**5. Mempersiapkan Build Environment**

* Spesifikasi Minimal Sistem

Untuk bisa meng-compile OS Android, sistem yang kita gunakan harus memenuhi beberapa syarat di bawah ini :

* + Menggunakan OS Linux atau Mac
  + Untuk OS Linux disarankan Ubuntu 10.04 ke atas
  + OS 64 bit dibutuhkan untuk meng-compile Android versi Gingerbread (2.3) dan ke atas
  + Sudah terinstal Phyton 2,5 – 2,7
  + Sudah terinstal GNU Make 3.81 – 3.82
  + JDK 6 untuk meng-compile Android versi Gingerbread ke atas dan JDK 5 untuk meng-compile Android versi Froyo dan sebelumnya
  + Git 1.7 atau yang lebih mutakhir.
* Download Package yang Dibutuhkan

Setelah syarat umum di atas terpenuhi, maka selanjutnya kita perlu menginstal beberapa package yang dibutuhkan pada saat proses compiling nanti.

**Untuk Ubuntu 10.04 – 11.10**

Buka aplikasi Terminal, lalu ketikkan :

sudo apt-get install git-core gnupg flex bison gperf build-essential \

zip curl zlib1g-dev libc6-dev lib32ncurses5-dev ia32-libs \

x11proto-core-dev libx11-dev lib32readline5-dev lib32z-dev \

libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown \

libxml2-utils xsltproc sun-java6-jdk

Lalu jika anda menggunakan Ubuntu 10.10, ketikkan :

$ sudo ln -s /usr/lib32/mesa/libGL.so.1 /usr/lib32/mesa/libGL.so

Jika anda menggunakan Ubuntu 11.04, ketikkan :

$ sudo apt-get install libx11-dev:i386

**Untuk Ubuntu 12.04 – 12.10**

Buka aplikasi Terminal, lalu ketikkan :

$ sudo apt-get install git-core gnupg flex bison gperf build-essential \

zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev \

libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 \

libgl1-mesa-dev g++-multilib mingw32 openjdk-6-jdk tofrodos \

python-markdown libxml2-utils xsltproc zlib1g-dev:i386

$ sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gnu/libGL.so

* Instal SDK Android

Setelah kita selesai meng-instal package yang dibutuhkan, kita perlu meng-instal SDK Android juga agar kita bisa berkomunikasi dengan device yang kita gunakan, maupun menggunakan emulator Android bawaan dari SDK tersebut.

* + Pertama kita harus men-download file : <http://dl.google.com/android/android-sdk_r21-linux.tgz>
  + Lalu ekstrak file tersebut di direktori home anda
  + Ubah nama folder hasil ekstrak tadi menjadi android-sdk

SDK Android sudah ter-instal di folder ~/android-sdk.

Kita masih perlu melakukan beberapa hal agar mempermudah akses SDK Android melalui Terminal. Buka aplikasi Terminal, pastikan anda berada di direktori ~ , lalu ketikkan gedit .bashrc , tambahkan beberapa line berikut ini pada file .bashrc yang dibuka dengan Text Editor tadi :

#AndroidDev PATH

export PATH=${PATH}:/home/user/android-sdk/platform-tools

export PATH=${PATH}:/home/user/android-sdk/tools

lalu save file tersebut. Langkah ini bertujuan agar kita bisa mengakses ADB, Fastboot, dan DDMS melalui terminal secara langsung tanpa harus menulis path di mana ketiga program ini berada. Namun secara default, SDK Android yang di-download tadi tidak menyediakan ADB, sehingga kita perlu men-downloadnya melalui Android SDK Manager.

Pada Terminal, masuk ke folder ~/android-sdk /tools, lalu ketik ./android, maka Android SDK Manager akan muncul.Setelah semua repository tersinkron, install/download Android SDK Platform Tools, setelah semua proses selesai, maka Platform Tools tersebut telah terinstal.

