

## LAMPIRAN A INSTALASI OMNET++

OMNeT ++ (*Objective Modular Network Testbed in C ++*) adalah sebuah *framework* simulasi yang berbasis C++ yang dapat digunakan untuk membangun simulasi jaringan. OMNeT++ dapat digunakan secara gratis untuk simulasi nonkomersial seperti pada institusi akademis dan untuk pengajaran. Dalam simulasi ini, OMNeT++ digunakan sebagai simulator untuk menjalankan simulasi berdasarkan dari *workspace* CCN-Lite. Langkah-langkah instalasi simulator adalah sebagai berikut:

1. Langkah pertama adalah mengunduh *file* OMNeT++ pada alamat <http://omnetpp.org>. Pilih dan unduh *file* OMNeT++ versi 4.5.
2. Setelah *file* terunduh, buka terminal dan tuliskan *command* untuk mengekstrak *file*:

```
$ tar xvfz omnetpp-4.5.0-src.tgz
```

Perintah `tar` digunakan untuk mengekstrak *file* berbentuk `.rar/.tgz`. Kemudian maksud dari perintah opsional `xvfz` adalah X untuk mengekstrak *file* dari berkas, V untuk menampilkan proses ekstraksi, F untuk argumen yang mengikuti nama *file* dan, Z untuk metode dekompresi.

3. Masuk ke direktori OMNeT++ dengan perintah :

```
$ cd omnetpp-4.5  
$ . setenv
```

Perintah `setenv` digunakan untuk memeriksa dimana folder `omnetpp-4.5` disimpan setelah diekstrak.

4. Setelah mengetahui dimana tempatkan folder `omnetpp-4.5`, ditahap ini dilakukan penentuan jalur agar IDE dapat dikenali oleh sistem

```
$ export PATH=$HOME/omnetpp-4.5.0/bin:$PATH
```

Apabila perintah diatas ingin disimpan secara permanen, maka lakukan edit pada *file* `.bashrc` dengan perintah

```
$ gedit ~/.bashrc
```

5. Lakukan pembaharuan sistem dengan perintah :

```
$ sudo apt-get update
```

6. Tahap ini adalah bagian yang diperlukan oleh IDE OMNeT++ agar dapat dijalankan dengan baik dan sistem dapat bekerja dengan selayaknya.

```
$ sudo apt-get install build-essential gcc g++ bison flex perl
qt5-default tcl-dev tk-dev libxml2-dev zlib1g-dev default-jre
doxygen graphviz libwebkitgtk-3.0-0
```

Penjelasan dari paket yang diinstal seperti pada Tabel 4.1 dibawah adalah sebagai berikut:

**Table Penjelasan Command**

Build Essential , gcc , g++ , bison , flex , perl	Paket ini dibutuhkan untuk meng- <i>compile</i> model dari simulasi OMNeT++ dan juga sebagai <i>tools</i> .
Tcl-dev , tk-dev	Dibutuhkan oleh bagian dari simulasi Tkenv <i>runtime</i> .
Libxml2-dev , zlib1g-dev	Sebagai <i>xml parser</i> yang dibutuhkan untuk membaca <i>file .xml</i> .
Default-jre	Untuk menginstal <i>java runtime</i> agar dapat menjalankan simulasi IDE berbasis <i>Eclipse</i> .
Doxygen , graphviz	Paket ini digunakan untuk dokumentasi dari fitur NED yang ada pada IDE.

7. Ketika instalasi paket telah selesai, dilanjutkan dengan melakukan instalasi paket MPI (*Message Passing Interfaces*) :

```
$ sudo apt-get install openmpi-bin libopenmpi-dev
```

Gunanya untuk membantu waktu eksekusi simulasi pemrosesan paralel.

8. Setelah semua bagian telah terpasang, maka dilanjutkan dengan konfigurasi omnetpp dengan perintah :

```
$ cd omnetpp-4.5
$ ./configure
```

Perintah ini digunakan untuk mendeteksi semua perangkat lunak yang telah dipasang dan dikonfigurasi oleh sistem. Hasil dari konfigurasi diatas akan dituliskan kedalam *file* bernama *Makefile.inc*.

