

Noprianto

# Optical Character Recognition di Linux

**D**i topik utama kali ini, kita akan membahas seputar Optical Character Recognition di Linux.

Saat ini, banyak pengguna komputer dari berbagai kalangan yang ingin menggunakan Linux. Namun, sebagian dari mereka selama ini telah bekerja dengan aplikasi tertentu di Windows, dan ingin tahu apakah fungsionalitas yang mereka dapatkan saat ini juga tersedia di Linux.

Saat ini, di dunia Linux dan *open source*, terdapat banyak sekali aplikasi yang bisa digunakan. Mulai dari paket *office*, *database*, *server*, sampai pengembangan aplikasi. Serunya, hampir semua aplikasi tersebut bisa didapatkan dengan bebas, tanpa harus membayar biaya lisensi sepeser pun. Kita pun dapat membagi aplikasi yang kita miliki, ke rekan kita yang ingin menggunakannya.

Memang, secara alami ketersediaan suatu software di Linux, apalagi untuk aplikasi yang tidak memiliki basis pengguna yang besar, umumnya tidak terlalu mudah diketahui. Jika kita bicara paket *office*, mudah sekali menemukan bahwa kita memiliki OpenOffice.org, KOffice, Abiword,

Gnumeric dan lain sebagainya. Kita pun tahu bahwa OpenOffice.org misalnya, memiliki kompatibilitas yang sangat baik dengan paket MS Office. Namun, bagaimana kalau kita ingin mencari aplikasi untuk Optical Character Recognition, seperti pada topik utama kita kali ini? Informasi yang kita dapatkan tidaklah selengkap ketika kita mencari paket *office*. Selain masalah basis pengguna yang lebih kecil, para *developer* program OCR juga tidak memasang iklan ataupun mengadakan kegiatan seremonial lainnya.

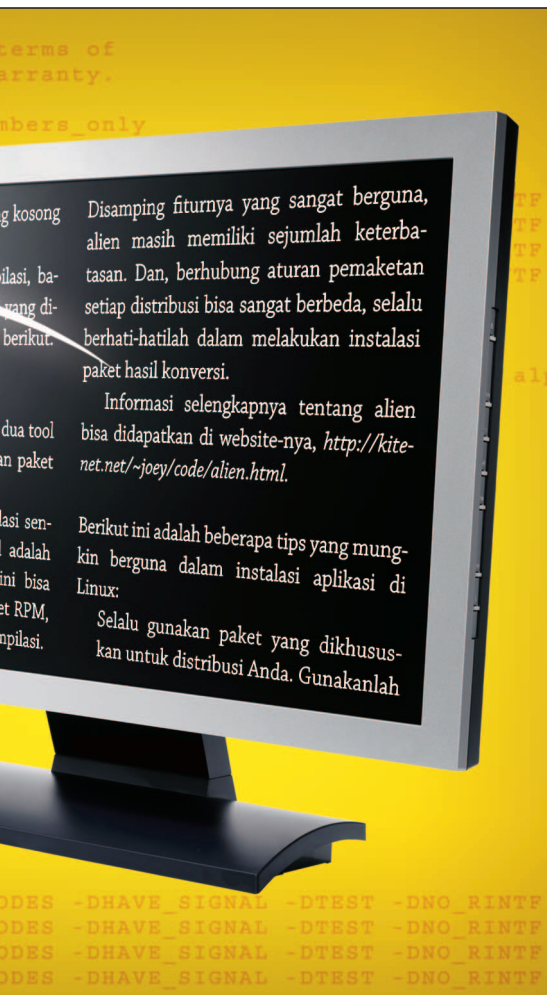
Ketika ditemukan pun, masih mengangkat sifat sebagian besar program *free/open source*. Terkadang kita menemukan bahwa program-program tersebut hanya datang dengan satu atau dua fungsionalitas tertentu, walaupun barangkali fungsionalitas yang diberikan bisa diandalkan. Dalam kasus OCR misalnya, jarang kita temukan software yang komplit, *all in one*, mulai dari *scanning*, melakukan pembacaan teks, sampai melakukan konversi ke format ter-

tentu. Sebagian besar dari mereka memfokuskan diri pada fungsi utama, yaitu melakukan pembacaan teks dari gambar yang telah tersedia.

