

Pemanfaatan ED-255EK *Embedded Education Platform* sebagai Modul Praktikum *Embedded System* Berbasis Linux

I Gst. Md. Ngurah Bimantara¹, I Gusti Agung Pt Raka Agung², Lie Jasa³

Abstract— As technology develops, the application of embedded systems in everyday life also grows, for example is a automatic washing machines, electric stoves, radios, and etc. This provides an indirect claim to understand the basic configuration of the embedded system application itself. Learning about the basics of embedded system application can be done using ED-255EK *Embedded Education Platform*. ED-255EK is powered by an Intel XScale PXA255 CPU that is compatible with ARM Version 5TE ISA, Linux operating system, adopts RISC super-pipeline, and operates at 200/300/400 Mhz with low power consumption. How to operate ED-255EK is to configure the operating system first, includes upgrades the package of Linux system, configure the network and serial communication. Then configuration on ED-255EK which includes downloading file system and setting IP Address. The final step is the configuration of the module, which includes the installation of driver files and application programs. The input data provided will be displayed on the ED-255EK module and serialized also displayed on the laptop screen. The modules used are LED, LCD, DOT-Matrix, FND, Key-pad, DIP-Switch, Stepper, IrDA, RF, and uWave modules.

Intisari— Seiring perkembangan teknologi, pengaplikasian *embedded system* dalam kehidupan sehari-hari juga ikut berkembang, seperti halnya mesin cuci otomatis, kompor listrik, radio, dan lain sebagainya. Hal ini memberikan tuntutan secara tidak langsung untuk memahami konfigurasi dasar dari pengaplikasian *embedded system* itu sendiri. Pembelajaran tentang dasar dari pengaplikasian *embedded system* dapat dilakukan dengan menggunakan ED-255EK *Embedded Education Platform*. ED-255EK didukung dengan CPU Intel XScale PXA255 yang memiliki kompatibel dengan ARM Versi 5TE ISA, sistem operasi Linux, mengadopsi *super-pipeline* RISC, serta beroperasi pada 200/300/400 Mhz dengan konsumsi daya yang rendah. Cara pengoperasian ED-255EK adalah dengan melakukan konfigurasi pada Sistem operasi terlebih dahulu, diantaranya upgrade *package*, konfigurasi jaringan dan komunikasi secara serial. Dilanjutkan dengan konfigurasi pada ED-255EK yang meliputi pengunduhan *file system* dan pengaturan IP Address. Langkah terakhir adalah pengkonfigurasi modul, yang meliputi pemasangan *file driver* dan program aplikasi. Data *input* yang diberikan akan ditampilkan pada modul ED-255EK dan secara serial juga ditampilkan pada layar laptop. Modul-modul yang digunakan diantaranya adalah modul LED, LCD, DOT-Matrix, FND, Key-pad, DIP-Switch, Motor Stepper, IrDA, RF, dan uWave.

Kata Kunci— *Embedded system*, ED-255EK, Linux, Praktikum.

¹Mahasiswa Program Studi Teknik Elektro, Jalan Padma No. 159 Penatih Denpasar 80238 INDONESIA (tlp: 081933021148; e-mail: ngurahbimantara@gmail.com)

^{2, 3}Dosen Program Studi Teknik Elektro Fakultas Teknik Universitas Udayana, Jln. Jalan Kampus Bukit Jimbaran 80361 INDONESIA (telp: 0361-703315; fax: 0361-4321; e-mail: puturaka@ee.unud.ac.id, liejasa@unud.ac.id)

I. PENDAHULUAN

Embedded system merupakan sebuah sistem yang dirancang khusus untuk melakukan perintah yang lebih spesifik guna meningkatkan kinerja sebuah alat. Banyak hal yang dapat dijumpai dalam kehidupan sehari-hari yang merupakan implementasi dari *embedded system* itu sendiri. Pengimplementasian *embedded system* terus berkembang seiring dengan perkembangan teknologi. Tantangan utama dalam merancang dan menganalisa *embedded system* berasal dari interaksinya dengan proses fisik[1], oleh karena itu, pelatihan dasar tentang *embedded system* sangat diperlukan guna untuk mengembangkan *embedded system* kedepannya.

Pada penelitian sebelumnya telah dibuat rancangan modul praktikum sistem tertanam berbasis Arduino. Penelitian ini menggunakan dua buah mikrokontroler, yaitu mikrokontroler Atmega8535 dan mikrokontroler Arduino. Penelitian ini menguji kecepatan antara dua buah mikrokontroler dalam menyelesaikan sebuah intruksi[2]. Penelitian selanjutnya dilakukan perancangan sistem *embedded* Linux berbasis ARM. Pada penelitian ini dilakukan pembuatan *single board computer* AT91RM9200A berbasis mikrokontroler ARM AT91RM9200 serta pembuatan *software* yang meliputi konfigurasi *bootloader*, *kernel*, dan *root filesystem*. *Software* ini selanjutnya di-*compile* menggunakan *cross compiler*[3]. Penelitian lainnya yaitu pembuatan prototipe modul pembelajaran *embedded system* berbasis Arduino. Penelitian ini memuat tentang hasil pembuatan prototipe modul pembelajaran guna menunjang matakuliah *embedded system* di prodi Teknik Elektro UWIKA. Konsep penelitiannya mengintegrasikan tiap-tiap komponen menjadi satu kesatuan dalam sebuah PCB dan dikendalikan menggunakan mikrokontroler Arduino[4]. Penelitian sebelumnya sebagian besar menggunakan mikrokontroler Arduino karena mikrokontroler Arduino bersifat open source untuk pengendali mikro *single board*, sehingga memudahkan pengguna elektronik dalam berbagai bidang[5].

