

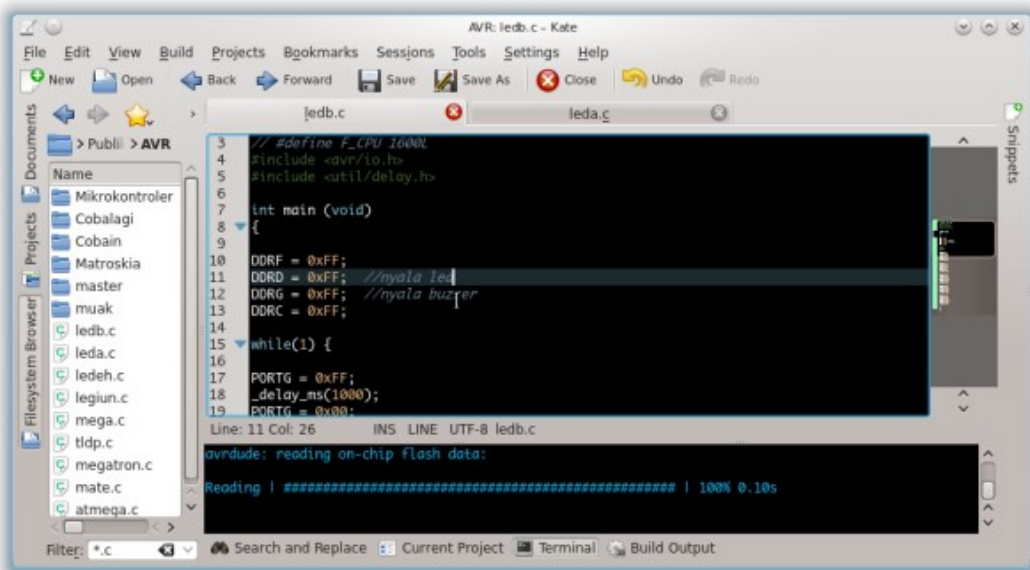
# Pengantar Pemrograman AVR – Kate Sebagai Pengganti Codevision di Linux

Bismillahirrahmanirrahim.



Kita melakukan *compile*, *build*, atau *burn* di Codevision AVR di Windows cukup dengan satu klik. Namun seringkali kita tidak mengetahui apa sesungguhnya yang terjadi ketika klik itu dilakukan. Ternyata, hal itu dilakukan dengan konfigurasi antara tombol atau shortcut key dengan perintah. Tulisan ini menjelaskan cara untuk mengonfigurasi build options dari Kate Text Editor supaya dapat melakukan pemrograman AVR di Linux. Dengan menerapkan tutorial ini, Anda dapat mengonfigurasi pula Gedit, Geany, Codeblocks, atau editor lain supaya menjadi “Codevision AVR” Anda di Linux.

Semoga tulisan ini bermanfaat.



## Daftar Isi

Pengantar Pemrograman AVR – Kate Sebagai Pengganti Codevision di Linux.....	1
1. Selayang Pandang.....	2
2. Langkah-Langkah Konfigurasi.....	2
3. Cara Bekerja.....	3
4. Penjelasan Kode.....	3
5. Penutup.....	3
6. Referensi.....	4
7. Tentang Dokumen Ini.....	4
8. Tentang Penulis.....	4

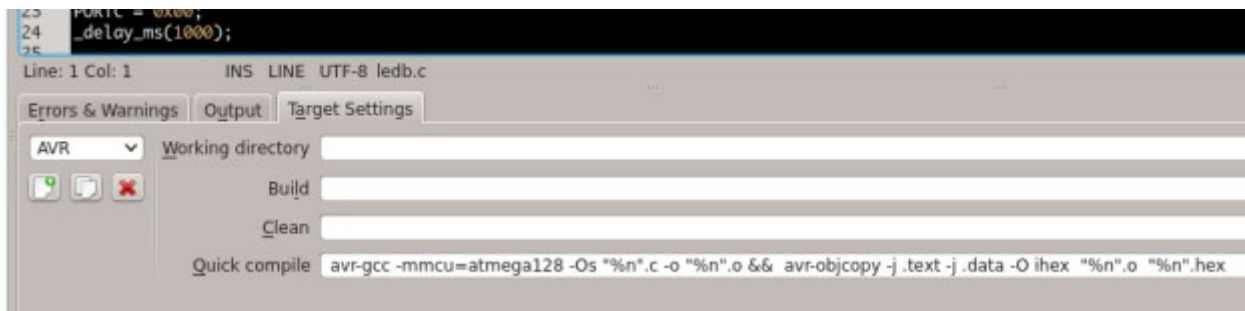
# 1. Selayang Pandang

Diharapkan Anda akan memperoleh yang seperti ini setelah mempraktikkan tutorial ini.

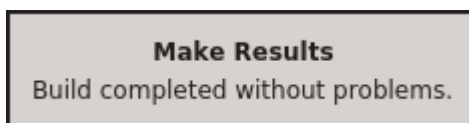
1. Ctrl+B : kompilasi dan *build* program .c menjadi .hex.
2. Pembakaran .hex ke Atmega dilakukan dari jendela Kate tanpa membuka Terminal terpisah.
3. Sama persis atau minimal mendekati *workflow* di Codevision AVR.