Untungnya, kita bisa menggabungkan satu per satu fungsionalitas ke dalam satu 'suite aplikasi' buatan kita. Bahkan, dengan kreasi kita sendiri menggunakan berbagai *tool* yang tersedia, kita bisa membangun solusi tertentu, yang di dunia *proprietary* barangkali bisa bernilai ekonomi cukup besar. Jangan lupa, semuanya bisa kita dapatkan tanpa harus membayar sepeser pun biaya lisensi.

Di dalam tulisan ini, kita akan membahas secara sekilas apa yang dimaksud dengan OCR, tool-tool apa saja yang tersedia untuk OCR di Linux, dan beberapa hal terkait seperti dukungan *scanner* di Linux, dan tool untuk bekerja dengan format gambar. Kita akan membahas pula pembuatan solusi OCR custom menggunakan tool yang tersedia di dunia *open source*. Khusus untuk pembahasan tool OCR yang tersedia,





kami akan membahas pula bagaimana tool tersebut dapat membaca teks dari gambar contoh yang kami sediakan. Anda bisa melihat sendiri, seberapa akurat tool tersebut bekerja.

Dengan demikian, apabila Anda membutuhkan fungsionalitas OCR dan menemukan bahwa tool-tool yang tersedia rupanya bisa Anda andalkan, Anda bisa segera membuat rencana migrasi ke Linux :)

## SEKILAS TENTANG OCR

Optical Character Recognition adalah kemampuan/fungsionalitas untuk menerjemahkan gambar berisikan tulisan tangan atau tulisan hasil ketik komputer menjadi teks yang bisa diedit. Umumnya, gambar tersebut didapatkan dari hasil *scan*.

Implementasi OCR sendiri sangatlah luas. Dalam bentuk sederhana, OCR bisa dilakukan diantaranya untuk:

- mengambil teks dari gambar-gambar hasil scan sebuah buku, untuk diformat kembali.

- komputerisasi formulir, di mana formulir yang diisi oleh user di-scan, dan isinya diambil menggunakan program OCR untuk diproses lebih lanjut.
- Penerjemah otomatis, di mana hasil scan langsung diterjemahkan ke bahasa lain, disimpan di sebuah file teks untuk dapat diedit lagi.

Walau kedengarannya sangat menyenangkan, OCR sendiri bukanlah hal yang sederhana, bahkan sangat rumit. Kemampuan program OCR untuk mengenali teks sangat tergantung pada faktor internal seperti algoritma yang digunakan, dan seberapa program tersebut telah 'di-training', ataupun pada faktor eksternal seperti seberapa banyak *noise* yang terdapat pada gambar, yang bisa menghambat pembacaan.

OCR sendiri, di ilmu komputer, umumnya dipelajari pada Artificial Intelligence, Pattern Recognition, atau Machine Vision/Computer Vision.

## BERBAGAI TOOL YANG TERSEDIA

Di bagian ini, kita akan membahas berbagai tool yang tersedia, dengan pembahasan difokuskan pada fungsionalitas utama setiap program, yaitu OCR. Untuk program *front end*, di mana fungsionalitas OCR didapatkan dari berbagai program OCR lain, kita akan membahas program OCR yang digunakan. Untuk tambahan, kita juga akan membahas sekilas fitur program front end tersebut.

Untuk menguji, kita akan menyediakan dua gambar, yaitu *sample\_a.pgm* (raw) dan *sample\_b.pgm* (raw), yang untuk lebih singkatnya kita sebut sebagai *sample\_a* dan *sample\_b*. *Sample\_a* berisikan sejumlah karakter a sampai z, A sampai Z, 0 - 9, dan berbagai tanda baca, disusun sedemikian rupa sehingga cukup longgar untuk dibaca (tidak ada karakter yang berdempetan). Sementara, *sample\_b* berisikan rangkaian kata yang disusun menjadi kalimat. Kedua file tersebut kita buat dengan GIMP. Sebagai catatan, *font* yang dipergunakan adalah *monospace*.

Memang, ini adalah kondisi sangat ideal, yang hampir tidak dapat ditemukan di dunia nyata untuk tugas-tugas yang melibatkan OCR. Kita tahu bahwa sebagian besar kegiatan OCR melibatkan dokumen

hasil scan. Namun, kita menggunakan *sample* tersebut untuk benar-benar menguji kemampuan OCR yang dimiliki oleh program yang dibahas. Soal seberapa mampu mengurangi *noise*, kita tidak membahasnya di dalam tulisan ini.