Penelitian kali ini menggunakan ED-255EK yang dilengkapi dengan prosesor Intel XScale PXA255 yang kompatibel dengan ARM versi 5TE ISA, menggunakan sistem operasi Linux, mengadopsi *super-pipeline* RISC, serta dapat beroperasi pada 200/300/400 MHz dengan konsumsi daya yang rendah. ED-255EK dirancang khusus sebagai modul pembelajaran *embedded system* sehingga sangat mudah, praktis, dan efisien digunakan sebagai media pembelajaran tentang *embedded system* berbasis Linux.

II. KAJIAN PUSTAKA

A. *Embedded System*

Embedded System terdiri dari perangkat keras (*hardware*) yang meliputi mikroprosesor atau mikrokontroler dan



komponen I/O yang terpasang di dalamnya, serta terdiri dari perangkat lunak (*software*) yang merupakan penggerak sistem dari *embedded* itu sendiri. *Embedded System* memiliki ciri-ciri diantaranya adalah memiliki *computing power* atau dilengkapi dengan prosesor, bekerja di lingkungan luar ruangan IT, dan memiliki tugas yang lebih spesifik. Sedangkan karakteristik dari *embedded system* adalah dirancang untuk melakukan tugas tertentu, tidak selalu didesain sebagai perangkat mandiri, dan *firmware* (instruksi program) disimpan ke dalam ROM.

B. Sistem Operasi Linux

Linux merupakan salah satu sistem operasi yang berbasis Unix yang dikembangkan oleh Linus Torvalds. Linux memiliki beberapa kelebihan diantaranya adalah *open source*, *freeware*, minimal *hardware*, skalabilitas, stabilitas, *shared libraries*, *non-fragmentasi*, dan kebal virus. Sedangkan kekurangan dari sistem operasi Linux adalah banyaknya *user* yang masih belum terbiasa menggunakan Linux sehingga struktur direktori atau hak akses masih membingungkan *user*.

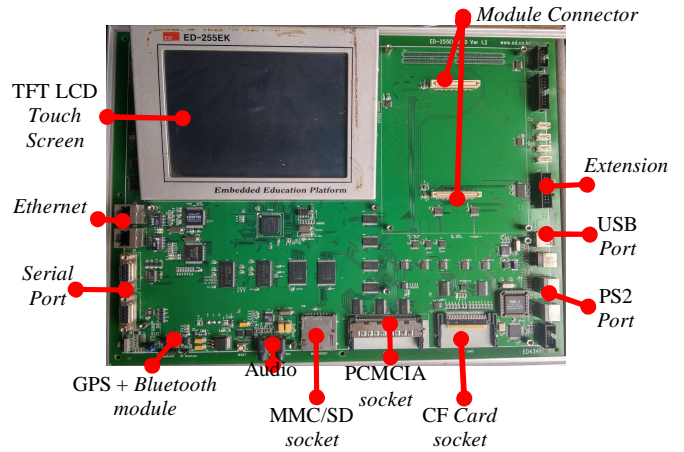
Seiring dengan perkembangan teknologi, Linux telah banyak memiliki distro, salah satunya adalah Fedora. Linux Fedora merupakan sebuah distro Linux yang berbasis RPM dan yum yang dikembangkan oleh Fedora Project. Keunggulan Linux Fedora dibandingkan dengan distro lainnya adalah memiliki *future* yang banyak dengan tampilan *desktop* yang memukau sehingga populer di Amerika. Linux Fedora juga memiliki sistem keamanan yang baik dengan adanya SELinux sehingga tidak mudah terkena virus serta terdapat *libreoffice* yang siap digunakan untuk perkantoran. Dibalik keunggulan itu, Linux Fedora juga memiliki beberapa kekurangan, seperti Tidak sesetabil Debian atau Slakware untuk server, tidak semudah atau semutakhir Suse atau Mandrake untuk penggunaan desktop, multimedia yang kurang secara default, serta Fedora merupakan distribusi untuk keperluan umum. Pada penelitian ini menggunakan Linux Fedora karena Linux Fedora memiliki kernel yang sesuai dengan kernel yang dibutuhkan untuk konfigurasi ED-255EK. Pada penelitian ini digunakan sistem operasi Linux Fedora 9.0.

