Membangun Paket Debian yang Bersih dengan pbuilder

stwn

Konferensi BlankOn #2, 31 Juli 2010M
Universitas Surabaya

nama: Iwan Setiawan
panggilan daring: stwn

Aktivitas terkini:
te.unsoed.ac.id dan kuliaz.org

Blankon?

Baru menjadi pengguna BlankOn
(selama 3 hari)

Kesan?

- memasang BlankOn di tablet
- wacom aktif dan terkonfigurasi otomatis
- konfigurasi *active protection system* (aps)

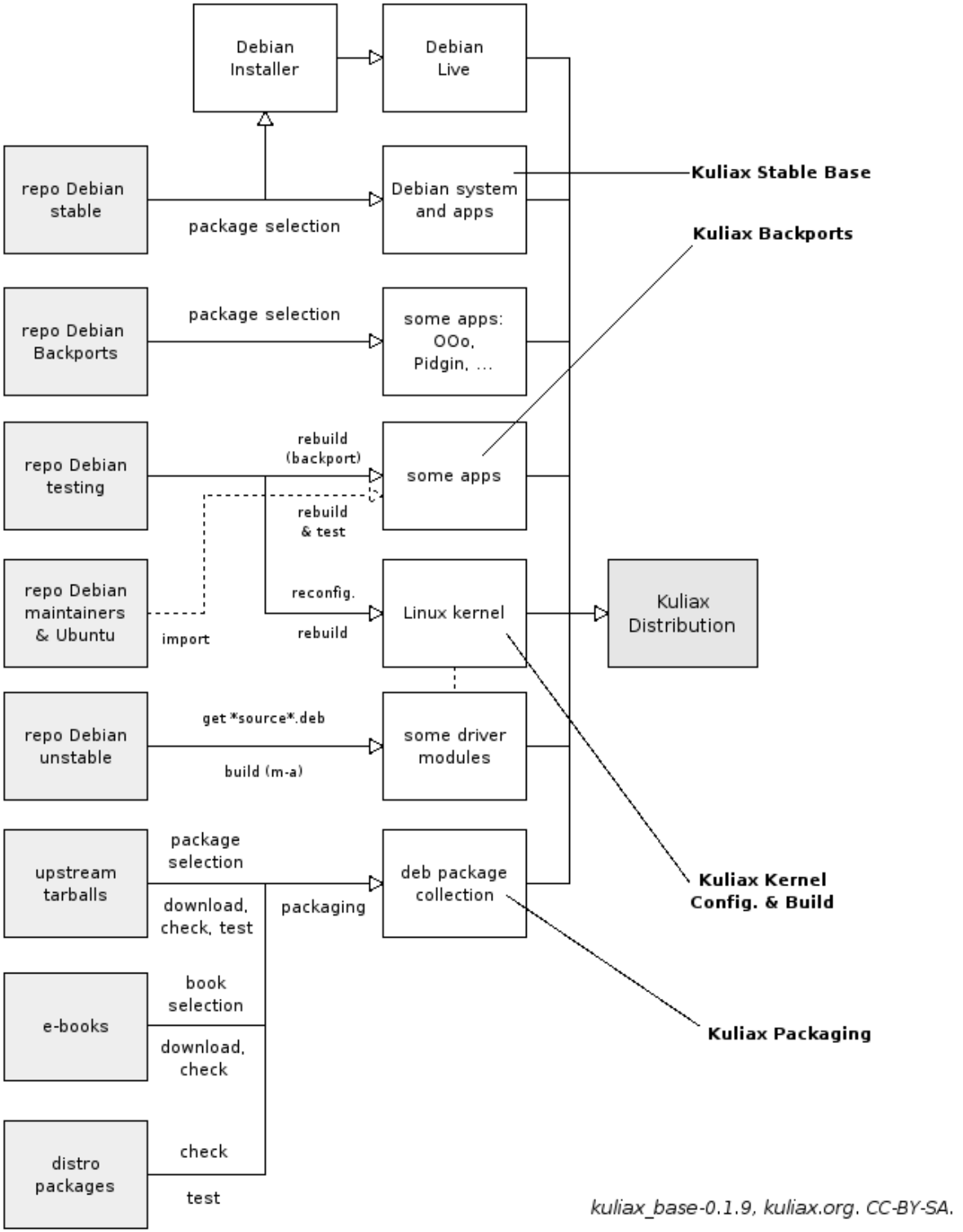
Membangun Paket Debian yang Bersih dengan pbuilder

