ISSN : 2407 – 1846
e-ISSN : 2460 – 8416

Website : jurnal.ftumj.ac.id/index.php/semnastek

PEMBUATAN APLIKASI PACKAGE DEPENDENCY MERGER BERBASIS SISTEM OPERASI DEBIAN GNU/LINUX

Hendra[1*], Rahmad Ade Putra
Jurusan Teknik Informatika, Fakultas Teknik, Universitas Muhammadiya Jakarta
Jl. Cempaka Putih Tengah No. 27 Jakarta Pusat 10510
[*]Hendra@ftumj.ac.id

ABSTRAK

Kemajuan di bidang teknologi jaringan komputer menyangkut perangkat keras, perangkat lunak maupun database dari jaringan tersebut. Banyak cara yang di gunakan untuk membuat komunikasi semakin mudah, Salah satu perubahan paling menonjol pada teknologi komunikasi jaringan computer adalah penggunaan teknologi wireless. Sistem keamanan yang paling umum digunakan pada wireless LAN adalah dengan metode enkripsi *Wired Equivalent Privacy* (WEP). WEP menggunakan satu kunci enkripsi yang digunakan bersama-sama oleh para pengguna wireless. Hal ini menyebabkan metode WEP tidak cocok diterapkan pada hotspot yang dipasang di tempat-tempat umum karena lubang keamanan yang dimiliki WEP sangat rentan di susupi, sehingga dapat dimasuki oleh pihak-pihak yang tidak berkepentingan. Dengan adanya server *Remote Access Dial In User* (RADIUS) yang bertindak sebagai *authentication, authorization and accounting* membuat administrator dapat memantau dan mengetahui identitas yang jelas dari user. RADIUS merupakan sebuah protocol yang memungkinkan keamanan jaringan wireless melakukan autentikasi, autorisasi, dan akuntansi untuk meremote user yang ingin mengakses suatu sistem atau layanan dari pusat server jaringan komputer..

Kata kunci: linux, wireless, jaringan, keamanan, server

ABSTRACT

Progress in the field of computer networking technology involves hardware, software or database from the network. Many ways are used to make communication easier, One of the most notable changes in computer network communication technology is the use of wireless technology. Security systems most commonly used in wireless LAN is the encryption method Wired Equivalent Privacy (WEP). WEP uses an encryption key that is shared by wireless users. This causes the WEP method was not suitable to be applied to the hotspot installed in public places because of security holes that WEP is vulnerable owned infiltrated, so it can be entered into by parties who are not interested. With the server Remote Access Dial In User (RADIUS), which acts as an authentication, authorization and accounting make the administrator can monitor and know the distinct identity of the user. RADIUS is a protocol that allows the security of wireless networks perform authentication, authorization, and accounting for users who want to remotely access a system or service from the central server computer network.

Keywords : *linux, wireless, network, security, server*

PENDAHULUAN

Sistem operasi berbasis Debian GNU/Linux merupakan sebuah sistem operasi yang memiliki lisensi publik yaitu General Public License. dimana setiap penggunanya mendapatkan lisensi yang menjamin kebebasan untuk menggunakan, mempelajari, membagikan dan juga melakukan modifikasi terhadap aplikasi yang memiliki lisensi berbasis General Public License. Sistem operasi Debian GNU/Linux dikembangkan oleh banyak software developer di berbagai negara dan juga dibantu oleh para relawan yang ada didalam melakukan software testing, sehingga sistem operasi Debian GNU/Linux dapat berkembang dengan sangat cepat. Pada

ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.ftumj.ac.id/index.php/semnastek

sistem operasi Debian GNU/Linux dibutuhkan banyak aplikasi third-party untuk membangun sistem operasi itu sendiri. Untuk melakukan pembaruan aplikasi third-party dibutuhkan package dependency yang diminta baik oleh third-party itu sendiri maupun sistem operasi yang digunakan.

Sistem operasi Debian GNU/Linux membutuhkan konektifitas internet yang memadai guna dapat melakukan pengunduhan paket dependensi dan juga sistem dependensi ketika pengguna memerlukannya. Pada saat pengguna sistem operasi Debian GNU/Linux berada pada lokasi dengan tingkat konektifitas internet yang terbatas akan menjadi sebuah hambatan ketika pengguna tersebut akan melakukan pengunduhan paket dependensi dan juga sistem dependensi.

Package dependency pada sistem operasi Debian GNU/Linux adalah paket aplikasi lain yang dibutuhkan aplikasi third-party untuk dapat memenuhi requirement yang diminta aplikasi third-party. Kebutuhan paket dependensi dan banyaknya paket dependensi pada aplikasi third-party itu sendiri berbeda-beda. Ketika akan melakukan pembaruan direktori cache untuk aplikasi third-party dan juga melakukan pembaruan direktori cache terhadap aplikasi yang telah ada pada sistem operasi Debian GNU/Linux, komputer tersebut haruslah terhubung ke dalam online repository archive yang banyak terdapat di jaringan internet terbuka guna mendapatkan paket utama serta paket dependensi terbaru yang telah di rekomendasikan oleh pengembang sistem operasi Debian GNU/Linux. Oleh karena itu untuk kelancaran pembaruan direktori cache diperlukan jaringan internet yang memadai. Akan tetapi tingkat kestabilan kecepatan internet dan juga besarnya pemberian kuota data pada setiap wilayah di Indonesia umumnya berbeda-beda tergantung kebijakan dari masing-masing internet service provider(ISP) yang digunakan. Tidak stabilnya kecepatan internet dan kurangnya kuota data dapat mengganggu jika sebuah komputer dengan sistem operasi Debian GNU/Linux akan melakukan pembaruan direktori cache untuk aplikasi third-party dan juga melakukan pembaruan direktori cache terhadap system dependency yang digunakan. Padahal pembaruan direktori cache diperlukan untuk menyimpan package dependency beserta aplikasi third-party dan juga system dependency guna keperluan penggunaan offline, yaitu saat komputer tersebut tidak terhubung kedalam online repository archive.