C. ED-255EK

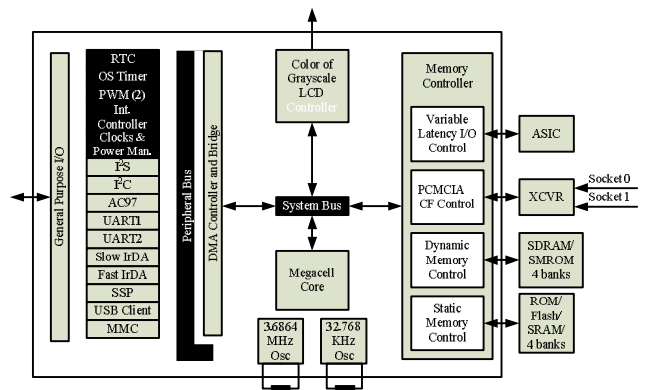
ED-255EK merupakan papan sirkuit yang telah ditanamkan prosesor Intel XScale PXA255 dan banyak *embedded operational system* dari *embedded Linux system* dan WIN CE. ED-255EK dilengkapi *port-port* seperti yang ditunjukkan pada Gambar 1. Dalam pengkonfigurasian ED-255EK hanya digunakan dua port, yaitu ethernet *port* dan serial *port*.

D. Intel XScale PXA255

XScale merupakan mikroarsitektur yang dirancang oleh Intel untuk *central processing units* dengan menggunakan arsitektur ARM versi 5TE ISA. XScale menggunakan *seven-stage integer* dan *eight-stage memory super-pipelined microarchitecture*. Keluarga XScale merupakan prosesor 32 bit yang diproduksi dengan proses 0.8µm 0.13µm dan memiliki cache data dan cache instruksi masing-masing sebesar 32 KB. Diagram blok yang ditunjukkan pada Gambar 2 merupakan struktur dari Intel XScale PXA255.



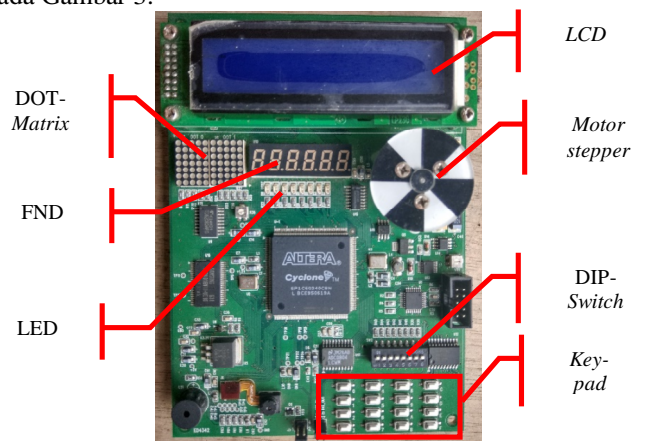
Gambar 1: ED-255EK



Gambar 2: Diagram blok PXA255

E. ED-255FPGA

ED-255FPGA merupakan papan sirkuit yang berisikan modul-modul seperti LED (*Light Emitting Diode*), LCD (*Liquid Crystal Display*), DOT-Matrix, FND (*Flexible Numeric Display*), motor stepper, key-pad, DIP-Switch, ADC, kamera CMOS, dan buzzer. Pada penelitian ini hanya digunakan tujuh modul, diantaranya LED, LCD, DOT-Matrix, FND, motor stepper, key-pad, dan DIP-Switch. ED-255FPGA ditunjukkan pada Gambar 3.

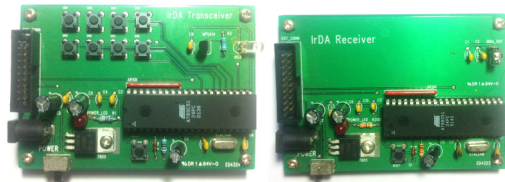


Gambar 3: ED-255FPGA

F. Extension Module

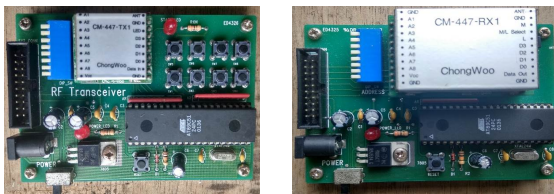
Extension module merupakan modul tambahan pada ED-255EK yang digunakan untuk melakukan transmisi data. Pada

penelitian ini menggunakan tiga buah modul *extension*, diantaranya adalah modul IrDA, modul RF, dan modul uWave. Modul IrDA merupakan modul yang dirancang untuk melakukan pengiriman dan penerimaan data dengan menggunakan inframerah. Modul IrDA menggunakan sensor CL-1L5R untuk mengirim sinyal inframerah yang ditanamkan pada modul *transceiver* dan sensor KSM603LM untuk menerima sinyal inframerah yang ditanamkan pada modul *receiver*. Modul IrDA ditunjukkan pada Gambar 4.



