompilasi adalah proses mengumpulkan dan mengubah kode sumber menjadi file yang bisa dijalankan. Software/Firmware memiliki 2 bentuk yaitu kode sumber (*source*) dan *binary*. Source dapat dibaca oleh manusia, namun belum bisa dimengerti oleh CPU. Sebaliknya, binary tidak bisa dipahami manusia, namun bisa dipahami CPU dan dijalankan isinya. Berikut secara ringkas diagram kompilasi:



Gambar 9: Diagram Proses Kompilasi

2.0.2 Struktur Dasar C

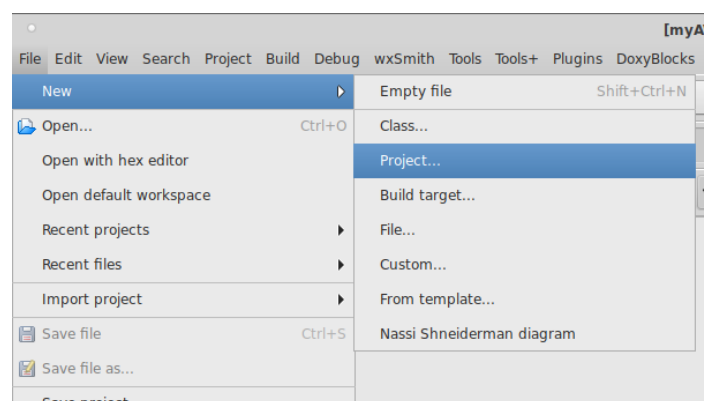
Bahasa C secara umum memiliki 4 bagian wajib, yaitu:

- Preprocessor
- Functions
- Variables
- Statement (Expressions atau Operation).

Setiap baris *statement* diakhiri dengan tanda titik koma (;).

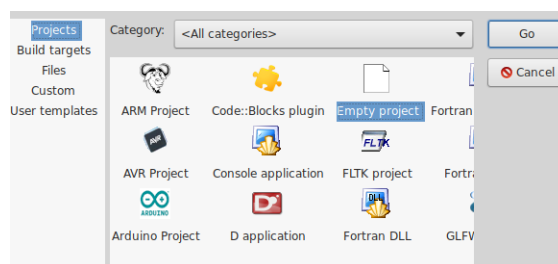
Berikut adalah contoh dasar yang dapat dijalankan di Code::Blocks.

1. Buka Code::Blocks.
2. Buat project baru.



Gambar 10: membuat project baru

3. Pilih Empty Project.



Gambar 11: membuat project kosong

4. Isi nama project.

Empty project

Please select the folder where you want the new project to be created as well as its title.

Project title:
coba

Folder to create project in:
/home/achmadi/Desktop/testC

Project filename:
coba.cbp

Resulting filename:
/home/achmadi/Desktop/testC/coba/coba.cbp

Gambar 12: nama dan tempat project

5. Pilih compiler **GNU GCC** dan isi build target **all** seperti dibawah

Empty project

Please select the compiler to use and which configurations you want enabled in your project.

Compiler:
GNU GCC Compiler

☐ Create "Debug" configuration: Debug

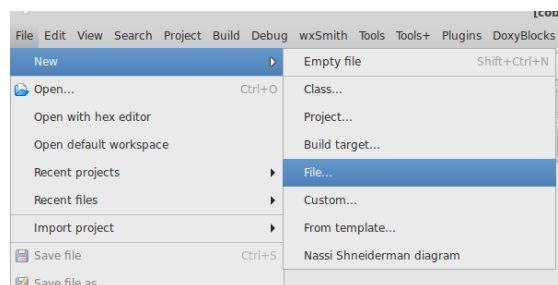
"Debug" options
Output dir.: bin/Debug/
Objects output dir.: obj/Debug/