Selanjutnya kita perlu memberikan ijin kepada program ADB agar bisa mengakses port USB, karena pada Ubuntu, hanya aplikasi aplikasi tertentu yang bisa mengakses port USB pada computer kita.

Buka Terminal, lalu ketik sudo gedit /etc/udev/rules.d/51-android.rules . Isikan beberapa baris berikut ke dalam file tersebut, dengan mengganti kata username dengan nama user anda di Ubuntu :

# adb protocol on passion (Nexus One)

SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4e12", MODE="0600", OWNER="<username>"

# fastboot protocol on passion (Nexus One)

SUBSYSTEM=="usb", ATTR{idVendor}=="0bb4", ATTR{idProduct}=="0fff", MODE="0600", OWNER="<username>"

# adb protocol on crespo/crespo4g (Nexus S)

SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4e22", MODE="0600", OWNER="<username>"

# fastboot protocol on crespo/crespo4g (Nexus S)

SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4e20", MODE="0600", OWNER="<username>"

# adb protocol on stingray/wingray (Xoom)

SUBSYSTEM=="usb", ATTR{idVendor}=="22b8", ATTR{idProduct}=="70a9", MODE="0600", OWNER="<username>"

# fastboot protocol on stingray/wingray (Xoom)

SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="708c", MODE="0600", OWNER="<username>"

# adb protocol on maguro/toro (Galaxy Nexus)

SUBSYSTEM=="usb", ATTR{idVendor}=="04e8", ATTR{idProduct}=="6860", MODE="0600", OWNER="<username>"

# fastboot protocol on maguro/toro (Galaxy Nexus)

SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4e30", MODE="0600", OWNER="<username>"

# adb protocol on panda (PandaBoard)

SUBSYSTEM=="usb", ATTR{idVendor}=="0451", ATTR{idProduct}=="d101", MODE="0600", OWNER="<username>"

# fastboot protocol on panda (PandaBoard)

SUBSYSTEM=="usb", ATTR{idVendor}=="0451", ATTR{idProduct}=="d022", MODE="0600", OWNER="<username>"

# usbboot protocol on panda (PandaBoard)

SUBSYSTEM=="usb", ATTR{idVendor}=="0451", ATTR{idProduct}=="d00f", MODE="0600", OWNER="<username>"

# usbboot protocol on panda (PandaBoard ES)

SUBSYSTEM=="usb", ATTR{idVendor}=="0451", ATTR{idProduct}=="d010", MODE="0600", OWNER="<username>"

# adb protocol on grouper (Nexus 7)

SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4e42", MODE="0600", OWNER="<username>"

# fastboot protocol on grouper (Nexus 7)

SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4e40", MODE="0600", OWNER="<username>"

Lalu simpan file tersebut. Langkah ini akan efektif setelah kita melakukan reboot.

Note : Jika device anda masih tidak bisa diakses oleh ADB, pastikan Debugging Mode tercentang di pengaturan device anda, lalu ketikkan lsusb di Terminal, lihat apakah ada device anda tertera di situ, catat nomor id Vendor dan id Productnya, lalu tambahkan line baru pada file di atas, copy line sebelumnya, lalu ganti value dari idVendor dan idProduct dengan yang sesuai dengan device kita tadi.

* Download Source Code dari CyanogenMod9

Setelah langkah langkah di atas selesai, maka kita siap untuk mendownload source code CyanogenMod 9. Buka Terminal lalu ketikkan :

mkdir -p ~/bin

mkdir -p ~/android/system

untuk membuat folder ~/bin dan ~/android/system. Setelah itu ketikkan :

curl https://dl-ssl.google.com/dl/googlesource/git-repo/repo > ~/bin/repo

chmod a+x ~/bin/repo

untuk mendownload file script repo buatan Google yang akan mendownload semua source code Android, setelah itu pindah ke folder ~/android/system, lalu ketikkan :

repo init -u git://github.com/CyanogenMod/android.git -b ics

repo sync -j4

untuk mendownload source code CM9 secara keseluruhan.