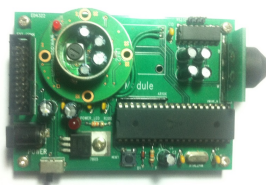
(a) (b)
 Gambar 4: Modul IrDA (a) *transceiver* dan (b) *receiver*

Modul RF merupakan modul yang dirancang untuk melakukan komunikasi data dengan menggunakan gelombang radio. Dibutuhkan dua buah modul untuk melakukan komunikasi data dengan menggunakan gelombang radio, yaitu modul pemancar gelombang dan modul penerima gelombang[6]. Modul RF pada penelitian ini menggunakan perangkat CM-447-TX1 untuk mengirim gelombang dan perangkat CM-447-RX1 untuk menerima gelombang. Modul RF ditunjukkan pada Gambar 5.



(a) (b)
 Gambar 5: Modul RF (a) *transceiver* dan (b) *receiver*

Modul uWave merupakan modul yang dirancang untuk mengirim dan menerima gelombang ultrasonik. Prinsip kerjanya yaitu pemancar gelombang akan mengirimkan sinyal kemudian diukur waktu yang dibutuhkan sampai pada datangnya pantulan objek[5]. Modul uWave pada penelitian ini menggunakan sensor ST-203L untuk mengirim dan menerima gelombang ultrasonik. Modul uWave ditunjukkan pada Gambar 6.



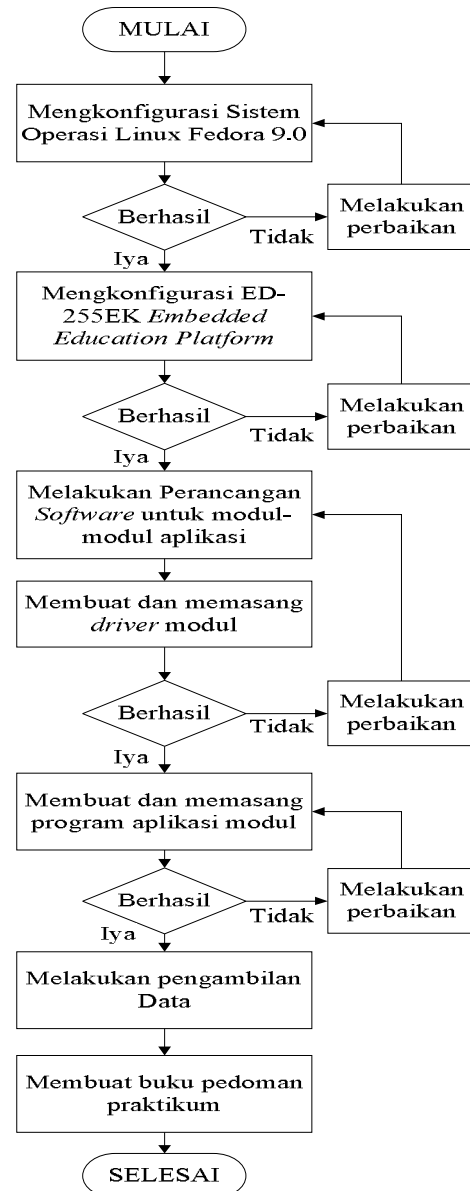
Gambar 6: Modul uWave

III. METODE PENELITIAN

Perancangan diagram alir pemanfaatan ED-255EK *Embedded Education Platform* sebagai modul praktikum *embedded system* berbasis Linux digunakan sebagai pedoman

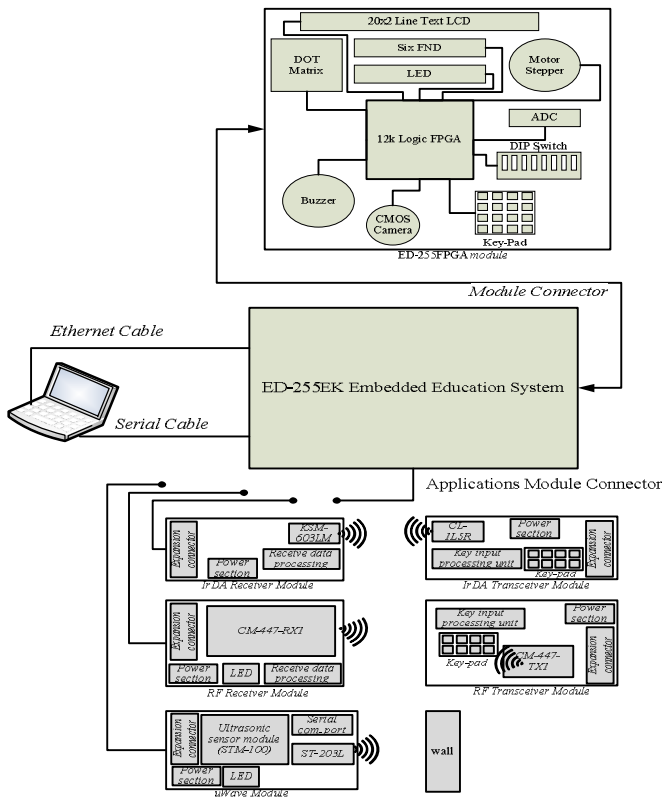
langkah-langkah yang akan dilakukan dalam penelitian. Diagram alir penelitian ditunjukkan pada Gambar 7.

