

LAMPIRAN A

PETUNJUK PENGGUNAAN

A.1. Petunjuk Penggunaan *VGA Adapter*



1. Sambungkan kabel VGA dari proyektor digital ke *port* VGA yang tersedia pada *VGA Adapter*.
2. Hidupkan *VGA Adapter* melalui saklar ON/OFF di bagian atas perangkat.
3. Pastikan indikator *power* menyala warna biru



4. Tunggu sekitar 20 detik sampai proyektor digital menampilkan *Welcome Screen* yang menandakan *VGA Adapter* sudah siap untuk dioperasikan.



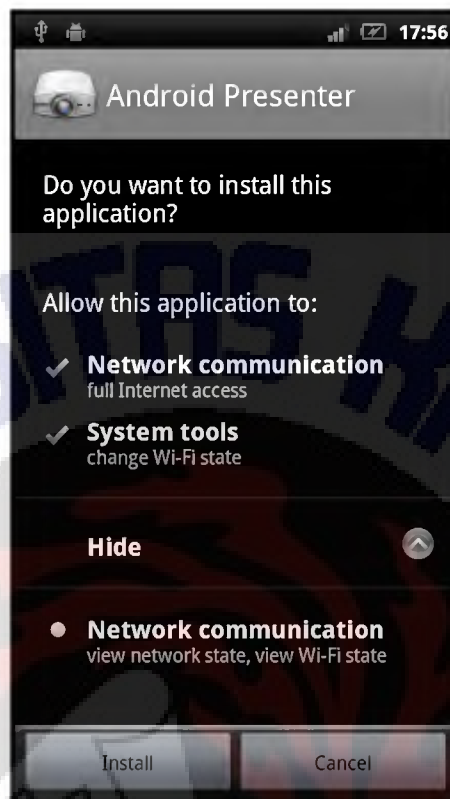
WELCOME SCREEN

5. Perangkat *VGA Adapter* siap untuk digunakan. *VGA Adapter* dihubungkan dengan *smartphone* Android (aplikasi *Android Presenter*) atau *notebook* (aplikasi *Desktop Presenter*).

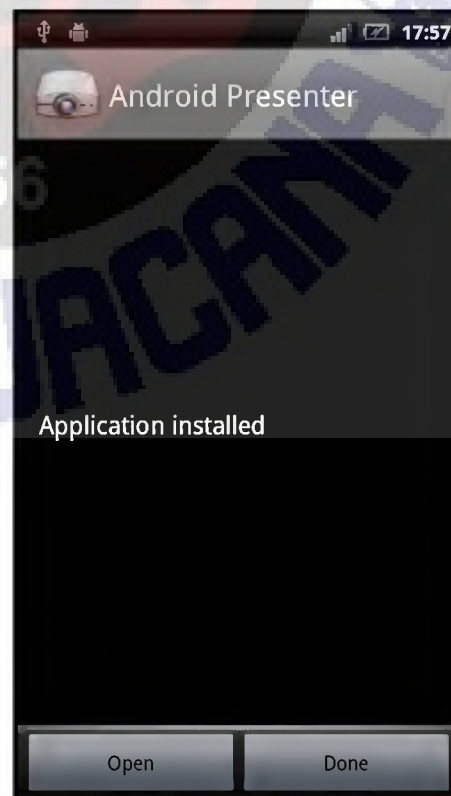
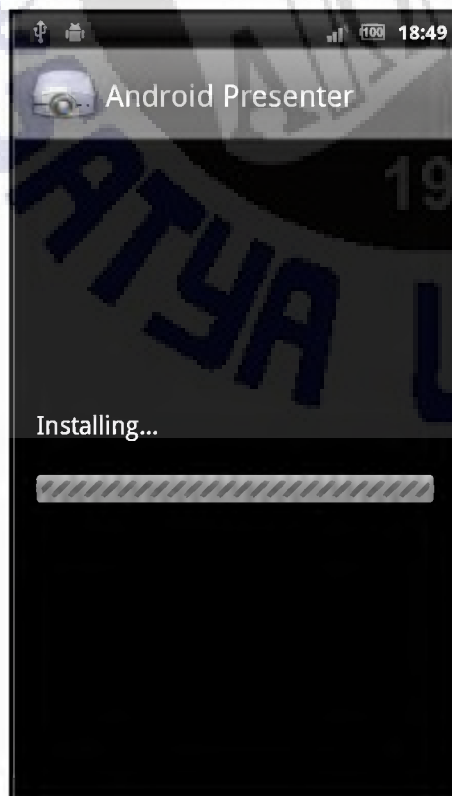


A.2. Petunjuk Penggunaan Aplikasi *Mobile Android (Android Presenter)*

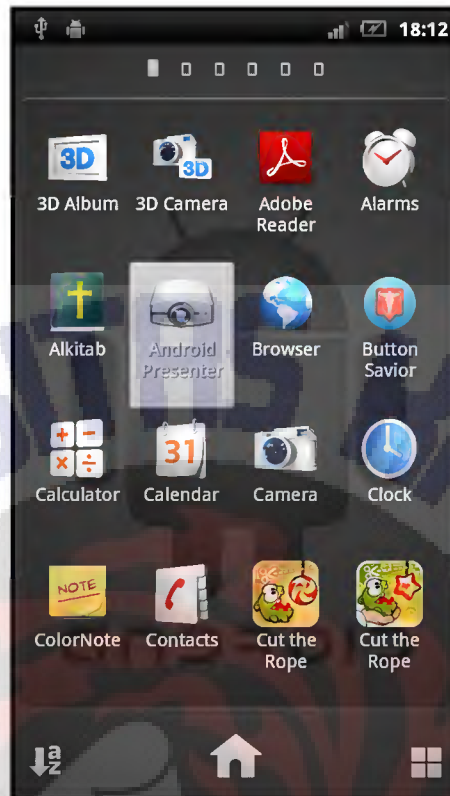
1. *Install* aplikasi *Android Presenter_1.0.apk*.



2. Tunggu sampai proses instalasi selesai.

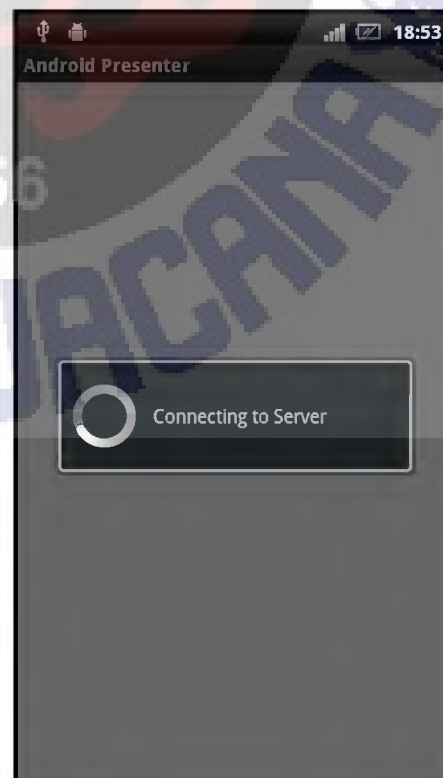
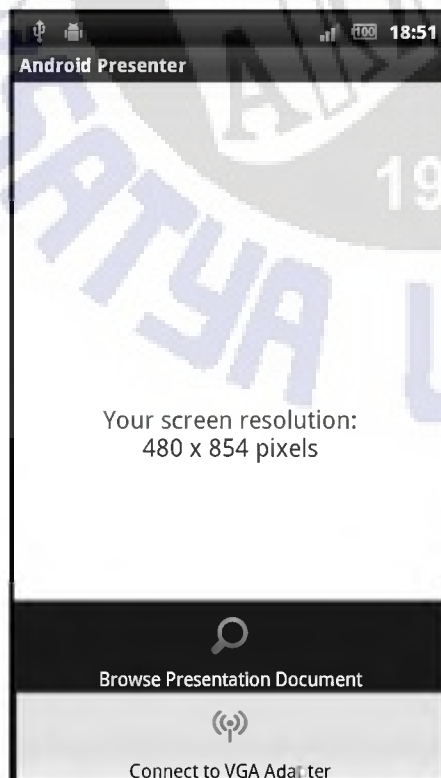


3. Jalankan aplikasi *Android Presenter* lewat *application launcher*.

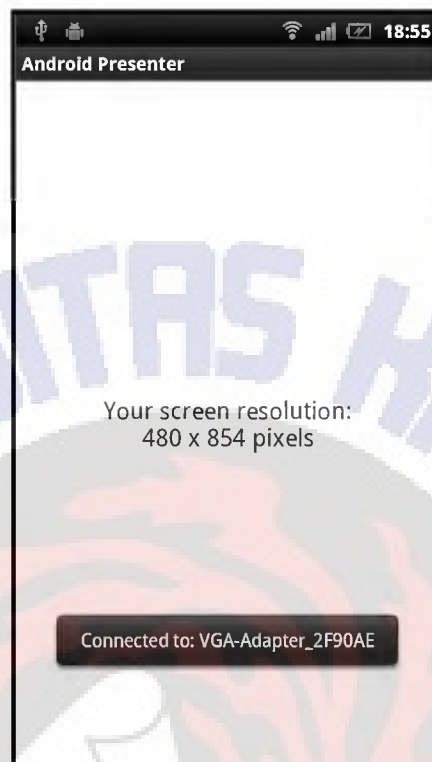


4. Koneksikan *smartphone* Android dengan *VGA Adapter*.

Tekan tombol menu pada *smartphone* Android, pilih menu *Connect to VGA Adapter*.

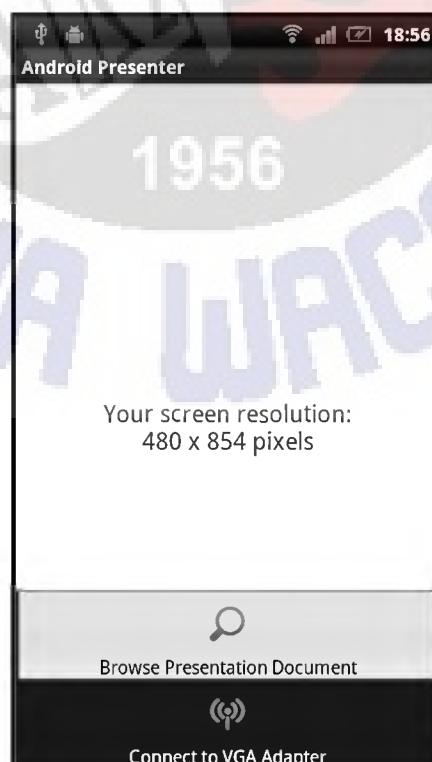


5. Tunggu sampai muncul notifikasi *Connected to: VGA-Adapter_xxxxxx* yang menandakan *smartphone* Android sudah terhubung dengan *VGA Adapter*.

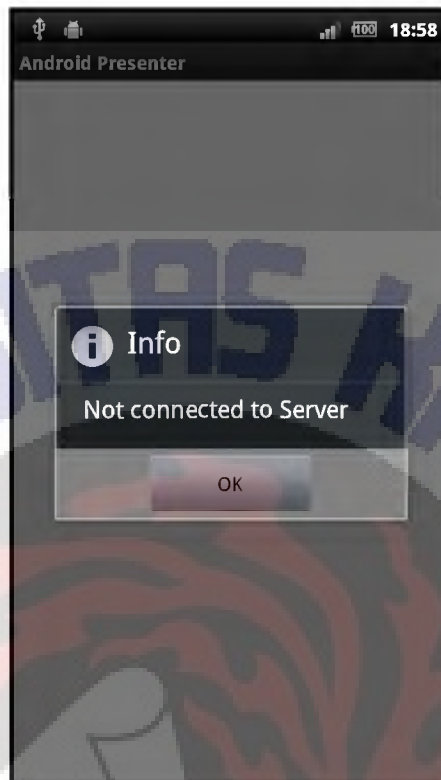


6. Cari dokumen presentasi yang akan ditampilkan.

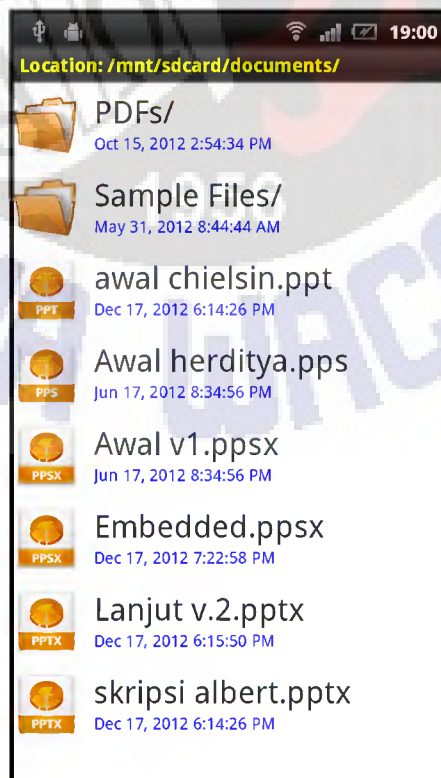
Tekan tombol menu, pilih menu *Browse Presentation Document*.