Untuk memastikan sitem operasi Debian GNU/Linux dapat melakukan pengumpulan package dependency yang akan dijadikan satu file yang terkompresi beserta aplikasi third-party utama atau saat pengumpulan system dependency yang akan dijadikan satu file yang terkompresi, tanpa harus terhubung ke dalam online repository archive secara langsung yang berada di internet, maka diperlukan aplikasi package dependency merger. Aplikasi tersebut berfungsi untuk mengunduh serta menyatukan paket-paket dependensi yang diperlukan aplikasi thrid-party, sehingga dapat digunakan saat komputer tersebut terhubung dengan koneksi internet yang memiliki batasan tertentu ataupun juga tidak terhubung ke internet. Oleh karena itu dalam penelitian ini akan dibuat aplikasi package dependency merger berbasis Debian GNU/Linux. Pembuatan aplikasi menggunakan bahasa pemrograman Python, semetara itu untuk merancang tampilan antar muka digunakan Qt designer. Perancangan aplikasi package dependency merger berdasarkan dengan metode advanced packaging tool yaitu dengan menggunakan pustaka python-apt yang telah ada pada setiap sistem operasi berbasis Debian GNU/Linux. Aplikasi tersebut berfungsi untuk mengunduh serta menyatukan paket-paket dependensi yang diperlukan kedalam sebuah file berbentuk arsip dengan menggunakan file signature text yang dibuat pada komputer offline, sehingga dapat digunakan ketika komputer tidak terhubung ke jaringan internet atau memiliki kuota yang terbatas.

METODE

Metodologi penelitian yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Studi Literatur

Dilakukan dengan cara membaca data-data yang diperlukan untuk membangun sebuah rancangan yang berkaitan dengan penelitian ini, seperti: mempelajari cara kerja sistem operasi debian GNU/Linux, melalui artikel, melalui buku terkait dan juga beberapa sumber dari internet.

2. Studi Observasi

ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.ftumj.ac.id/index.php/semnastek

Metode ini dilakukan dengan cara mendapatkan data dan informasi yang digunakan sebagai dasar pembuatan aplikasi dengan cara mengikuti seminar dan juga kuliah umum tentang sistem operasi berbasis Linux.

3. Perancangan Aplikasi

Hasil yang diperoleh pada proses analisis kemudian dilanjutkan pada tahap perancangan. Pada perancangan penulis menggunakan suatu model perancangan sistem yakni Unified Modeling Language (UML). Pemodelan digunakan dikarenakan mudah dipahami dan juga telah terintegrasi. Perancangan dilakukan untuk menentukan permasalahan mengenai bahasa pemrograman yang digunakan, input/process/output terhadap aplikasi yang dibuat, dan juga permasalahan teknik yang akan diimplementasikan.

4. Desain Aplikasi

Desain aplikasi diterapkan sebelum proses pembuatan aplikasi guna mendapatkan tampilan berupa user interface yang baik.

5. Pembuatan Aplikasi

Pembuatan aplikasi dengan menggunakan software yang dibutuhkan.

6. Uji coba Aplikasi

Proses mengenai penerapan implementasi aplikasi yang dibuat sesuai dengan hasil analisis pada tahapan sebelumnya, hasil uji coba aplikasi package dependency merger. Pengujian dilakukan dengan pembaharuan meta package information index, pembaharuan package dependency cache, pembaharuan system dependency cache pada sistem operasi Debian GNU/Linux yang memiliki keterbatasan untuk terhubung ke jaringan internet, sehingga memastikan bahwa komputer tersebut dapat melakukan sistem pembaruan guna melakukan instalasi aplikasi third-party dan juga melakukan upgrade sistem melalui metode advaced packaging tool yang telah ada tanpa harus terhubung langsung ke online repository archive.

HASIL DAN PEMBAHASAN

Analisis Kebutuhan Aplikasi

Analisa merupakan suatu penguraian perancangan aplikasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan yang ada dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisa juga merupakan suatu tahap pemahaman terhadap aplikasi yang dibuat. Tahap ini bertujuan untuk mengetahui mekanisme aplikasi, proses-proses yang terlibat dalam aplikasi serta hubungan-hubungan antar proses.

Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang diperlukan untuk membuat aplikasi *package dependency merger* adalah sebagai berikut:

1. *Processor* yang digunakan adalah Intel Core 2 Duo.
2. RAM 4 GB DDR3, untuk mempercepat eksekusi aplikasi.
3. GPU Nvidia GT630 untuk mengolah antar muka dengan Qt.
4. Hard disk space 10 GB.
5. *Internet connection* minimal 1 Mbps untuk uji coba *download.*
6. Dibutuhkan minimal 2 buah komputer.

Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan penulis untuk membuat aplikasi *package dependency merge* adalah sebagai berikut :

1) Menggunakan sistem operasi berbasis Debian GNU/Linux.
2) Eric Python *Integrated Development Environment* 6.0.1 untuk membuat aplikasi *package dependency merger.*
3) Python 2.7 sebagai bahasa pemrograman yang digunakan.
4) Python 2.7 *standard library* menggunakan *standard library* yang telah ada didalamnya.
5) Python APT, *built-in library* yang terdapat pada sistem operasi Debian GNU/Linux sesuai dengan Debian python *policy*.
6) K-*Desktop Environment* sebagai tampilan desktop yang digunakan untuk menjalankan antar muka pada Qt.
7) Qt *Designer* untuk merancang tampilan antar muka yang digunakan untuk menjalankan *package dependency merger.*

Unsur Yang Melatarbelakangi *Package Dependency Merger*

Pada aplikasi package dependency merger terdapat dua unsur yang melatar belakangi masalah yang ada. Pertama, kebutuhan ketika pengguna sistem operasi berbasis Debian GNU/Linux terhubung kedalam jaringan *internet* yang memiliki

ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.ftumj.ac.id/index.php/semnastek

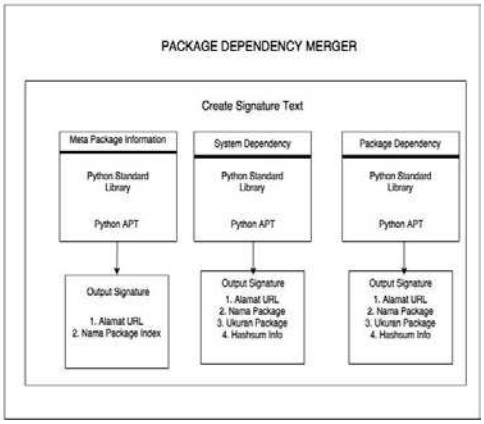
batasan tertentu ataupun saat tidak terbuhung atau *offline.* Kedua, ketika pengguna debian GNU/Linux ingin melakukan pengunduhan melalui komputer berbasis sistem operasi Unix-like lainnya yang mana komputer tersebut terdapat bahasa pemrograman python yang merupakan bahasa pemrograman yang telah diadopsi menjadi standard dari Linux Standard Base.

Package dependency merger dapat mengunduh setiap paket yang dibutuhkan oleh penguna menggunakan file signature text yang telah dibuat menggunakan metode *Advanced Packaging Tool* berdasarkan *meta package index* yang terdapat pada sistem operasi Debian GNU/Linux.

Penggunaan aplikasi *package dependency merger* hanya dapat digunakan ketika pengguna ingin mengunduh paket dependensi pada sistem operasi debian GNU/Linux membuat file berbentuk signature text terlebih dahulu pada komputer offline, lalu menggunakannya dengan menggunakan komputer lainnya khusus sistem operasi berbasis Unix-like. Aplikasi ini tidak dapat dijalankan pada komputer berbasis Microsoft Windows.

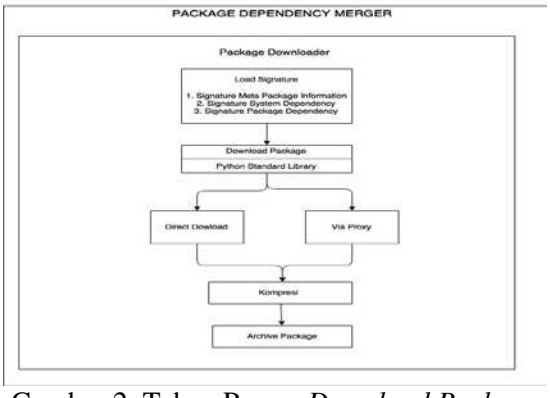
Tahap Pembuatan Komponen *Package Dependency Merger*

Package dependency dan juga meta *package information index* merupakan komponen penting dalam sistem operasi berbasis Debian GNU/Linux. Setiap aplikasi dan library memiliki hubunbgan keterkaitan antara setiap paket aplikasi lainnya, misalnya ketika user akan melakukan pemasangan aplikasi *Wine*, maka aplikasi tersebut membutuhkan banyak aplikasi terkait yang harus dipenuhi terlebih dahulu untuk melakukan tahap instalasi. Pada saat komputer yang digunakan terhubung kedalam jaringan *internet* yang tidak memiliki batasan hal itu tentu mudah untuk dilakukan, tetapi hal tersebut menjadi suatu hambatan ketika komputer yang akan diperbarui direktori pada sebuah aplikasi tersebut memiliki batasan pada konektifitas internet yang digunakan, dikarenakan sistem operasi berbasis Debian GNU/Linux membutuhkan koneksi *internet* yang handal untuk terhubung ke *online repository archive* guna mendapatkan sinkronisasi terhadap *meta package infomartion index* dan juga untuk melakukan pengunduhan setiap *package dependency*, baik itu untuk keperluan saat melakukan pembaruan direktori *cache* untuk *package dependency* pada aplikasi *third-party* atau saat pembaruan direktori *cache* pada *system dependency*.



Gambar 1. Tahap Pembuatan *File Signature* Pada Aplikasi *Package Dependency Merger.*

Pada alur yang terdapat dari gambar diagram diatas adalah pada saat komputer dengan sistem operasi Debian GNU/Linux yang memiliki batasan koneksi *internet* akan melakukan pengunduhan untuk mendapatkan *meta package information index* atau *package dependency* atau *system dependency* yang mana diatara ketiga *output* tersebut berupa *file signature* yang digunakan oleh komputer lainnya berbasis Unix-like untuk mengunduh salah satu output tersebut saat berada di wilayah dengan tingkat konektifitas internet yang memadai.

