

BAB 2

TINJAUAN PUSTAKA

2.1. Kondisi Praktikum Sistem Mikroprosesor di Jurusan Teknik Elektro

Praktikum sistem mikroprosesor ini juga meliputi proyek pembuatan *file program* menggunakan perangkat lunak *Keil* di dalam sistem operasi *Windows*, namun dikarenakan pada praktikum ini, masih menggunakan sistem operasi *Windows* dan perangkat lunak *Keil* yang bajakan, maka masalah tersebut dapat diatasi dengan menggunakan perangkat lunak *Kontrollerlab* yang berjalan diatas sistem operasi yang *opensource* yaitu *Ubuntu*.

Pada proses pembuatan file program, bahasa program yang digunakan yaitu bahasa program level rendah yaitu *assembly*, sejak berkembangnya bahasa program level menengah menuju ke tinggi yaitu bahasa C, karena mendekati bahasa yang dimengerti manusia dan banyak pemakai yang memanfaatkan bahasa program tersebut, maka bahasa C digunakan sebagai pengganti bahasa *assembly*.

Port yang digunakan sebagai kabel penghubung dari perangkat lunak ke perangkat keras pada praktikum ini yaitu *port serial* dan *paralel*, sejak berkembangnya *port USB (Universal Serial Bus)* yang lebih baik dalam kecepatan pengiriman dan kepraktisannya, maka *port USB* digunakan sebagai pengganti *port serial* dan *paralel*.

Dalam simulasinya, perangkat keras yang dipakai sebagai alat uji pada praktikum ini yaitu memori EEPROM (*Electrically Erasable Programmable*

Read Only Memory) mikrokontroler keluarga MCS51 (*MicroChipSeries51*) salah satunya IC AT89C51 dan AT89S51, sejak berkembangnya IC Atmel Mega AVR (*Alf (Egil Bogen) and Vegard (Wollan)'s Risc processor*) seperti ATMega8535 yang lebih unggul dalam kemudahan pemakaian, memori yang besar dan harganya yang terjangkau, maka AVR digunakan sebagai pengganti dari mikrokontroler MCS51.

2.2. Perbandingan Mikrokontroler MCS51 dengan Mikrokontroler AVR

Komponen *chip* mikrokontroler yang sangat terkenal dan sering digunakan untuk membangun sebuah perangkat elektronik kendali seperti robotika yakni, mikrokontroler jenis keluarga MCS51 yaitu AT89C51, AT89S51 dan mikrokontroler AVR yaitu ATMega8535, ATMega8, ATMega16. Berikut dibawah ini merupakan perbandingan fitur dan kemampuan antara mikrokontroler keluarga MCS51 dengan mikrokontroler AVR⁽¹⁴⁾:

1. Kecepatan

Untuk mengeksekusi 1 buah instruksi mikrokontroler AVR hanya memerlukan 1 clock saja, sedangkan MCS51 dalam hal ini AT89S51 memerlukan 12 clock. Jadi dalam hal ini AVR lebih cepat dari MCS51

2. Bahasa Pemrograman

Bahasa pemrograman yang digunakan oleh AVR adalah bahasa C (salah satu bahasa program yang sering digunakan untuk pemrograman mikrokontroler), sehingga lebih mudah dipelajari daripada bahasa

assembly. Bahasa *assembly* merupakan bahasa program yang lebih dekat ke bahasa mesin sehingga lebih cepat eksekusinya. Namun, pihak Atmel telah mengoptimasi bahasa C yang digunakan untuk AVR sehingga lebih cepat dan efisien.

3. Memori yang lebih besar

Untuk keluarga mikrokontroler AVR sudah banyak yang memiliki memori internal yang relatif besar. Misal untuk seri ATmega16: 16 Kb (*Flash memory*), 512 Bytes (EEPROM), 1 Kb (RAM), sedangkan untuk AT89S51 memiliki 4 Kb (*Flash memory*), 128 bytes (RAM). Dari sini terlihat kapasitas memori AVR lebih besar daripada AT89S51. AVR juga memiliki EEPROM internal sehingga tidak perlu memakai EEPROM eksternal seperti pada AT89S51.

4. Efisiensi Perangkat Keras

Dengan AVR, dapat mengurangi penggunaan komponen pendukung seperti EEPROM eksternal (baca poin yang ke-3 tentang memori) bahkan tidak perlu lagi menggunakan EEPROM eksternal. Untuk sumber clock pada AVR telah tersedia secara terintegrasi, karena didalamnya sudah terdapat XTAL yang bisa diaktifkan sehingga tidak perlu memakai XTAL tambahan lagi. Untuk nilainya bisa dikalibrasi sendiri apakah 4 Mhz, 1 Mhz atau lainnya. Sebagai catatan untuk kebutuhan akurasi XTAL eksternal masih perlu diatur. Akan tetapi jika akurasi tidak terlalu signifikan, cukup XTAL internal saja yang digunakan.

Bagi yang memerlukan ADC untuk konversi sinyal analog ke digital tidak perlu lagi memakai ADC eksternal (ADC 0804) karena untuk AVR seri ATmega16 sudah ada ADC internal 10 bit.

5. Fitur-Fitur Tambahan

Mikrokontroler AVR memiliki fitur-fitur tambahan lainnya yang tidak terdapat pada mikrokontroler MCS51. Diantaranya :

- a. RTC dengan *oscilator* terpisah.
- b. PWM (*Pulse Width Modulation*).
- c. ADC 10 bit *internal*.
- d. *Master / slave SPI Serial interface*.
- e. *On chip analog comparator*.

2.3. Pengenalan Mikrokontroler AVR⁽¹²⁾

Pada saat ini penggunaan mikrokontroler dapat kita temui pada berbagai peralatan, misalnya peralatan yang terdapat dirumah, seperti telepon digital, *microwave oven*, televisi, mesin cuci, sistem keamanan rumah, PDA, dan lain-lain yang semisalnya. Mikrokontroler dapat kita gunakan untuk berbagai aplikasi misalnya untuk pengendalian, otomasi industri, akuisisi data, telekomunikasi dan lain-lain. Keuntungan menggunakan mikrokontroler yaitu harganya murah, dapat diprogram berulang kali, dan dapat kita program sesuai dengan keinginan.

