

PANDUAN DASAR PEMAKETAN BLANKON

Disusun oleh:

Komunitas Pengembang BlankOn

DAFTAR ISI

PANDUAN DASAR PEMAKETAN BLANKON.....	1
BAGAIMANA MEMAKETKAN DI BLANKON.....	2
A. Pengantar.....	2
B. Prasyarat.....	3
C. Memulai.....	5
1. Paket Binary dan Paket Source.....	5
2. Peralatan Pemaketan.....	5
3. Mulai membangun paket.....	8
I. Membangun paket dengan dpkg-buildpackage.....	9
II. Membangun paket dengan pbuilder.....	13

BAGAIMANA MEMAKETKAN DI BLANKON

Pemaketan adalah membuat paket Debian dari kode sumber agar mampu dipasang dalam direktori system dan menggunakan konfigurasi yang telah ditentukan.

A. Pengantar

Panduan ini ditujukan bagi Anda yang ingin membuat dan mengelola paket BlankOn. Meskipun beberapa konsep dalam panduan ini dapat digunakan untuk membuat paket binary untuk penggunaan pribadi, panduan ini didesain untuk mereka yang ingin mendistribusikan paket ke, dan, untuk orang lain. Panduan ini dapat berguna untuk distribusi lain yang berbasis Debian.

Mungkin ada beberapa alasan yang mendasari Anda ketika belajar cara memaketkan program untuk BlankOn:

- Membuat dan memperbaiki paket BlankOn adalah salah satu cara yang bagus untuk berkontribusi bagi komunitas BlankOn.
- Cara ini merupakan cara yang baik untuk mengetahui bagaimana BlankOn dan aplikasi yang telah Anda instal bekerja.
- Mungkin Anda ingin menginstal sebuah paket yang tidak ada dalam lumbung paket BlankOn.
- Kenapa panduan pemaketan ini diperlukan?

Karena cara ini paling efisien untuk membuat suatu paket yang belum ada dalam lumbung paket. Atau mungkin selama ini kita hanya mengenal cara manual dalam hal menginstall paket baru yang belum ada dalam lumbung paket *make install*. Sayangnya cara manual ini

hanya bersifat sementara saat kita memasangnya dan tidak bisa kita bagi ke teman atau ke distro basis Debian lainnya.

Semoga setelah selesai membaca panduan ini Anda memiliki perkakas dan pengetahuan yang cukup untuk melakukan semua itu.

B. Prasyarat

Panduan ini mengasumsikan bahwa kita sudah memiliki pengetahuan yang cukup untuk membangun dan menginstal perangkat lunak dari source pada suatu distribusi Linux. Panduan ini juga menggunakan *Command Line Interface (CLI)* di seluruh bagian, jadi Anda harus terbiasa untuk menggunakan console terminal.

Banyak perkakas yang dapat membantu pemeliharaan paket dalam berkerja, namun pada penjelasan berikut tidak memberikan penjelasan yang mendetail, karena setiap perkakas memiliki dokumentasi komprehensif sendiri.

Setidaknya Anda juga dapat menggunakan hal berikut:

- **make:**

GNU Make adalah perkakas yang sangat penting untuk membangun perangkat lunak. Digunakan untuk mengubah tugas kompilasi yang kompleks menjadi mudah. Sangat penting untuk tahu bagaimana menggunakannya karena kami akan menyimpan informasi mengenai proses pembuatan paket di dalam Makefile. Dokumentasi tersedia pada situs GNU [<http://www.gnu.org/software/make/manual/make.html>].

- **./configure:**

Skrip ini disertakan di hampir semua source Linux, khususnya di perangkat lunak yang ditulis menggunakan bahasa kompilasi seperti C dan C++. Skrip ini digunakan untuk membuat Makefile (berkas yang digunakan oleh make) yang terkonfigurasi sesuai dengan sistem

komputer yang Anda gunakan. Alat pembuatan paket standar Debian menggunakan skrip ini, jadi Anda diharapkan mengetahui apa yang skrip configure lakukan. Informasi tentang configure dapat ditemukan dalam dokumentasi make.

- **apt/dpkg:**

Selain digunakan untuk menginstal program apt dan dpkg memiliki banyak fitur yang berguna untuk pembuatan paket.

- **apt-cache dump:**

Menampilkan daftar setiap paket dalam cache. Paket ini sangat berguna apabila digunakan bersama grep pipe seperti apt-cache dump | grep foo untuk mencari paket yang nama atau dependencies-nya menyertakan "foo".

- **apt-cache policy:**

Menampilkan daftar repository (main/extras/restricted/extras-restricted) dimana suatu paket tersebut ada.