Pemanfaatan ED-255EK sebagai modul praktikum *embedded system* dimulai dengan melakukan konfigurasi sistem operasi Linux yang terdiri dari penambahan *package* Linux, konfigurasi jaringan dan komunikasi data secara serial. Jika konfigurasi sistem operasi telah berhasil, dilanjutkan dengan melakukan konfigurasi ED-255EK yang meliputi pengaturan IP-Address dan pengunduhan file sistem ED-255EK. Pengkonfigurasi ED-255EK dilakukan dengan cara menghubungkan ED-255EK ke Laptop dengan menggunakan kabel *ethernet* dan kabel *serial*. Konfigurasi pemasangan modul-modul pada ED-255EK ditunjukkan pada Gambar 8.



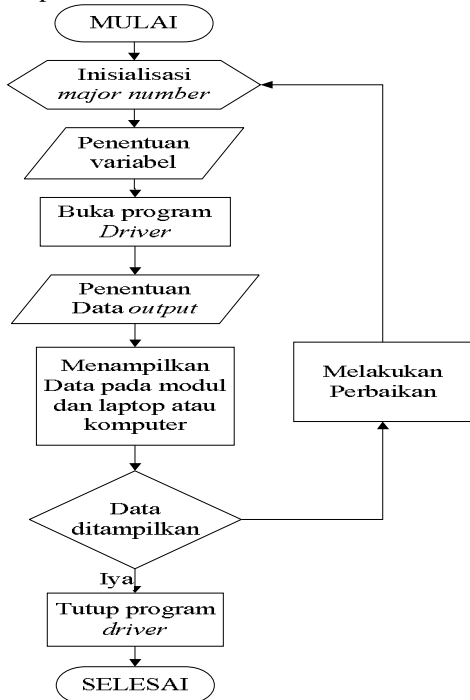
Gambar 7: Diagram alir penelitian pemanfaatan ED-255EK





Gambar 8: Konfigurasi pemasangan modul-modul pada ED-255EK

Perancangan perangkat lunak (*software*) untuk modul-modul aplikasi pada penelitian ini menggunakan terminal Linux. Perancangan *software* meliputi pembuatan file *driver* dan program aplikasi masing-masing modul. Diagram alir untuk perancangan *software* masing-masing modul aplikasi ditunjukkan pada Gambar 9.



Gambar 9: Diagram alir perancangan *software*

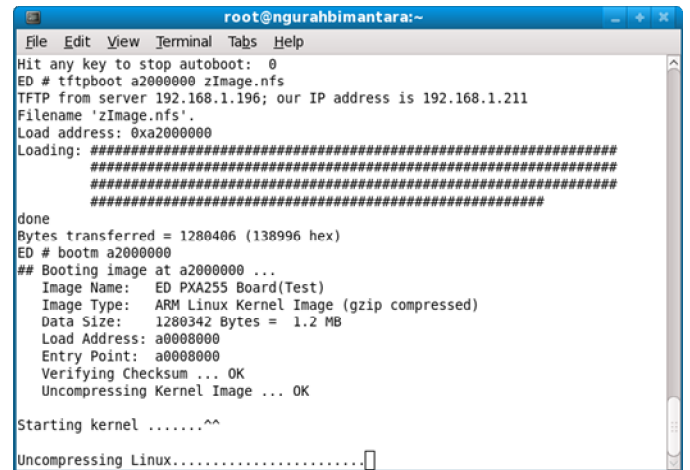
IV. HASIL DAN PEMBAHASAN

A. Konfigurasi Sistem Operasi Linux

Konfigurasi sistem operasi Linux merupakan langkah pertama yang dilakukan untuk memanfaatkan ED-255EK *embedded education platform* sebagai modul praktikum *embedded system* berbasis Linux. Tahap pengkonfigurasi dimulai dengan penambahan *package* Linux, diantaranya “*ftp*”, “*ftp-server*”, “*kernel-devel*”, dan *gcc*. Penambahan *package* Linux dilakukan dengan memasukkan perintah “*yum install*” pada kernel Linux. Langkah selanjutnya yaitu membuat direktori *tftpboot* dengan menggunakan perintah “*mkdir*” dan mengedit file *tftp* dengan menggunakan perintah “*nano*”. Dilanjutkan dengan pengekseskuan TFTP Demon dengan menggunakan perintah “*/etc/init.d/xinetd restart*”. Konfigurasi dilanjutkan dengan melakukan pengaturan pada NFS Server. File NFS Server dapat disalin dengan menggunakan perintah “*cp*” dan mengekstraknya dengan menggunakan perintah “*tar -xzvf*”. Pada file *exports* ditambahkan “*/etc/ED255_NFS_ROOT/192.168.1.211(rw,no_root_squash,no_all_squash)*”. Langkah selanjutnya adalah melakukan *restart* NFS Server dengan menggunakan perintah “*/etc/init.d/nfs restart*” dan *disabled Firewall* dan SELinux. Langkah terakhir adalah pengaturan serial komunikasi antara ED-255EK dengan Laptop. Perintah yang digunakan adalah “*minicom -s*”.