Atmel AVR adalah jenis mikrokontroler yang paling sering dipakai dalam bidang elektronika dan instrumentasi. Mikrokontroler AVR ini memiliki arsitektur RISC (*Reduce Instruction Set Computing*) delapan bit, di mana semua instruksi dikemas dalam kode 16-bit (16 bits word) dan sebagian besar instruksi dieksekusi

dalam 1 (satu) siklus *clock*, maka dari itu AVR memiliki kecepatan eksekusi program lebih cepat daripada mikrokontroler jenis lainnya.

Nama AVR sendiri berasal dari "*Alf (Egil Bogen) and Vegard (Wollan)'s Risc processor*" dimana *Alf Egil Bogen* dan *Vegard Wollan* adalah dua penemu berkebangsaan *Norwegia* yang menemukan mikrokontroler AVR yang kemudian diproduksi oleh *Atmel*.

Secara umum, AVR dapat dikelompokkan dalam 10 kelas, yaitu:

1. Keluarga AVR Otomotif
2. Keluarga AVR *Z-Link*
3. Keluarga AVR Manajemen Batere
4. Keluarga AVR CAN
5. Keluarga AVR LCD
6. Keluarga AVR Pencahayaan
7. Keluarga TinyAVR
8. Keluarga MegaAVR
 - a. ATmega8535
 - b. ATmega16
 - c. ATmega32
 - d. ATmega8
9. Keluarga AVR USB
10. Keluarga XMEGA

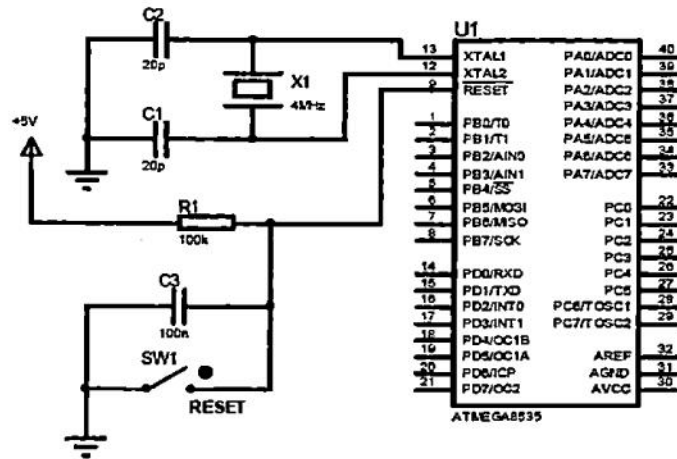
lunak yang bernama *AVRStudio*. Selain itu, terdapat beberapa *cross compiler* dari pihak ketiga yang dapat digunakan seperti *CodeVisionAVR Compiler* atau *ICC AVR* (id.wikipedia.org, 2009). Perangkat lunak tersebut diatas dijalankan pada sistem operasi *Windows* namun tidak gratis.

ATMega8535 merupakan salah satu mikrokontroler 8 bit buatan *Atmel* untuk keluarga AVR yang diproduksi secara massal pada tahun 2006. Karena merupakan keluarga AVR, maka *ATMega8535* juga menggunakan arsitektur RISC.

2.3.1. Keterangan Singkat AVR *ATMega8535*⁽²⁾

Secara singkat *ATMega8535* memiliki beberapa kemampuan:

1. Sistem mikrokontroler 8 bit berbasis RISC (*Reduce Instruction Set Computing*) dengan kecepatan maksimal 16 MHz.
2. Memiliki memori flash 8 KB, SRAM sebesar 512 byte dan EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 byte.
3. Memiliki ADC (Pengubah analog-ke-digital) *internal* dengan ketelitian 10 bit sebanyak 8 saluran.
4. Memiliki PWM (*Pulse Wide Modulation*) *internal* sebanyak 4 saluran.
5. Portal komunikasi serial (USART [*Universal Serial Asynchronous Receiver-Transmitter*]) dengan kecepatan maksimal 2,5 Mbps.
6. Enam pilihan *mode sleep*, untuk menghemat penggunaan daya listrik.



Gambar 2.1. Rangkaian Dasar ATmega8535

2.3.2. Konfigurasi Pin Mikrokontroler AVR ATmega8535⁽²⁾

ATmega8535 memiliki 40 pin untuk model PDIP, dan 44 pin untuk model TQFP dan PLCC. Nama-nama pin pada mikrokontroler ini adalah :

1. VCC untuk tegangan pencatu daya positif.
2. GND untuk tegangan pencatu daya negatif.
3. *PortA* (PA0 - PA7) sebagai *port Input/Output* dan memiliki kemampuan lain yaitu sebagai *input* untuk ADC.
4. *PortB* (PB0 - PB7) sebagai *port Input/Output* dan juga memiliki kemampuan yang lain.
5. *PortC* (PC0 - PC7) sebagai *port Input/Output* untuk ATmega8535.
6. *PortD* (PD0 - PD7) sebagai *port Input/Output* dan juga memiliki kemampuan yang lain.
7. *RESET* untuk melakukan *reset* program dalam mikrokontroler.
8. XTAL1 dan XTAL2 untuk input pembangkit sinyal *clock*.
9. AVCC untuk pin masukan tegangan pencatu daya untuk ADC.
10. AREF untuk pin tegangan referensi ADC.