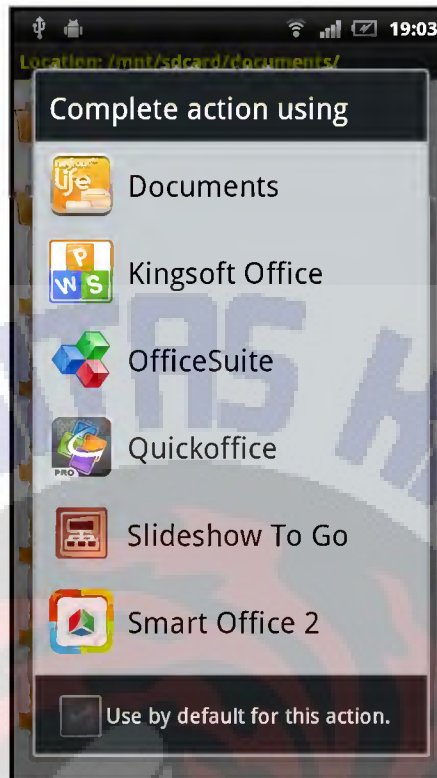
Jika muncul peringatan *Not connected to Server*, pastikan *smartphone* Android sudah terhubung ke *VGA Adapter* sesuai dengan Langkah 4 dan Langkah 5.



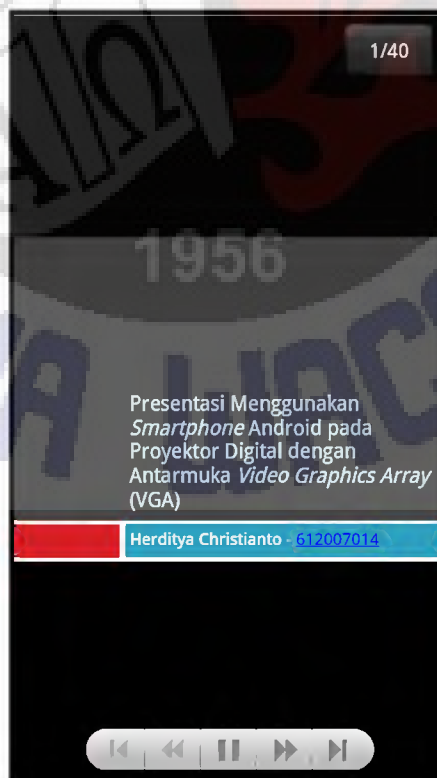
7. Pilih dokumen presentasi lewat *file explorer* yang tersedia.



8. Pilih aplikasi *office* untuk menampilkan presentasi.



9. Presentasi siap ditampilkan.



A.3. Petunjuk Penggunaan Aplikasi *Desktop Presenter*

1. *Install* aplikasi *Java Runtime Environment* (JRE).

Sistem Operasi	Langkah Instalasi
Windows (32-bit)	<ul style="list-style-type: none">✓ Unduh <i>installer</i>: http://download.oracle.com/otn-pub/java/jdk/7u10-b18/jre-7u10-windows-i586.exe✓ Jalankan <i>installer</i>, ikuti instruksi yang tersedia sampai selesai.
Windows (64-bit)	<ul style="list-style-type: none">✓ Unduh <i>installer</i>: http://download.oracle.com/otn-pub/java/jdk/7u10-b18/jre-7u10-windows-x64.exe✓ Jalankan <i>installer</i>, ikuti instruksi yang tersedia sampai selesai.
Linux (Ubuntu)	<ul style="list-style-type: none">✓ Jalankan <i>command</i> berikut dari <i>Terminal</i>: <code>sudo apt-get install openjdk-7-jre</code>

2. Periksa versi dari instalasi JRE lewat *Command Prompt* (Windows) atau *Terminal* (Linux) dengan perintah berikut ini:

```
java -version
```

Bila JRE sudah terpasang dengan benar maka akan muncul keluaran seperti berikut ini (keluaran tergantung pada versi JRE yang di-*install*):

```
java version "1.6.0_37"
```

```
Java(TM) SE Runtime Environment (build 1.6.0_37-b06)
```

```
Java HotSpot(TM) Client VM (build 20.12-b01, mixed mode,  
sharing)
```

3. Hubungkan komputer dengan *VGA Adapter* lewat *Wi-Fi* secara manual. Berikut ini parameter jaringan *Wi-Fi* pada *VGA Adapter*:

✓ SSID : VGA-Adapter_XXXXXX

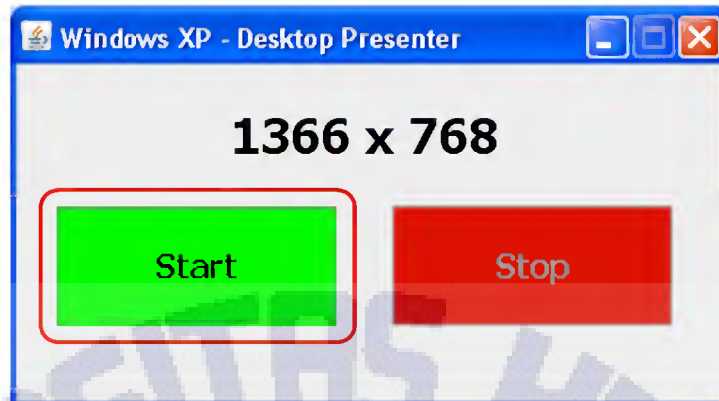
✓ Password : hereditya

4. Jalankan aplikasi *Desktop Presenter*.

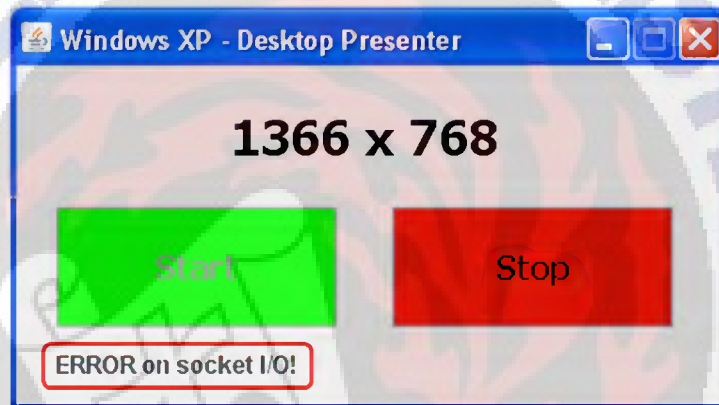
✓ Windows : *Double-click* pada file *Desktop_Presenter.jar*.

✓ Linux : Jalankan *Terminal* pada direktori file *Desktop_Presenter.jar*, jalankan perintah `java -jar Desktop_Presenter.jar`.

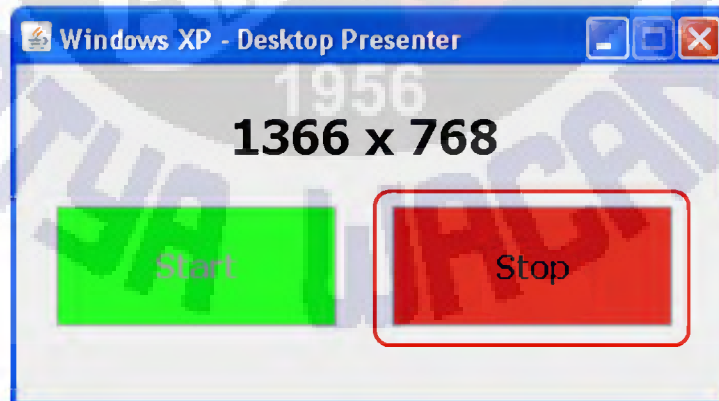
5. Klik *Start* atau tombol warna hijau untuk memulai aplikasi.



Apabila muncul pesan *error* di bagian bawah, pastikan komputer sudah terhubung dengan *VGA Adapter* lewat *Wi-Fi* menggunakan parameter pada Langkah 3.



6. Klik *Stop* atau tombol warna merah untuk menghentikan aplikasi.



LAMPIRAN B

ANDROID NATIVE DEVELOPMENT KIT

B.1. Pengenalan NDK

NDK merupakan *toolset* yang memungkinkan pengembangan aplikasi Android menggunakan bahasa pemrograman *native* seperti C dan C++. NDK berisi seperangkat *cross-toolchains* seperti *compiler* dan *linker* yang dapat menghasilkan *native ARM binaries* untuk *platform* Android. NDK juga berisi seperangkat *system headers* untuk *native API* yang dapat digunakan dalam pengembangan aplikasi.

NDK merupakan bagian dari SDK dan dapat digunakan bersama-sama dengan SDK yang menggunakan bahasa pemrograman Java. NDK dapat digunakan pada bagian aplikasi yang tidak dapat dibangun dengan SDK karena keterbatasan dalam Java, atau pada aplikasi yang memiliki algoritma pemrograman yang kompleks dan sangat bergantung pada kemampuan prosesor. NDK juga memungkinkan penggunaan pustaka-pustaka yang ditulis dalam bahasa C/C++ serta kemudahan dalam *porting* dan *code reuse* di antara *project* Android atau *project* non-Android.

B.2. Instalasi NDK

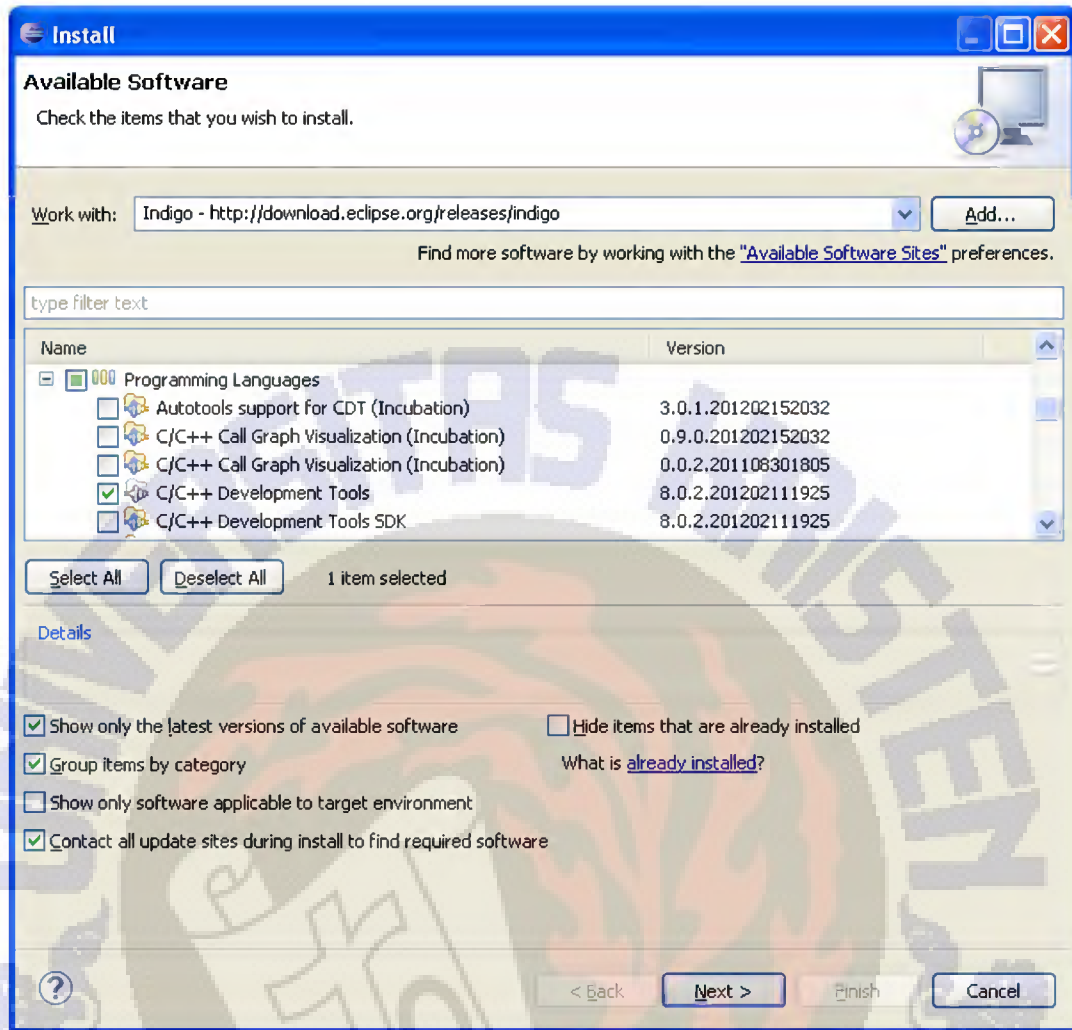
Sebelum instalasi NDK dapat dilakukan, beberapa *software* di bawah ini harus sudah terlebih dahulu ter-*install*:

- *Java Development Kit* (JDK)
- Android SDK
- Eclipse IDE
- Cygwin 1.7 dan GNU Make 3.81 (untuk Windows)

Jika *software-software* di atas sudah terpasang, maka langkah-langkah instalasi NDK adalah sebagai berikut:

1. Instalasi *C/C++ Development Tools* pada Eclipse IDE, pada lampiran ini menggunakan Eclipse Indigo (3.7).

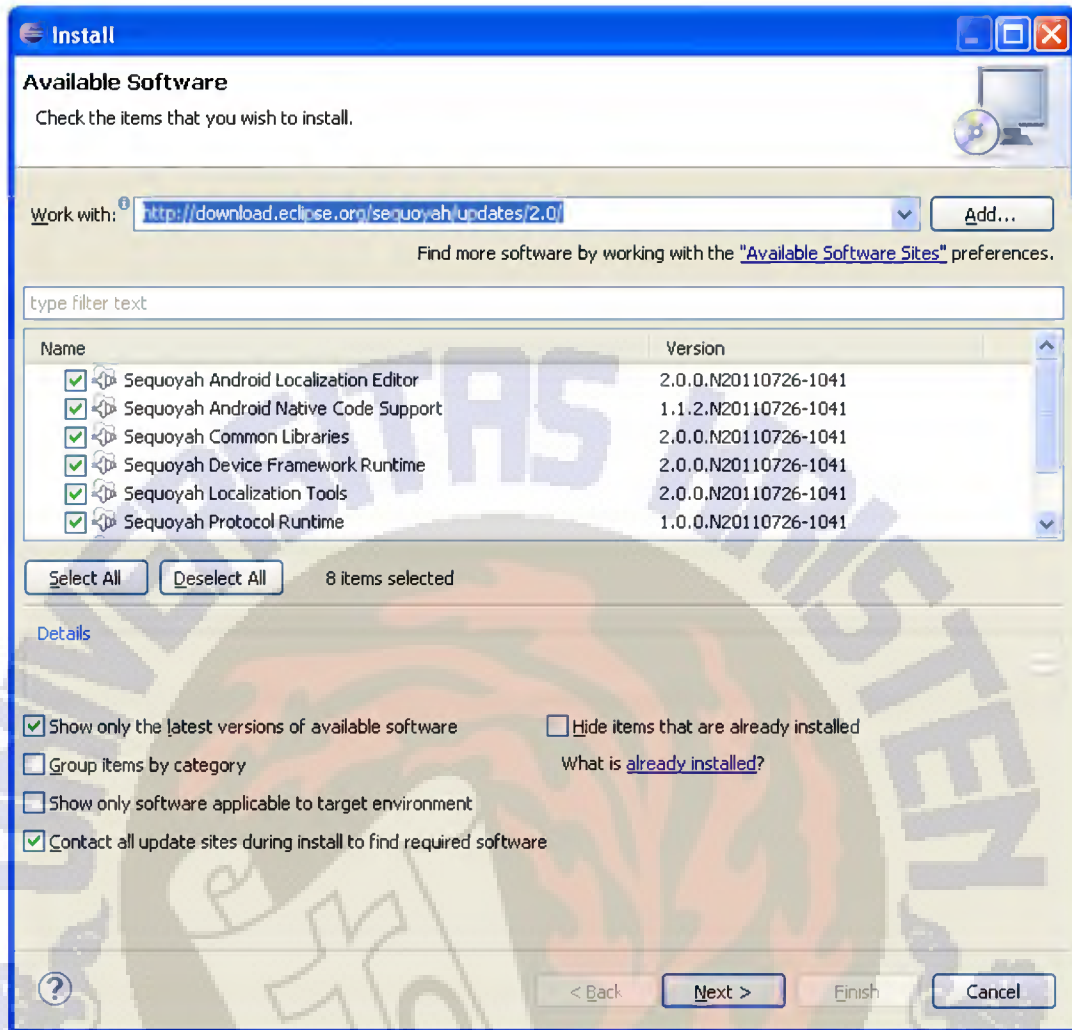
Pada Eclipse IDE, masuk ke menu *Help* → *Install New Software*. Pada kolom *Work with*, pilih *Indigo* (<http://download.eclipse.org/releases/indigo>). Pilih *Programming Languages*, kemudian beri cek pada *C/C++ Development Tools*.



Klik *Next* dan *Install*.

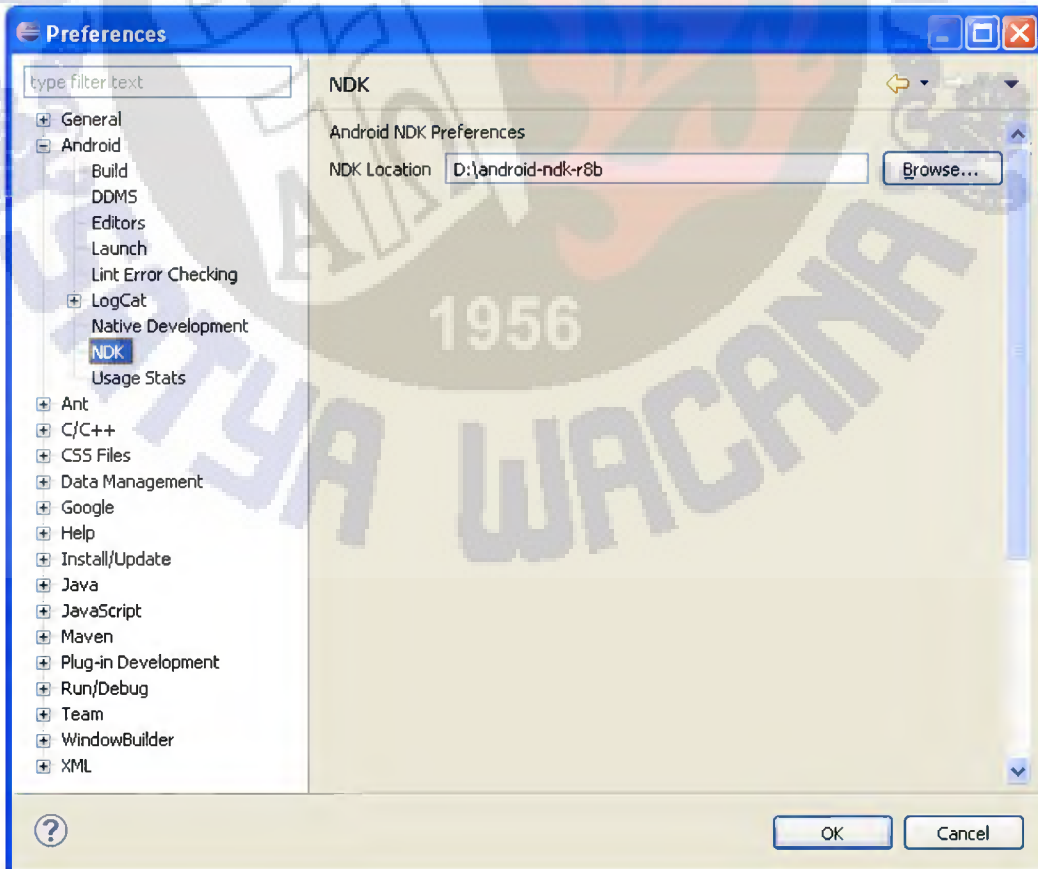
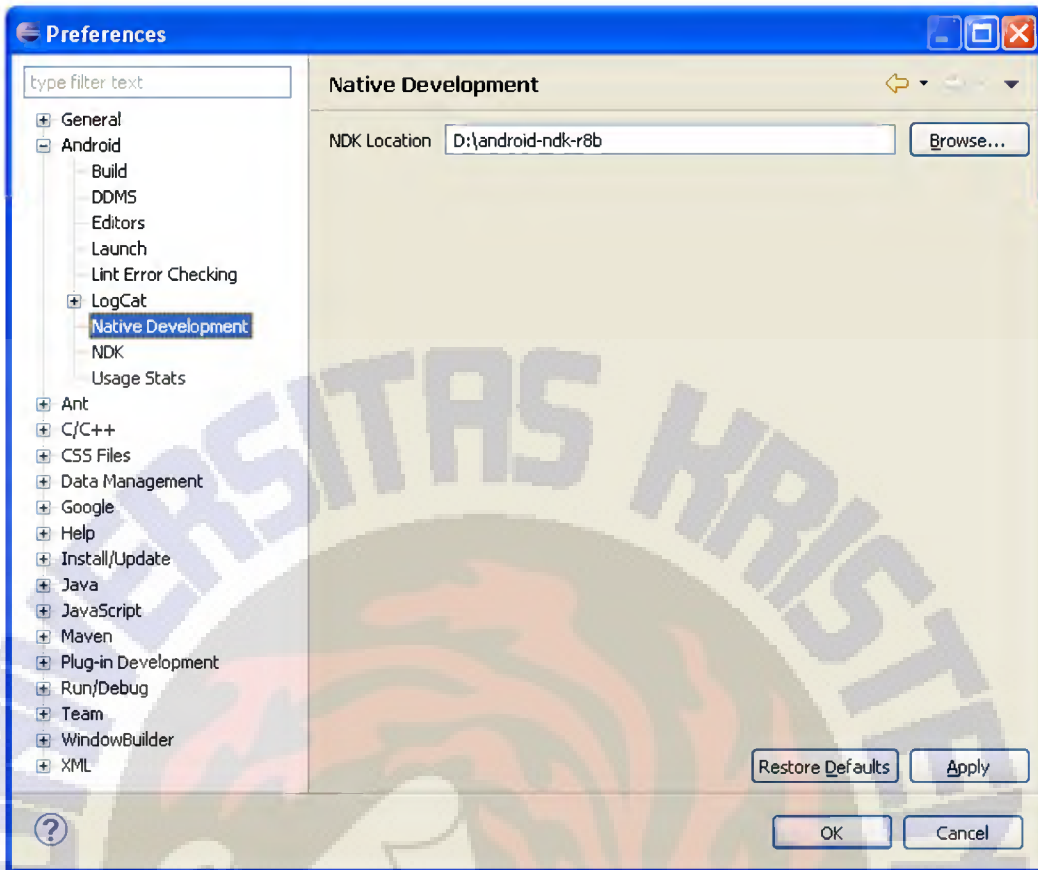
2. Instalasi *plug-in Sequoyah* pada Eclipse IDE. *Plug-in* ini digunakan untuk memudahkan dalam pembuatan *project* dengan NDK.

Pada Eclipse IDE, masuk ke menu *Help* → *Install New Software*. Pada kolom *Work with*, masukkan <http://download.eclipse.org/sequoyah/updates/2.0/>. Hilangkan cek pada *Group items by category*, kemudian klik *Select All*.



Klik *Next* dan *Install*.

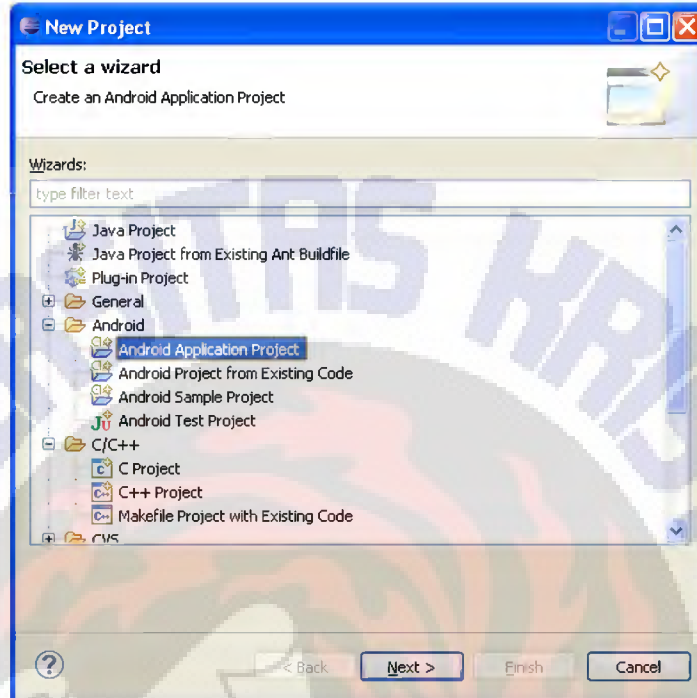
3. Instalasi NDK, pada lampiran ini menggunakan *Android NDK Revision 8b*. NDK dapat diunduh dari <http://developer.android.com/tools/sdk/ndk/index.html>.
4. Ekstrak berkas instalasi NDK ke dalam sebuah direktori, misal `C:\android-ndk-r8b`.
5. Tambahkan direktori instalasi NDK ke dalam *path* pada *system variable*.
 (*Command prompt Windows*) `set PATH=%PATH%; C:\android-ndk-r8b`
 (*Linux Terminal*) `set PATH=${PATH}:/username/android-ndk-r8b`
6. Tambahkan direktori instalasi NDK ke **Android Preferences pada Eclipse IDE**.
 Pada Eclipse IDE, masuk ke menu *Window* → *Preferences* kemudian pilih *Android*. Pada kolom *NDK Location* pada *Native Development* dan *NDK* masing-masing diisi dengan direktori instalasi NDK.



