

# Pengantar Pemrograman AVR – Tutorial Avrdude di Linux untuk Membakar HEX ke Atmega128 Melalui USBASP

Bismillahirrahmanirrahim.



Tutorial sederhana ini sejatinya adalah catatan pengalaman saya setelah berhasil membakar (download) .hex ke dalam chip Atmega128 melalui kabel USBASP dengan program avrdude di Ubuntu. Tutorial ini – walaupun sederhana– ditujukan sebagai pijakan awal Anda untuk memrogram mikrokontroler AVR dengan hanya menggunakan Linux. Tutorial ini diharap memberi gambaran Anda untuk menggantikan semua aplikasi Windows yang biasa dipakai semisal untuk IDE:

Codevision AVR, AVR Studio, dan Atmel Studio; dan untuk burner: Khazama dan Extreme Burner. Tutorial ini menjelaskan cara penggunaan avrdude sebagai pengganti Khazama atau Extreme Burner di Linux. Perlu Anda ketahui sebetulnya Khazama sendiri hanya GUI untuk avrdude yang dibuat khusus untuk USBASP<sup>[1]</sup> sedangkan Extreme Burner sendiri sudah memiliki versi Linux<sup>[2]</sup>. Jadi, yang berperan dalam pembakaran .hex di sistem operasi apa pun adalah avrdude. Semoga tutorial ini bermanfaat.

## Daftar Isi

Pengantar Pemrograman AVR – Tutorial Avrdude di Linux untuk Membakar HEX ke Atmega128 Melalui USBASP.....	1
1. Prinsip Pemrograman AVR di Linux.....	2
2. Hardware yang Dibutuhkan.....	2
3. Software yang Dibutuhkan.....	3
4. Spesifikasi Sistem.....	4
5. Langkah-Langkah Download HEX ke Atmega.....	5
6. Perintah untuk Download HEX.....	5
7. Hasil Download.....	6
8. Rangkuman.....	8
9. Tentang Tulisan Ini.....	8
10. Ucapan Terima Kasih.....	8
11. Referensi.....	8
12. Tentang Dokumen Ini.....	9
13. Tentang Penulis.....	9

# 1. Prinsip Pemrograman AVR di Linux

1. Tulis kode di IDE (seperti Codeblocks).
2. Kompilasikan jadi .o dengan avr-gcc.
3. Link jadi .hex dengan avr-objcopy<sup>[3][4]</sup>.
4. Bakar ke Atmega dengan avrdude.

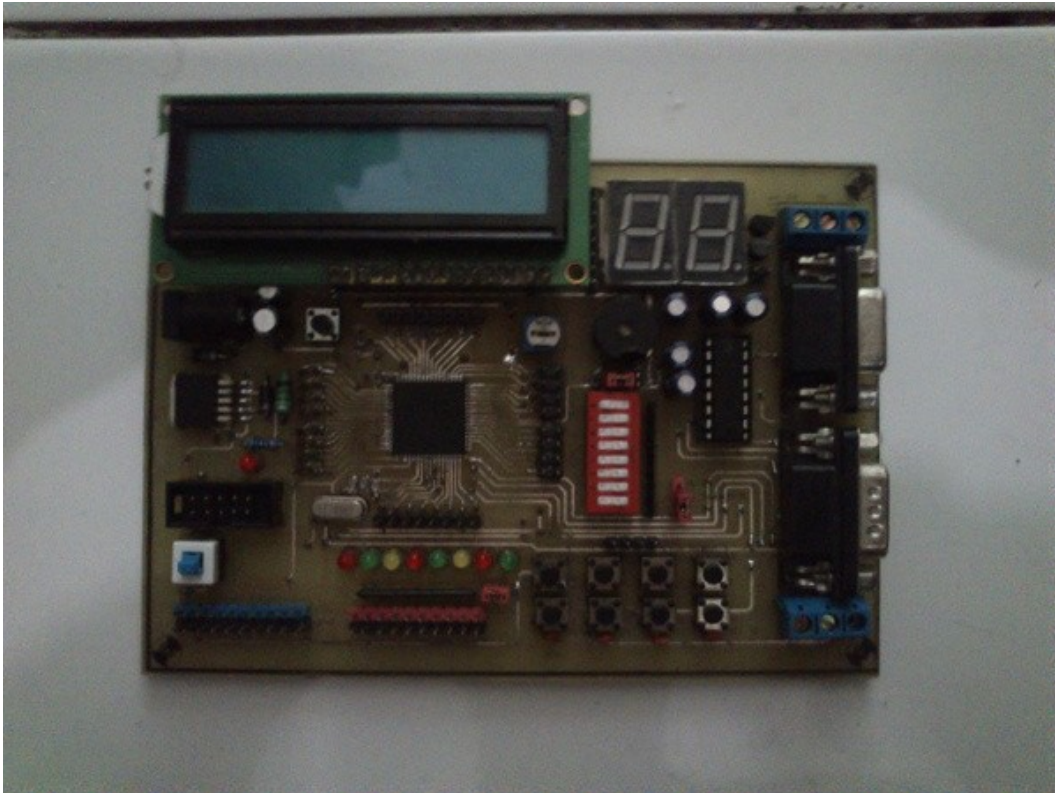
Sebenarnya 4 langkah ini sama dengan prinsip pemrograman di Windows. Hanya saja semuanya Anda lakukan dengan IDE seperti Atmel Studio sehingga tidak tampak langkah-langkahnya. Untuk tidak disalahpahami, IDE di Linux semisal Codeblocks juga canggih (terdapat fitur syntax highlighter + code completion) layaknya Anda peroleh di Atmel Studio sehingga mudah menulis kode C. Tulisan ini hanya berfokus pada langkah ke-4.