| | | | |
|-----------------|----|----|-------------|
| (XCK/TO) PB0 | 1 | 40 | PA0 (ADC0) |
| (T1) PB1 | 2 | 39 | PA1 (ADC1) |
| (INT2/AI10) PB2 | 3 | 38 | PA2 (ADC2) |
| (OC0/AI11) PB3 | 4 | 37 | PA3 (ADC3) |
| (SS) PB4 | 5 | 36 | PA4 (ADC4) |
| (MOSI) PB5 | 6 | 35 | PA5 (ADC5) |
| (MISO) PB6 | 7 | 34 | PA6 (ADC6) |
| (SCK) PD7 | 8 | 33 | PA7 (ADC7) |
| RESET | 9 | 32 | AREF |
| VCC | 10 | 31 | GND |
| GND | 11 | 30 | AVCC |
| XTAL2 | 12 | 29 | PC7 (TOSC3) |
| XTAL1 | 13 | 28 | PC8 (TOSC1) |
| (RXD) PD0 | 14 | 27 | PC5 |
| (TXD) PD1 | 15 | 26 | PC4 |
| (INT0) PD2 | 16 | 25 | PC3 |
| (INT1) PD3 | 17 | 24 | PC2 |
| (OC1B) PD4 | 18 | 23 | PC1 (SDA) |
| (OC1A) PD5 | 19 | 22 | PC0 (SCL) |
| (ICP1) PD6 | 20 | 21 | PD7 (OC2) |



Gambar 2.2. Keterangan Pin pada Mikrokontroler AVR ATmega8535⁽¹⁴⁾

2.3.3. Pengisian Program pada Mikrokontroler AVR

Untuk melakukan pemrograman dalam mikrokontroler AVR, terdiri atas *compiler* dan *downloader*. *Compiler*, yaitu aplikasi yang fungsinya untuk menerjemahkan program baik yang ditulis dalam bahasa-bahasa pemrograman tingkat tinggi (seperti C, Basic, Pascal) maupun yang ditulis dalam bahasa pemrograman tingkat rendah (seperti *assembly*) menjadi kode-kode mesin (dalam bentuk *hexadecimal*). *Downloader* adalah aplikasi yang digunakan untuk merekam kode-kode mesin yang telah dikonversi sebelumnya oleh *Compiler*, untuk selanjutnya dimasukkan ke dalam memori mikrokontroler.⁽¹⁴⁾

Berdasarkan pengetahuan diatas, maka dalam *platform* apa saja sebenarnya dimungkinkan untuk membuat aplikasi yang bisa berkomunikasi dengan mikrokontroler, asalkan kedua fitur tadi tersedia (*Compiler* dan *Downloader*).⁽¹⁴⁾

Atmel telah menyediakan perangkat lunak khusus yang dapat diunduh dari website resmi *Atmel*, perangkat lunak tersebut adalah *AVRStudio*, perangkat lunak

ini menggunakan bahasa *assembly* sebagai bahasa perantaranya. Selain *AVRStudio*, ada beberapa perangkat lunak pihak ketiga yang dapat digunakan untuk membuat program pada *AVR*. Perangkat lunak dari pihak ketiga ini menggunakan bahasa pemrograman tingkat tinggi seperti bahasa C, *Java*, atau *Basic*. Untuk melakukan pemindahan dari komputer ke dalam *chip*, dapat digunakan beberapa cara seperti menggunakan kabel JTAG atau menggunakan STK buatan *Atmel*.⁽⁴⁾

Selain menggunakan *Windows Operating System*, ternyata pemrograman mikrokontroler AVR juga dapat dilakukan pada *Linux Operating System*. Pada *Linux* juga sudah tersedia perangkat lunak yang dapat digunakan untuk melakukan pemrograman, salah satunya adalah perangkat lunak *Kontrollerlab*. *Kontrollerlab* merupakan perangkat lunak pemrograman *Linux* berbasis GUI (*Graphic User Interface*) yang dapat digunakan dengan mudah sebagaimana layaknya kita menggunakan perangkat lunak pemrograman mikrokontroler AVR di *Windows*. *Kontrollerlab* sudah menyediakan *feature-feature* yang sangat lengkap.⁽⁴⁾

Selain menggunakan perangkat lunak berbasis GUI, pemrograman mikrokontroler AVR di *Linux* juga dapat dilakukan dengan menggunakan *console* (*command prompt* pada sistem operasi *windows/DOS*) yaitu *konsole* untuk versi KDE desktop dan *terminal* untuk versi GNOME desktop. Salah satu kelebihan pemrograman dengan menggunakan konsol adalah kita dapat lebih memahami perintah-perintah yang digunakan dalam proses pemrograman mikrokontroler AVR tersebut walaupun terlihat sulit untuk dilakukan, karena banyak perintah-

perintah di dalam konsol atau *terminal* yang harus dipelajari dan dihapalkan.⁽⁴⁾

2.4. Perbandingan Antara Sistem Operasi *Windows* dengan *Linux Ubuntu*⁽⁶⁾

Sistem operasi dapat digunakan sebagai alat bantu pembuatan program pada mikrokontroler, yang sering digunakan adalah *Windows*, namun dapat juga menggunakan yang *opensource* yakni *Linux* seperti *Ubuntu*. Perbedaan mendasar antara sistem operasi *Windows* dan *Linux* adalah sebagai berikut :

1. *Windows* lebih rentan terhadap *virus* dan *malware* sedangkan *Linux* masih tergolong aman dari *virus*.
2. *Windows* lebih banyak membutuhkan *resource* memori ketika menjalankan beberapa aplikasi, sedangkan *Linux* tidak membutuhkan *resource* memori yang besar ketika menjalankan lebih dari dua aplikasi secara bersamaan, karena *Linux* sudah terdapat memori *swap* sebagai penambah memori komputer agar bisa lebih cepat memproses data.
3. *Windows* dan perangkat lunaknya bersifat *proprietary* (berpemilik) dan komersil sedangkan *Linux* gratis dan *opensource*, pengguna komputer hanya tinggal *download* perangkat lunak-perangkat lunak *Linux* di internet jika koneksi internetnya baik.