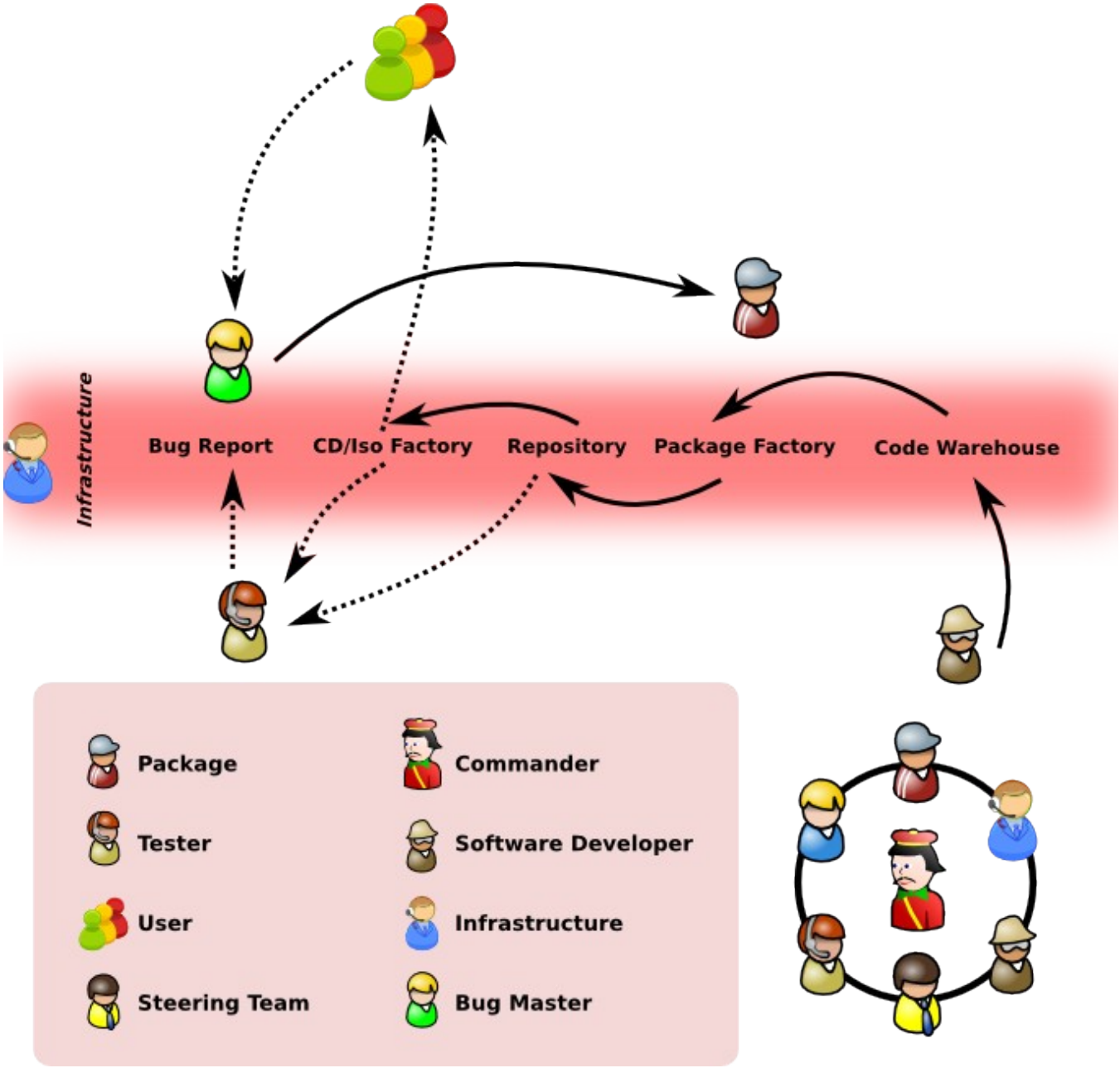
Membangun? **Paket?** Debian? Bersih? pbuilder?

Kenapa paket?

memudahkan pengelolaan program
(dalam sistem GNU/Linux kita)

Kenapa membangun paket?





dari presentasi somat: oo-presentasi-blankon.odp

paket program tidak ada di dalam repo distro

paket di dalam repo distro *jadul* (kurang termutakhirkan)

inqin membcuat paket Debian dari program buatan sendiri

ada kesalahan pada paket dalam repo distro

kita ingin berkontribusi ke komunitas perangkat lunak bebas

iseng saja daripada nganggur :P

Hasilnya..