☒ Create "Release" configuration: all

"Release" options
Output dir.: bin/
Objects output dir.: obj/

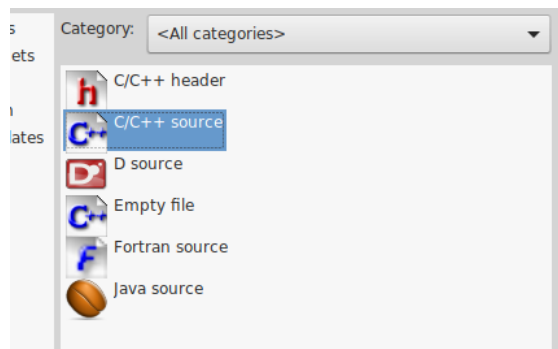
Gambar 13: pilih compiler

6. Buat File baru



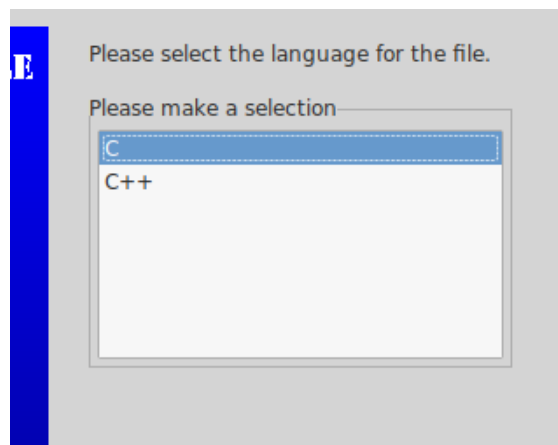
Gambar 14: buat file baru

7. Pilih file C/C++ Sources



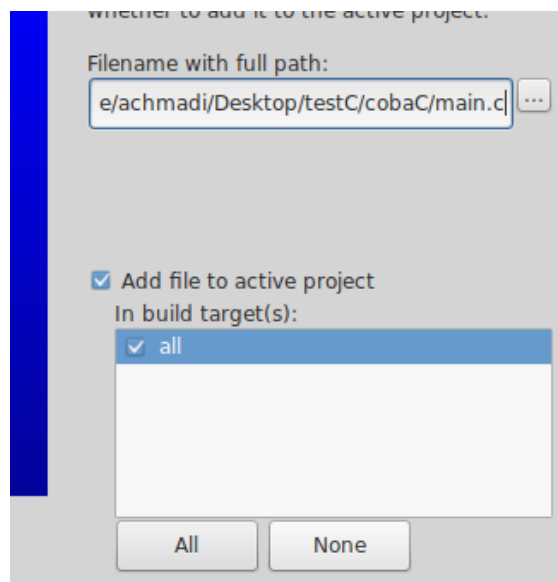
Gambar 15: C/C++ source

8. Pilih tipe C



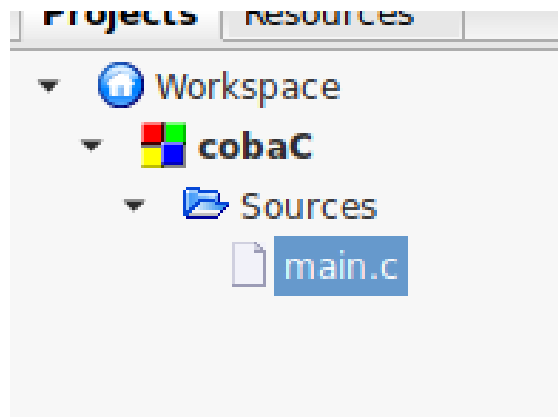
Gambar 16: C source

9. isi alamat folder project tadi diakhiri dengan **main.c**. Jangan lupa centang **all** untuk menambahkan file ke



Gambar 17: nama dan alamat file

10. hasil akhir **main.c**.