## 2. Hardware yang Dibutuhkan

1. Perangkat kabel USBASP (bukan USBISP MKII).
2. Perangkat minimum system Atmega128.
3. Laptop dengan sistem operasi Linux (saya pakai Ubuntu).



*Illustration 1: USBASP*



*Illustration 2: Minimum System Atmega128*

### **3. Software yang Dibutuhkan**

1. avrdude
2. avr-gcc
3. binutils
4. avr-libc<sup>[3]</sup>
5. Terminal
6. Penyunting teks apa pun (sebagai ganti Atmel Studio untuk menulis kodenya)

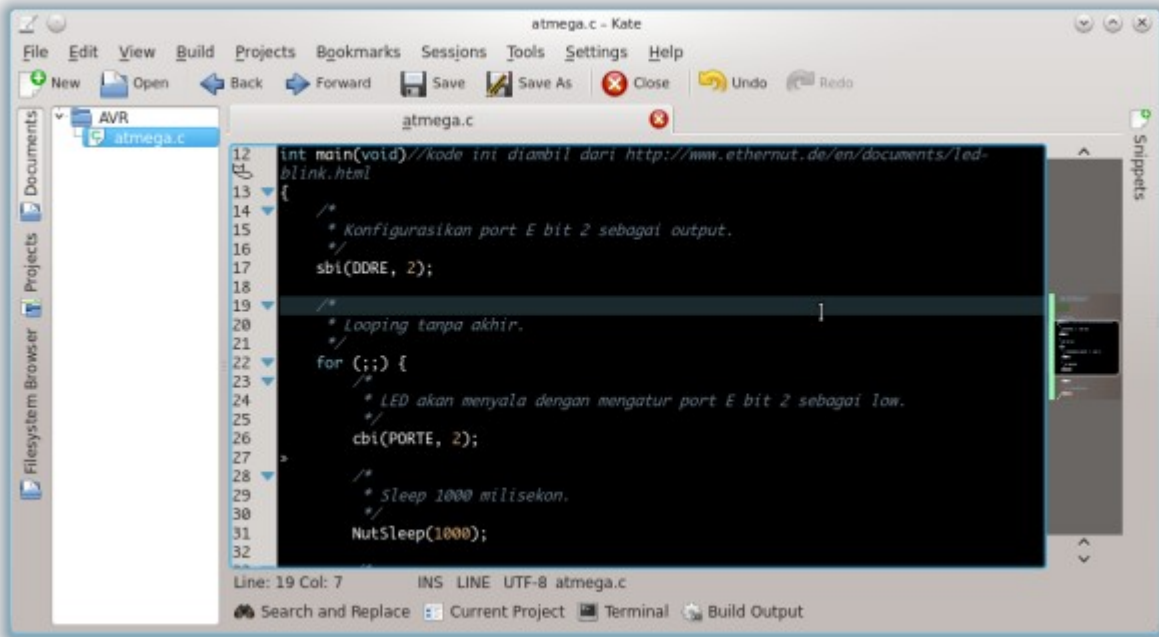


Illustration 3: Penyunting teks Kate untuk pemrograman AVR

Perlu saya tuliskan di sini biar kita tidak salah kaprah. Yang dilakukan oleh Atmel Studio sendiri sebetulnya hanyalah, hanyalah, memanggil **avrdude** untuk membakar ke chip Atmega. Jadi, sebetulnya yang diperlukan itu avrdude bukan Atmel Studio-nya. Lagipula, Atmel Studio sendiri hanya memakai **avr-gcc** untuk melakukan kompilasi kode C ke OBJ/ELF dan **avr-objcopy** linking ke HEX. Walhamdulillah, justru avrdude dan avr-gcc itu asalnya dibuat untuk Linux. Mereka open source software. Maka semestinya bisa juga kita lakukan hal yang sama. Tidak harus dengan Windows. Karena di sini saya hanya menunjukkan cara membakar, saya tidak menjelaskan penulisan kode pada penyunting teks. Namun saya jelaskan bahwa saya bisa memakai aplikasi Gedit, Geany, atau Kate yang terinstal di Ubuntu saya untuk melakukannya. Anda punya? Pakailah.

Seluruh kebutuhan di atas dapat diperoleh dalam 1 perintah saja di Ubuntu yaitu ***sudo apt-get install avrdude avr-gcc avr-libc binutils***.

## 4. Spesifikasi Sistem

Supaya tidak kebingungan, maka saya sebut di sini spesifikasi mesin saya.

- Laptop ASUS X44C 32 bit dengan satu port USB 2.0 dan satu port USB 3.0.