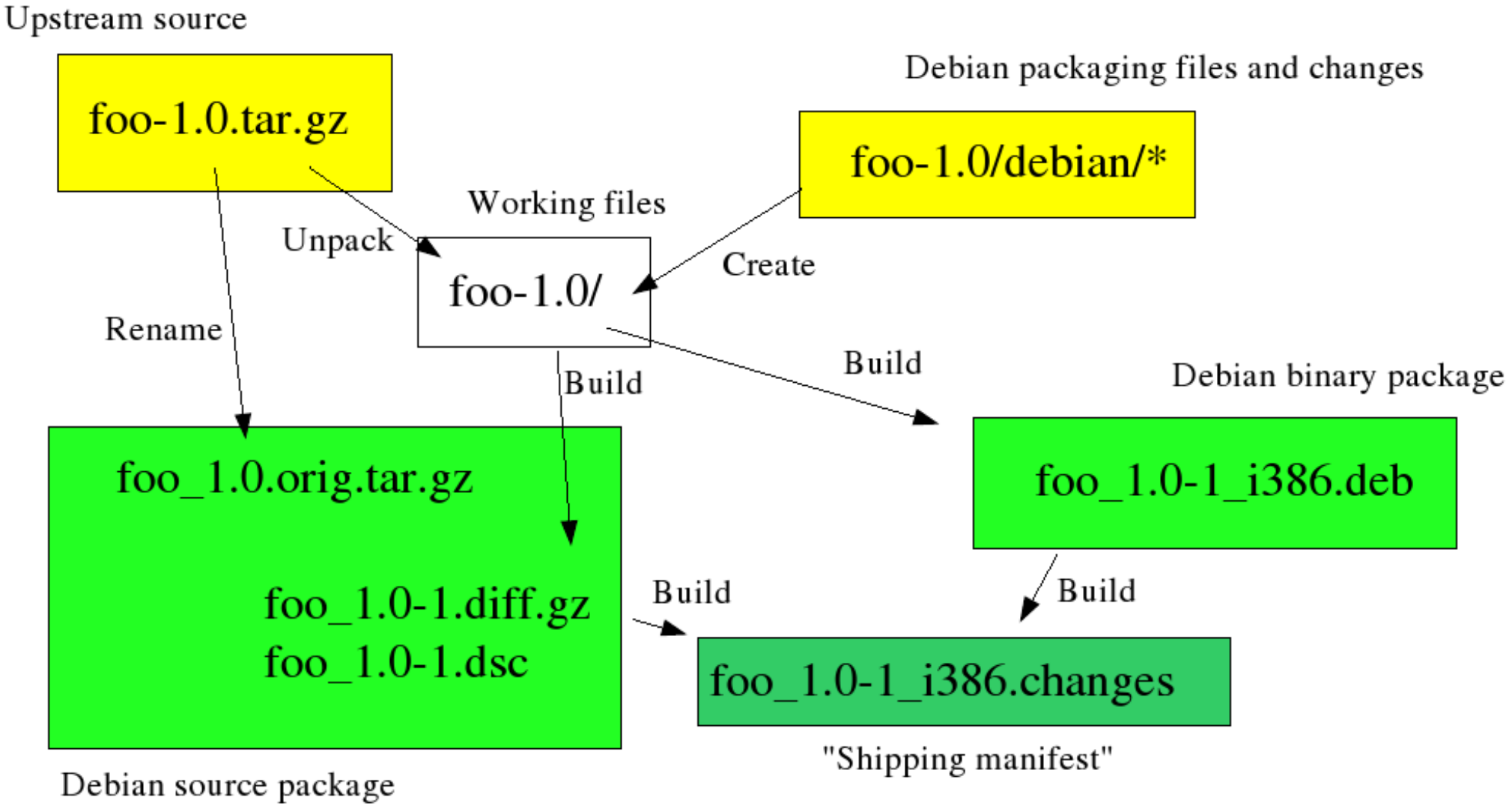
bermanfaat untuk kita :D

juga orang lain :)

kita senang, orang lain senang

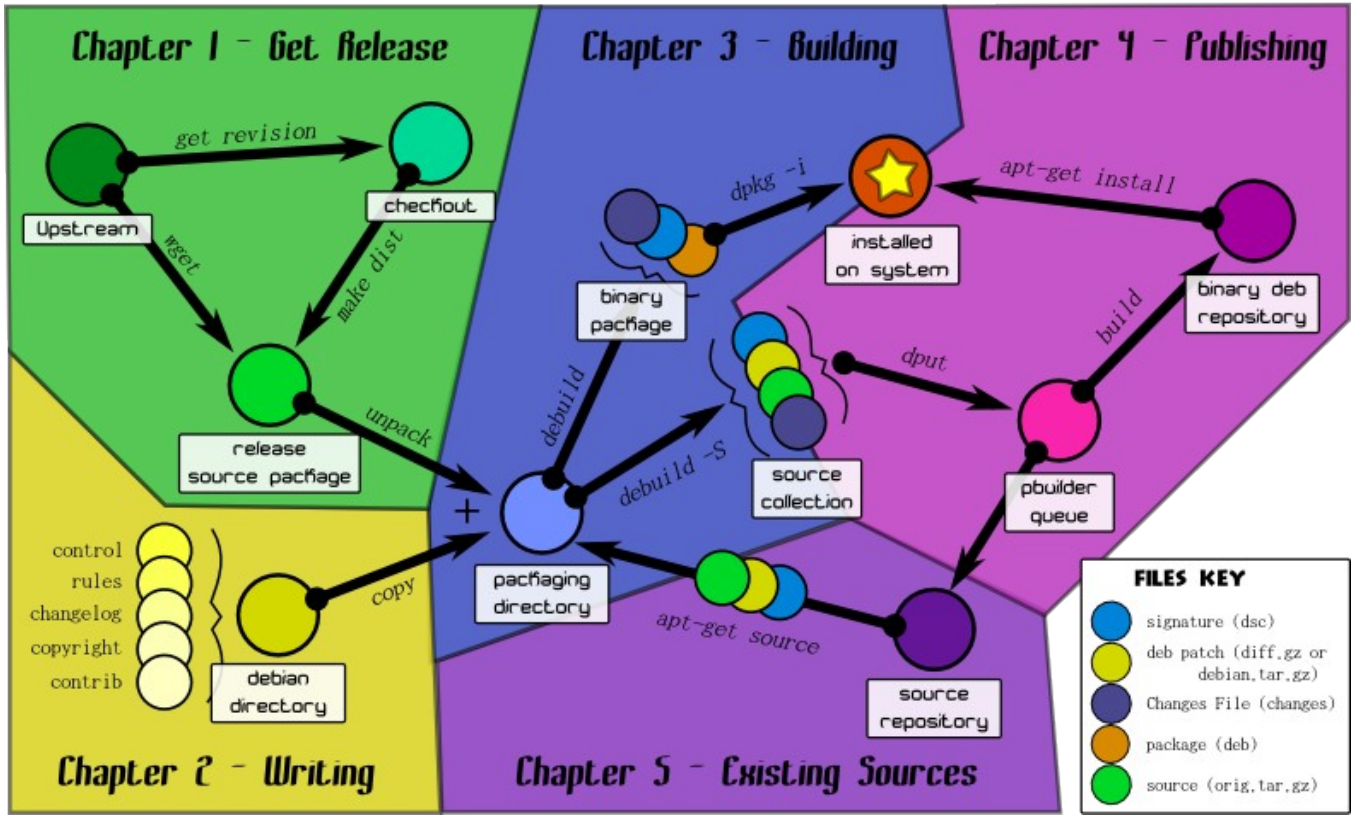
Pemaketan Debian?