## GOOCR

GOOCR adalah program OCR yang dikembangkan oleh Joerg Schulenburg, dan tim pengembang lainnya. Program ini dilisensikan di bawah GPL. Sampai tulisan ini dibuat, GOOCR telah mencapai versi 0.44, yang dirilis pada tanggal 1 Maret 2007 yang lalu. GOOCR sendiri telah dikembangkan cukup lama, sejak tahun 2000. Walaupun nama yang digunakan dalam *source code* adalah GOOCR, alamat *website*-nya menggunakan nama JOOCR. Hal ini disebabkan pada waktu registrasi, nama GOOCR telah digunakan oleh pihak lain. Tulisan ini akan tetap menggunakan nama GOOCR.

Untuk pengembangan aplikasi berbasis GOOCR, awalnya proyek ini menyiapkan *libgocr*. Namun, sayangnya hingga saat tulisan ini dibuat, *libgocr* tidak dikembangkan lagi. Dengan demikian, developer yang ingin mengembangkan aplikasi berbasis GOOCR untuk membangun front end yang mudah digunakan, dengan tetap memanggil *binary* GOOCR.

Untuk menggunakan GOOCR, *download*-lah GOOCR dari *website*-nya, <http://jocr.sourceforge.net/>. Setelah itu, lakukanlah langkah-langkah berikut ini:

- Ekstrak arsip *source code*, dan masuklah ke direktori hasil ekstraksi:

```
$ tar zxvf gocr-0.44.tar.gz
```

```
$ cd gocr-0.44
```

- Konfigurasi:

```
$ ./configure
```

- Kompilasi:

```
$ make
```

```
make -C src all
```

```
make[1]: Entering directory `./tmp/utama/gocr/gocr-0.44/src'
```

```
gcc -g -O2 -I../include -DHAVE_CONFIG_H -c -o pgm2asc.o pgm2asc.c
```

```
gcc -g -O2 -I../include -DHAVE_CONFIG_H -c -o box.o box.c
```

```
gcc -g -O2 -I../include -DHAVE_CONFIG_H -c -o database.o database.c
```

```
gcc -g -O2 -I../include -DHAVE_
CONFIG_H -c -o detect.o detect.c
...
...
...
make[1]: Leaving directory `/tmp/
utama/gocr/gocr-0.44/src'
--- for help do: make help ---
--- for documentation do: make
doc ---
--- for examples do: make
examples ---
```

● Instalasi:

```
# make install
```

Setelah instalasi dilakukan, Anda dapat menjalankan binary gocr yang dihasilkan:

```
$ gocr
Optical Character Recognition --
- gocr 0.44
use option -h for help
```

Berikut ini, kita akan mencoba untuk menguji GOOCR menggunakan sample\_a:

```
$ gocr sample_a.pgm
ABCDEF
GHIJKL
MNOPQR
STUVWX
YZ
abcdef
ghijkl
mnopqr
stuvwx
YZ
__J2345
6789-!
#_on
_x() _
+â i?/\'
```

sample\_a.

```
A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z
a b c d e f
g h i j k l
m n o p q r
s t u v w x
y z
0 1 2 3 4 5
6 7 8 9 ~ !
@ # $ % ^
& * ( ) _ -
+ = ? / \ `
' " , . ; :
```

Sementara, berikut ini kita menguji GOOCR dengan sample\_b:

```
$ gocr sample_b.pgm
Linux ada Ia h sistem
aperasi ya ng bebas
digunakan.
Saat in i, terdapat
sangat banyak distri
busi Linux yang bisa
digunakan.
Sebagai contoh:
- Debian GNU/Linux
- OpenSUSE
- Fedora
- Slackware
dan masih
banyak lagi.
Selengkapnya
bisa dibaca di
http://www.distrowatch.com
Se Ia mat menggu na ka n
LINUX!
```

**Format JPEG**

Dari contoh-contoh tersebut, kita menggunakan format PGM. Bagi Anda yang ingin langsung menggunakan format JPEG, gunakanlah program djpeg, yang datang bersama paket libjpeg:

```
# djpeg sample_a.jpg | gocr -
ABCDEF
GHIJKL
MNOPQR
STUVWX
YZ
...
...
...
```

sample\_b.