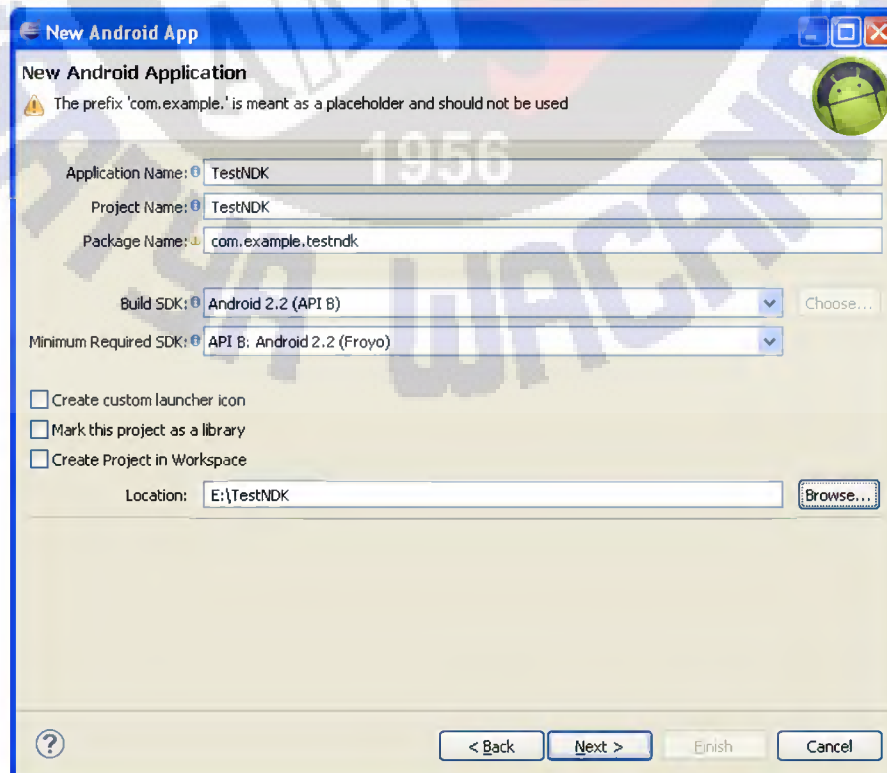
Klik *Apply* kemudian *OK*.

B.3. Pembuatan *Project* dengan NDK

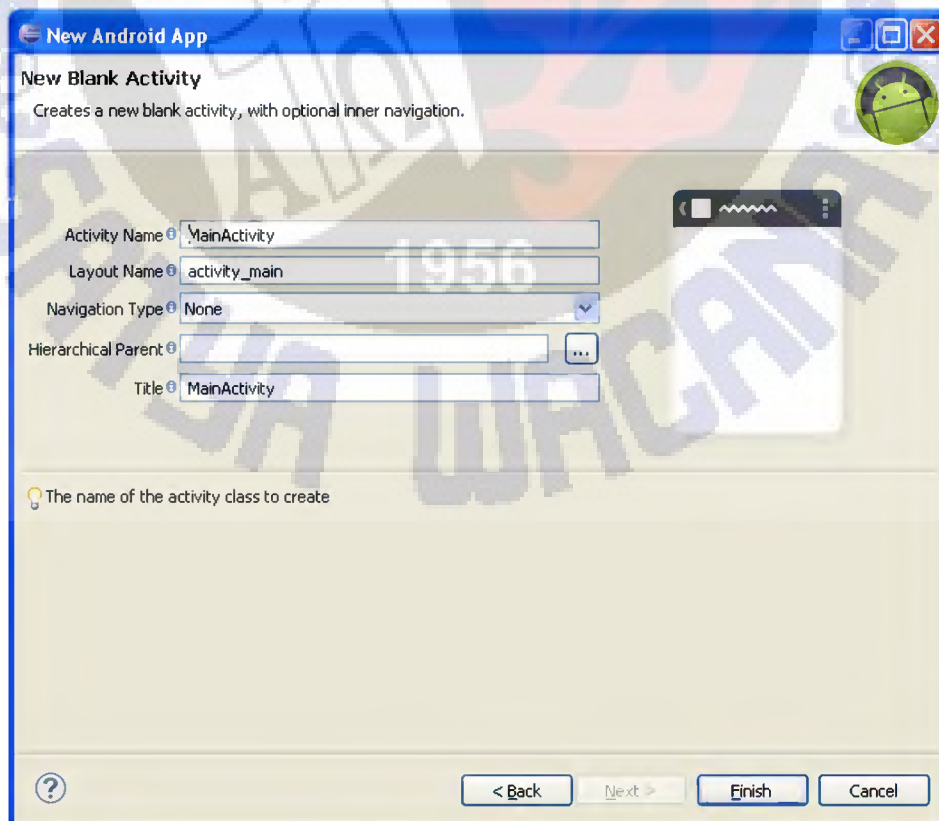
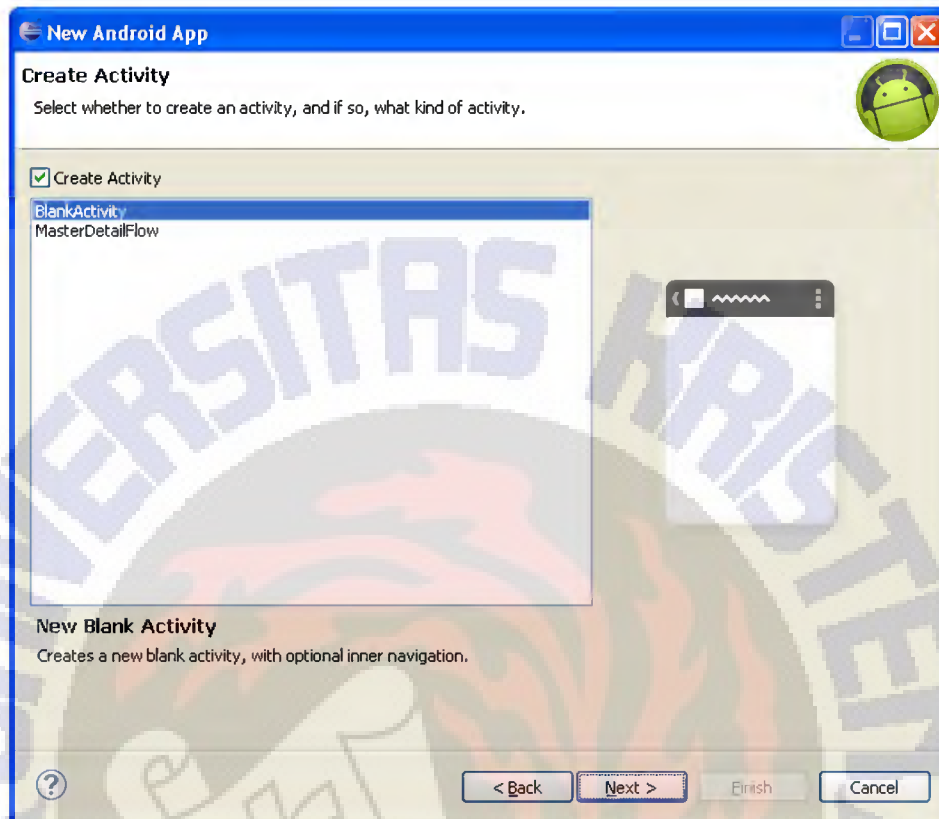
1. Pada Eclipse IDE, masuk ke menu *File* → *New* → *Project* akan muncul jendela *New Project*, pilih *Android Application Project* kemudian klik *Next*.



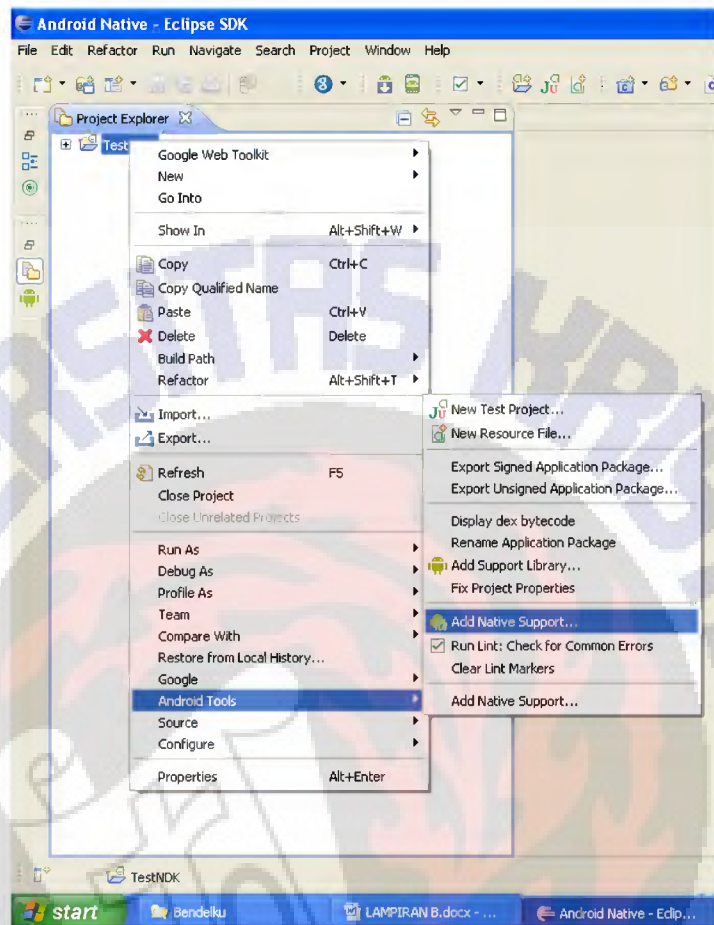
2. Masukkan *Application Name*, *Build SDK* dan *Minimum Required SDK*. Masukkan lokasi penyimpanan *project*, lokasi ini tidak boleh mengandung spasi. Klik *Next*.



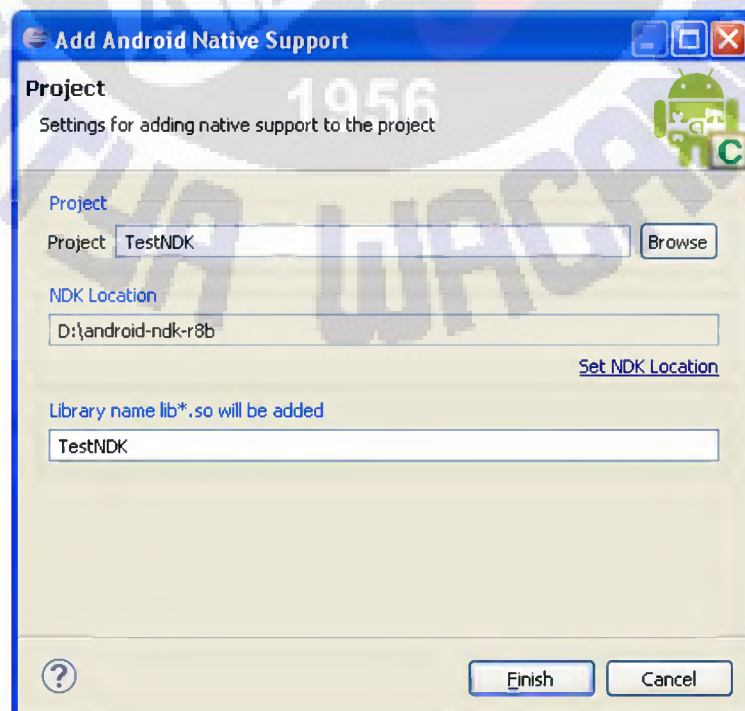
3. Klik *Next* pada *Create Activity*, kemudian klik *Finish* pada *New Blank Activity*.
Project baru sudah berhasil dibuat.