B. Konfigurasi ED-255EK

Konfigurasi ED-255EK meliputi pengaturan IP Address dan pengunduhan file sistem ED-255EK. Pengaturan IP Address dilakukan dengan cara masuk pada sistem ED-255EK menggunakan perintah “*minicom*”. Perintah yang dimasukkan adalah “*editenv*” untuk mengatur IP Address pada alamat 192.168.1.211. Langkah selanjutnya adalah pengunduhan file sistem ED-255EK dengan menggunakan perintah “*tftpboot a2000000 zImage.nfs*”. File sistem akan tersimpan pada alamat a2000000 dan untuk melakukan *booting* dari alamat a2000000 digunakan perintah “*bootm a2000000*”. Hasil pengunduhan dan *booting* sistem ditunjukkan pada Gambar 10.



Gambar 10: Pengunduhan dan *booting* sistem ED-255EK

C. Konfigurasi Modul-Modul Aplikasi

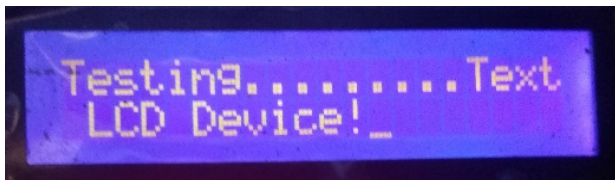
Dalam pengkonfigurasi modul-modul aplikasi, dibutuhkan file *driver* dan program aplikasi yang disimpan pada masing-masing direktori modul. Pembuatan direktori menggunakan perintah “*mkdir*”. File *driver* dan program

aplikasi dibuat dengan menggunakan format “.c”, seperti led.c, led.c, dot.c, dan lain sebagainya. File driver dan program aplikasi akan di-*crosscompiling* menjadi format file “.o” dengan menggunakan perintah “make”. File driver dan program aplikasi dengan format “.o” disalin ke ED-255EK dengan menggunakan perintah “make install”. Langkah selanjutnya adalah pembuatan *device file* untuk masing-masing modul aplikasi dengan menggunakan perintah “mknod”. File driver masing-masing modul dipasang pada ED-255EK dengan menggunakan perintah “insmod”. Program aplikasi masing-masing modul dieksekusi dengan menggunakan perintah “./”. Pada program aplikasi modul LED diberikan nilai data *input* 0x00ff. Nilai data ini merupakan bentuk bilangan heksadesimal, sehingga jika dikonversikan ke dalam bentuk biner maka akan diperoleh data 0000 0000 1111 1111. Selanjutnya data dalam bentuk biner akan dibaca oleh sistem ED-255EK yang selanjutnya ditampilkan pada data *output* pada modul LED, seperti yang ditunjukkan pada Gambar 11.



Gambar 11: Hasil *running* program aplikasi modul LED

Pada program aplikasi modul LCD diberikan instruksi untuk melakukan pembacaan posisi dan kondisi kursor pada modul LCD. Posisi dan kondisi kursor sebelumnya sudah diatur pada program aplikasi, sehingga eksekusi program aplikasi akan menampilkan posisi dan kondisi kursor pada modul LCD dan pada layar laptop. Hasil *running* program aplikasi modul LCD ditunjukkan pada Gambar 12, dan pada layar laptop ditunjukkan pada Gambar 13.



Gambar 12: Hasil *running* program aplikasi modul LCD

```

**** LCD Device Driver Information ****
* Line = 2 *
* Column = 12 *
* flag = 1 *
* Blink = OFF *
*****
Testing Complete. Press any key!
    
```

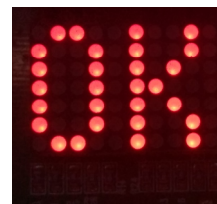
Gambar 13: Hasil *running* program aplikasi modul LCD pada *Host PC*

Pada program aplikasi modul *DOT-Matrix* diberikan nilai data *input* yaitu { 0x0192, 0x00252, 0x0254, 0x0258, 0x0254, 0x0252, 0x0192 }. Nilai data ini merupakan bentuk bilangan heksadesimal. Jika dikonversikan ke dalam bentuk biner maka akan diperoleh TABEL I. Prinsip kerja *DOT-Matrix* yaitu menggunakan sistem *scanning* kolom. *Scanning* kolom merupakan suatu sistem dimana dari sekian banyak kolom, hanya satu kolom yang menyala dan secara bergantian. Jika

diperlambat, akan terlihat perpindahan nyala LED dari satu kolom ke kolom lainnya, namun karena *display* dari *DOT-Matrix* sangat cepat, maka terlihat semua LED menyala secara bersamaan [7]. Data hasil dari *running* program aplikasi modul *DOT-Matrix* ditunjukkan pada Gambar 14.