Linux adalah sistem operasi yang bebas digunakan. Saat ini, terdapat sangat banyak distri busi Linux yang bisa digunakan. Sebagai contoh:  
- Debian GNU/Linux  
- OpenSUSE  
- Fedora  
- Slackware  
dan masih banyak lagi. Selengkapnya bisa dibaca di <http://www.distrowatch.com>  
Selamat menggunakan LINUX!

Untuk format-format gambar lain, lihatlah pada pembahasan tool konversi format gambar.

**Format output**

Bagi Anda yang berkepentingan untuk langsung menyimpan hasil pembacaan ke format HTML, TeX, ataupun XML, opsi -f bisa dipergunakan. Contoh:

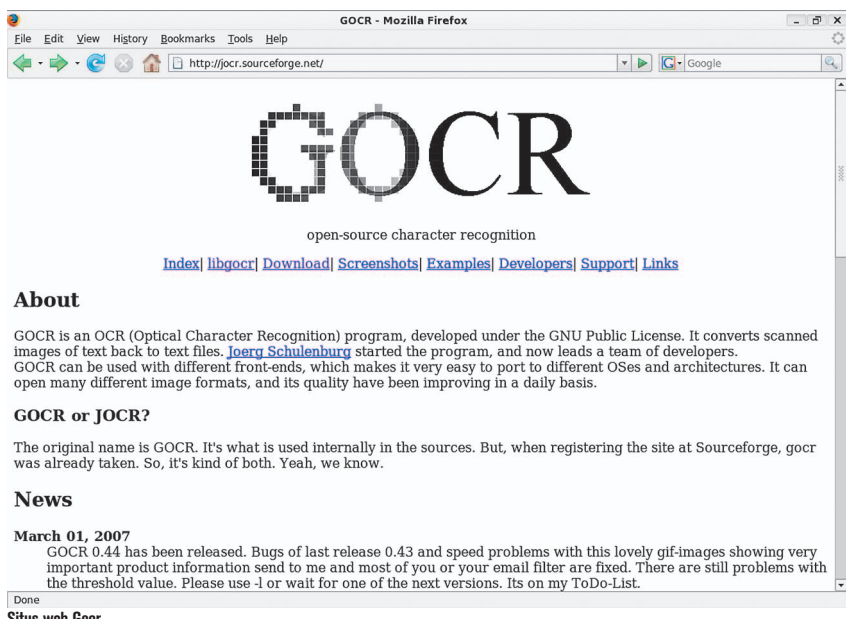
```
$ gocr sample_a.pgm -f HTML
ABCDEF<br />
GHIJKL<br />
MNOPQR<br />
STUVWX<br />
YZ<br />
...
...
...
```

GOOCR sendiri masih dapat diatur agar bekerja lebih baik lagi. Silakan baca manual penggunaan, atau berikan opsi -h untuk melihat opsi lain yang bisa diberikan.

**OCRAD**

OCRAD, yang menyebut dirinya sebagai The GNU OCR adalah program OCR yang dikembangkan oleh Antonio Diaz Diaz. Selain dapat membaca teks dari gambar, program ini juga datang dengan *layout analyzer* yang dapat memisahkan kolom atau blok teks, yang umum ditemukan pada hasil cetakan. Sampai pada saat tulisan ini dibuat, OCRAD telah mencapai versi 0.16.

Untuk pengembangan aplikasi berbasis OCRAD, gunakanlah OCRAD sebagai *back end* dari aplikasi Anda. Penulis memiliki beberapa kesempatan membangun solusi memanfaatkan OCRAD, dan mendapatkan bahwa OCRAD sangat fleksibel untuk



digunakan.

Download-lah OCRAD dari berbagai *mirror* GNU, atau kunjungilah website OCRAD di <http://www.gnu.org/software/ocrad/ocrad.html>. Setelah men-download, lakukanlah langkah-langkah berikut ini:

- Ekstrak arsip source code, dan masuklah ke direktori hasil ekstraksi:

```
$ tar jxvf ocrad-0.16.tar.bz2
$ cd ocrad-0.16
```

- Konfigurasi:

```
$ ./configure
VPATH = .
prefix = /usr/local
exec_prefix = $(prefix)
bindir = $(exec_prefix)/bin
datadir = $(prefix)/share
infodir = $(datadir)/info
mandir = $(datadir)/man
OK. Now you can run make.
```