2.4.1. Pengenalan sistem operasi *Windows*⁽¹²⁾

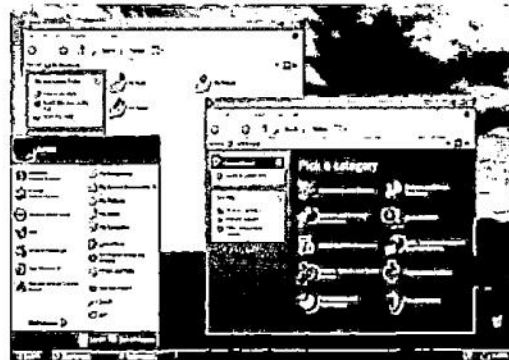
Windows XP adalah jajaran sistem operasi berbasis grafis yang dibuat oleh *Microsoft* untuk digunakan pada komputer pribadi, yang mencakup komputer rumah dan *desktop* bisnis, laptop, dan pusat media (*Media Center*). Nama "XP"

adalah kependekan dari "*Experience*". *Windows XP* merupakan penerus *Windows 2000 Professional* dan *Windows Me*, dan merupakan versi sistem operasi *Windows* pertama yang berorientasi konsumen yang dibangun di atas *kernel* dan arsitektur *Windows NT*. *Windows XP* pertama kali dirilis pada 25 Oktober 2001, dan lebih dari 400 juta salinan instalasi digunakan pada Januari 2006, menurut perkiraan seorang analis IDC.

Windows XP digantikan oleh *Windows Vista*, yang dirilis untuk pengguna *volume license* pada 8 November 2006, dan di seluruh dunia untuk masyarakat umum pada tanggal 30 Januari 2007. Banyak *Original Equipment Manufacturer* (OEM) dan juga penjual ritel menghentikan produksi perangkat dengan *Windows XP* pada tanggal 30 Juni 2008. *Microsoft* sendiri terus menjual *Windows XP* melalui *Custom-built PC* (OEM kecil yang menjual komputer rakitan) sampai dengan 31 Januari 2009.



Gambar 2.3. Lambang *Windows XP* dari *Microsoft*



Gambar 2.4. Tampilan *Desktop Windows XP*

Windows XP sebelumnya dikenal dengan kode sandi "*Whistler*", yang mulai dikembangkan oleh para pengembang *Microsoft* pada pertengahan tahun 2000-an. Bersamaan dengan proyek ini, *Microsoft* juga tengah menggarap proyek *Windows* generasi baru penerus *Windows* Me (*Millennium Edition*) yang dinamakan dengan kode sandi "*Windows Neptune*" yang diproyeksikan sebagai "*Windows* NT versi rumahan".

Setelah *Windows* ME dianggap kurang sukses menyaingi kesuksesan *Windows* 98, *Microsoft* pun akhirnya memutuskan untuk mengawinkan dua buah sistem operasi *Windows* tersebut (sistem operasi berbasis *Windows* NT dan sistem operasi berbasis *Windows* 9x) ke dalam sebuah produk. Itulah yang kita kenal sekarang dengan *Windows* XP.⁽¹²⁾

2.4.2. Pengenalan Sistem Operasi *Linux* dan *Ubuntu*⁽¹²⁾

Linux merupakan salah satu sistem operasi alternatif pengganti sistem operasi *Windows*, sistem operasi *Linux* dan perangkat lunaknya biasa disebut *opensource*, dimana kita dapat dengan leluasa melihat, mempelajari bahkan juga memperbolehkan untuk mengubah serta memperbaiki sistem operasi tersebut, semuanya legal. Selain itu *Linux* juga gratis dan dapat diperbanyak tanpa harus was-was disebut sebagai pembajak perangkat lunak. Hal mendasar inilah yang menjadikan *Linux* berkembang dengan sangat pesat dewasa ini.

Linux awalnya merupakan suatu proyek hobi yang dikerjakan *Linus Torvalds*, seorang mahasiswa Universitas *Helsinki, Finlandia*. Pada saat itu *Linus Torvalds* menggunakan suatu sistem operasi berbasis *Unix* yang dinamakan *Minix*

yang dibuat oleh *Andy Tanenbaum*. Ia merasa bahwa *Minix* memiliki banyak kelemahan. Oleh karena itu ia kemudian membuat suatu *cloning* dari sistem operasi berbasis *Unix* tersebut dan mempublikasikan *release* pertama pada bulan Oktober 1991, yaitu *release 0.10* dengan nama *Linux* yang sebagian besar dibuat dengan menggunakan bahasa C.

Berikut adalah beberapa keistimewaan yang dimiliki oleh *Linux* :⁽⁶⁾

1. *Kernel Linux* dibangun untuk prosesor *intel 80386 mode* terproteksi yang dapat digunakan *multitasking*.
2. *Multiuser*, *Linux* mengizinkan *shared executables* dimana bila ada lebih dari satu aplikasi dijalankan, baik satu pengguna menjalankan *task* yang sama atau banyak pengguna menjalankan *task* yang sama, semua *task* dapat membagi pakai memori yang sama. Metode ini dinamakan *copy-on-write pages* yang membuat penggunaan RAM menjadi efisien.
3. Full 32 bit.
4. Sebagian besar ditulis dalam bahasa C.
5. Mendukung berbagai macam bahasa program seperti *Java, Python, Perl, C, C++* dan lain-lain.
6. Mendukung implementasi lengkap untuk TCP/IP.
7. Mengenali berbagai macam file sistem seperti FAT, FAT32, NTFS, MINIX-1, ISO9660 dan lain-lain.
8. Mendukung hampir semua fasilitas yang dimiliki oleh sistem operasi *UNIX*

9. Mendukung perangkat lunak GNU (*Gnu Not Unix*).
10. Memiliki fasilitas grafik yang disebut GUI (*Graphical User Interface*) yang dipergunakan untuk *X-Window* (windownya *Linux*) dari MIT.
11. *Swap space* (memori bayangan) yang dapat digunakan jika komputer mempunyai memori yang terbatas.
12. Dukungan terhadap *dynamically shared library* yang dapat memperkecil ukuran dari program.
13. Keamanan sistem operasi *Linux* lebih tinggi, karena setiap masuk pada bagian yang sensitif harus disertai *password*.

Ubuntu merupakan salah satu distribusi *Linux* yang berbasiskan *Debian*.