- **apt-cache show:**

Menampilkan informasi mengenai paket binary.

- **apt-cache showsrc:**

Menampilkan informasi mengenai paket source.

- **apt-cache rdepends:**

Menampilkan reverse dependencies untuk sebuah paket (yang dibutuhkan paket untuk melakukan query).

- **dpkg -S:**

Menampilkan daftar paket binary yang dimiliki oleh suatu berkas tertentu.

- **dpkg -I:**

Menampilkan daftar paket yang baru saja dipasang. Perintah ini mirip dengan apt-cache dump tetapi hanya untuk paket yang dipasang.

- **dpkg -c:**

Menampilkan daftar isi dari paket binary. Berguna untuk memastikan bahwa berkas sudah terinstal di tempat yang benar.

- **dpkg -f:**

Menampilkan berkas control untuk paket binary. Berguna untuk memastikan bahwa dependencies sudah benar.

- **Grep-dctrl :**

Mencari informasi khusus di dalam paket. Merupakan penggunaan khusus dari paket grep (tapi tidak terinstal secara baku).

C. Memulai

1. Paket Binary dan Paket Source

Sebagian besar pengguna distribusi berbasis Debian seperti halnya BlankOn tidak akan pernah berurusan dengan kode sumber sebenarnya yang digunakan untuk membuat aplikasi di komputer mereka. Kode sumber melainkan dikompilasi menjadi paket binary dari paket source yang berisi kode sumber itu sendiri dan aturan untuk membuat paket binary. Pemaket mengunggah paket source dengan perubahan yang telah mereka lakukan untuk sistem build lalu mengkompilasi paket binary untuk setiap arsitektur komputer. Sebuah sistem yang terpisah lalu mendistribusikan paket .deb dan source yang telah berubah ke lumbung paket.

2. Peralatan Pemaketan

Ada banyak perkakas yang dibuat khusus untuk pemaketan di sistem berbasis Debian. Kebanyakan dari perkakas tersebut tidak penting untuk

membuat paket namun sangat berguna dan sering mengotomatisasi pekerjaan yang diulang. Halaman man dan info yang menyertai mereka adalah sumber informasi yang berharga. Daftar paket berikut dirasa berguna untuk memulai pemaketan:

- **build-essential**

Metapackage yang bergantung pada libc6-dev, gcc, g++, make, and dpkg-dev. Satu paket yang mungkin kurang Anda kenali adalah dpkg-dev. Paket ini mengandung perkakas seperti dpkg-buildpackage dan dpkg-source yang digunakan untuk membuat, membongkar, dan membangun paket source maupun binary.

- **Devscripts**

Paket devscripts berisi banyak program yang membantu beragam pekerjaan pengembang Debian:

- **debuild** memungkinkan menghasilkan sebuah paket (dengan **dpkg-buildpackage**) dan menjalankan **Lintian** untuk memeriksa kesesuaian dengan kebijakan Debian.
- **debclean** membersihkan paket sumber setelah paket binari dihasilkan,
- **dch** memungkinkan penyuntingan cepat dan mudah dari berkas **debian/changelog** dalam paket sumber,
- **uscan** memeriksa apakah terdapat versi baru perangkat lunak yang dirilis oleh penulis upstream; hal ini membutuhkan berkas **debian/watch** dengan deskripsi dari lokasi rilis,
- **debi** memungkinkan memasang (dengan **dpkg -i**) paket Debian yang baru saja dibuat, dan menghindari mengetik nama lengkap dan path,

- Hal yang mirip, **debc** memungkinkan pencarian isi dari paket yang baru dibuat (dengan **dpkg -c**), tanpa harus mengetik nama lengkap dan path,
- **bts** mengendalikan sistem pelacakan kutu(bug) dari command line; program ini secara otomatis menghasilkan pesan seperlunya.
- **debrelease** mengunggah paket yang baru dibuat ke server remote, tanpa harus mengisi nama lengkap dan path dari berkas **.changes** terkait.
- **debsign** menandakan berkas ***.dsc** dan ***.changes**
- **uupdate** proses pembuatan dari revisi baru paket saat versi upstream telah dirilis.

