

# Panduan Pemaketan BlankOn



Di susun  
Tim Pengembang BlankOn



*Persembahan dari BlankOn untuk Anak Negeri*

*E-book ini masih dalam tahap revisi yang akan terus direvisi*

Halaman Pengembang BlankOn : <http://dev.blankonlinux.or.id/>

Milis Pengembang BlankOn : <http://groups.google.com/group/BlankOn-dev>

Milis Pengguna : <http://groups.google.com/group/BlankOn>

Tiket : <http://dev.blankonlinux.or.id/report>

Chat Room Blankon : server *irc.freenode.net* / kanal : *#blankon*

# Panduan Pemaketan BlankOn

Panduan Pemaketan BlankOn adalah pengantar dalam program pembuatan paket untuk BlankOn dan distribusi linux berbasis Debian.

## Bab I. Pengantar

Panduan ini ditujukan bagi Anda yang ingin membuat dan mengelola paket BlankOn. Walaupun 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. Walau panduan ini ditulis untuk digunakan pada distribusi BlankOn Linux, namun panduan ini dapat berguna untuk distribusi lain yang berbasis dari 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 di repository BlankOn.
- Kenapa panduan pemaketan ini diperlukan?  
Karena cara ini paling efisien untuk membuat suatu paket yang belum ada di repository, mungkin selama ini kita hanya mengenal cara manual dalam hal menginstall paket baru yang belum ada di repository `./configure` (`./autogen.sh`), `make` dan `sudo make install`. Sayangnya cara manual ini hanya bersifat sementara disaat kita menginstallnya dan tidak bisa kita share ke distro (debian dan turunannya) lain.

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

## Prasyarat

Panduan ini mengasumsikan bahwa pembaca 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 nyaman untuk menggunakan terminal. 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 binar y.
  - **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 diinstal. Perintah ini mirip dengan `apt-cache dump` tetapi hanya untuk paket yang diinstal.
  - **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 paketgrep (tapi tidak terinstal secara baku)
- **diff**: Program diff dapat digunakan untuk membandingkan dua berkas dan membuat patch. Contoh yang sering dipakai yaitu `diff -rN file.old file.new > file.diff`. Perintah ini akan membuat diff (rekrusif jika ada direktori) yang menampilkan perubahan, atau “delta”, diantara dua buah berkas.
- **patch** : Program patch digunakan untuk menerapkan suatu patch (biasanya dibuat dengan diff atau program lain) ke suatu berkas atau direktori. Untuk menerapkan patch yang dibuat di atas, Anda dapat mengetik `patch -p0 < file.diff`. Opsi `-p` memberitahu patch berapa banyak harus men-strip path dari nama berkas dalam patch. `-p0` artinya tidak men-strip apapun, atau tetap menggunakan path.

## Bab II. Memulai

### 1. Paket Biner 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 malah dikompilasi menjadi paket binary dari paket source yang berisi kode sumber itu sendiri dan aturan untuk membuat paket binary. Pemaket meng-upload 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 berkas .deb binary dan source yang telah berubah ke repository mirror.

### 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* adalah 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 dan binary.
- *devscripts* memuat banyak skrip yang membuat pekerjaan pengelolaan paket menjadi mudah. Beberapa yang sering digunakan adalah `debdiff`, `dch`, `debuild`, dan `debsign`.

- *debhelper dan dh-make*  
adalah 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*  
adalah pengganti dari PGP yang lengkap dan bebas untuk digunakan menandatangani berkas secara digital (termasuk juga paket).
- *fakeroot*  
mensimulasi 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.

Install peralatan :

```
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](mailto:joe.hacker@bungker.com) (e-mail = e-mail Anda)

Passphrase: hanyatest (buat password anda)



## I. Tahap pertama dengan *dpkg-buildpackage*

Cara ini tergantung serba manual, dimana kita harus menginstall dependenciesnya terlebih dahulu

```
sudo apt-get build-dep ed
```

- Mulai bermain dengan source untuk di paket kan :

```
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 isiemailkita -f ../ed-1.6.tar.gz
```

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

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

**(Pilih "s")**

```
cd debian
```

```
ls
```

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

```
ls
```

- (Masih di direktori debian, tulis):

```
dch -e
```

**(Lalu isi seperti berikut):**

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

```
* Initial release (Closes: #nnnn) <nnnn is the bug number of  
your ITP>
```

```
-- Joe Hacker <joe.hacker@bungker.com> Tue, 29 May 2012  
20:36:22 +0700
```

- selanjutnya

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
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?
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/>.
```

Lalu pindah ke direktori sebelumnya dengan cara ketik:

```
cd ..
```

- Terakhir langsung jalankan perintah untuk membangun pakatnya  
`dpkg-buildpackage -rfakeroot`  
jika diminta Passphrase, isi dengan Passphrase yang anda buat tadi.

## II. Tahap kedua dengan *pbuilder*

*Pbuilder* = *personal package builder*

### Kenapa menggunakan *Pbuilder* ?

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 dapat dengan lancar jaya dipasang dan dibangun pada sistem Debian lain misal akan diunggah ke pabrik 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 dependency terinstal di sistem Anda terlebih dahulu. Karena *pbuilder* memungkinkan pemaket untuk memeriksa dependency pembangun karena paket dibuat dalam instalasi BlankOn minimal, dan ketergantungan pembangun di-download menurut berkas debian/control.

Berikut adalah panduan singkat cara menginstal, menggunakan, dan meng-update 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.

- Membuat informasi untuk *pbuilder*

```
nano /etc/pbuilderrc
```

(isi sesuai distro yang anda inginkan)

```
COMPONENTS="main restricted extras extras-restricted"  
DISTRIBUTION=rote  
MIRROR_SITE=http://arsip.blankonlinux.or.id/blankon/  
DEBOOTSTRAP_OPTS=('--keyring' '/usr/share/keyrings/blankon-  
archive-keyring.gpg')
```

- Mulai membangun *pbuilder*

```
sudo pbuilder create
```

Tunggu sampai proses selesai membangun BlankOn (distro) minimal

- Membangun paket di pbuilder

Caranya sama seperti diatas, namun jika 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 direcktory

```
cd /var/cache/pbuilder/result/  
ls
```

Catatan :

- E-book ini masih dalam versi jahitan (alpha version)
- Jika ada masukan silahkan di tiketkan

*Sumber referensi :*  
*Debian Package guide*  
*Ubuntu Package guide*  
*Pemaketan BlankOn*