- Kompilasi:

```
$ make
g++ -Wall -W -O2 -c -o arg_parser.o arg_parser.cc
g++ -Wall -W -O2 -c -o common.o common.cc
g++ -Wall -W -O2 -c -o rational.o rational.cc
g++ -Wall -W -O2 -c -o rectangle.o rectangle.cc
...
...
...
```

```
g++ -Wall -W -O2 -o ocrad
arg_parser.o common.o rational.o
rectangle.o track.o ucs.o
page_image.o page_image_io.o
page_image_layout.o bitmap.o
block.o profile.o feats.o feats_test0.o
feats_test1.o character.o character_r11.o
character_r12.o character_r13.o textline.o
textline_r2.o textblock.o textpage.o
main.o
```

- Instalasi:

```
# make install
```

Setelah instalasi dilakukan, Anda dapat menjalankan binary ocrad yang dihasilkan:

```
$ ocrad -V
GNU Ocrad version 0.16
Copyright (C) 2006 Antonio Diaz Diaz.
This program is free software; you may
redistribute it under the terms of
the GNU General Public License.
This program has absolutely no
warranty.
```

Berikut ini, kita akan mencoba untuk menguji OCRAD menggunakan `sample_a`:

```
$ ocrad sample_a.pgm
A B C D E F
G H I J K L
M N O p O R
s T u v w x
Y z
a b c d e f
```

```
g h i j k L
m n o p q r
s t u v w x
y z
o l z 3 4 s
6 7 8 9 - !
@ # $ % ^
b * ( ) -
+ = ? / \ .
| | | . .
_ . _ .
```

Berikut ini adalah pengujian dengan `sample_b`:

```
$ ocrad sample_b.pgm
Llinux adalah |_|_lem
opera_| yang beba_
dlgunakan
Saal ln1. lerdapal
_angal banyak dl_lrl
bu_| Llinux yang bl_a
dlgunakan
Sebagai conloh
- Deblan GNU/Linux
- OpensUSE
- Fedora
- Slackware
dan ma_lh
banyak lagl
Selengkapnya
bl_a dl baca dl
hwp //www dl_lrowalch com
Selamal menggunakan
LINUX1
```

**Filter**

Program OCRAD dapat pula digunakan untuk mengambil teks berdasarkan filter tertentu. Berikut ini adalah perintah untuk mendapatkan filter yang didukung:

```
$ ocrad --filter=help
Valid filter names are: none
letters letters_only numbers
numbers_only
```

Sebagai contoh, kita bisa meminta OCRAD untuk hanya mengambil angka saja (`numbers_only`):

```
$ ocrad --filter=numbers_only sample_a.pgm
4 8
6 1 7 1
0 0
5 7
2
6
```

```

9 1
0 4
5
2
0 1 2 3 4 5
6 7 8 9

6

1 1 1

```

Tentu saja, filtrasi tetap berdasarkan kemampuan OCRAD untuk membaca teks. Pada contoh filtrasi `numbers_only`, hasil pembacaan tidaklah sesuai harapan, walau semua angka yang sesungguhnya bisa diambil.

## Region

Hal paling menarik dari program ini adalah kemampuan untuk mengambil teks pada *region rectangle* tertentu pada gambar. Dengan demikian, kita bisa membaca bagian tertentu saja dari gambar. Region rectangle kita spesifikasikan dengan koordinat: *left, top, right, bottom*. Untuk mendapatkan koordinat region, kita bisa mempergunakan program GIMP dengan mengarahkan *pointer mouse* ke bagian yang diinginkan. Koordinat akan tampil pada sudut kiri bawah window.

Sebagai contoh, masih menggunakan `sample_a`, kita bisa mengambil region huruf kecil saja, di mana region huruf kecil adalah 200,155,365,300.

Contoh:

```

$ ocrad sample_a.pgm --
crop=200,155,365,300

a b c d e f
g h i j k l
m n o p q r
s t u v w x
y z

```

## LOCR

LOCR adalah program OCR yang mengusung slogan *An Optical Character Recognition Program for Linux*. LOCR dikembangkan oleh Miguel A. Lerma pada musim panas 2000. Versi LOCR pada saat tulisan ini dibuat adalah 0.1.0.

Untuk pengembangan aplikasi berbasis LOCR, gunakanlah LOCR sebagai *back end* dari aplikasi Anda.