Proyek *Ubuntu* resmi disponsori oleh *Canonical.Ltd* yang merupakan perusahaan milik seorang kosmonot asal Afrika Selatan, Mark Shuttleworth. Nama *Ubuntu* diambil dari nama sebuah konsep ideologi di Afrika Selatan, "*Ubuntu*" berasal dari bahasa kuno Afrika, yang berarti "rasa perikemanusiaan terhadap sesama manusia". Tujuan dari distribusi *Linux Ubuntu* adalah membawa semangat yang terkandung di dalam Filosofi *Ubuntu* ke dalam dunia perangkat lunak. *Ubuntu* adalah sistem operasi lengkap berbasis *Linux*, tersedia secara bebas dan mempunyai dukungan baik yang berasal dari komunitas maupun tenaga ahli profesional.⁽¹²⁾



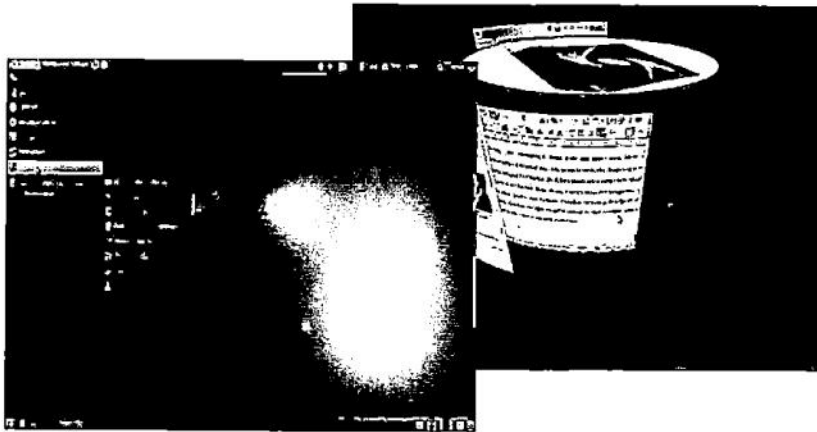
Gambar 2.5. *Ubuntu Linux* Beserta Logonya.

Ubuntu mempunyai filosofi sebagai berikut :⁽¹¹⁾

1. bahwa perangkat lunak harus tersedia dengan bebas biaya.
2. bahwa perangkat lunak tersebut harus dapat digunakan dalam bahasa lokal masing-masing dan untuk orang-orang yang mempunyai keterbatasan fisik, dan
3. bahwa pengguna harus mempunyai kebebasan untuk mengubah perangkat lunak sesuai dengan apa yang mereka butuhkan.

Perihal kebebasan inilah yang membuat *Ubuntu* berbeda dari perangkat lunak berpemilik (*proprietary*), bukan hanya peralatan yang dibutuhkan tersedia secara bebas biaya, tetapi kita juga mempunyai hak untuk memodifikasi perangkat lunak sampai perangkat lunak tersebut bekerja sesuai dengan yang diinginkan.

Setiap rilis mempunyai nama kode dan nomor versi. Nomor versi berdasarkan tahun dan bulan dari rilis. Sebagai contoh, rilis *Ubuntu* yang pertama, 4.10, dirilis tanggal 20 Oktober 2004. Rilis *ubuntu* keluar setiap 6 bulan sekali tiap bulan April dan Oktober. Rilis *ubuntu* biasanya terdiri dari berbagai edisi, yaitu edisi *Desktop*, *Server*, dan *Netbook*. Perbedaan mendasar pada ketiganya adalah pada versi *desktop* terdapat tampilan *desktop manager* sedangkan pada edisi *server* tidak ada, hanya *command prompt* saja.



Gambar 2.6. Desktop Ubuntu versi 10.04 LTS dan Desktop effect Compiz Fusion.

Tabel 2.1. Daftar rilis Ubuntu yang telah dirilis.

| Versi | Tanggal Rilis | Nama kode | Sub Versi | Didukung sampai |
|---------------------------------|-----------------|-------------------------|----------------|---|
| 4.10 | 20 Oktober 2004 | <i>Warty Warthog</i> | <i>Sounder</i> | 30 April 2006 |
| 5.04 | 8 April 2005 | <i>Hoary Hedgehog</i> | <i>Array</i> | 31 Oktober 2006 |
| 5.10 | 13 Oktober 2005 | <i>Breezy Badger</i> | <i>Colony</i> | 13 April 2007 |
| 6.06 LTS (Long Term Support) | 1 Juni 2006 | <i>Dapper Drake</i> | <i>Flight</i> | Juni 2009 (desktop) Juni 2011 (server) |
| 6.10 | 26 Oktober 2006 | <i>Edgy Eft</i> | <i>Knot</i> | April 2008 |
| 7.04 | 19 April 2007 | <i>Feisty Fawn</i> | <i>Herd</i> | Oktober 2008 |
| 7.10 | 18 Oktober 2007 | <i>Gutsy Gibbon</i> | <i>Tribe</i> | April 2009 |
| 8.04 LTS (Long Term Support) | 21 April 2008 | <i>Hardy Heron</i> | <i>Siege</i> | April 2011 (desktop) April 2013 (server) |
| 8.10 | 30 Oktober 2008 | <i>Intrepid Ibex</i> | | |
| 9.04 | 04/23/09 | <i>Jaunty Jackalope</i> | | |
| 9.10 | 10/29/09 | <i>Karmic Koala</i> | | |
| 10.04 LTS | 04/29/10 | <i>Lucid Lynx</i> | | |

Release code Ubuntu terdiri dari dua kata yang berupa *adjective* (kata sifat) yang diikuti dengan nama hewan yang disusun secara alfabetis setiap

rilisnya, kecuali versi 6.06 ke bawah (4.10; 5.04; 5.10). Misalnya *Hardy Heron*, terdiri dari kata sifat *hardy* (*Bold; brave; stout; daring; resolute; intrepid*. [1913 *Webster*]) dan *heron* (sejenis burung).

CD Ubuntu dapat diperoleh secara bebas/gratis dalam bentuk iso maupun kepingan cd melalui *shipit*. Bagi pengguna Internet di Indonesia dapat mengunduh berkas Ubuntu lebih cepat dari cermin-cermin yang ada di Indonesia dibandingkan mengunduh dari situs luar.⁽¹²⁾

2.5. Bahasa *Assembly* dan Bahasa C⁽⁷⁾

Secara umum bahasa pemrograman dikelompokkan menjadi dua jenis, pertama, bahasa tingkat tinggi yang artinya bahasa pemrograman tersebut ditulis dengan bahasa yang mirip dengan bahasa manusia, contoh bahasa tingkat tinggi ini adalah bahasa C++, C#, *Java*, *Go*, *Perl*, *Python*, dll.