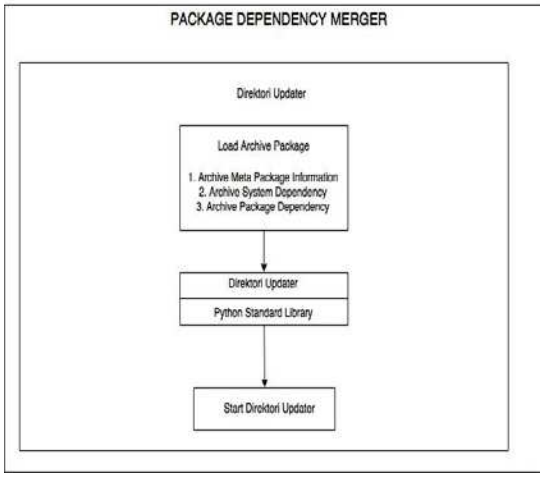


Gambar 2. Tahap Proses *Download Package* Pada Aplikasi *Package Dependency Merger.*

Pada gambar 2 merupakan proses saat pengunduhan *meta package information index* atau *package dependency* atau *system dependency.* Dari ketiga pilihan tersebut, setiap

ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.ftumj.ac.id/index.php/semnastek

paket yang akan diunduh dapat menggunakan metode *download* secara langsung (*direct download)* atau bisa menggunakan pilihan *web proxy* sebagai perantara. Kemudian hasil keluaran yang didapat berupa paket-paket yang dikompresi kedalam file arsip dengan format .Zip atau Bz.2.



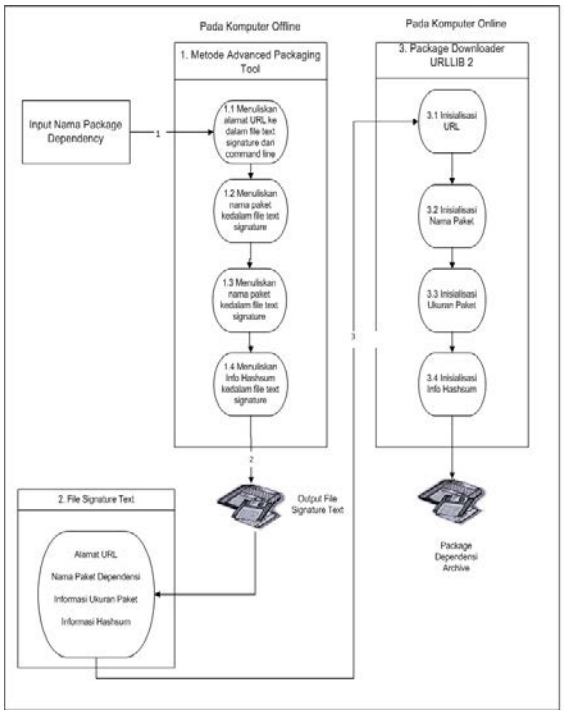
Gambar 3.Tahap Pembaruan Direktori Pada *Aplikasi Package Dependecy Merger.*

Pada gambar 3 adalah proses untuk pembaruan direktori *cache* atau *index* direktori dengan menggunakan paket-paket yang telah diunduh dan dikompresi pada tahap pengunduhan. Proses pembaruan direktori *cache* untuk *package dependency* atau pembaruan direktori *cache* untuk *system dependency* terletak pada direktori /var/cache/archives, yang mana direktori tersebut merupakan standard baku dan telah ditetapkan oleh para pengembang Debian GNU/Linux. Sementara untuk pembaruan *index* direktori yang digunakan oleh *meta package information index* terdapat pada direktori /var/lib/apt/lists.

Selain dari tahap analisa yang dilakukan diatas, penulis juga melakukan analisa yang meliputi deskripsi umum perangkat lunak yang digunakan, analisa kebutuhan perangkat lunak yang digunakan dan deskripsi kebutuhan fungsional.

Proses Perancangan Aplikasi

Pada penjelasan proses perancangan Aplikasi dapat dijelaskan bahwa aplikasi ini memiliki 2 proses utama, yaitu proses pada saat komputer *offline* dan *direct download* :



Gambar 4. Proses Perancangan Aplikasi Pada Saat Komputer *Offline* Dan *Direct Download*

Proses perancangan aplikasi ini adalah :