Overview of packaging process



dari Lars Wirzenius, Debian Packaging School: Lesson 1

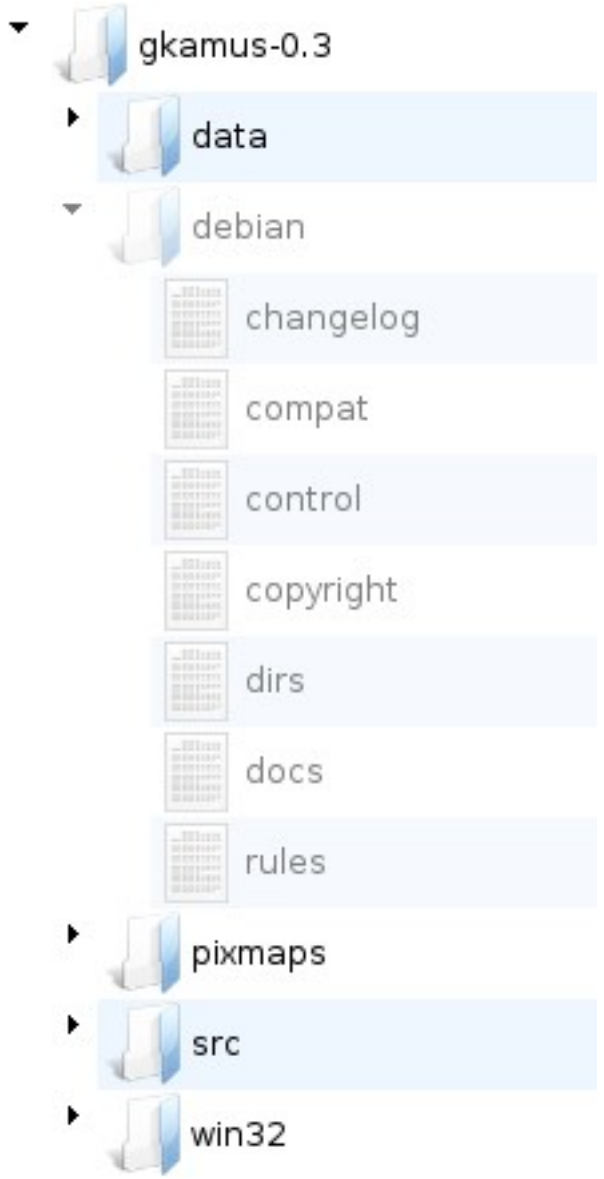
Contents

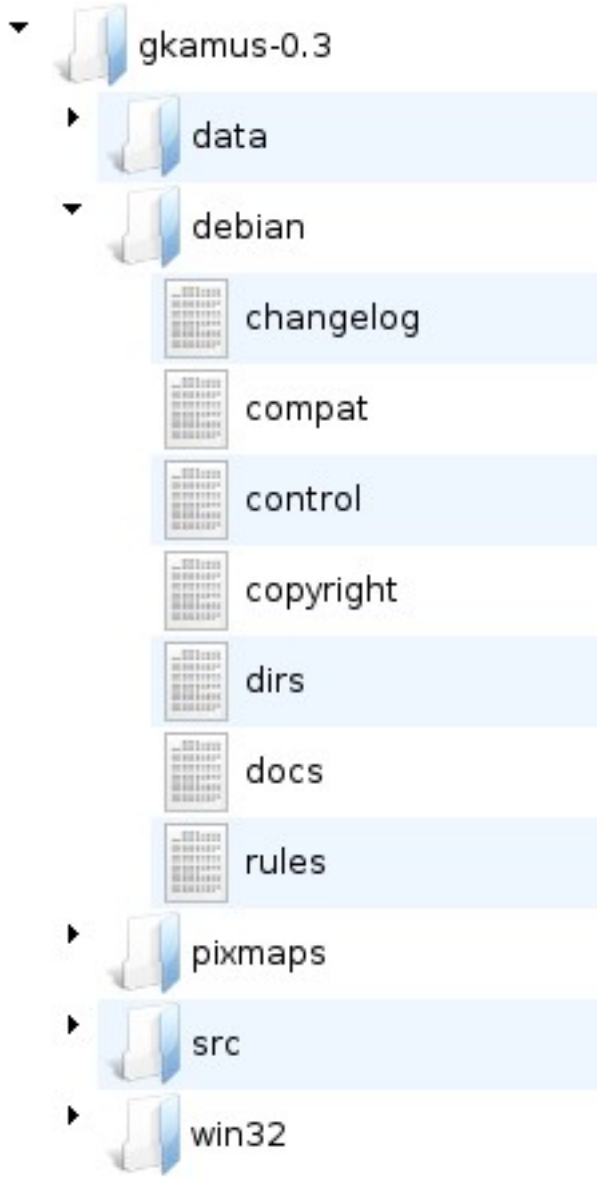


dari Deb Package Contents http://doctormo.org/2010/04/19/deb-package-contents/

debian/{changelog,control,copyright,rules}
(dirs,preinst,postinst,prerm,postrm,...)

selain berkas-berkas sumber


gkamus-0.3
data
debian
changelog
compat
control
copyright
dirs
docs
rules
pixmaps
src
win32


gkamus-0.3
data
debian
changelog
compat
control
copyright
dirs
docs
rules
pixmaps
src
win32

paket sumber: .orig.tar.gz, .diff.gz, .dsc