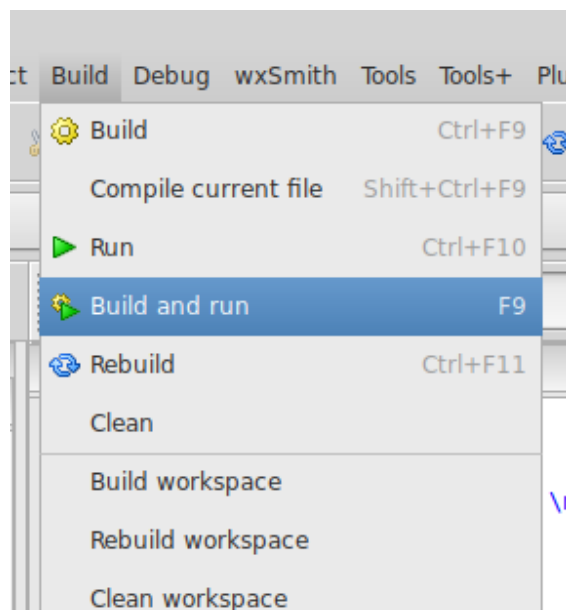
Gambar 18: file main.c

11. masukkan kode berikut. Tips: Manfaatkan Code Completion.

```
#include <stdio.h>

int main() {
    printf("Hello, World! \n");
    return 0;
}
```

12. Tekan menu Build->Build and Run.



Gambar 19: Build and Run

2.0.3 Komentar

Komentar adalah bagian dari kode sumber yang bukan bagian dari pemrograman dan tidak dikompilasi compiler. Komentar ini berfungsi sebagai penanda, dokumentasi, penjelasan, atau apa saja yang dapat membantu memahami source yang ditulis. Bisa juga sekedar "non-aktifkan" sebagian kode.

Berikut contoh komentar:

- Per Baris

```
// contoh komentar
unsigned int vt=0; //sebelah ini bukan komentar
// per baris
```

- Per Block

```
/* contoh komentar
unsigned int vt=0; (sebelah ini ikut jadi komentar)
per baris */
```

2.0.4 Tipe Data

Tipe data dalam bahasa C diukur dengan satuan **bit**. Nilai 1 bit adalah nilai terkecil dalam memory yang bisa diisi antara 1 atau 0 (salah satu). Berikut satuan tipe data yang sering dipakai:

- Bit. Satuan terkecil yang hanya berisi 1 atau 0
- Nibble. Senilai dengan 4bit.
- Byte. Senilai dengan 8bit.

Selanjutnya dalam alokasi variable dalam memory, dikenal istilah tipe data. Tipe data adalah deklarasi jenis variable berdasarkan ukurannya. Setiap tipe data memiliki ukuran maximal yang jika nilai nya melewatinya akan terjadi *overflow* (kembali ke 0). Berikut tipe data yang umum:

name	type	bit-size	range
boolean	bool	1	0~1
character	unsigned char	8	0~255
character	char	8	-127~128
integer	unsigned int	16	0~65.535
integer	int	16	-32,768~32,767
long	unsigned long	32	0~4,294,967,295
long	long	32	±2,147,483,647
float	float	32	±2,147,483,647
double	double	64	±9,223,372,036,854,775,807

Untuk tipe double dan float, mampu menyimpan bilangan *floating-point* (decimal). Akan tetapi standar floating-point setiap chip arsitektur berbeda-beda sehingga perlu diperhatikan.

Tujuan setelah mengenal tipe data adalah untuk mendeklarasikan tipe variabel sesuai kebutuhan. Dalam bahasa C/C++, variable wajib dideklarasikan terlebih dahulu agar sistem dapat memesan ukuran memori secara pas. Pola deklarasi variable:

```
<rentang> <tipe data> variabel;
```

Penjelasan:

- <rentang> adalah tipe rentang yang dipakai. Pilihannya adalah **unsigned** yaitu dari 0 hingga maximal. Dan **signed** atau **kosong** saja untuk minus setengah maximal hingga plus setengah maximal.
- <tipe data> adalah tipe data sesuai rentang.
- variabel adalah nama variable dengan aturan:
 - tidak diawali angka, tapi boleh underscore (_).
 - tidak boleh ada spasi,koma,titik, tapi boleh underscore (_).
 - tidak konflik nama dengan yang lain di lingkungan pemrograman.

Sebagai contoh:

```
unsigned int foo;
foo = 5;
int foomin;
foomin = -5;
```