- Ubuntu 12.04 32 bit.
- Terinstal aplikasi IDE Codeblocks, CodeLite, Geany, Eclipse, serta penyunting teks mutakhir Kate di Ubuntu saya.
- Terinstal GNOME Terminal dan Konsole di Ubuntu saya.

## 5. Langkah-Langkah Download HEX ke Atmega

1. Tancapkan kabel USBASP pada port USB 2.0.
2. Tancapkan kabel USBASP sisi ujung pada *minimum system* Atmega.
3. Jalankan perintah avrdude.

## 6. Perintah untuk *Download* HEX

```
# avrdude -c usbasp -P usb -p atmega128 -U flash:w:namaprogramnya.hex
```

Penjelasan Perintah<sup>[5]</sup>

Perlu diketahui sebelumnya. Pertama, perintah inilah yang digunakan oleh Atmel Studio untuk melakukan pembakaran. Kedua, karena Anda menggunakan Linux, Anda harus melakukannya dalam mode root (tanda # di atas). Perintah untuk menjadi root adalah su. Sekarang, mari perhatikan apa maksud perintah di atas.

**avrdude** : nama perintah untuk membakar kode .hex ke chip Atmega

**-c usbasp** : menentukan kabel downloader yang Anda gunakan. Karena kita pakai USBASP, maka kita juga tentukan usbasp. Awas, besar kecil huruf harus diperhatikan. Jika salah menentukan opsi ini, avrdude akan salah juga mengenali jalan mana yang harus ditempuh untuk mengirim .hex ke Atmega. Pilihan nama kabel untuk opsi ini sangat banyak. Jika Anda memakai kabel jenis lain semacam USBISP MKII, Anda bisa membaca *datasheet* nama opsinya dengan ketik perintah man avrdude di Terminal.

**-P usb** : menentukan jalur dari kabel downloader yang tertancap di komputer Anda. Karena saya menggunakan USBASP dan ia menggunakan USB, maka saya tulis usb. Mungkin seumur hidup Anda tidak akan mengubah opsi ini karena selalu menggunakan USB.