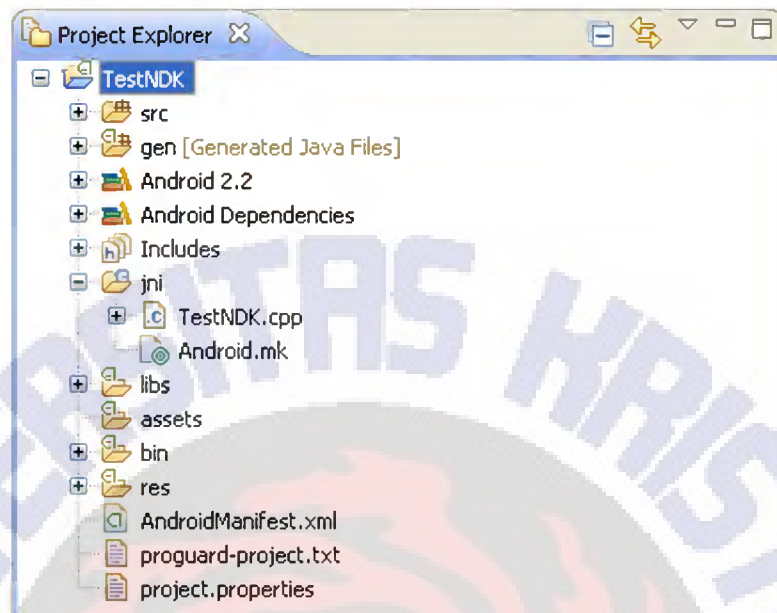
4. Pada *Package Explorer*, klik kanan pada *project* yang baru dibuat, pilih *Android Tools*
→ *Add Native Support*.



Klik *Finish* pada jendela yang muncul.



5. Eclipse IDE akan membuat sebuah *folder* `jni/` pada direktori *project* Android. Secara *default*, *folder* ini akan berisi dua *file* yaitu `TestNDK.cpp` dan `Android.mk`.



B.4. Aplikasi *Hello World* dengan NDK

1. Ganti nama *file* `TestNDK.cpp` dalam *folder* `jni/` menjadi `TestNDK.c`. Ubah isi `TestNDK.c` menjadi seperti di bawah ini:

```
#include <string.h>
#include <jni.h>

JNIEXPORT jstring JNICALL
Java_com_example_testndk_MainActivity_stringFromC(
    JNIEnv * env, jobject obj) {
    return (*env)->NewStringUTF(env, "Hello World from C");
}
```

2. Ubah isi *file* `Android.mk` dalam *folder* `jni/` menjadi seperti di bawah ini:

```
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_MODULE     := TestNDK
LOCAL_SRC_FILES  := TestNDK.c

include $(BUILD_SHARED_LIBRARY)
```

3. Ubah isi *file* `activity_main.xml` dalam *folder* `res/layout/` menjadi seperti di bawah ini:

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent" >

  <TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:text="@string/hello_world"
    tools:context=".MainActivity" />

</RelativeLayout>
```

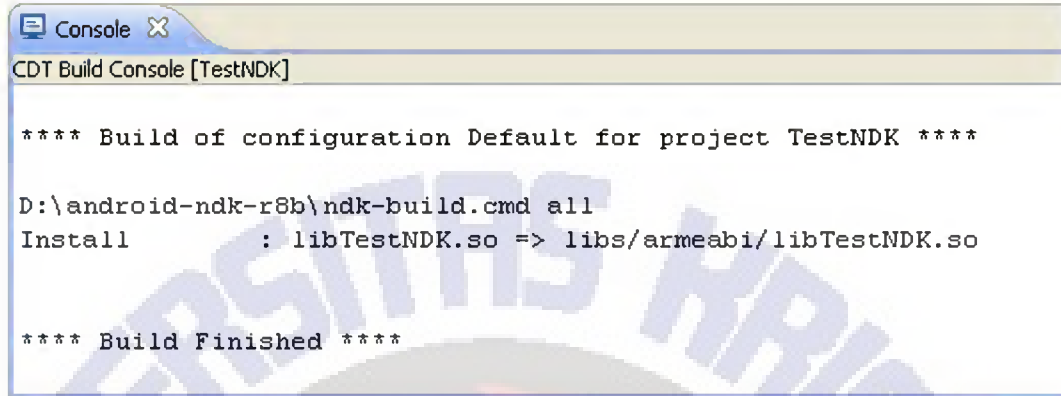
4. Ubah ini *file* `MainActivity.java` dalam *folder* `src/com/example/testndk/` menjadi seperti di bawah ini:

```
package com.example.testndk;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.TextView;

public class MainActivity extends Activity {
    TextView text;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        text = (TextView) findViewById(R.id.textView);
        text.setText(stringFromC());
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
    public native String stringFromC();
    static {
        System.loadLibrary("TestNDK");
    }
}
```

5. *Build project* lewat menu *Project* → *Build All* atau dengan tombol *Ctrl+B*. Buka jendela *Console* lewat menu *Window* → *Show View* → *Console*, jika *build* berhasil dilakukan maka akan muncul pesan sebagai berikut:



```
CDT Build Console [TestNDK]

**** Build of configuration Default for project TestNDK ****

D:\android-ndk-r8b\ndk-build.cmd all
Install      : libTestNDK.so => libs/armeabi/libTestNDK.so

**** Build Finished ****
```

6. *Run project* lewat menu *Run* → *Run* atau dengan tombol *Ctrl+F11*. Berikut ini tampilan aplikasi *Hello World* dengan menggunakan NDK pada *emulator* Android:



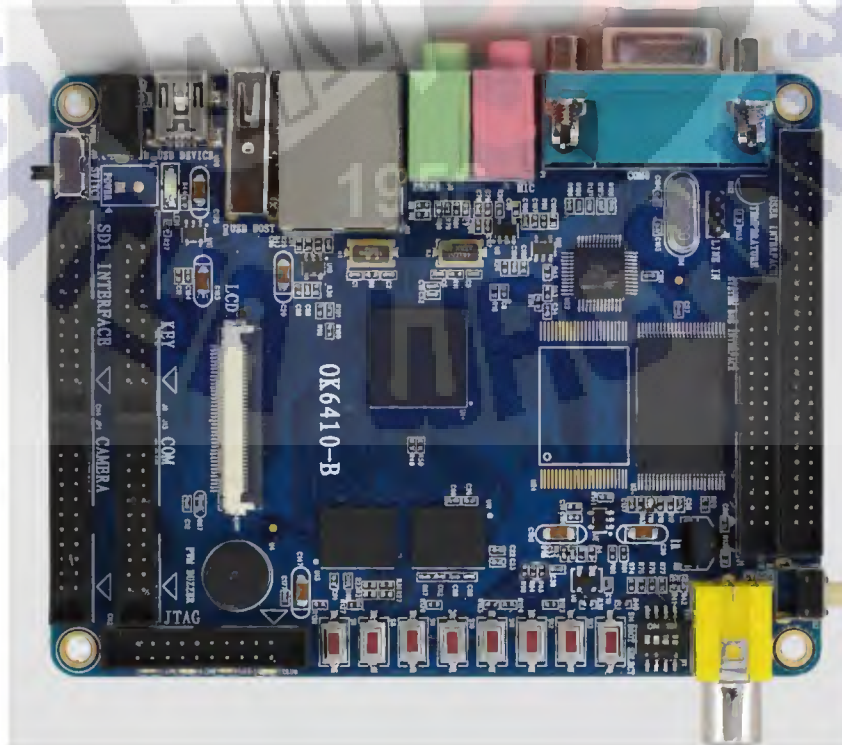
LAMPIRAN C

DEVELOPMENT BOARD OK6410-B

C.1. Pengenalan *Development Board* OK6410-B [15, h.4]

OK6410-B merupakan *development board* untuk *Single Board Computer* (SBC) dari Forlinx Embedded. OK6410-B sendiri merupakan versi *single board* dari varian OK6410. *Development board* ini menggunakan mikroprosesor S3C6410 yang berbasis ARM1176JZF-S *core* dan masih termasuk dalam keluarga ARM11. OK6410-B dapat digunakan untuk keperluan edukasi, aplikasi dan *development system* untuk sistem *embedded* berbasis ARM. Dengan dimensi *board* 81 x 105 mm, OK6410-B adalah *development board* untuk mikroprosesor S3C6410 yang terkecil sampai tulisan ini dibuat.

Karena menggunakan mikroprosesor berbasis ARM11, OK6410-B mendukung sistem operasi yang diperuntukkan untuk perangkat *embedded* yaitu Embedded Linux, Windows CE dan Android. Ketiga sistem operasi tersebut masing-masing telah dilengkapi dengan *Board Support Packages* (BSP) dan *basic drivers* untuk seluruh komponen yang terdapat pada OK6410-B. Dukungan terhadap sistem operasi tersebut memudahkan dalam pembuatan perangkat lunak untuk aplikasi berbasis sistem *embedded*.



***Development Board* OK6410-B**

C.2. Fitur Perangkat Keras OK6410-B [15, h.5]

Berikut ini adalah seluruh fitur perangkat keras yang terdapat pada *development board* OK6410-B:

- PCB 6 *layer*, kinerja stabil dan sudah melewati uji elektromagnetis.
- Samsung S3C6410 *microcontroller* berbasis ARM1176JZF-S *core*, frekuensi kerja @ 533/667 MHz.
- 256 MB *Mobile* DDR RAM.
- 2 GB NAND Flash.
- Sumber detak 12 MHz, 48 MHz, 27 MHz, 32.768 KHz.
- Catu daya 5 V.
- Satu tombol *reset* dengan *chip* khusus.
- 4-bit DIP *switch* untuk pemilihan mode *boot*.
- Empat *port* serial, termasuk satu 5-wire RS232 (DB9) dan tiga 3-wire TTL (konektor 20 pin 2.0mm-pitch).
- Satu *port* Ethernet 100M dengan DM9000AE dan indikator.
- Satu USB *Host Interface* yang mendukung protokol USB1.1.
- Satu USB *Slave* (MINI-USB) *Interface* mendukung protokol USB2.0.
- Satu slot *high-speed* SD Card mendukung SD Memory dan SDIO.
- Satu Wi-Fi *expansion interface*.
- Dua soket standar 3.5mm untuk *Stereo Audio I/O*.
- Satu *audio Line-In interface*.
- LCD dan *touch screen interface* untuk TFT LCD 3.5", 4.3", 5.6", 5.7", 7", 8".
- Satu CVBS *output interface* (PAL/NTSC)
- Satu konektor *CMOS camera* mendukung ITU-R BT601/656 mode 8-bit.
- *Built-in* RTC (*Real Time Clock*) dengan baterai *back-up*.
- Satu JTAG *interface* menggunakan konektor 2 x 10 pin.
- Satu sensor temperature digital (DS18B20).
- Satu *Infrared receiver*.
- Empat *user* LED.

- Satu *buzzer*.
- Satu konektor *keypad* 8x8.
- Satu *port expansion interface* serial 2 x 10 pin. Termasuk tiga *port* serial TTL dan 6 GPIO.
- Satu 2 x 15 pin *system bus interface*.
- Satu 2 x 25 GPIO *interface*.

C.3. Instalasi Embedded Linux 3.0.1 pada OK6410-B [16, h.10-18]

(petunjuk pada bagian ini dilakukan pada Windows XP)

Berikut ini langkah-langkah untuk instalasi sistem operasi Embedded Linux 3.0.1 pada *development board* OK6410-B menggunakan SD Card:

1. Format SD Card dengan format FAT32.
2. Jalankan program `SD_Writer.exe`.



3. Klik *Scan*, lalu pilih lokasi SD Card pada pilihan *SD Volume*.
4. Pada pilihan *SD Type*, pilih *Auto*.
5. Pada pilihan *OS Type*, pilih *Linux*.

6. Klik *Select Boot*, pada jendela yang muncul cari file *mmc.bin*, klik *Open*.



7. Klik *Program*, jika sukses maka akan muncul pesan “It’s OK”, klik *Quit*.



8. Salin tiga file *u-boot.bin* (*bootloader*), *zImage* (*kernel*), *cramfs* (*filesystem*) ke dalam SD Card.

9. Masukkan SD Card ke OK6410-B.

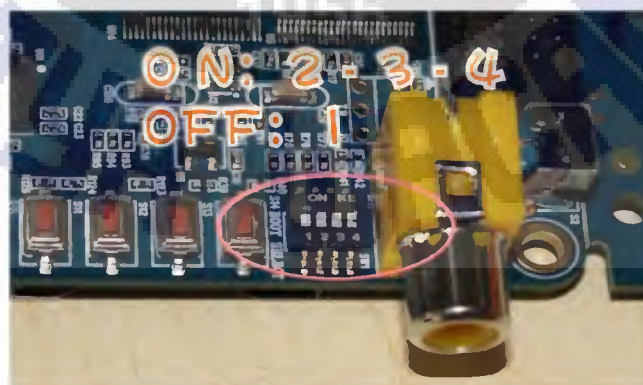


10. Pasang catu daya 5 V pada *development board* OK6410-B.



11. Atur DIP *switch* untuk masuk ke mode *boot* SD Card seperti pada tabel berikut:

Nomor Pin	Pin 1	Pin 2	Pin 3	Pin4
Nilai Pin	0	1	1	1



12. Hubungkan OK6410-B dengan komputer melalui kabel serial.



13. Jalankan program `dnw.exe`.

Atur konfigurasi serial pada menu *Configuration* → *Options*, pilih *Baud Rate* 115200 dan *COM port* sesuaikan dengan nomor *port* serial komputer.

Lalu klik menu *Serial Port* → *Connect*.

14. Nyalakan *power switch*, pada program DNW akan muncul tampilan seperti berikut:

```
U-Boot 1.1.6 (May 13 2011 - 16:11:02) for SMDK6410
*****|
**   u-boot 1.1.6   **
**   Updated for TE6410 Board   **
**   Version 1.0 (10-01-15)   **
**   OEM: Forlinx Embedded   **
**   Web: http://www.witech.com.cn   **
*****|

CPU:      S3C6410 @532MHz
          Fclk = 532MHz, Hclk = 133MHz, Pclk = 66MHz, Serial = CLKUART (SYNC
Mode)
Board:    SMDK6410
DRAM:     256 MB
Flash:    0 kB
NAND:     select s3c_nand_oob_mlc_128
2048 MB
```


15. Proses instalasi akan segera berjalan, tunggu hingga seluruh proses selesai.

Jika proses instalasi sudah selesai maka akan muncul tampilan seperti berikut:

```
Writing data at 0x1f85000 -- 86
Writing data at 0x1fd7000 -- 87
Writing data at 0x2028800 -- 88
Writing data at 0x207a800 -- 89
Writing data at 0x20cc800 -- 90
Writing data at 0x211e800 -- 91
Writing data at 0x2170800 -- 92
Writing data at 0x21c2800 -- 93
Writing data at 0x2214000 -- 94
Writing data at 0x2266000 -- 95
Writing data at 0x22b8000 -- 96
Writing data at 0x230a000 -- 97
Writing data at 0x235c000 -- 98
Writing data at 0x23ae000 -- 99
Writing data at 0x23ff800 -- 100bytes written: OK
SMDK6410 #
```

16. Matikan *power switch*, atur *DIP switch* untuk masuk ke mode *boot NAND Flash* seperti pada tabel berikut:

Nomor Pin	Pin 1	Pin 2	Pin 3	Pin4
Nilai Pin	0	1	0	0



17. Nyalakan *power switch*, sistem operasi Embedded Linux siap untuk digunakan.

C.4. Instalasi *Cross-Compiler* GCC 4.2.2 [16, h.101]

(petunjuk pada bagian ini dilakukan pada *Linux Ubuntu*)

Cross-compiler GCC 4.2.2 dapat digunakan untuk pembuatan program serta kompilasi *kernel* dan *bootloader* pada *development board* OK6410-B. Berikut ini langkah-langkah instalasi *cross-compiler* GCC 4.2.2 pada komputer dengan sistem operasi *Linux Ubuntu*:

1. Jalankan program *Terminal* *Linux*.
2. Masuk sebagai *superuser* dengan perintah berikut ini:

```
sudo su
```

Masukkan *password* administrator.

3. Buat *folder* baru pada direktori `/usr/local/arm` dengan perintah berikut ini:

```
mkdir /usr/local/arm
```

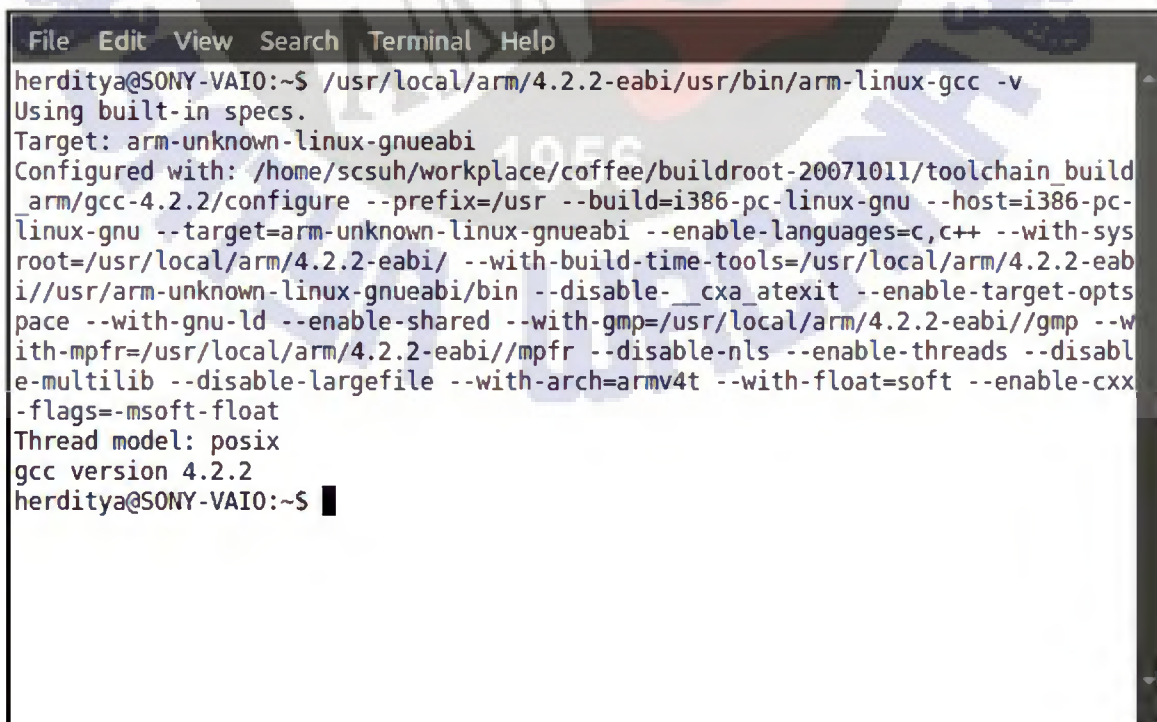
4. Ekstrak *file* `cross-4.2.2-eabi.tar.bz2` dengan perintah berikut ini:

```
tar jxvf cross-4.2.2-eabi.tar.bz2 -C /usr/local/arm/
```

5. Masukkan perintah berikut ini untuk memeriksa instalasi *cross-compiler*:

```
/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-gcc -v
```

Jika instalasi berhasil maka akan muncul tampilan seperti berikut:



```
File Edit View Search Terminal Help
herditya@SONY-VAIO:~$ /usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-gcc -v
Using built-in specs.
Target: arm-unknown-linux-gnueabi
Configured with: /home/scsuh/workplace/coffee/buildroot-20071011/toolchain_build_arm/gcc-4.2.2/configure --prefix=/usr --build=i386-pc-linux-gnu --host=i386-pc-linux-gnu --target=arm-unknown-linux-gnueabi --enable-languages=c,c++ --with-sysroot=/usr/local/arm/4.2.2-eabi/ --with-build-time-tools=/usr/local/arm/4.2.2-eabi//usr/arm-unknown-linux-gnueabi/bin --disable-_cxa_atexit --enable-target-optspace --with-gnu-ld --enable-shared --with-gmp=/usr/local/arm/4.2.2-eabi//gmp --with-mpfr=/usr/local/arm/4.2.2-eabi//mpfr --disable-nls --enable-threads --disable-multilib --disable-largefile --with-arch=armv4t --with-float=soft --enable-cxx-flags=-msoft-float
Thread model: posix
gcc version 4.2.2
herditya@SONY-VAIO:~$ █
```

6. Tambahkan direktori instalasi *cross-compiler* ke dalam *path* pada *system variable* dengan perintah berikut:

```
gedit /etc/profile.d/arm.sh
```

Jendela program *gedit* akan terbuka, tuliskan perintah berikut ini:

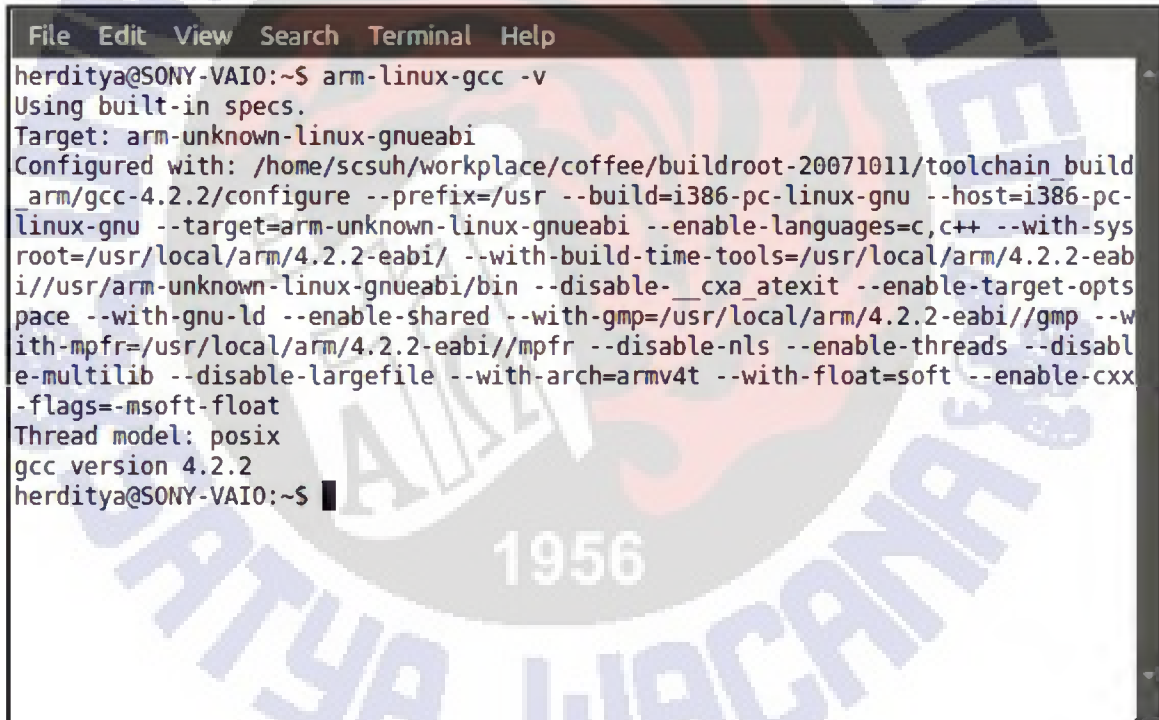
```
export PATH=$PATH:/usr/local/arm/4.2.2-eabi/usr/bin
```

Simpan dengan masuk ke menu *File* → *Save* atau dengan tombol *Ctrl+S*.

7. *Restart* komputer.
8. Masukkan perintah berikut dalam *Terminal*:

```
arm-linux-gcc -v
```

Apabila muncul tampilan seperti pada Langkah 5 maka *cross-compiler* sudah siap untuk digunakan.