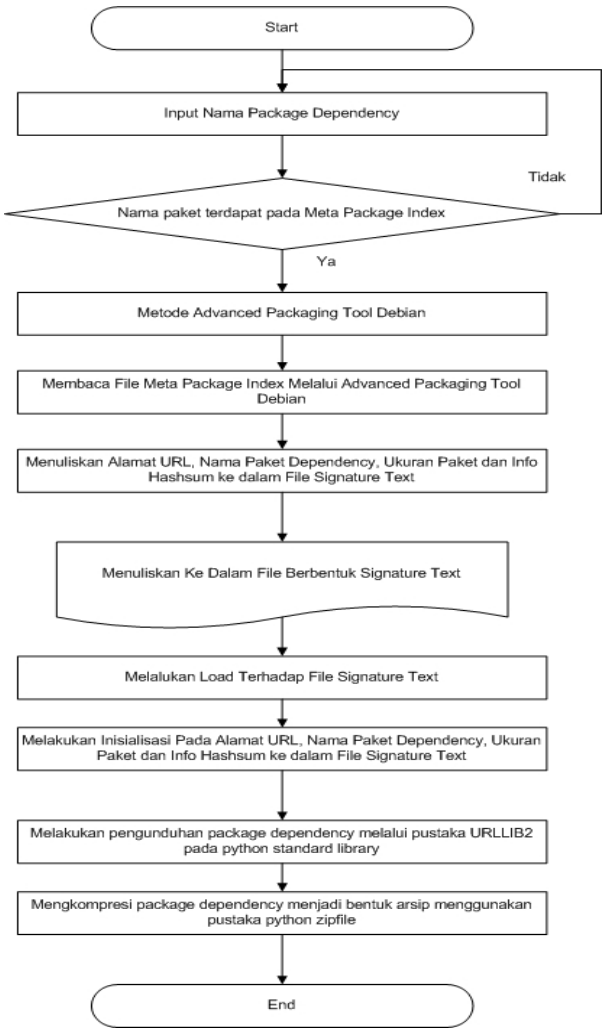
1. *Input* nama paket dependensi yang diperlukan.
2. Proses 1 akan menuliskan alamat URL, nama paket dependensi, ukuran paket, dan juga info *hashsum* berdasarkan *meta package index* pada komputer *offline* guna mendapatkan info pada *file signature text* dengan *metode advanced packaging tool* untuk mengunduh melalui direct download.
3. Proses 3 akan melakukan tahap inisialisasi alamat URL, nama paket dependensi, ukuran paket, dan juga info *hashsum* untuk melakukan *direct download* dengan pustaka URLLIB2 pada python standard library.
4. Output paket berupa file dependensi yang terkompresi dilakukan dengan menggunakan pustaka Zipfile yang terdapat *pada python standard library.*

Flowchart Proses Aplikasi

Flowchart proses aplikasi menjelaskan tentang alur prosedur dari aplikasi yang dibuat

ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.ftumj.ac.id/index.php/semnastek

dengan menggunakan metode *Advanced Packaging Tool* pada aplikasi *package dependency merger:*



Gambar 5. *Flowchart* Proses Metode Pada Aplikasi *Package Dependency Merger*

Pada gambar 5, menggambarkan *flowchart* proses metode pada aplikasi yang dibuat. Pada gambar tersebut dijelaskan bahwa jika nama paket dependensi yang dimasukan benar, maka aplikasi dapat menemukannya pada *meta package index.* Setelah nama paket dependensi ditemukan, maka paket tersebut akan diinisialisasi menggunakan metode *Advanced Packaging Tool* pada komputer *offline* beserta dengan seluruh paket lainnya yang dibutuhkan paket tersebut.

Setelah dilakukan inisialisasi maka paket tersebut akan dituliskan kedalam file berbentuk *text signature* guna keperluan pengunduhan saat menggunakan komputer yang terhubung kedalam jaringan *internet.*

Proses pengunduhan dilakukan dengan melakukan tahap load terhadap *file signature* yang dibuat pada komputer *offlin,* lalu kemudian file tersebut dibaca oleh aplikasi dan diunduh dengan menggunakan pustaka URLLIB2 pada *python standard library*.

Setelah file diunduh aplikasi akan mengkompresi kedalam archive berbentuk .zip dengan menggunakan pustaka ZipFile pada *python standard library*.

Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan proses yang terjadi didalam aplikasi yang dibuat ini. Proses tersebut dijelaskan dalam *use case diagram*, *class diagram*, *package diagram*, dan *component diagram.*

Use Case Diagram

Use case diagram menggambarkan proses yang ada di dalam aplikasi ini. *Use case diagram* aplikasi ini digambarkan pada gambar dibawah ini :

Gambar 6. *Use Case Diagram* Aplikasi *Package Dependency Merger.*

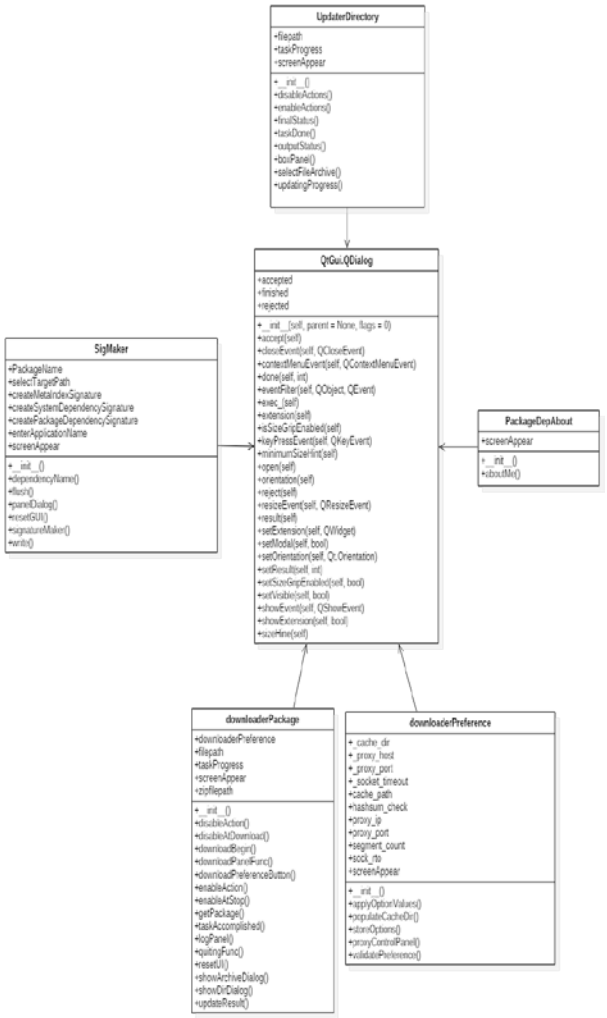
Gambar 3 diatas mendeskripsikan *Use Case Diagram* pada aplikasi *package dependency merger* adalah sebagai berikut:
Deskripsi *Use Case Package Dependency Merger*:

ISSN : 2407 – 1846
e-ISSN : 2460 – 8416

Website : jurnal.ftumj.ac.id/index.php/semnastek

Class Diagram

Class diagram digunakan untuk menggambarkan class atau rancangan blueprint pada object. *Class* yang ada pada *aplikasi package dependency merger* adalah berupa kontainer untuk pembuatan *front-end* antar muka pada aplikasi. Pada *class diagram* yang terdapat pada aplikasi *package dependency merger* digambarkan sebagai berikut :



Gambar 7. *Class Diagram* Aplikasi *Package Dependency Merger.*

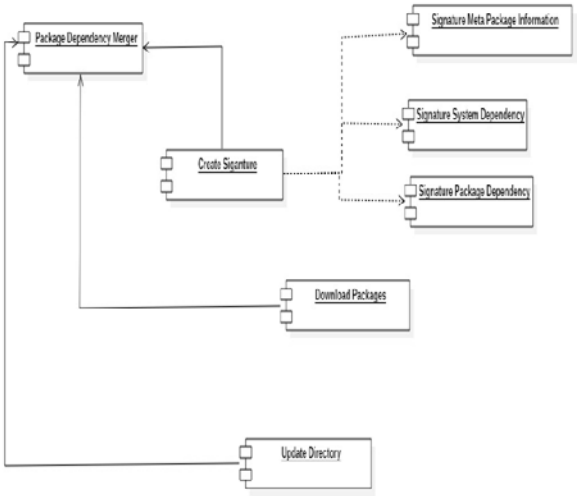
Package Diagram

Package diagram digunakan untuk menggambarkan struktur hirarki model pada aplikasi *package dependency merger*. Struktur hirarki yang terdapat didalamnya berupa modul aplikasi yang dibuat dalam bahasa pemrograman python, sehingga dapat digunakan sebagai komponen-komponen untuk membangun aplikasi. Pada *package diagram* yang terdapat pada aplikasi *package dependency merger* adalah sebagai berikut:

Gambar 8. *Package Diagram* Aplikasi *Package Dependency Merger.*

Component Diagram

Component diagram yang terdapat pada *package dependency merger* digunakan untuk menggambarkan struktur keterkaitan antara satu komponen dengan komponen lainnya. Komponen yang digambarkan baik berupa fungsi-fungsi yang ada pada aplikasi *package dependency merger* dan *output* yang dihasilkannya. *Component diagram* dapat digambarkan pada gambar berikut :



Gambar 9. *Component Diagram* Aplikasi *Package Dependency Merger.*

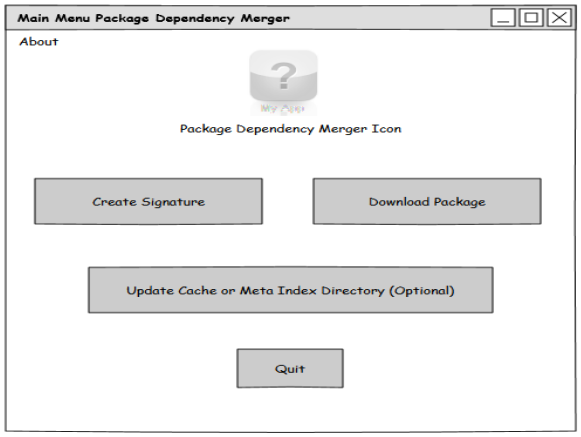
ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.ftumj.ac.id/index.php/semnastek

Perancangan Tampilan Antarmuka

Perancangan tampilan antarmuka adalah kumpulan dari operasi-operasi yang menggambarkan aplikasi *package dependency merger* berupa komponen objek.

Perancangan Tampilan Menu Utama

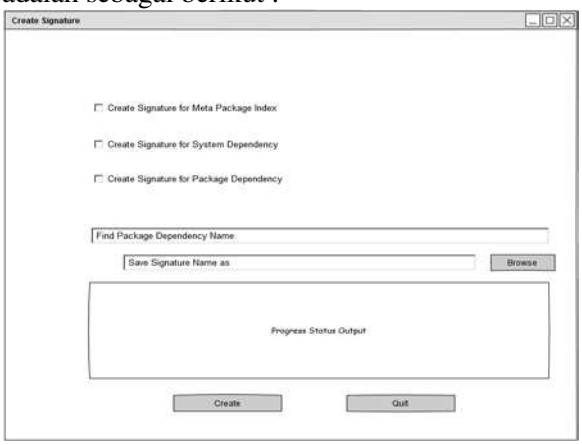
Tampilan menu utama adalah tampilan keseluruhan dari aplikasi *package dependency merger.* Untuk perancangan tampilan menu utama dalam aplikasi *package dependency merger* adalah sebagai berikut:



Gambar 10. Tampilan *Menu* Utama *Package Dependency Merger.*

Perancangan Tampilan *Create Signature*

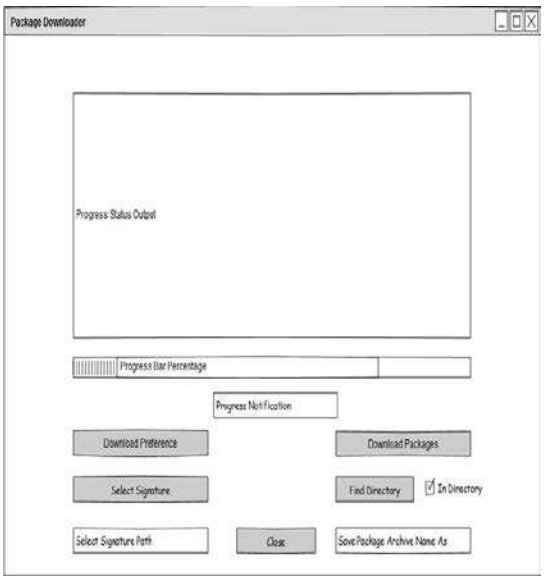
Tampilan *create signature* adalah tampilan yang digunakan untuk membuat *signature* yang akan digunakan untuk membuat *signature meta package information* atau membuat *signature system dependency* atau membuat *signature package dependency*. Untuk perancangan tampilan *create signature* pada aplikasi *package dependency merger* adalah sebagai berikut :



Gambar 11. Tampilan *Menu Create Signature*.

Perancangan Tampilan *Package Downloader*

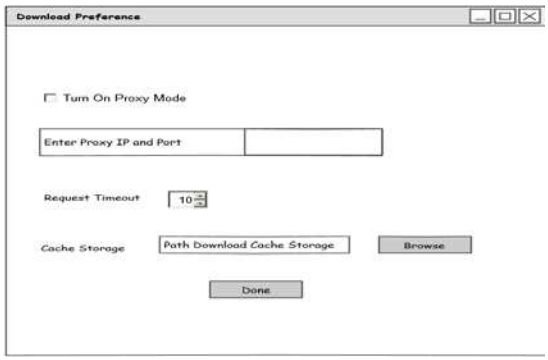
Tampilan *package downloader* adalah *menu* tampilan yang akan digunakan untuk mengunduh paket yang terdapat pada *file signature*. Desain tampilan *package downloader* adalah sebagai berikut:



Gambar 12. Tampilan *Package Downloader*.

Perancangan Tampilan *Download Options*

Tampilan *download options* adalah tampilan yang akan digunakan untuk opsional pengunduhan. Pada *download options* terdapat opsional untuk menggunakan *proxy*, pada pilihan penggunaan *proxy* menggunakan alamat IP dan *port* yang diminta. Fitur lain yang terdapat pada *download options* adalah opsional untuk menyimpan *cache* yang didapat pada saat pengunduhan paket, *cache* disimpan melalui *folder* direktori yang dibuat. Berikut ini adalah desain tamplian *download options* pada aplikasi *package dependency merger*:

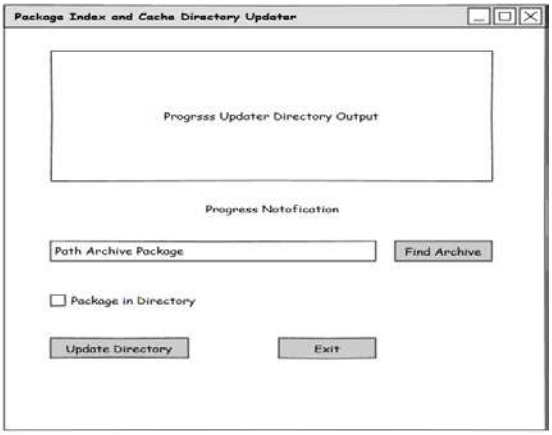


Gambar 13. Tampilan *Download Options*.

ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.ftumj.ac.id/index.php/semnastek

Perancangan Tamplian Pembaruan Direktori

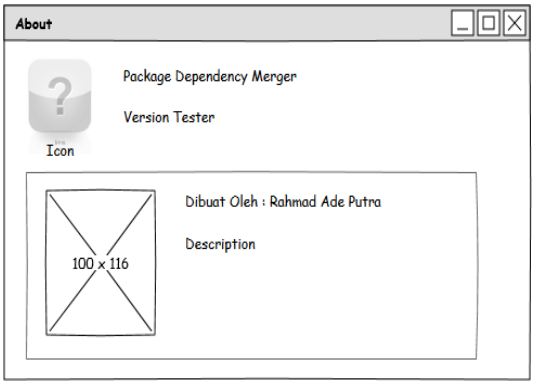
Tamplian pembaruan direktori digunakan untuk pembaruan direktori yang terdapat pada *package information index* dan juga *cache direktori*, pembaruan direktori dilakukan sehingga paket-paket yang diunduh dapat ditempatkan pada direktori yang telah distandarisasi oleh sistem operasi Debian GNU/Linux. Berikut ini adalah desain tamplian pembaruan direktori pada aplikasi *package dependency merger*:



Gambar 14. Tampilan Pembaruan Direktori.

Perancangan Tampilan *About*

Tampilan *about package dependency merger* adalah *form* yang digunakan untuk menampilkan biodata tentang pembuat aplikasi. Desain untuk tampilan *about package dependency merger* adalah sebagai berikut:



Gambar 15. Tampilan *About Package Dependency Merger.*

Hasil merupakan bagian utama artikel ilmiah, berisi : hasil bersih tanpa proses analisis data, hasil pengujian hipotesis. Hasil dapat disajikan dengan table atau grafik, untuk memperjelas hasil secara verbal

Pembahasan merupakan bagian terpenting dari keseluruhan isi artikel ilmiah. Tujuan pembahasan adalah: Menjawab masalah penelitian, menafsirkan temuan-temuan, mengintegrasikan temuan dari penelitian ke dalam kumpulan pengetahuan yang telah ada dan menyusun teori baru atau memodifikasi teori yang sudah ada.