debian/control: Build-Depends, Depends

debian/control

Source: cnet
Section: net
Priority: optional
Maintainer: Iwan Setiawan <stwn@keuliax.org>
Build-Depends: debhelper (>= 7), tcl8.4-dev, tk8.4-dev
Standards-Version: 3.7.3
Homepage: http://www.csse.uwa.edu.au/cnet/

Package: cnet
Architecture: any
Depends: ${shlibs:Depends}
Description: a network protocol simulation program

Kenapa pbuilder?

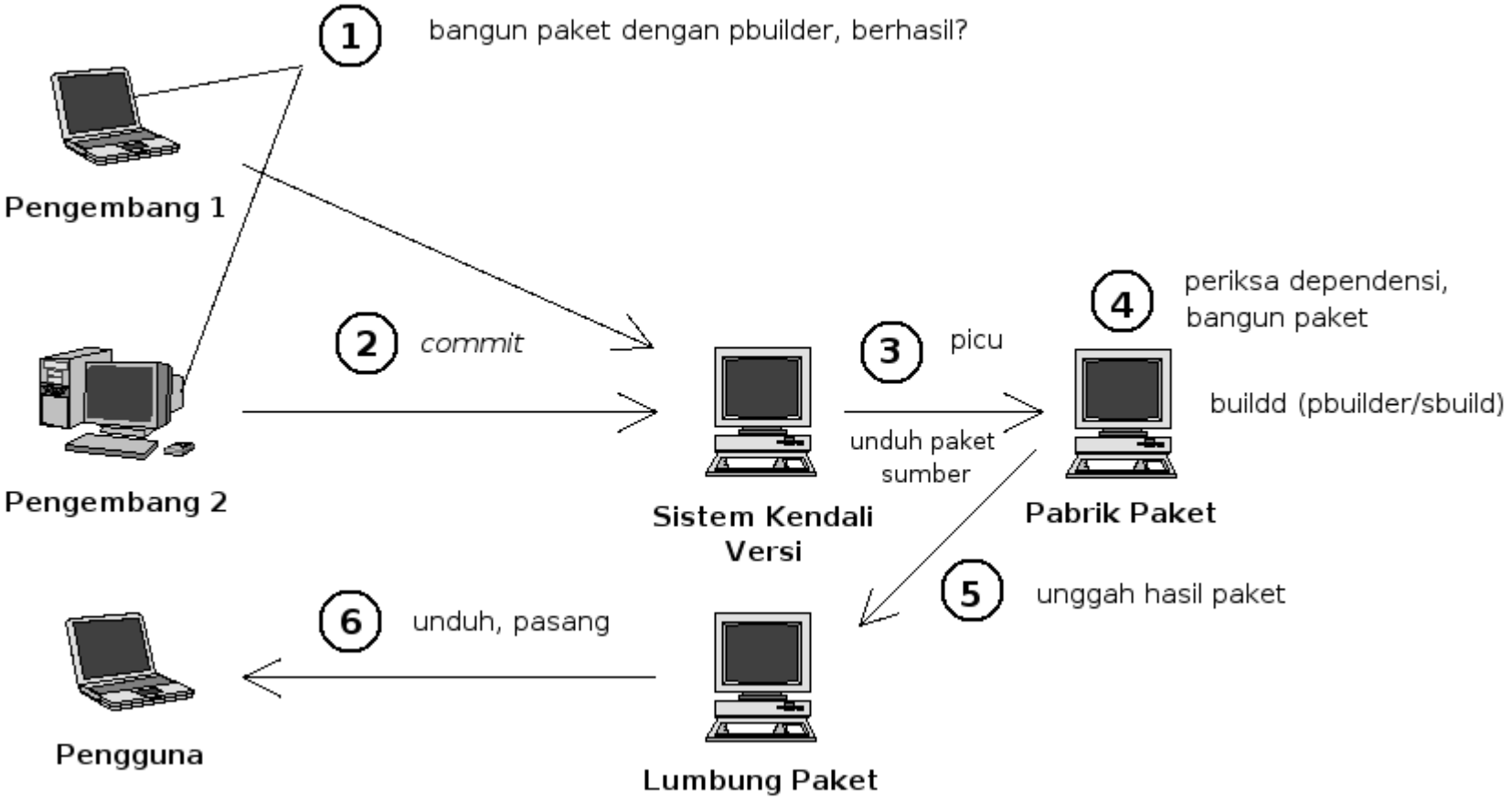
pbuilder = *personal package builder*

sistem pembangun paket Debian otomatis

memudahkan proses membangun paket Debian secara otomatis di dalam lingkungan yang bersih