## 2. Langkah-Langkah Konfigurasi

1. Aktifkan Build Plugin pada Settings > Configure Kate... > Plugins > Build Plugin.
2. Akan muncul satu tombol Build Output di bagian bawah Kate. Klik.
3. Isikan kode berikut ke dalam kotak isian Quick compile: `avr-gcc -mmcu=atmega128 -Os "%n".c -o "%n".o && avr-objcopy -j .text -j .data -O ihex "%n".o "%n".hex` (bacalah penjelasan kode di bawah).



4. Asosiasikan aksi Quick compile dengan Ctrl+B pada Settings > Configure Shortcuts > klik pada kolom setelah Quick Compile. Jika terdapat konflik dengan suatu shortcut lain, paksa saja dengan OK/reassign.
5. Sekarang cobalah kompilasikan kode program AVR Atmega yang telah Anda buat menjadi .hex dengan menekan Ctrl+B setelah menyimpan ketikannya. Jika semua konfigurasi benar, maka Kate akan menampilkan pesan bahwa Build completed **without problems**. Jika belum punya kode sumber C, silakan unduh pada akhir tulisan ini.



6. Sekarang, untuk membuat Kate bisa membakar .hex, asosiasikan fokus. Ini gunanya untuk memindahkan fokus kerja Anda sekarang dari teks ke Terminal bawaan Kate dan sebaliknya. Buka Settings > Configure Shortcuts > cari Focus Terminal dengan search > beri shortcut key F2.

## 3. Cara Bekerja

Sekarang Anda perlu tahu bagaimana langkah kerja dengan Kate untuk melakukan pemrograman C AVR (yakni compile, build, dan burn). Berikut ini langkah-langkahnya.

1. Tulis kode lalu simpan.
2. Ctrl+B untuk membuat .hex.
3. F2 untuk membuka Terminal.
4. Panggil perintah avrdude yang sudah pernah diketik (harus sebagai root).
5. Cek hardware Atmega Anda.
6. F2 untuk kembali kepada kode.
7. Istimewanya Terminal di Linux, Anda bisa mencari perintah yang sudah pernah diketik dengan tombol panah atas. Jalankan perintah dengan Enter. Jadi, tidak perlu mengetik lagi perintah avrdude yang panjang itu setiap akan membakar .hex. Inilah workflow saya yang sudah saya lakukan sendiri untuk melakukan pemrograman + pembakaran kode AVR. Sangat ringkas dan hemat waktu. Jika Anda berhasil mengonfigurasi Kate sehingga jadi seperti di atas, maka selamat! Anda telah mendapatkan Codevision AVR secara gratis hanya dengan Kate.

## 4. Penjelasan Kode

Di atas saya sebutkan Anda harus menuliskan kode. Sekarang saya harus bertanggung jawab menjelaskan maksudnya.

1. Kode di atas adalah perintah untuk kompilasi dan untuk linking.
2. Kode tersebut terdiri dari dua perintah terpisah yang dipersatukan dengan **&&**.
3. Bisa Anda tebak, perintah pertama adalah `avr-gcc -mmcu=atmega128 -Os "%n".c -o "%n".o` dan perintah kedua adalah `avr-objcopy -j .text -j .data -O ihex "%n".o "%n".hex`.
4. Yang perlu Anda perhatikan cuma tanda %n pada kedua perintah. Apa maksudnya? Maksudnya adalah mendeteksi nama file Anda sekarang. Jika tidak begini, Anda harus menulis ulang Build Option pada setiap file .c Anda dengan nama file-nya.
5. Opsi `-mmcu=atmega128` harus Anda ganti dengan chip Atmega yang sesuai. Kalau Anda menggunakan Atmega8, maka ganti namanya dengan `atmega8`. Ini harus diperhatikan untuk semua chip Atmega. Jika salah, maka saya tidak bertanggung jawab akan apa yang terjadi.

## 5. Penutup

Dengan cara yang sama, Anda bisa membuat Codevision AVR Anda sendiri dengan text editor yang sudah ada di Linux. Semoga tulisan ini bermanfaat.

## 6. Referensi

1. <http://www.tldp.org/HOWTO/Avr-Microcontrollers-in-Linux-Howto/x207.html>
2. <http://pikirsa.wordpress.com/2011/08/28/geany-untuk-pemrograman-atmel-avr-di-gnu-linux-ubuntu/>

## 7. Tentang Dokumen Ini

Dokumen ini adalah versi PDF dari tulisan asli

<http://malsasa.wordpress.com/2014/01/16/pengganti-codevision-avr-di-linux/>. Ditulis dengan Libreoffice Writer di Ubuntu 12.04. Fonta yang dipakai adalah Ubuntu 12 pt.

Dokumen ini mulai ditulis pada 28 Januari 2014 dan diterbitkan pertama kali pada 28 Januari 2014. Dokumen ini masih berisi kekurangan dan tidak ditutup kemungkinan untuk direvisi nantinya.

## 8. Tentang Penulis

Penulis adalah warga Forum Ubuntu Indonesia. Penulis mendukung penggunaan perangkat lunak legal (terutama FOSS) untuk masyarakat. Penulis menyediakan buku-buku panduan Linux untuk pemula maupun ahli untuk diunduh secara gratis<sup>1</sup>. Penulis bisa dihubungi via SMS di nomor 0896 7923 7257.

---

1 <http://malsasa.wordpress.com/pdf>