Beberapa yang sering digunakan adalah debdiff, dch, debuild, dan debsign:

- **debhelper dan dh-make**

Skrip untuk mengotomatisasi tugas pembuatan paket. dh-make dapat digunakan untuk membuat "debianization" dan menyediakan banyak berkas contoh,

- **diff dan patch**

Digunakan untuk membuat dan menerapkan patch, berturut-turut. Kedua aplikasi tersebut sering digunakan dalam pembuatan paket karena lebih mudah, bersih dan efisien untuk menampilkan perubahan kecil sebagai patch daripada harus menggunakan banyak salinan berkas,

- **gnupg**

- Pengganti dari PGP yang lengkap dan bebas untuk digunakan menandatangani berkas secara digital (termasuk juga paket),

- **fakeroot**

Melakukan simulasi bagaimana cara menjalankan perintah dengan hak akses root. Hal ini sangat berguna bila Anda ingin membuat paket binary dari pengguna biasa,

- **lintian**

Membedah paket Debian, pelaporan bug, dan pelanggaran Kebijakan. Paket ini berisi pemeriksaan otomatis untuk banyak aspek dari Kebijakan Debian seperti halnya error umum,

- **pbuilder**

Membangun sistem chroot dan membuat paket di dalam chroot. Merupakan sistem yang ideal untuk digunakan jika sebelumnya sebuah paket diperiksa apakah telah memiliki dependency yang tepat serta membangun paket yang bersih dan siap untuk diuji serta didistribusikan.

Sebelum membangun paket, diwajibkan untuk memasang pustaka paket berikut ini :

```
sudo apt-get install devscripts build-essential fakeroot debhelper gnupg pbuilder dh-make dpkg-dev ubuntu-dev-tools
```

3. Mulai membangun paket

- Informasi pribadi :

```
$ nano ~/.bashrc
(isi pada baris terakhir)
export DEBFULLNAME="Joe Hacker" (sesuaikan dengan nama Anda)
export DEBMAIL="joe.hacker@bungker.com" (sesuaikan dengan e-mail Anda)
```

- Memeriksa informasi pribadi yang telah dibuat

```
$ source ~/.bashrc  
$ export | grep DEB
```

- Membuat kunci pribadi

```
$ gpg -gen-key  
(Buat)  
Real name: Joe Hacker (Nama = Nama Anda)  
E-mail address: joe.hacker@bungker.com (e-mail = e-mail Anda)  
Passphrase: hanyatest (buat password anda)
```

Ada 2 cara yang umum digunakan dalam membangun paket :

I. Membangun paket dengan dpkg-buildpackage

Cara ini tergantung serba manual, dimana kita harus memasang dependensi(paket ketergantungan) terlebih dahulu dan juga cara ini bisa membuat system kita kotor dengan library yang tidak dibutuhkan oleh system kita, serta cara ini bersifat pribadi tidak dianjurkan untuk mendistribusikan paket yang telah berhasil dibuat.

Mengambil kode sumber paket:

```
$ mkdir ~/Pemaketan  
$ cd ~/Pemaketan  
$ wget ftp://ftp.gnu.org/pub/gnu/ed/ed-1.6.tar.gz  
$ tar zxvf ed-1.6.tar.gz  
$ cd ed-1.6
```

```
$ ls
```

```
$ dh_make -e isi-email-kita -f ../ed-1.6.tar.gz
```

Type of package: single binary, multiple binary, library, kernel module or cdb?

```
[s/m/l/k/b] s
```

(Pilih "s")

```
$ cd debian
```

```
$ ls
```

```
$ rm *.ex *.EX docs info README.*
```

```
$ ls
```

Secara default kita menggunakan editor text nano

```
$ cd ../
```

```
$ dch -e
```

Isi seperti berikut:

```
ed (1.6-1) unstable; urgency=low
```

```
* Initial release
```

```
-- Joe Hacker <joe.hacker@bungker.com> 20:36:22 +0700
```

Tekan Ctrl + X untuk menyimpan dan tekan Y untuk konfirmasi.

```
$ cd debian/
```

```
$ nano control
```

(Lengkapi seperti berikut):

Source: ed

Section: editors

Priority: extra

Maintainer: Joe Hacker <joe.hacker@bungker.com>

Build-Depends: debhelper (>= 8.0.0), autotools-dev

Standards-Version: 3.9.2

Homepage: <http://www.gnu.org/software/ed/>

Vcs-Git: [git://github.com/joe/ed.git](http://github.com/joe/ed.git)

Vcs-Browser: <https://joe.hacker@github.com/joe/ed.git>

Package: ed

Architecture: any

Depends: \${shlibs:Depends}, \${misc:Depends}

Description: classic UNIX line editor

The ed is a line-oriented text editor. It is used to create, display, modify and otherwise manipulate text files.