**-p atmega128** : menentukan chip yang akan ditulisi kode .hex. Pilihan nama chip untuk opsi ini sangat banyak. Silakan ketik perintah man avrdude.

**-U flash:w:namaprogramnya.hex** : menentukan perilaku dari kegiatan download yang kita lakukan, apakah sekadar dikirim lalu ditolak begitu saja oleh Atmega atau melakukan penulisan (flash) ke dalamnya. Opsi flash maksudnya melakukan flash ke dalam chip sedangkan w maksudnya write (menulis) kode .hex. Ini opsi paling penting dalam perintah avrdude ini. Jika Anda tidak menentukannya, maka usaha Anda sia-sia. Jika ingin mempelajari perintah avrdude lebih lanjut, silakan ketik perintah man avrdude di Terminal.

# 7. Hasil Download

Berikut ini bukti yang saya salin dari Konsole. Lihat, sama persis dengan Atmel Studio Anda. Skrinshot Konsole dan foto LED yang menyala juga saya sertakan.

```
Reading | ##### | 100% 0.00s
avrdude: Device signature = 0x1e9702
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: warning: cannot set sck period. please check for usbasp firmware update.
avrdude: reading input file "namaprogramnya.hex"
avrdude: input file namaprogramnya.hex auto detected as Intel Hex
avrdude: writing flash (4306 bytes):

Writing | ##### | 100% 1.70s
avrdude: 4306 bytes of flash written
avrdude: verifying flash memory against namaprogramnya.hex:
avrdude: load data flash data from input file namaprogramnya.hex:
avrdude: input file namaprogramnya.hex auto detected as Intel Hex
avrdude: input file namaprogramnya.hex contains 4306 bytes
avrdude: reading on-chip flash data:

Reading | ##### | 100% 1.49s
avrdude: verifying ...
avrdude: 4306 bytes of flash verified

avrdude: safemode: Fuses OK

avrdude done. Thank you.

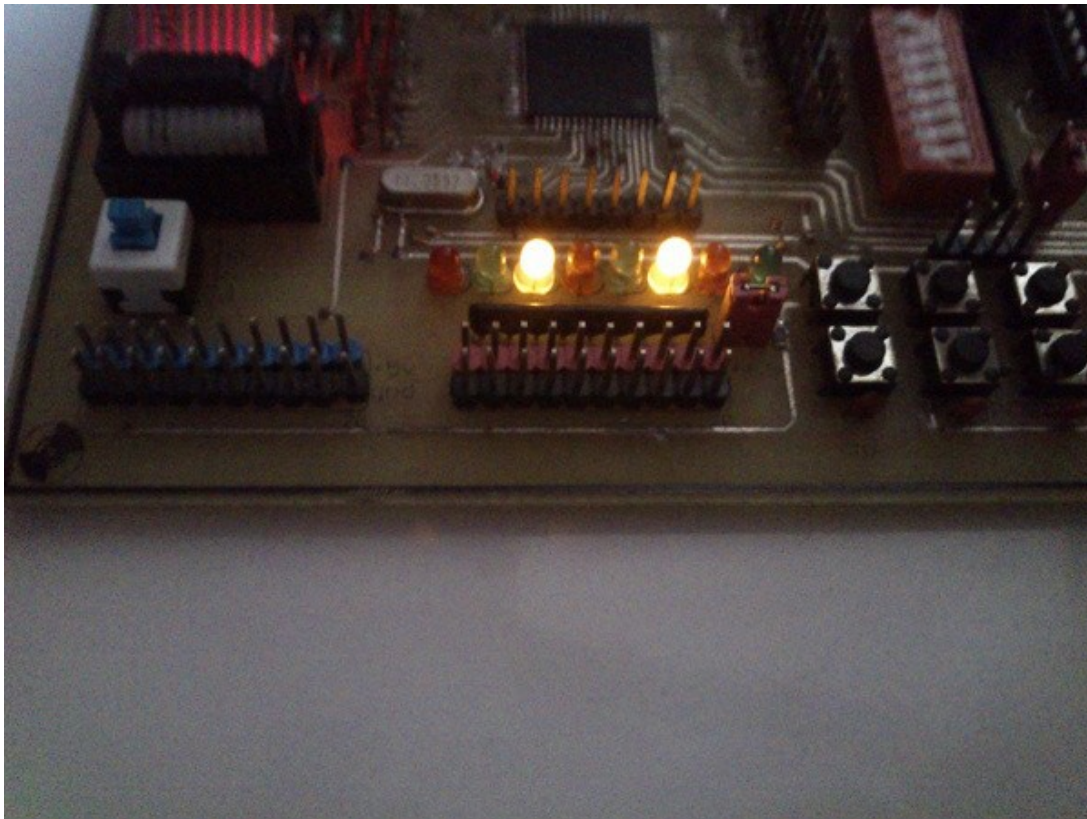
root@master:/home/master/Publik/AVR#
```

```
AVR: avrdude - Konsole
File Edit View Bookmarks Settings Help
AVR: avrdude
root@master:/home/master/Publik/AVR# avrdude -c usbasp -P usb -p atmega128 -U flash:w:nanaprogramnya.hex
avrdude: warning: cannot set sck period. please check for usbasp firmware update.
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s
avrdude: Device signature = 0x1e9702
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: warning: cannot set sck period. please check for usbasp firmware update.
avrdude: reading input file "nanaprogramnya.hex"
avrdude: input file nanaprogramnya.hex auto detected as Intel Hex
avrdude: writing flash (4306 bytes):

Writing | ##### | 97% 2.51s
```