bersih dari "kotoran" 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


1
bangun paket dengan pbuilder, berhasil?
Pengembang 1
2
commit
Pengembang 2
3
picu
unduh paket sumber
Sistem Kendali Versi
4
periksa dependensi, bangun paket
buildd (pbuilder/sbuild)
Pabrik Paket
5
unggah hasil paket
6
unduh, pasang
Pengguna
Lumbung Paket

lingkungan yang bersih di pbuilder didapatkan dengan:

chroot melalui (c)debootstrap dan paket-paket minimal/dasar pembangun yang dipasang di dalamnya (essential, apt, build-essential)

Komponen essential Debian dan BlankOn berbeda

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

Beberapa keuntungan pbuilder

dapat menggunakan sistem yang berbeda dari target
dapat memeriksa dan memenuhi dependensi pembangunan paket
dapat digunakan sebagai alat pengujicoba paket

ketika kita menggunakan sistem pembangun normal, bukan chroot, akan kita dapati bahwa paket yang kita bangun mungkin dirakit dengan pustaka atau program versi berbeda dari target sistem

Contoh:

kita ingin membangun paket untuk ombilin, maka kita harus menggunakan sistem ombilin yang standar dan bersih

jika kita membangun paket tersebut di nanggar misalnya, bisa jadi paket kita "terikat" dengan program dan pustaka yang ada di dalamnya sehingga menjadikannya tidak bersih/tidak dapat dipasang/dibangun kembali pada target ombilin

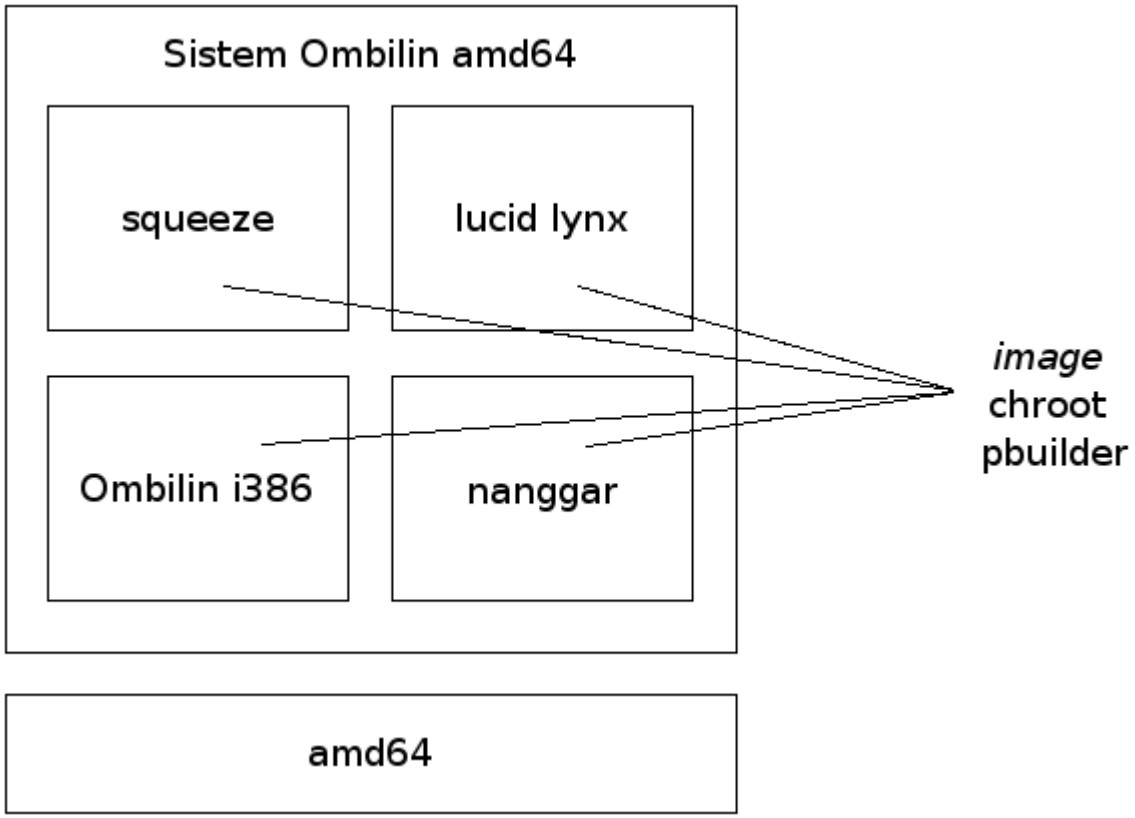
Misal kita memiliki sistem:

- nanggar dan ingin membangun paket untuk ombilin
- ombilin dan ingin membangun paket untuk nanggar
- nanggar dan ingin membangun paket untuk lucid (Ubuntu)
- ombilin dan ingin membangun paket untuk squeeze (Debian)
- ombilin amd64 dan ingin membangun paket untuk ombilin i386

Debian GNU/Linux	
4.0	Etch
5.0	Lenny
6.0	Squeeze

Ubuntu	
9.10	Karmic
10.04	Lucid
10.10	Maverick

BlankOn	
4.1	Meuligoe
5.0	Nanggar
6.0	Ombilin


Sistem Ombilin amd64
squeeze
lucid lynx
Ombilin i386
nanggar
image
chroot
pbuilder
amd64

Dengan cara biasa, untuk memenuhi dependensi ombilin, nanggar harus kita upgrade sebagian paketnya ke ombilin, dan hal ini membuat sistem kita tidak bersih dan tidak "asli" lagi seperti yang kita inginkan

Dengan pbuilder maka hal2 tersebut tidak perlu dilakukan

image sistem dasar pbuilder, tidak akan diubah, karena setiap pembangunan, sistem akan diekstrak dari arsip base.tgz

Pbuilder dapat dikonfigurasi dengan baris perintah atau berkas konfigurasi (/etc/pbuilderrc atau ${HOME}/.pbuilderrc)

Pbuilder tidak menguji coba program tetapi menguji coba bahwa sebuah paket program dapat dibangun dan dependensinya terpenuhi

pbuilder {create, build, update}

Kebutuhan membangun paket Debian dengan pbuilder

- komputer* terpasang BlankOn
- akses** lumbung paket BlankOn, ~32 GB(?)
- ruang cakram/*disk* yang cukup
- (c)debootstrap, pbuilder, dan paket pengembangan***
- peladen web: apache2/lighttpd untuk lumbung lokal

* dengan spesifikasi perangkat keras yang memadai. semakin canggih dan besar kapasitasnya, semakin bagus

** dalam penyimpan, jaringan lokal, atau Internet. Serta dibutuhkan konfigurasi jaringan & lumbung yang benar

*** build-essential, debhelper, devscripts, dh-make, fakeroot, gpg, dan paket lain yang dibutuhkan

/etc/apt/sources.list:

deb http://lumbung/blankon ombilin main extras restricted extras-restricted

Langkah-Langkah Membangun Paket Debian dengan pbuilder

1. konfigurasi
2. membuat *image* sistem dasar chroot
3. mengunduh paket sumber, dan modifikasi
4. membangun paket dengan pbuilder
5. ujicoba paket

1. Konfigurasi

Konfigurasi (1)

```
$ nano .bashrc
```

```
export DEBFULLNAME="Iwan Setiawan"
export DEBEMAIL="stwn@kuliax.org"
export EDITOR="vim"
export PATH="/usr/local/bin:/usr/bin:/bin:$HOME/tmp/dev/scripts"
```

```
$ source .bashrc
```

CATATAN: .bashrc dapat pula diganti dengan .bash_profile :)

Konfigurasi (2)

```
$ gpg --gen-key
```

```
Please select what kind of key you want:
    (1) DSA and Elgamal (default)
    (2) DSA (sign only)
    (5) RSA (sign only)
Your selection?
DSA keypair will have 1024 bits.
ELG-E keys may be between 1024 and 4096 bits long.
```

Konfigurasi (3)

What keysize do you want? (2048)
Requested keysize is 2048 bits

Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
Key is valid for? (0) 12m
Key expires at Mon 03 Jan 2011 08:20:24 AM UTC
Is this correct? (y/N) y

Konfigurasi (4)

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Iwan Setiawan
Email address: stwn@kuliax.org
Comment: stwn
You selected this USER-ID:
    "Iwan Setiawan (stwn) <stwn@kuliax.org>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O

Konfigurasi (5)

We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.

```
.++++++++++++++.++++++++++...+++++.++++++++++++++.+++++.++++++++
+..++++++++++
++++++++..+++++++++++++++.++++++++++...++++++++++++++++++++.++++++
++++++++++++++
>+++++..+++++>.+++++.....+++++^^^
```

gpg: key XXXXXXXX marked as ultimately trusted public and secret key created and signed.

Konfigurasi (6)

```
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2011-01-03
pub 1024X/XXXXXXXX 2010-01-08 [expires: 2011-01-03]
      Key fingerprint = 5X70 93X5 756A XXXX 4164  8264 6X87
232X X2X1 6887
uid    Iwan Setiawan (stwn) <stwn@kuliax.org>
sub    2048x/4XXX3954 2010-01-08 [expires: 2011-01-03]
```

Konfigurasi (7)

```
$ vim .pbuilderrc

MIRRORSITE=http://localhost/blankon/
DEBOOTSTRAP="debootstrap"
DISTRIBUTION="ombilin"
COMPONENTS="main extras restricted extras-restricted"
AUTO_DEBSIGN=yes
APTCACHE=$HOME/blankon-dev/pbuilder/cache/
BUILDPLACE=$HOME/blankon-dev/pbuilder/build/
BUILDRESULT=$HOME/blankon-dev/repo/blankon/
BINDMOUNTS=$HOME/blankon-dev/repo/blankon/
HOOKDIR=$HOME/blankon-dev/pbuilder/hook/
```