Kedua ialah bahasa tingkat rendah yaitu bahasa yang lebih dekat dengan bahasa mesin, dimana perintah-perintah yang dibuat biasanya lebih mendetail agar bisa langsung diproses oleh mesin, contoh bahasa tingkat rendah ini adalah bahasa *assembly*.

| | | |
|-------|-------|--------|
| | ORG | 000H |
| | MOV | A, #0 |
| | MOV | B, #1H |
| MAIN: | ACALL | STEP |
| | LCALL | CALC |
| | AJMP | MAIN |
| STEP: | INC | A |
| | RET | |
| CALC: | PUSH | ACC |
| | INC | B |
| | MUL | AB |
| | MOV | B,A |
| | POP | ACC |
| | RET | |
| | END | |

Gambar 2.7. Contoh Penulisan *Listing Program Assembly*.

Dalam proses pengekseskuan, bahasa tingkat tinggi lebih lambat dibandingkan dengan bahasa tingkat rendah. Hal ini disebabkan bahasa tingkat tinggi harus melewati lebih banyak proses penerjemahan. Sedangkan untuk bahasa C dapat juga tergolong dalam bahasa tingkat menengah, karena masih mempunyai ciri-ciri bahasa tingkat rendah dan sudah mendekati bahasa manusia, sehingga mempunyai kecepatan dalam pengekseskuan dan mempunyai kemudahan dalam penulisan *script*-nya.

Bahasa C merupakan pengembangan lebih lanjut dari bahasa BCPL yang dikembangkan oleh *Martin Richards* pada tahun 1976. Bahasa BCPL ini memberikan ide pada *Ken Thompson* yang kemudian mengembangkan bahasa pemrograman yang disebut sebagai bahasa B pada tahun 1970. Perkembangan selanjutnya dari bahasa B adalah bahasa C oleh *Dennis Ritchie* sekitar tahun 1970-an. Di *Bell Laboratories Inc.*⁽⁶⁾

Bahasa C pertama kali digunakan pada komputer *Digital Equipment Cooperation PDP-11* yang menggunakan sistem operasi *UNIX*. Bahasa C adalah bahasa standar, bahasa pemrograman dimana sebuah program yang ditulis dengan versi Bahasa C tertentu akan dapat dikompilasi dengan versi Bahasa C yang lain dengan sedikit modifikasi. Standar Bahasa C adalah standar dari *UNIX*.

Contoh proses pengeksekusian salah satu file yang menggunakan bahasa C di dalam sistem operasi *Linux* misalnya *Ubuntu*, Penulisan file program dapat menggunakan aplikasi *Gedit* atau menggunakan aplikasi editor lain seperti *pico*, *nano* atau *vim*, setelah file disimpan, file tersebut dapat dieksekusi lewat terminal.

```

/*program 1*/
/*test*/
main()
{
printf("percobaan Bahasa C di Linux\n");
}

```

Gambar 2.8. Penulisan Bahasa Program C Menggunakan *Text Editor*

Kemudian contoh sederhana program Bahasa C diatas disimpan dengan nama *test.c*, kemudian untuk melakukan eksekusi, buka *terminal* atau konsolnya *Linux* dan lakukan dengan perintah `gcc -o nama_file nama_file.c`, contoh untuk program di atas ialah dengan mengetik pada *terminal*, `gcc -o test test.c` atau `gcc test.c -o test`. Kemudian ketik `./test` untuk menjalankan program yang telah dibuat.

```

pakoek@pakoek-laptop: ~$ gcc -o test test.c
pakoek@pakoek-laptop: ~$ ./test
pakoek@pakoek-laptop: ~$ percobaan Bahasa C di Linux
pakoek@pakoek-laptop: ~$

```

Gambar 2.9. Penulisan Hasil *Output Program* pada *Terminal*.⁽⁶⁾

2.6. Media Penghubung Antara Perangkat Lunak Dengan Mikrokontroler

UART atau *Universal Asynchronous Receiver-Transmitter* adalah bagian perangkat keras komputer yang menerjemahkan antara *bit-bit parallel* data dan *bit-bit serial*. UART biasanya berupa sirkuit terintegrasi yang digunakan untuk komunikasi *serial* pada komputer atau *port serial* perangkat periperal. UART sekarang ini termasuk di dalam beberapa mikrokontroler.

Media penghubung kabel sebagai komunikasi antara komputer dengan mikrokontroler yang sering digunakan ada tiga jenis, yaitu *Port Parallel*, *Serial*, USB. *Port parallel* dan *port* USB merupakan media penghubung tercepat dalam menerima dan mengirimkan data.

2.6.1. *Parallel Port*⁽¹²⁾

Port paralel (DB-25) adalah salah satu jenis soket pada personal komputer untuk berkomunikasi dengan peralatan luar seperti printer model lama. Karena itu *parallel port* sering juga disebut *printer port*. Perusahaan yang memperkenalkan *port* ini adalah *Centronic*, maka *port* ini juga disebut dengan *Centronics port*.

Semua data, kontrol, dan status dari *port parallel* berhubungan dengan *register-register* yang ada di dalam komputer. Dengan mengakses langsung *register-register* tersebut, masukan dan keluaran dari *port parallel* dapat diatur.