*Illustration 4: Proses pembakaran .hex dengan avrdude*



*Illustration 5: LED Menyala*

## 8. Rangkuman

1. Tancapkan kabel USBASP.
2. Jalankan perintah *flashing* avrdude.

Proses bakar HEX di Linux secara garis besar adalah demikian.

## 9. Tentang Tulisan Ini

Tulisan ini adalah catatan pribadi saya ketika membakar .hex ke Atmega dengan avrdude. Keberhasilan tersebut terjadi setelah mencoba perintah avrdude yang disebutkan di salah satu laman forum AVR Freaks<sup>[6]</sup>. Sebelum keberhasilan ini, saya menganggap mustahil melakukan pemrograman AVR di Linux karena semua orang selalu menggunakan Windows. Namun pada akhirnya, tulisan ini adalah *proof of concept* untuk saya sendiri bahwa Linux dapat menggantikan Windows untuk melakukannya. Tulisan ini dibuat dengan harapan kita beralih dari penggunaan aplikasi bajakan dan beralih kepada yang legal walaupun banyak tantangan karena di situ terdapat barakah. Tulisan ini ditulis seluruhnya di Ubuntu 12.04 dengan menggunakan FOSS secara total baik Android untuk mengambil foto maupun transfer datanya ke Ubuntu saya serta memakai WordPress untuk menyusunnya. Tulisan ini disusun tanpa melibatkan Windows dan aplikasinya sedikit pun. Semoga tulisan ini menjadi pijakan awal Anda untuk melakukan pemrograman AVR lebih lanjut di Linux.

## 10. Ucapan Terima Kasih

Terima kasih untuk Akang Muhammad Arif Wicaksono yang telah meminjamkan semua hardware plus hape Sony Ericsson untuk mengambil foto.

## 11. Referensi

[1] <http://jlectronique.org/ProgrammateurAVREN.htm>

[2] <http://extremeelectronics.co.in/avr-tutorials/gui-software-for-usbasp-based-usb-avr-programmers/>

[3] <http://www.tldp.org/HOWTO/Avr-Microcontrollers-in-Linux-Howto/x207.html>

[4] <http://www.ethernut.de/en/documents/led-blink.html>

[5] <http://www.micahcarrick.com/tutorials/avr-microcontroller-tutorial/getting-started.html>

[6] <http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&p=561733>



## 12. Tentang Dokumen Ini

Dokumen ini adalah versi PDF dari tulisan asli <http://malsasa.wordpress.com/2014/01/11/pengantar-pemrograman-avr-di-linux-avrdude-hex-usbasp/>. 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.

## 13. 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.

---

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