Tabel 1. Elemental compositions of sampling sites

Site	TiO_2 (wt%)	Al_2O_3 (wt%)	MnO (wt%)	MgO (wt%)	Na_2O (wt%)
GIJ	0.5	16.4	0.19	2.74	3.00
GPW	0.78	19.0	0.18	4.57	2.55
GSR	0.62	16.3	0.17	3.09	3.09
KLB	0.67	15.7	0.14	5.07	2.59
KSG	1.90	17.1	0.15	3.79	3.33
PWH	0.58	20.9	0.12	1.55	3.00
SKP	0.68	17.8	0.16	3.12	2.75

Tabel dibuat dengan lebar garis 1 pt dan *tables caption* (keterangan tabel) diletakkan di atas tabel. Keterangan tabel yang terdiri lebih dari 2 baris ditulis menggunakan spasi 1.

Garis-garis tabel diutamakan garis horizontal saja sedangkan garis vertikal dihilangkan.

KESIMPULAN DAN SARAN

Simpulan dari pembuatan aplikasi *package dependency merger* adalah sebagai berikut:

1. Aplikasi *Package Dependency Merger* dapat digunakan komputer berbasis Debian GNU/Linux untuk memperoleh *meta package information* dari *online repository*

ISSN : 2407 – 1846
e-ISSN : 2460 – 8416
Website : jurnal.ftumj.ac.id/index.php/semnastek

archive. Pengunduhan dilakukan melalui komputer lain berbasis system operasi Unix dengan menggunakan *file signature text* yang dibuat terlebih dahulu.
2. Aplikasi *Package Dependency Merger* dapat digunakan ketika komputer berbasis Debian GNU/Linux tidak terhubung ke internet guna memperoleh *system dependency*. Dimana paket diunduh melalui komputer lain berbasis system operasi Unix dengan cara membuat *file signature text* terlebih dahulu pada komputer yang *offline.*
3. Aplikasi *Package Dependency Merger* digunakan oleh pengguna system operasi Debian GNU/Linux yang memiliki batasan *internet* untuk medapatkan *package dependency*. Dimana *package dependency* dibutuhkan oleh aplikasi *third-party*.
4. Aplikasi *Package Dependency Merger* dapat memeriksa info hashsum yang ada pada *system dependency* atau *package dependency* yang diunduh. Pemeriksaan dilakukan dengan membandingkan info hashsum yang ada di *file signature text* dengan paket yang diunduh guna mengurangi resiko paket yang korup.
5. Bila terdapat file yang korup maka proses dari pengunduhan paket harus diulang ke tahap sebelumnya, dikarenakan aplikasi ini langsung mengkompres file ke dalam archive.
6. Proses pengunduhan terhadap *meta package information* atau *system dependency* atau *package dependency* yang dilakukan oleh komputer lain dapat menggunakan *web proxy*.
7. Dapat meperbarui direktori *cache* komputer yang tidak terhubung ke *internet* baik berupa *system dependency* atau *package dependency.*

5.2. Saran

Saran yang penulis berikan adalah sebagai berikut:

1. Dibutuhkan PyQ4 sebagai tampilan utama antarmuka dengan menggunakan KDE *desktop*.
2. Penggunaan aplikasi ini harus dengan menggunakan internet mobile broadband atau koneksi internet lainnya jika tidak ditemukannya hotspot terdekat pada saat proses pengunduhan paket dependensi.
3. Harus mencari kecepatan *internet* yang memadai guna memperoleh paket-paket yang diunduh pada saat menggunakan *Downloader Packages* pada aplikasi *Package Dependency Merger*.
4. Menggunakan *premium* atau *private proxy* menggunakan *web proxy* untuk mengunduh paket-paket melalui komputer lain jika mengaktikan fitur *web proxy*. Hal tersebut berguna untuk memperoleh kecepatan yang maksimal serta mengurangi paket yang korup.

DAFTAR PUSTAKA

Gagne Greg, Operating System Concepts 8th Edition, 2010, Wiley, New York.

Cesati Marco, Understanding the Linux Kernel, 3rd Edition, 2005, O'Reilly Media, Sebastopol, CA.

Hertzog RaphaÃl, The Debian Administrator's Handbook, 2010, Freexian, Arizona.

Negus Christopher, Linux Bible 8th Edition, 2013, Wiley, New York.

Attending Linux Conference, refspecs.linuxfoundation.org, 2015, Tokyo, Japan.

Tanenbaum Andrew S, Computer Networks 4th Edition, 2003, Pearson Education, Vallejo, CA.

Bestavros, Web Caching and Content Delivery 1st Edition, 2008, Elsevier, North Holland.

Phillips Dusty, Python 3 Object Oriented Programming, 2010, Packt Publishing, Birmingham.

Hellmann Doug, The Python Standard Library, 2011, Addison-Wesley, Boston.

Summerfield Mark, Rapid GUI Programming with Python and Qt, 2007, Prentice Hall, New Jersey.

Dharwiyanti Sri, Pengantar Unified Modelling Language (UML), 2003, Ilmukomputer, Jakarta

Sholiq, Pemodelan Sistem Informasi Berorientasi Objek Dengan UML, 2006, Graha Ilmu, Yogyakarta

Simarmata Janner, Rekayasa Perangkat Lunak, 2010, Andi Publisher, Yogyakarta