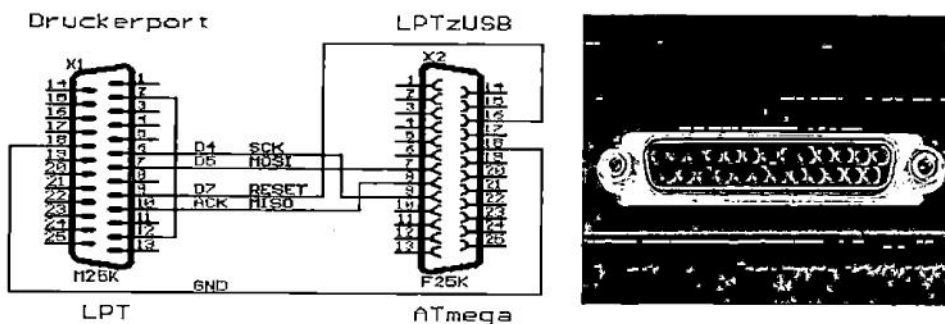
Register-register pada *port parallel* adalah:

1. Register data
2. Register status
3. Register kontrol

Pada umumnya di komputer personal alamat dasar LPT1 adalah 0x378 (378 *hexadecimal*) dan LPT2 adalah 0x278. Alamat dari ketiga register tersebut diatas dapat ditentukan dengan menjumlahkan alamat dasar dari port paralel dengan bilangan desimal tertentu. Misalnya kita ingin mengakses register data dari port paralel LPT1, alamat register datanya sama dengan alamat dasar dari LPT1 yaitu 0x378. Sedangkan alamat register status sama dengan alamat register dasar + 1 atau 0x379 dan alamat register kontrolnya sama dengan alamat register dasar + 2 atau 0x37A. Hal tersebut berlaku juga pada LPT2. Untuk lebih jelasnya, dapat dilihat ditabel dibawah ini:

Tabel 2.2. Register kontrol pada *parallel port*.

| Register | LPT1 | LPT2 |
|-------------------------------------|-------|-------|
| Register data (alamat dasar + 0) | 0x378 | 0x278 |
| Register status (alamat dasar + 1) | 0x379 | 0x279 |
| Register kontrol (alamat dasar + 2) | 0x37A | 0x27A |



Gambar 2.10. *Port* Paralel Dalam Bentuk Simbol dan Bentuk Fisiknya.

2.6.2. USB port⁽¹²⁾

Universal Serial Bus (USB) adalah standar *bus serial* untuk perangkat penghubung, biasanya kepada komputer namun juga digunakan di peralatan

lainnya seperti konsol permainan, ponsel dan PDA. Sistem USB mempunyai desain yang *asimetris*, yang terdiri dari pengontrol *host* dan beberapa peralatan terhubung yang berbentuk pohon dengan menggunakan peralatan *hub* yang khusus.

Desain USB ditujukan untuk menghilangkan perlunya penambahan *expansion card* ke ISA komputer atau bus PCI, dan memperbaiki kemampuan *plug-and-play* (pasang-dan-mainkan) dengan memperbolehkan peralatan-peralatan ditukar atau ditambah ke sistem tanpa perlu *mereboot* komputer. Ketika USB dipasang, ia langsung dikenal sistem komputer dan memroses *device driver* yang diperlukan untuk menjalankannya.

USB dapat menghubungkan peralatan tambahan komputer seperti *mouse*, *keyboard*, pemindai gambar, kamera digital, printer, hard disk, dan komponen *networking*. USB kini telah menjadi standar bagi peralatan multimedia seperti pemindai gambar dan kamera digital.

Versi terbaru (hingga Januari 2005) USB adalah versi 2.0. Perbedaan paling mencolok antara versi baru dan lama adalah kecepatan transfer yang jauh meningkat. Kecepatan transfer data USB dibagi menjadi tiga, antara lain:

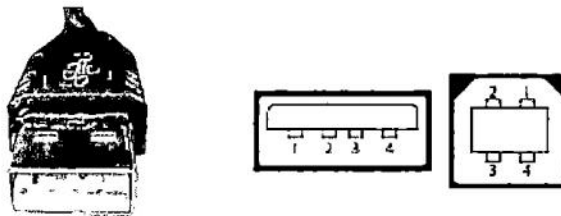
- High speed data dengan frekuensi clock 480.00Mb/s dan tolerasi pensinyalan data pada $\pm 500\text{ppm}$.
- Full speed data dengan frekuensi clock 12.000Mb/s dan tolerasi pensinyalan data pada $\pm 0.25\%$ atau 2,500ppm.
- Low speed data dengan frekuensi clock 1.50Mb/s dan tolerasi pensinyalan data pada $\pm 1.5\%$ atau 15,000ppm.

2.6.2.1. Persinyalan USB

USB adalah *host-centric bus* di mana host/terminal induk memulai semua transaksi. Paket pertama/penanda (*token*) awal dihasilkan oleh host untuk menjelaskan apakah paket yang mengikutinya akan dibaca atau ditulis dan apa tujuan dari perangkat dan titik akhir. Paket berikutnya adalah data paket yang diikuti oleh *handshaking packet* yang melaporkan apakah data atau penanda sudah diterima dengan baik atau pun titik akhir gagal menerima data dengan baik.

Setiap proses transaksi pada USB terdiri atas:

- Paket token/sinyal penanda (Header yang menjelaskan data yang mengikutinya)
- Pilihan paket data (termasuk tingkat muatan) dan
- Status paket (pemberitahuan hasil transaksi dan untuk koreksi kesalahan)



Gambar 2.11. Port USB dan Nomor kaki pada port USB

Tabel 2.3. Penetapan kaki pada port USB

| Kaki | Fungsi |
|-------|-------------------------|
| 1 | V_{BUS} (4.75–5.25 V) |
| 2 | D ⁻ |
| 3 | D ⁺ |
| 4 | GND |
| Shell | Shield |

2.6.2.2. Paket data umum USB

Data di bus USB disalurkan dengan cara mendahulukan *Least Significant Bit*(LSB). Paket-paket USB terdiri dari data-data berikut ini:

- *Sync*

Semua paket harus diawali dengan data *sync*. *Sync* adalah data 8 bit untuk *low* dan *full speed* atau data 32 bit untuk *high speed* yang digunakan untuk mensinkronkan *clock* dari penerima dengan pemancar. Dua *bit* terakhir mengindikasikan dimana data PID dimulai.

- PID (*Packet Identity*/Identitas paket)

Adalah *field* untuk menandakan tipe dari paket yang sedang dikirim.