9. Setelah konfigurasi telah dilakukan maka tahap terakhir adalah mem-*compile file* OMNeT++ dengan perintah *make*. Perintah ini akan memakan waktu sekitar 30 menit dan perintah ini dilakukan di dalam direktori omnetpp-4.5.

```
$ make
```

10. Untuk menjalankan IDE ini dapat memanggil dari *command line* dengan perintah :

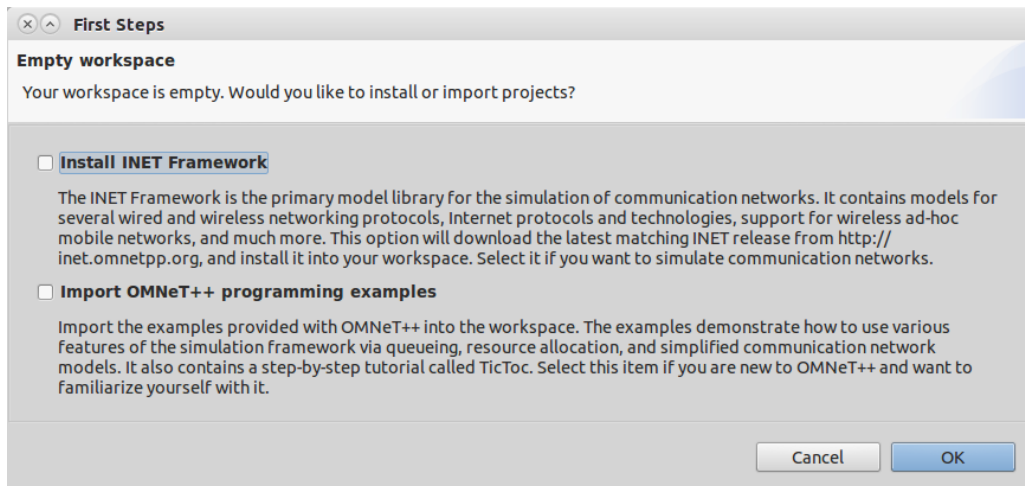
```
$ omnetpp
```

Atau dapat dilakukan dengan cara memasang ikon *desktop* atau *menu item*.

```
$ make install-menu-item
```

```
$ make install-dekstop-icon
```

Pada saat membuka IDE ini akan muncul *windows first steps* dimana menampilkan pilihan instalasi INET secara otomatis menggunakan koneksi Internet dan *import* contoh program OMNeT++, karena di dalam simulasi ini menggunakan INET yang telah ditentukan dan juga untuk meringankan jalannya program ketika melakukan *build project* maka kedua pilihan tidak di centang seperti pada dibawah.



**Tampilan *First Steps* Pada OMNeT++**

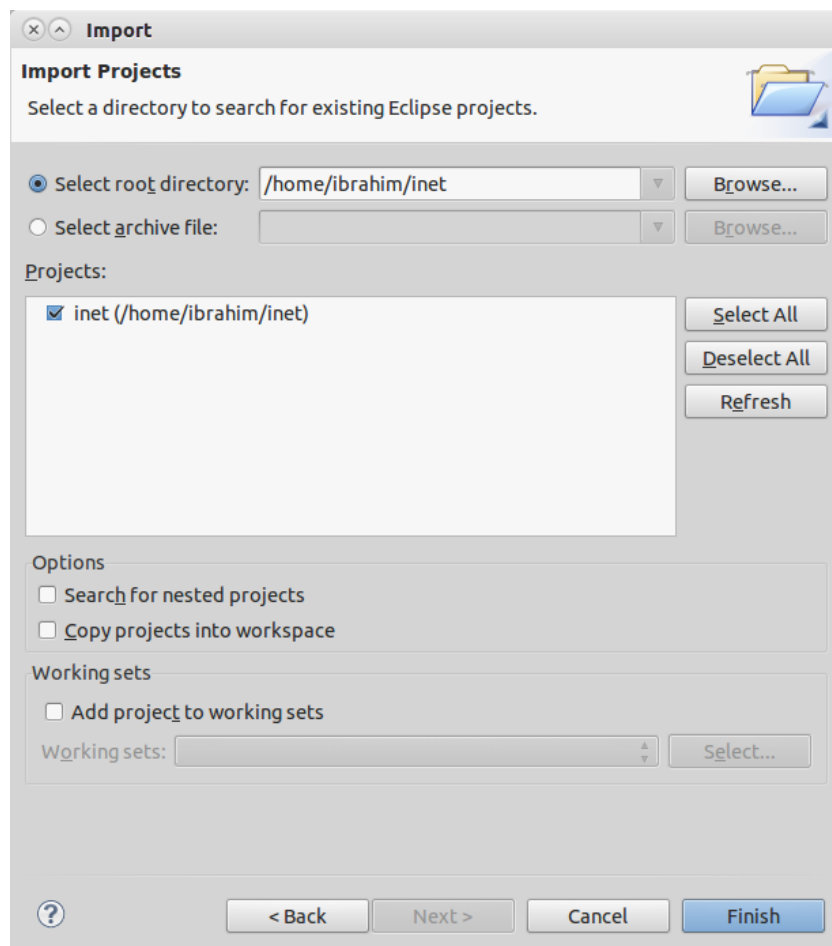
## LAMPIRAN B INSTALASI INET FRAMEWORK

INET *Framework* adalah sebuah *library* model open-source yang digunakan pada simulator OMNeT++. Dalam penelitian ini, INET *Framework* sangat dibutuhkan oleh CCN-Lite karena bagian-bagian node yang dipakai diambil dari *library* ini. Agar dapat dijalankan, perlu langkah-langkah instalasi INET yang sebagai berikut:

1. Langkah pertama adalah mengunduh dari website <http://inet.omnetpp.org/Download> , kemudian pilih inet 2.4.0.
2. Buka terminal dan tulis perintah untuk melakukan ekstrak berkas INET

```
$ tar xvfz inet-2.4.0-src.tgz
```

3. Buka IDE OMNeT++ dan lakukan *import project* melalui *File -> Import -> General -> Existing Projects to the Workspace ->* pilih “*select root directory*”, *browse ->* cari folder INET yang telah di ekstrak.



### Tampilan Ketika Melakukan *Import Workspace*

Akan muncul *workspace* INET, klik kanan dan pilih *build project*.

## LAMPIRAN C INSTALASI CCN-LITE

CCN-Lite adalah sebuah projek yang merepresentasikan protokol CCN dan NDN dimana projek ini dapat digunakan berbagai *platform* seperti UNIX, Linux kernel, OMNeT++, Android, Arduino (Uno dan AtMega328, KiB RAM), RFDuino (32KiB RAM) dan Docker. Dalam penelitian ini penulis menggunakan simulator OMNeT++ untuk mengetahui bagaimana kelengkapan dari CCN-Lite apabila dibandingkan dengan arsitektur CCN yang sebenarnya. Terdapat dua langkah untuk mengambil berkas CCN-Lite yaitu dengan mengunduh berkas secara *offline* atau menggunakan perintah *gitclone* dari github. Penulis melakukan cara kedua yaitu dengan perintah *gitclone*, alasan memilih cara kedua karena ketika ada *bug* atau *error* maka yang pertama kali diperbaiki adalah yang berada pada github. Berikut adalah langkah-langkah instalasi CCN-Lite:

1. Langkah pertama adalah memasang openSSL.

```
$ sudo apt-get install libssl-dev
```

OpenSSL sendiri adalah sebuah *toolkit* kriptografi yang mengimplementasikan *Secure Socket Layer* dan *Transport Layer Security* serta terkait dengan protokol jaringan standar kriptografi yang dibutuhkan oleh keduanya.

2. Lakukan *gitclone* CCN-Lite.

```
$ gitclone https://github.com/cn-uofbasel/ccn-lite
```

Perintah *gitclone* ini digunakan untuk meng-*copy file* yang berasal dari repositori Git.

3. Lakukan penyesuaian variabel dari CCN-Lite dan untuk penentuan jalur agar dapat dikenali oleh sistem.

```
$ export CCNL_HOME="`pwd`/ccn-lite"  
$ export PATH=$PATH:"$CCNL_HOME/bin"
```

Agar perintah diatas menjadi permanen, dapat ditambahkan juga kedalam *file* *.bashrc*.

4. Kemudian masuk ke direktori CCN-Lite untuk melakukan *build*

```
$ cd $CCNL_HOME/src  
$ make clean all
```

Perintah *make clean all* digunakan untuk menghapus semua bentuk temporary dan membentuk ulang *file* dari awal.

5. Setelah melakukan langkah diatas, tutup dan buka kembali terminal dan masuk kembali ke direktori *ccn-lite/src* untuk mem-*build* sebuah berkas bernama *ccn-lite-omnet*.

```
$ cd $CCNL_HOME/src  
$ make ccn-lite-omnet
```

Berkas ini terdapat di dalam directori *ccn-lite/src*.

6. Buka IDE OMNeT++ untuk melakukan import project dengan cara File -> Import -> General -> Existing projects into workspace -> pilih select archive *file* (karena *file* bentuk .tgz) -> browse dan pilih ccn-lite-omnet.tgz. Tidak lupa untuk mengaktifkan checkbox di dalam kolom project.
7. Muncul *workspace* ccn-lite, kemudian klik kanan -> Index -> Rebuild. Ini digunakan untuk mengatur kembali seperti awal apabila terdapat konfigurasi yang tidak beraturan.
8. Setelah selesai lakukan klik kanan -> build project. Ketika proses *build* selesai, maka project siap untuk dijalankan.