\$ nano copyright

(Lalu isi sebagai berikut):

Format: [http://svn.debian.org/wsvn/dep/web/deps/dep5.mdwn?](http://svn.debian.org/wsvn/dep/web/deps/dep5.mdwn?op=file&rev=174)

[op=file&rev=174](http://svn.debian.org/wsvn/dep/web/deps/dep5.mdwn?op=file&rev=174)

Upstream-Name: ed

Source: <http://www.gnu.org/software/ed/>

Files: *

Copyright: 1993, 1994, 2006, 2007, 2008, 2009, 2010, 2011 Free Software Foundation, Inc

2006, 2007, 2008, 2009 Backus <ericb@lsid.hp.com>

1993, Karl Berry <kb@cs.umb.edu>

1994, 2011 Theo Deraadt <deraadt@newt.fsa.ca>

2006, 2007 Kaveh R. Ghazi <ghazi@noc.rutgers.edu>

2010, 2011 Mike Haertel <mike@ichips.intel.com>

2011 Francois Pinard <pinard@iro.umontreal.ca>

1993, 1994 Rodney Ruddock <rodney@snowwhite.cis.uoguelph.ca>

License: GPL-3.0+

Files: debian/*

Copyright: 2011 Joe Hacker <joe.hacker@isp.com>

License: GPL-3.0+

License: GPL-3.0+

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This package is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Tekan Ctrl + X untuk menyimpan dan tekan Y untuk konfirmasi

```
$ cd ../
```

Periksa dependensi apakah sudah terpasang

```
$ dpkg-checkbuilddeps
```

Jika dependensi belum terpasang akan muncul dan silahkan dipasang, jika sudah terpasang tidak keluar apa-apa. Langsung lanjut membangun paket dengan menjalankan perintah :

```
$ dpkg-buildpackage -rfakeroot
```

Biasanya lintian akan meminta Passphrase, isi dengan Passphrase yang telah dibuat tadi.

II. Membangun paket dengan pbuilder

Pbuilder atau personal package builder.

"Mengapa menggunakan Pbuilder ?"

Pbuilder adalah Sistem pembangun paket Debian otomatis, memudahkan proses membangun paket Debian secara otomatis di dalam lingkungan yang bersih dari pustaka atau dependensi program yang tidak seharusnya dan dibangun pada sistem Debian lain misal akan diunggah ke lumbung paket BlankOn.

Dengan lingkungan yang bersih maka paket-paket yang dibangun dan dihasilkan dari paket sumber akan membutuhkan paket-paket yang sesuai/cocok di lumbung paket standar distro asal atau turunan.

Penggunaan pbuilder sebagai pembangun paket memungkinkan Anda untuk membangun paket dari dalam lingkungan chroot. Anda bisa saja membangun paket binary tanpa perlu menggunakan pbuilder, tapi Anda harus memiliki semua depedensi terinstal di sistem Anda terlebih dahulu. Karena pbuilder memungkinkan pemaket untuk memeriksa depedensi pembangun karena paket dibuat dalam instalasi system minimal, dan ketergantungan pembangun diunduh menurut berkas debian/control.

Berikut adalah panduan singkat cara menginstal, menggunakan, dan memperbaharui lingkungan pbuilder, tapi banyak detail dari penggunaan pbuilder yang tidak disertakan dalam panduan ini. Halaman manual pbuilder memiliki banyak informasi yang dapat digunakan sebagai rujukan bila Anda mengalami masalah atau membutuhkan informasi yang lebih mendetail.

a. Membuat informasi untuk pbuilder

```
$ nano /etc/pbuilderrc
```

(isi sesuai distro yang anda inginkan)

```
COMPONENTS="main restricted extras extras-restricted"
```

```
DISTRIBUTION=suroboyo
```

```
MIRRORSITE=http://arsip.blankonlinux.or.id/blankon/
```

```
DEBOOTSTRAP_OPTS=('--keyring' '/usr/share/keyrings/blankon-  
archive-keyring.gpg')
```

b. Mulai membangun pbuilder

```
$ sudo pbuilder create
```

Tunggu sampai proses selesai membangun BlankOn (distro) minimal

c. Memperbarui pbuilder

```
$ sudo pbuilder --update
```

c. Membangun paket di pbuilder

Caranya sama seperti diatas, namun yang pertama tadi kita menjalankan perintah

```
dpkg-buildpackage -rfakeroot
```

Untuk pbuilder, jalankan perintah :

```
$ debuild -S
```

tunggu sampai proses selesai, dan jalankan

```
$ cd ../
```

```
$ sudo pbuilder build ed_1.6-1.dsc
```

Untuk hasilnya bisa di cek di direcktori

```
$ cd /var/cache/pbuilder/result/
```

```
$ ls
```