TABEL I
KONVERSI NILAI DATA HEKSADesimal MENJADI BINNER

Heksadesimal	Binner
0x0192	0000 0001 1001 0010
0x0252	0000 0010 0101 0010
0x0254	0000 0010 0101 0100
0x0258	0000 0010 0101 1000
0x0254	0000 0010 0101 0100
0x0252	0000 0010 0101 0010
0x0192	0000 0001 1001 0010



Gambar 14: Hasil *running* program aplikasi modul *DOT-Matrix*

Pada program aplikasi modul FND diberikan nilai data *input* berupa angka 0, 1, 2, 3, 4, 5. Nilai data ini akan dituliskan pada modul FND dan dibaca pada *host PC*, sehingga saat *running* program aplikasi modul FND akan terlihat seperti Gambar 15 dan pada layar laptop seperti Gambar 16.

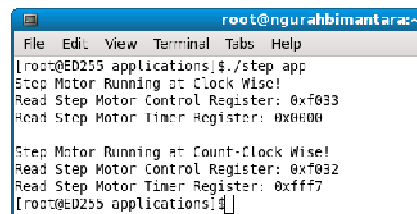


Gambar 15: Hasil *running* program aplikasi modul FND

```

FNDs displayed the sequential numbers!
Read data[0] = 0x0000
Read data[1] = 0x0001
Read data[2] = 0x0002
Read data[3] = 0x0003
Read data[4] = 0x0004
Read data[5] = 0x0005
    
```

Gambar 16: Hasil *running* program aplikasi modul FND pada *Host PC*



Gambar 17: Hasil *running* program aplikasi modul *motor stepper* pada *Host PC*

Pada program aplikasi modul *motor stepper* diberikan instruksi untuk melakukan pembacaan arah dan kecepatan putaran motor. Saat program aplikasi dieksekusi, modul *motor stepper* akan langsung berputar dan pada *host PC* akan



ditampilkan data hasil putaran dari modul *motor stepper* itu sendiri. Data hasil pada *host PC* ditunjukkan pada Gambar 17.

Pada program aplikasi modul *DIP-Switch* diberikan instruksi untuk melakukan pembacaan data secara kontinyu dari setiap saklar yang diubah-ubah. Pada modul *DIP-switch* terdapat 8 buah saklar yang dapat diubah-ubah. Kedelapan buah saklar ini akan memberikan nilai data dalam bentuk biner yang kemudian dikonversikan ke dalam bentuk heksadesimal. Data inilah yang akan ditampilkan pada *host PC* secara kontinyu. Data hasil *running* program modul *DIP-Switch* ditunjukkan pada TABEL II.

TABEL II
HASIL *RUNNING* PROGRAM APLIKASI MODUL *DIP-SWITCH*

DIP-Switch								Data Hasil
1	2	3	4	5	6	7	8	
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	0x0000
OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	0x0001
OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	0x0002
OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	0x0004
OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	0x0008
OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	0x0010
OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	0x0020
OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	0x0040
ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	0x0080

Pada program aplikasi modul *key-pad* diberikan instruksi untuk melakukan pembacaan data dari setiap tombol atau *push button* yang ditekan pada modul *key-pad*. Pada modul *key-pad* terdapat 16 tombol yang diberi nomor berdasarkan angka heksadesimal. Hasil dari *running* program modul *key-pad* ditunjukkan pada TABEL III.

TABEL III
HASIL *RUNNING* PROGRAM APLIKASI MODUL *KEY-PAD*

No	Key-pad	Output
1	Tombol 0	KEY-PAD: Pressed KEY_0
2	Tombol 1	KEY-PAD: Pressed KEY_1
3	Tombol 2	KEY-PAD: Pressed KEY_2
4	Tombol 3	KEY-PAD: Pressed KEY_3
5	Tombol 4	KEY-PAD: Pressed KEY_4
6	Tombol 5	KEY-PAD: Pressed KEY_5
7	Tombol 6	KEY-PAD: Pressed KEY_6
8	Tombol 7	KEY-PAD: Pressed KEY_7
9	Tombol 8	KEY-PAD: Pressed KEY_8
10	Tombol 9	KEY-PAD: Pressed KEY_9
11	Tombol A	KEY-PAD: Pressed KEY_A
12	Tombol B	KEY-PAD: Pressed KEY_B
13	Tombol C	KEY-PAD: Pressed KEY_C
14	Tombol D	KEY-PAD: Pressed KEY_D
15	Tombol E	KEY-PAD: Pressed KEY_E
16	Tombol F	KEY-PAD: Pressed KEY_F

Pada program aplikasi modul *IrDA* dan modul *RF*, diberikan instruksi yang sama dengan program aplikasi modul *key-pad*, hanya saja media komunikasi datanya yang berbeda. Pada modul *transceiver* *IrDA* dan *RF* terdapat modul *key-pad* yang diberinama SW1, SW2, SW3, SW4, SW5, SW6, SW7, dan SW8. Eksekusi program aplikasi modul *IrDA* dan modul *RF* ditunjukkan pada TABEL IV dan TABEL V.