```
File Edit View Search Terminal Help
herditya@SONY-VAIO:~$ arm-linux-gcc -v
Using built-in specs.
Target: arm-unknown-linux-gnueabi
Configured with: /home/scsuh/workplace/coffee/buildroot-20071011/toolchain_build
_arm/gcc-4.2.2/configure --prefix=/usr --build=i386-pc-linux-gnu --host=i386-pc-
linux-gnu --target=arm-unknown-linux-gnueabi --enable-languages=c,c++ --with-sys
root=/usr/local/arm/4.2.2-eabi/ --with-build-time-tools=/usr/local/arm/4.2.2-eab
i//usr/arm-unknown-linux-gnueabi/bin --disable-__cxa_atexit --enable-target-opts
pace --with-gnu-ld --enable-shared --with-gmp=/usr/local/arm/4.2.2-eabi//gmp --w
ith-mpfr=/usr/local/arm/4.2.2-eabi//mpfr --disable-nls --enable-threads --disabl
e-multilib --disable-largefile --with-arch=armv4t --with-float=soft --enable-cxx
-flags=-msoft-float
Thread model: posix
gcc version 4.2.2
herditya@SONY-VAIO:~$
```

C.5. Aplikasi *Hello World* pada OK6410-B [16, h.110-112]

(petunjuk pada bagian ini dilakukan pada *Linux Ubuntu*)

1. Buat sebuah dokumen teks baru, beri nama `hello.c`. Tuliskan kode berikut pada *file*

`hello.c`:

```
#include <stdio.h>

int main() {
    printf("Hello World from OK6410-B");
}
```

2. Jalankan *Terminal* dari lokasi direktori *file* `hello.c`, masukkan perintah berikut untuk meng-*compile file* `hello.c`:

```
arm-linux-gcc hello.c -o hello
```

Apabila kompilasi berhasil akan muncul sebuah *file* baru bernama `hello` di direktori yang sama dengan `hello.c`.

3. Salin *file* `hello` ke dalam SD Card, masukkan SD Card ke dalam OK6410-B.
4. Hubungkan OK6410-B dengan komputer melalui kabel serial.
Nyalakan *power switch* OK6410-B.
5. Tunggu hingga *booting* OK6410-B selesai, masuk ke *command shell* OK6410-B.
6. Masukkan dua baris perintah berikut ini untuk menjalankan program *Hello World*:

```
cd sdcard
./hello
```

Hasilnya akan tampak seperti gambar berikut:



```
/dev/ttyUSB0 - PuTTY
[root@FORLINUX6410]# cd sdcard
[root@FORLINUX6410]# ./hello
Hello World from OK6410-B
[root@FORLINUX6410]# █
```

C.6. Pengaturan Resolusi *Display* VGA pada OK6410-B [16, h.104]

(petunjuk pada bagian ini dilakukan pada *Linux Ubuntu*)

Pengaturan resolusi *display* VGA dapat dilakukan dengan mengkompilasi ulang *kernel* untuk OK6410-B. Langkah-langkahnya adalah sebagai berikut:

1. Jalankan program *Terminal* Linux.
2. Masuk sebagai *superuser* dengan perintah berikut ini:

```
sudo su
```

Masukkan *password* administrator.

3. Ekstrak *file* FORLINUX_linux-3.0.1.tar.gz dengan perintah berikut ini:

```
tar zxf FORLINUX_linux-3.0.1.tar.gz
```

Akan muncul *folder* baru dengan nama linux-3.0.1.

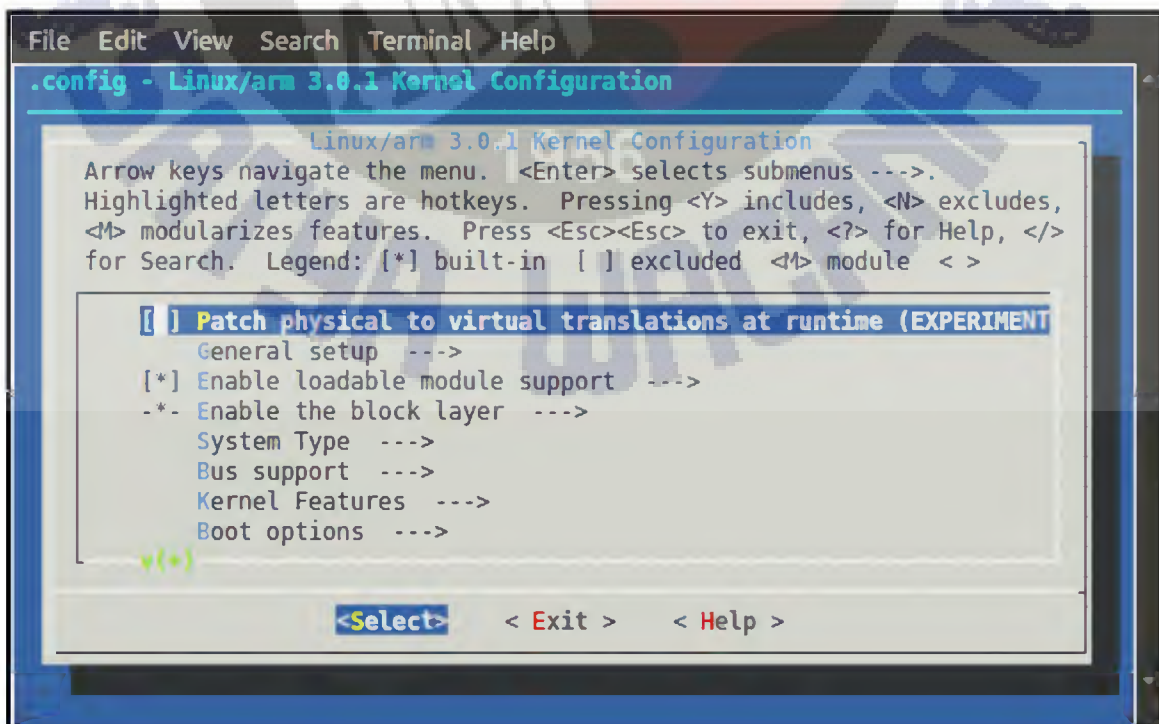
4. Masuk ke *folder* linux-3.0.1 dengan perintah berikut ini:

```
cd linux-3.0.1
```

5. Jalankan perintah berikut ini untuk menampilkan menu konfigurasi *kernel*:

```
make menuconfig
```

Akan muncul sebuah *user interface* dengan menu utama seperti berikut ini:



6. Untuk pengaturan resolusi *display* VGA, dari menu utama masuk ke *Device Drivers* → *Graphics support* → *Select LCD Type*.

```
File Edit View Search Terminal Help
.config - Linux/arm 3.0.1 Kernel Configuration

Linux/arm 3.0.1 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

^(-)
  Boot options --->
  CPU Power Management --->
  Floating point emulation --->
  Userspace binary formats --->
  Power management options --->
  [*] Networking support --->
  Device Drivers --->
  File systems --->

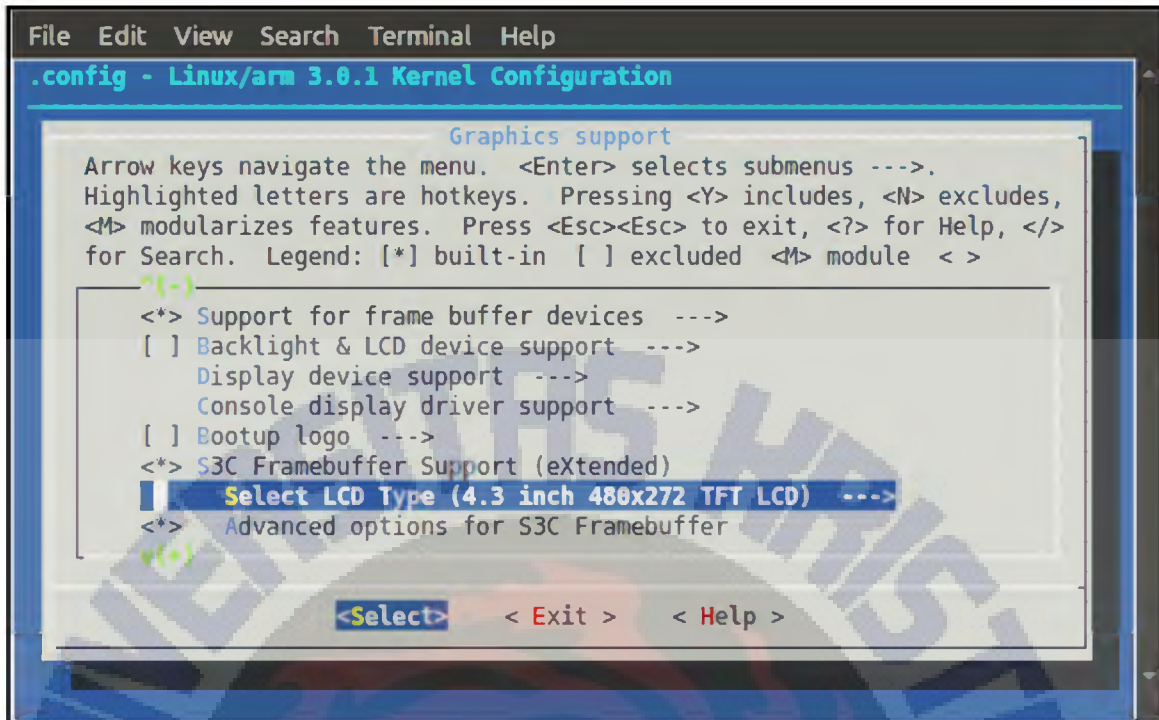
v(+)
  <Select> < Exit > < Help >
```

```
File Edit View Search Terminal Help
.config - Linux/arm 3.0.1 Kernel Configuration

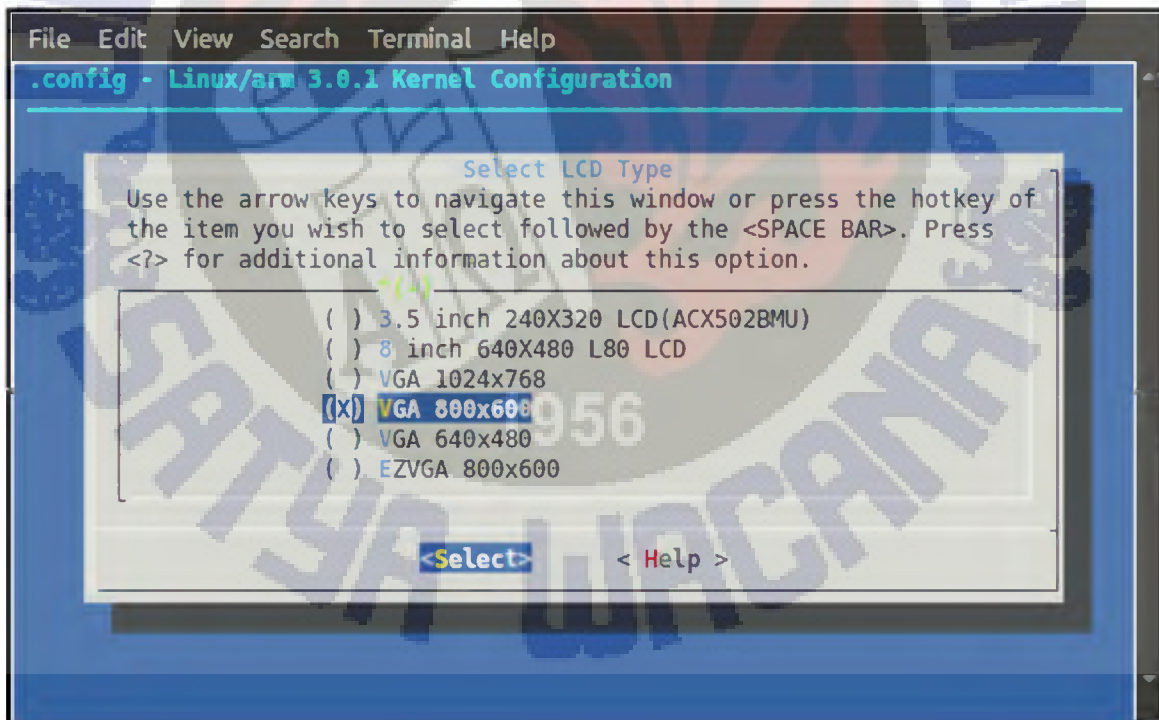
Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

^(-)
  [*] Watchdog Timer Support --->
  Sonics Silicon Backplane --->
  Broadcom specific AMBA --->
  [ ] Multifunction device drivers --->
  [ ] Voltage and Current Regulator Support --->
  <*> Multimedia support --->
  Graphics support --->
  <*> Sound card support --->