Untuk menggunakan LOCR, downloadlah LOCR dari <http://www.math.northwestern.edu/~mlerma/locr/locr-0.1.0.tgz>. Setelah

men-download, lakukanlah langkah-langkah berikut ini:

- Ekstrak arsip source code, dan masuklah ke direktori hasil ekstraksi:

```

$ tar zxvf lochr-0.1.0.tgz
lochr-0.1.0/
lochr-0.1.0/lochr.c
lochr-0.1.0/foo.ps
lochr-0.1.0/foo.txt
lochr-0.1.0/foo.pnm
lochr-0.1.0/foo.dat
lochr-0.1.0/gpl.txt
lochr-0.1.0/README

$ cd lochr-0.1.0

```

- Kompilasi:

```
$ gcc -o lochr lochr.c
```

- Instalasi:

Kopikanlah binary `lochr` yang dihasilkan ke `/usr/bin` atau lokasi lain yang Anda inginkan. Kopikan pula `foo.dat` ke lokasi *share*, misal `/usr/share/lochr/`. Atau, bagi Anda yang tidak ingin mengototir sistem, biarkan saja `lochr` di direktori source code.

Setelah instalasi dilakukan, Anda dapat menjalankan binary `lochr` yang dihasilkan:

```

$ lochr -h
LOCR Version 0.1.0 by Miguel A.
Lerma Copyright (C) 2000
Usage:
lochr [-aghlms] [ -d <datafile>
... ] [ -t <textfile> ] [<infile>
[<outfile>]]

```

Sebagai catatan, LOCR tidak bisa bekerja dengan file PGM, namun dapat bekerja dengan format PBM. Oleh karena itu, kita akan menggunakan versi PBM dari `sample_a` dan `sample_b`. Cara konversi format PGM ke PBM dapat dibaca pada pembahasan tool konversi format gambar.

Berikut ini, kita akan menguji LOCR dengan `sample_a` (format PBM):

```

$ lochr -d foo.dat sample_a.pbm
A 8 C O 5 F
G S I J K L
M N O P Q R
S T C V 4 X
Y Z
ò b c d e f
q h i j k l
M H O Q Q 7
S t u v w X

V `
Ô 1 2 3 4 5

6 7 8 9 - !
@ # $ % ^
b * ( ) -
+ = ? y Ñ *

I II y y
I * I *

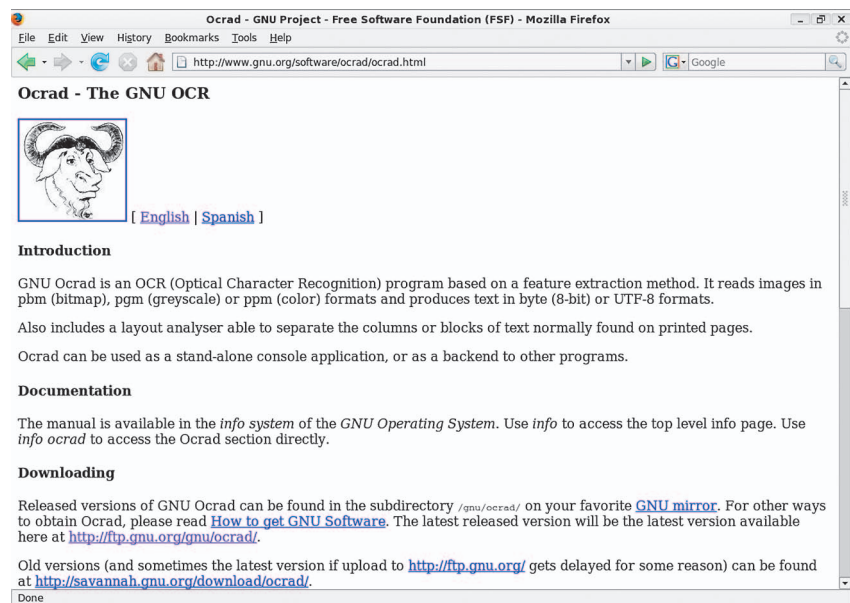
```

Berikut ini adalah pengujian dengan `sample_b` (format PBM):