Tabel dibawah ini menunjukkan nilai-nilai PID :

Tabel 2.4. Nilai PID pada port USB

| Group | Nilai PID | Identitas Paket |
|------------------|-----------|---------------------------------|
| <i>Token</i> | 0001 | OUT <i>Token</i> |
| <i>Token</i> | 1001 | IN <i>Token</i> |
| <i>Token</i> | 0101 | SOF <i>Token</i> |
| <i>Token</i> | 1101 | SETUP <i>Token</i> |
| Data | 0011 | DATA0 |
| Data | 1011 | DATA1 |
| Data | 0111 | DATA2 |
| Data | 1111 | MDATA |
| <i>Handshake</i> | 0010 | ACK <i>Handshake</i> |
| <i>Handshake</i> | 1010 | NAK <i>Handshake</i> |
| <i>Handshake</i> | 1110 | STALL <i>Handshake</i> |
| <i>Handshake</i> | 0110 | NYET (<i>No Response Yet</i>) |
| <i>Special</i> | 1100 | PREamble |
| <i>Special</i> | 1100 | ERR |
| <i>Special</i> | 1000 | <i>Split</i> |
| <i>Special</i> | 0100 | <i>Ping</i> |

Ada 4 bit PID data, supaya yakin diterima dengan benar, 4 bit di komplementasikan dan diulang, menjadikan 8 bit data PID. Hasil dari pengaturan tersebut adalah sebagai berikut:

| | | | | | | | |
|------|------|------|------|-------|-------|-------|-------|
| PID0 | PID1 | PID2 | PID3 | nPID0 | nPID1 | nPID2 | nPID3 |
|------|------|------|------|-------|-------|-------|-------|

- ADDR (*address*)

Bagian alamat dari peralatan dimana paket digunakan. Dengan lebar 7 bit, 127 peralatan dapat disambungkan. Alamat 0 tidak sah, peralatan yang belum terdaftar harus merespon paket yang dikirim ke alamat 0.

- ENDP (*End Point*)

Titik akhir dari *field* yang terdiri dari 4 bit, menjadikan 16 kemungkinan titik akhir. *Low speed devices*, hanya dapat mempunyai 2 tambahan *end point* pada puncak dari *pipe default*. (maksimal 4 *endpoints*)

- CRC

Cyclic Redundancy Check dijalankan pada data di dalam paket yang dikirim. Semua penanda (*token*) paket mempunyai sebuah 5 bit CRC ketika paket data mempunyai sebuah 16 bit CRC.

- EOP (*End of packet*)

Akhir dari paket yang disinyalkan dengan satu angka akhir 0 (*Single Ended Zero/SEO*) untuk kira-kira 2 kali bit diikuti oleh sebuah J 1 kali.

Data yang dikirim dalam bus USB adalah salah satu dari 4 bentuk, yaitu *control*, *interrupt*, *bulk*, atau *isochronous*.

2.6.2.3. Perancangan Peralatan yang Menggunakan USB⁽¹²⁾

Untuk membuat suatu peralatan yang dapat berkomunikasi dengan protokol USB tidak perlu harus mengetahui secara rinci protokol USB. Bahkan kadang tidak perlu pengetahuan tentang USB protokol sama sekali. Pengetahuan tentang USB protokol hanya diperlukan untuk mengetahui spesifikasi yang dibutuhkan untuk alat kita.

Pada kenyataannya untuk mengimplemetasikan USB protokol di FPGA ataupun perangkat bantu lain sangat tidak efisien dan banyak waktu terbuang untuk merancangnya. Menggunakan kontroler USB sangat lebih dianjurkan dalam membuat alat yang dapat berkomunikasi melalui protokol ini. Kontroler USB mempunyai banyak macam bentuk, dari *microcontroller* berbasis 8051 yang mempunyai *input output* USB secara langsung sampai pengubah *protocol* dari *serial* seperti *I2C bus* ke USB.

USB *controller* biasanya dijual dengan disertai berbagai fasilitas yang mempermudah pengembangan alat, diantaranya manual yang lengkap, *driver* untuk *Windows XP*, contoh code aplikasi untuk mengakses USB, contoh *code* untuk USB *controller*, dan skema rangkaian elektroniknya.

Dalam sisi pengembangan perangkat lunak untuk aplikasi dalam *personal computer*, komunikasi antar perangkat keras di dalam perangkat keras USB tidak terlalu diperhatikan karena *Windows* ataupun sistem operasi lain yang akan mengurusnya. Pengembang perangkat lunak hanya memberikan data yang akan dikirim ke alat USB di *buffer* penyimpan dan membaca data dari alat USB dari

buffer pembaca. Untuk *driver* pun kadang-kadang *Windows* sudah menyediakannya, kecuali untuk peralatan yang mempunyai spesifikasi khusus kita harus membuatnya sendiri.

2.7. Hipotesis

Saat ini sudah ada yang mampu mengimplementasikan praktek mikrokontroler AVR menggunakan sistem operasi *Linux*, terutama lewat *konsole* atau *terminal (command prompt)*, salah satunya Khoirul Umam. Beliau bekerja di PT. Nurul Fikri Cipta Inovasi sebagai Staff Teknik Komputer, konsentrasi mengajar beliau pada bidang *Microcontroller* dan *Robotic*.⁽⁹⁾

Kebanyakan mahasiswa khususnya yang hobi membuat program *file* untuk AVR, lebih cenderung menggunakan perangkat lunak dan sistem operasi yang ilegal, karena kemudahan penggunaan dan prosesnya lebih cepat dan efisien, berbeda jika menggunakan sistem operasi *Linux*, karena proses penulisan *file* program dapat dilihat, dipelajari, dan dipahami melalui *terminal*.

Maka solusi yang tepat agar dapat menghindari pembajakan perangkat lunak, yaitu dengan membeli sistem operasi yang berlisensi dan komersil atau jika tidak ada biaya, dapat menggunakan yang perangkat lunak gratis atau menggunakan sistem operasi alternatif seperti *Ubuntu*, yang mudah dipelajari bagi pemula yang ingin migrasi dari sistem operasi yang berbayar, menuju sistem operasi yang gratis dan *opensource* (kode terbuka).