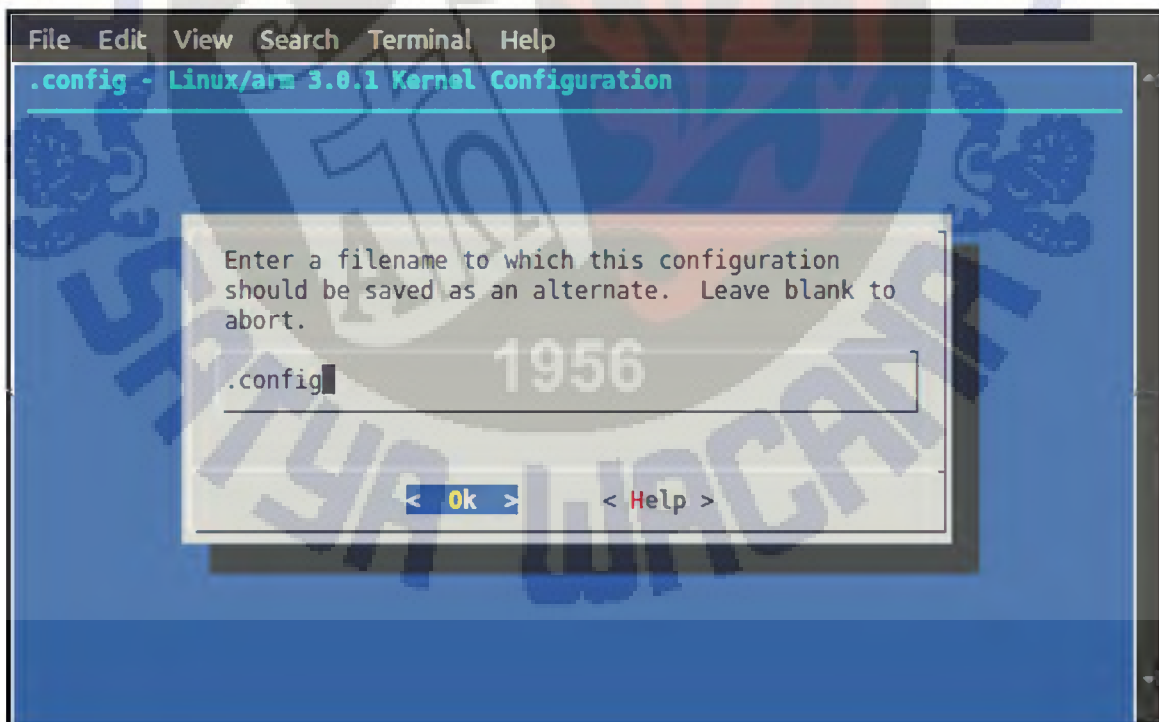
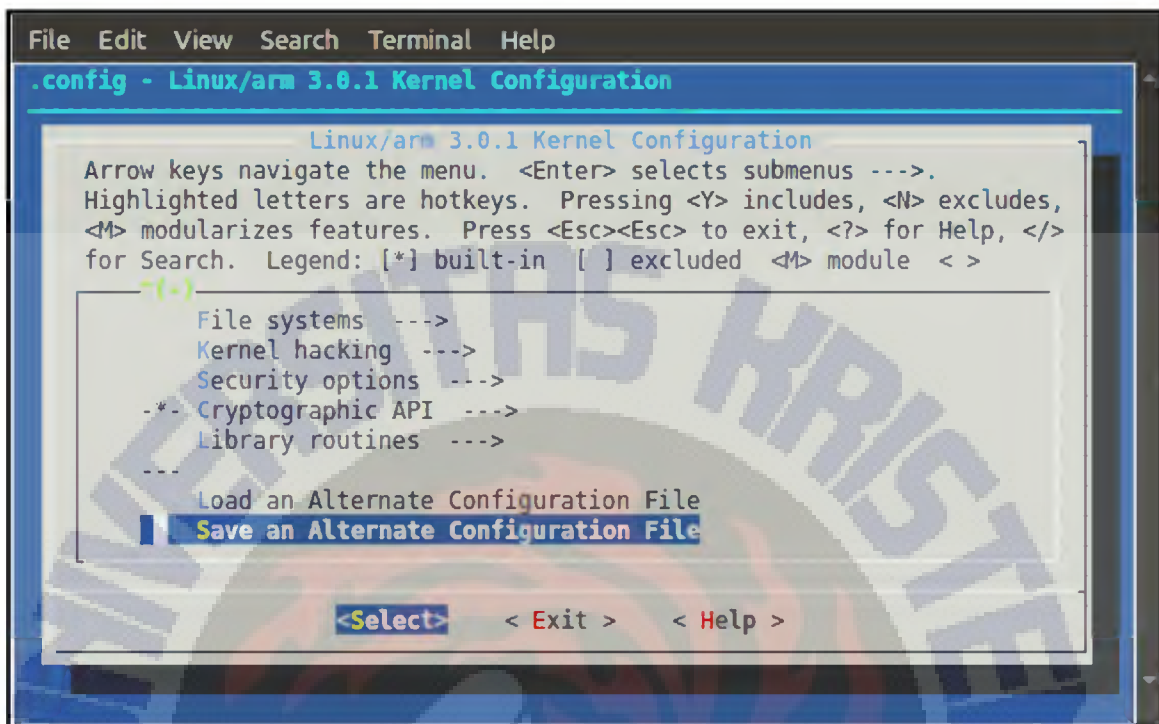
v(+)
  <Select> < Exit > < Help >
```



Pada jendela *Select LCD Type*, pilih *VGA 800x600*.



7. Kembali ke menu utama, pilih *Save an Alternate Configuration Type*, kemudian *Ok* pada jendela yang muncul.



8. Pilih *Exit* untuk keluar dari *user interface* dan kembali ke *Terminal*.
9. Masukkan perintah berikut ini untuk meng-*compile* konfigurasi *kernel*:

```
make zImage
```

Proses kompilasi *kernel* akan segera berjalan seperti berikut ini:

```
File Edit View Search Terminal Help
CC arch/arm/plat-samsung/dev-video.o
CC arch/arm/plat-samsung/dev-pwm.o
CC arch/arm/plat-samsung/dev-fimc0.o
CC arch/arm/plat-samsung/dev-fimc1.o
CC arch/arm/plat-samsung/setup-fimc0.o
CC arch/arm/plat-samsung/setup-fimc1.o
CC arch/arm/plat-samsung/dma.o
CC arch/arm/plat-samsung/wakeup-mask.o
LD arch/arm/plat-samsung/built-in.o
CC kernel/sched.o
CC kernel/fork.o
CC kernel/exec_domain.o
CC kernel/panic.o
CC kernel/printk.o
CC kernel/cpu.o
CC kernel/exit.o
CC kernel/itimer.o
TIMEC kernel/timeconst.h
CC kernel/time.o
CC kernel/softirq.o
CC kernel/resource.o
```

10. Jika proses kompilasi telah selesai maka tampilan pada *Terminal* seperti berikut ini:

```
File Edit View Search Terminal Help
LD .tmp_vmlinux1
KSYM .tmp_kallsyms1.S
AS .tmp_kallsyms1.o
LD .tmp_vmlinux2
KSYM .tmp_kallsyms2.S
AS .tmp_kallsyms2.o
LD vmlinux
SYSMAP System.map
SYSMAP .tmp_System.map
OBJCOPY arch/arm/boot/Image
Kernel: arch/arm/boot/Image is ready
AS arch/arm/boot/compressed/head.o
GZIP arch/arm/boot/compressed/piggy.gzip
AS arch/arm/boot/compressed/piggy.gzip.o
CC arch/arm/boot/compressed/misc.o
CC arch/arm/boot/compressed/decompress.o
SHIPPED arch/arm/boot/compressed/lib1funcs.S
AS arch/arm/boot/compressed/lib1funcs.o
LD arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
herditya@SONY-VAIO:~/Desktop/linux-3.0.1$
```

11. *File* hasil kompilasi *kernel* yaitu *zImage* berada di direktori `/arch/arm/boot` di dalam *folder* `linux-3.0.1`.

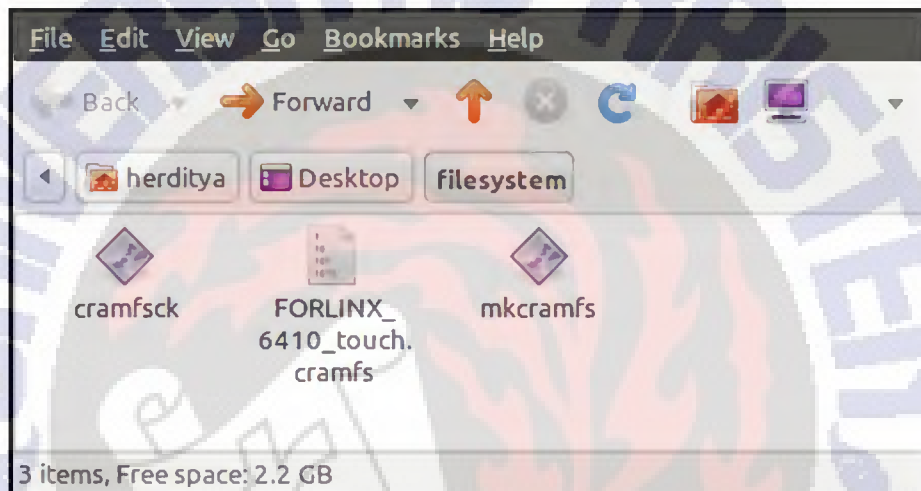
12. *Install* OK6410-B dengan *zImage* hasil kompilasi di atas seperti pada petunjuk **Instalasi Embedded Linux 3.0.1 pada OK6410-B** di bagian sebelumnya.

C.7. Pengaturan Program yang Dijalankan Saat *Startup* [16, h.182-183]

(petunjuk pada bagian ini dilakukan pada *Linux Ubuntu*)

Sebuah program dapat diatur untuk dijalankan secara otomatis saat *startup* atau saat OK6410-B selesai *booting*. Pengaturan ini dilakukan dengan memodifikasi *filesystem* untuk mendaftarkan program yang akan dijalankan saat *startup*. Langkah-langkah yang perlu dilakukan adalah sebagai berikut:

1. Salin *file* `cramfsck`, `mkcramfs` dan `FORLINUX_6410_touch.cramfs` ke dalam sebuah *folder*.



2. Jalankan program *Terminal* Linux.
3. Masuk sebagai *superuser* dengan perintah berikut ini:

```
sudo su
```

Masukkan *password* administrator.

4. Atur *permission* untuk *file* `cramfsck` dan `mkcramfs` dengan perintah sebagai berikut:

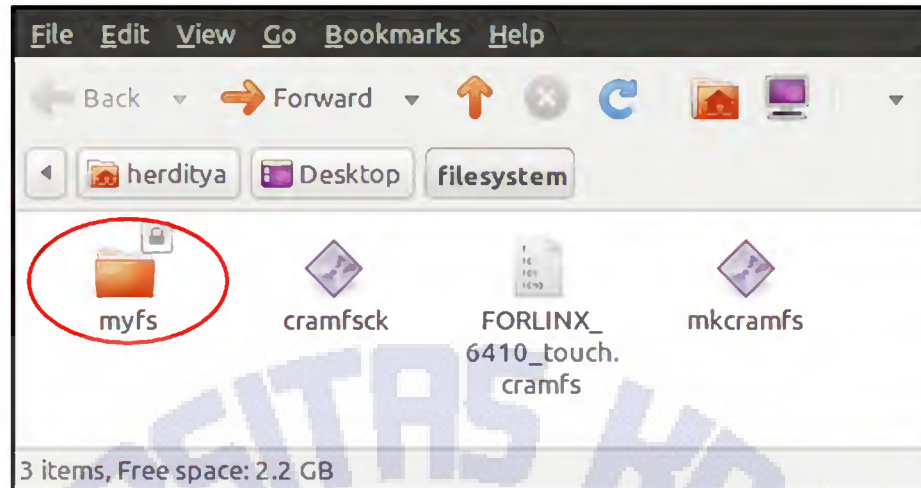
```
chmod 777 cramfsck
```

```
chmod 777 mkcramfs
```

5. Ekstrak *file* `FORLINUX_6410_touch.cramfs` ke dalam sebuah *folder*, misal `/myfs` dengan perintah berikut ini:

```
./cramfsck -x myfs FORLINUX_6410_touch.cramfs
```

Akan muncul sebuah *folder* baru dengan nama `myfs` yang berisi *filesystem* untuk Embedded Linux 3.0.1.



6. Buka *file* rcS yang ada di dalam *folder* /myfs/etc/init.d dengan *text editor*.

Misal program *hello* akan dijalankan saat *startup*, tambahkan baris perintah berikut di bagian akhir *file* rcS:

```
./sdcards/hello &
```

*Tambahkan tanda '#' pada baris /bin/qtopia & untuk menonaktifkan Qtopia (opsional).

*File program *hello* harus disalin ke dalam SD Card pada OK6410-B!

7. Simpan *file* rcS lewat menu *File* → *Save* atau dengan tombol Ctrl+S.

8. Kompilasi *folder* myfs menjadi *file* cramfs dengan perintah berikut:

```
./mkcramfs myfs/ cramfs
```

Jika kompilasi berhasil maka akan muncul *file* cramfs dan tampilan pada *Terminal* seperti berikut:

```
File Edit View Search Terminal Help
root@SONY-VAIO:/home/herditya/Desktop/filesystem# ./mkcramfs myfs/ cramfs
Directory data: 247400 bytes
Everything: 104048 kilobytes
Super block: 76 bytes
CRC: 788cc5cd
root@SONY-VAIO:/home/herditya/Desktop/filesystem# █
```

9. *Install* OK6410-B dengan *cramfs* hasil kompilasi di atas seperti pada petunjuk **Instalasi Embedded Linux 3.0.1 pada OK6410-B** di bagian sebelumnya.