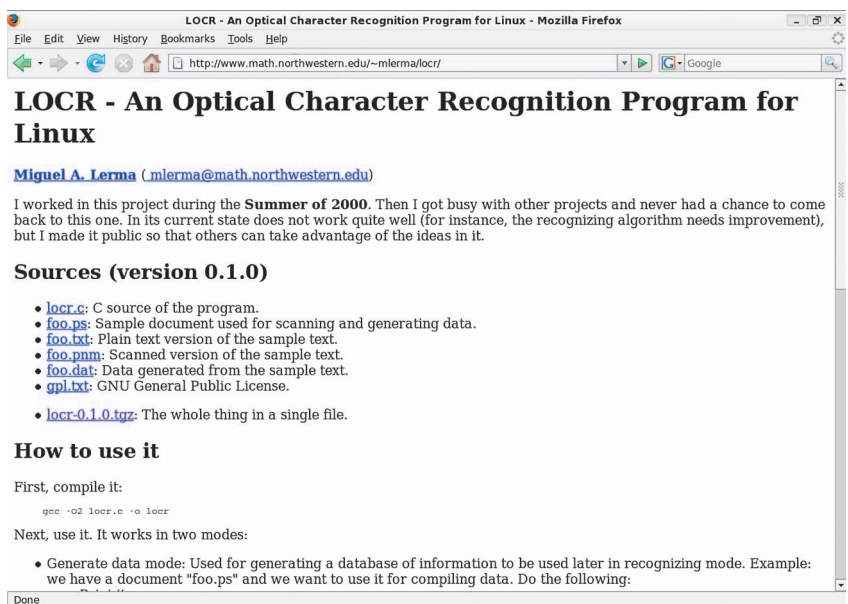
```

$ lochr -d foo.dat sample_b.pbm
L!uux ada Ha 5 s!stem
opecoS! yo uq bebaS

```



Situs web Ocrad.



Situs web locr.

```
d !qu u 6 k6 u .
Sa at ! u ! , te rd6 pot
Sa uQat b0 uy6 k d!Stc!
bu S ! L! u u X ya u q b !Sa
d !qu u a k6 u .
Se bD Qo ! co uto 5 :
^ Oeb!a u GNC/L!uux
~ OpeuSCS5
~ 7edo ra
~ SHack$a re
da u mos!5
ba uya k HoQ!.
SeHeuqk6 puya
b !sa d ! ba ca d !
5kp ://wwj d !StrowOtc 5 . co m

SeHa mat meuqqu uD ka u
LHMCX !
```

## CLARA OCR

Clara OCR adalah program OCR yang dikembangkan oleh Ricardo Ueda Karpiscek. Clara sendiri adalah singkatan dari Cooperative Lightweight chAracter Recognizer. Disebut *cooperative*, karena clara menawarkan *interface web* untuk *training* dan *revision*, sehingga clara dapat memanfaatkan revisi dari pengguna di Internet.

Dibandingkan dengan program lainnya, Clara OCR datang cukup lengkap, dengan interface GUI dan web interface (untuk mengumpulkan revisi). Pada saat tulisan ini dibuat, versi terbaru Clara OCR adalah clara-20031214. Informasi selengkapnya tentang Clara OCR bisa didapatkan di website-nya:

<http://www.geocities.com/claraocr/>.

Untuk menggunakan Clara OCR, download-lah program tersebut dari <http://www.geocities.com/claraocr/clara-20031214.tar.gz>. Setelah men-download, lakukanlah langkah-langkah berikut ini:

- Ekstrak arsip source code, dan masuklah ke direktori hasil ekstraksi:

```
$ tar zxvf clara-20031214.tar.gz
$ cd clara-20031214
```

- Kompilasi:

```
$ make
gcc -I/usr/X11R6/include -g
-Wall -DHAVE_POPEN -DMECHECK -
DFUN_CODES -DHAVE_SIGNAL -DTEST -
DNO_RINTF -c -o clara.o clara.c
...
...
gcc -o clara -g clara.o skel.
o event.o symbol.o pattern.o
pbm2cl.o cml.o welcome.o redraw.
o html.o alphabet.o revision.
o build.o consist.o pgmblock.o
preproc.o obd.o -L/usr/X11R6/lib
-LX11 -lm
```

- Instalasi:

```
# make install
```

Setelah instalasi dilakukan, Anda dapat menjalankan binary clara yang dihasilkan:

```
$ clara -v
this is Clara OCR version 20031214
compiled with HAVE_POPEN
```

```
compiled with MEMCHECK
```

```
compiled with FUN_CODES
```

Untuk menjalankan interface GUI clara, jalankanlah Clara tanpa parameter apapun:

```
$ clara
```

Untuk membuka gambar, lakukanlah langkah-langkah berikut ini:

- Masuklah ke direktori yang berisikan gambar.
- Jalankanlah clara tanpa parameter.
- Akseslah tab Page, dan pilihlah gambar dari list.