TABEL IV
HASIL *RUNNING* PROGRAM APLIKASI MODUL *IRDA*

No	Tombol Modul <i>IrDA</i> Transceiver	Data Hasil
1	SW1	Receive = 1

2	SW2	Receive = 2
3	SW3	Receive = 3
4	SW4	Receive = 4
5	SW5	Receive = 5
6	SW6	Receive = 6
7	SW7	Receive = 7
8	SW8	Receive = 8

TABEL V
HASIL *RUNNING* PROGRAM APLIKASI MODUL *RF*

No	Tombol Modul <i>RF</i> Transceiver	Data Hasil
1	SW1	Receive = 1
2	SW2	Receive = 2
3	SW3	Receive = 3
4	SW4	Receive = 4
5	SW5	Receive = 5
6	SW6	Receive = 6
7	SW7	Receive = 7
8	SW8	Receive = 8

Pada program aplikasi modul *uWave* diberikan instruksi untuk melakukan pengiriman dan penerimaan data berupa gelombang ultrasonik. Saat eksekusi program aplikasi, modul *uWave* akan memancarkan gelombang ultrasonik, kemudian akan menerima pantulan gelombang dari objek yang ada di depannya. Pantulan gelombang ultrasonik tersebut akan diproses pada AT89S51 yang selanjutnya akan diubah menjadi data berupa jarak antara sensor dengan objek. Pada penelitian ini, modul *uWave* tidak dapat memberikan data yang sesuai dengan jarak antara modul dengan objek, seperti yang ditunjukkan pada TABEL VI. Berdasarkan data hasil yang ditunjukkan pada TABEL VI, dapat disimpulkan bahwa modul *uWave* tidak dapat bekerja secara optimal dan tidak layak digunakan sebagai modul praktikum.

TABEL VI
HASIL *RUNNING* PROGRAM APLIKASI MODUL *UWAVE*

No	Jarak antara Modul <i>uWave</i> dengan Penghalang	
	Terukur (cm)	Terbaca (cm)
1	0	18
2	5	18
3	10	18
4	15	19
5	20	19
6	25	23
7	30	26
8	40	35
9	45	44
10	60	56

V. KESIMPULAN

Adapun simpulan yang diperoleh dari penelitian ini adalah sebagai berikut:

- ED-255EK *Embedded Education Platform* dapat dijadikan modul praktikum *embedded system* berbasis Linux.
- Pengaturan ED-255EK *Embedded Education Platform* meliputi konfigurasi Sistem Operasi, konfigurasi ED-255EK, serta konfigurasi file driver dan program aplikasi masing-masing modul.
- Pengendalian modul *I/O build-in* dan modul komunikasi data lainnya memerlukan file driver dan program aplikasi yang di-*compiling* dengan menggunakan Makefile masing-masing.

D. Modul uWave tidak dapat dijadikan modul praktikum karena tidak memberikan data yang kurang valid.

REFERENSI

- [1] Edward A. Lee and Sanjit A. Seshia, 2017, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, Second Edition, MIT Press, ISBN 978-0-262-53381-2.
- [2] Ulansari, Ramadhani, 2012. Rancangan Modul Praktikum Sistem Tertanam Berbasis Mikrokontroler Arduino. Skripsi Sistem Komputer. Ilmu Komputer dan Teknologi Universitas Gunadarma.
- [3] Andika Gunawan, Henry Hermawan, 2013. Perancangan Sistem Embedded Linux Berbasis ARM. Jurnal Ilmiah Mahasiswa Universitas Surabaya Vol.2 No.2
- [4] Arief Budijanto, Achmad Shoim, 2015. Prototipe Modul Pembelajaran Embedded System Berbasis Arduino. Seminar Nasional Sains dan Teknologi Terapan III 2015 Institut Teknologi Adhi Tama Surabaya. ISBN 978-602-98569-1-0.
- [5] Muhammad Namiruddin Al-Hasan, Cok Indra Partha, Yoga Divayana, "Rancang Bangun Pemandu Tuna Netra Menggunakan Sensor Ultrasonik Berbasis Mikrokontroler", Majalah Ilmiah Teknologi Elektro, Vol. 16, No.3, September – Desember 2017.
- [6] I Made Agung Pranata, Nyoman Pramaita, Nyoman Putra Sastra, "Sistem *Smart Traffic Light* Berbasis RFID Untuk Layanan Darurat", Majalah Ilmiah Teknologi Elektro, No. 3, September – Desember 2017.
- [7] Mustika, Dayu, 2012. Dot Matrix. <http://mustikadayu23.blogspot.co.id/2012/09/dot-matrix.html>



[Halaman ini sengaja dikosongkan]