Konfigurasi (8)

Buat *symlink* di direktori root peladen web:

```
$ sudo ln -s /media/repo/blankon /var/www/blankon
```

Buat direktori-direktori yang dibutuhkan:

```
$ mkdir -p $HOME/blankon-dev/pbuilder/{build,cache,hook}
$ mkdir $HOME/blankon-dev/{pkgs,repo}
```

Opsional: jika RAM besar misal 4 GB percepat kinerja dengan konfigurasi ini

```
$ sudo mount -t tmpfs tmpfs $BUILDPLACE
```

Membuat image dasar chroot

```
$ sudo pbuilder create
```

Hasil: /var/cache/pbuilder/base.tgz

Proses `pbuilder create`:

1. jalankan (c)debootstrap
2. salin konfigurasi lokal*
3. periksa hook
3. upgrade paket, jika ada
4. membuat arsip base.tgz

* hosts, resolv.conf, sources.list

pbuilder hook

Dengan hook kita dapat menambah fungsi-fungsi pada proses pbuilder

Terdapat aturan huruf awal pada setiap skrip/program yang akan kita jadikan hook pada pbuilder: A, B, C, D, E (man 8 pbuilder)

Ingat set mode eksekusi pada hook!

Contoh pbuilder hook

```
#!/bin/bash
# example file to be used with --hookdir
#
# run tests. Current directory is top of source-code.
#
# 2005, 2007 Junichi Uekawa
#
set -e

echo "Installing the prerequisites"
for PKG in $(ls /tmp/buildd/*.deb | sed -e's,.*/,,;s,_.*,,' ); do
     apt-get install -y --force-yes "$PKG" || true
     apt-get remove -y "$PKG" || true
done
# ignore the failures since they are not the prime interest

dpkg -i /tmp/buildd/*.deb
...
```

Mengunduh paket sumber

$ apt-get source pkg*

(pkg*{.orig.tar.gz,.diff.gz,.dsc})

atau salin berkas tersebut secara manual :D dan
jalankan dpkg-source -x pkg.dsc, atau gunakan dget URL/pkg.dsc

jika kita membuat sendiri paketnya
lakukan perintah `debuild -S -sa`
untuk membuat paket sumber

* ingat, harus ada baris deb-src di sources.list

dh_make*, sunting debian/, dch, debuild -S -sa, pbuilder build pkg.dsc

* untuk paket baru buatan sendiri yang belum ada direktori debian/

Membangun Paket

$ sudo pbuilder build pkg.dsc

Proses yang terjadi: ekstrak base.gz ke $BUILDPLACE, masuk chroot, memenuhi dependensi pembangunan, membangun paket

Hasil akan ada di $BUILDRESULT
Cara lain gunakan pdebuild sebagai penggganti debuild

Jika muncul pesan kesalahan bahwa program yang dirakit dan akan dibangun membutuhkan pustaka/program tertentu, sunting Build-Depends pada debian/control dan bangun ulang

Bagaimana cara memeriksa Depends? pada kasus yang umum sudah ada variabel ${shlibs:Depends} ;)

Jangan gunakan perangkat *removable* seperti perangkat penyimpan drive eksternal untuk direktori $BUILDPLACE dan $HOOKDIR, khususnya yang dikaitkan (*mount*) melalui HAL atau aplikasi seperti nautilus atau konqueror karena konfigurasi mount asalnya adalah **noexec** dan **nodev**

Untuk membersihkan direktori $BUILDPLACE dan $APTCACHE:

```
$ sudo pbuilder clean
```

pbuilder umum pula dijadikan alat pengujian apakah sebuah paket telah sesuai dependensi pembangunannya

Ujicoba paket

Manual dengan: dpkg -i pkg.deb/gdebi pkg.deb
atau dapat melalui *hook* misal yang melakukan:
pasang, hapus, *upgrade*, hapus total

pbuilder untuk *backport* paket

Umumnya membutuhkan proses *backport* dependensi paket lainnya (nama, versi, ...) dan penyesuaian :)

Pembangunan Paket Masal

Mungkinkah?

Mungkin, asal dependensi pembangunan terpenuhi tanpa interaksi pengembang, tidak ada paket dependensi yang membutuhkan jawaban yes/no/..., tidak ada pustaka dan alat pengembangan yang "aneh", ...

Yang dibutuhkan adalah pola bagaimana paket-paket dibangun dan skrip/program pengotomatisasi masukan-keluaran paket ke/dari pbuilder

Contoh-Contoh

/usr/share/doc/pbuilder/examples/

man 8 pbuilder

Daftar URL

http://kuliaz.org/devel#membangun_paket
http://netfort.gr.jp/~dancer/software/pbuilder-doc/
https://wiki.ubuntu.com/Packaging/Training/Logs/2009-06-25

Pertanyaan?

koreksi? masukan?

Lisensi

Surat elektronik

stwn@kuliax.org
stwn@unsoed.ac.id