Sebagai catatan, bacalah dokumentasi yang datang bersama clara untuk informasi selengkapnya.

## Kooka

Kooka merupakan program *raster image scan* untuk desktop KDE. Berikut ini adalah fitur-fitur Kooka:

### Scan

- Dukungan scan diperoleh dengan pustaka SANE.
- Penanganan opsi scan dengan dialog yang ramah.
- Mendukung *Preview* dan *Final Scan*.
- Pemilihan area scan yang interaktif, dan otomatis dalam panel preview.

### Image storage

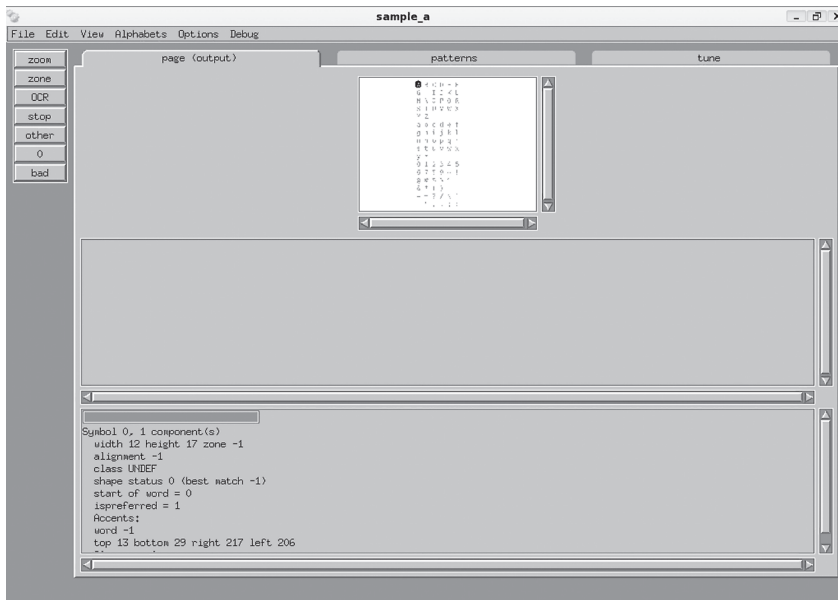
- Kooka akan membantu memilih format gambar yang cocok, dan membuat nama file gambar secara otomatis.
- *Image* disimpan di *image gallery*.

### Image Gallery

- Organisasi image yang mudah digunakan.
- Mendukung *Intelligent zooming*.
- Datang dengan kemampuan manipulasi image sederhana.
- Mendukung *printing* dalam ukuran *original* ataupun *custom*.
- Mendukung *drag and drop* dengan program KDE lain.

## OCR

- Mendukung GOCR.
- Bekerja dengan KADMOS.
- Menggunakan OCRAD sebagai *engine default* karena mendukung layout detection, dan mampu melakukan OCR dengan cukup baik.



Clara OCR.

## KDE Scan service

- libscan menyediakan scan *service* untuk semua aplikasi KDE yang membutuhkan.

Informasi selengkapnya tentang kooka bisa didapatkan di <http://kooka.kde.org/>.

## TOOL KONVERSI FORMAT GAMBAR

- Umumnya, program-program OCR yang dibahas di tulisan ini bekerja dengan format-format gambar berikut ini:
- PBM, merupakan format *Portable Bitmap* (hitam putih; 1 bit per pixel).
- PGM, merupakan format *Portable Graymap (grayscale)*; 8 bit per pixel).
- PPM, merupakan format *Portable Pixmap* (warna; 24 bit per pixel).

Sementara, PNM (*portable anymap*) adalah *generic reference* untuk tiga format tersebut.

Output dari program file untuk sample\_a:

```
$ file sample_a.pgm
sample_a.pgm: Netpbm PGM "rawbits"
image data
```

Lantas, bagaimana kalau sebagian besar dari file yang Anda miliki bukan dalam format tersebut? Salah satu solusinya adalah dengan menggunakan program konverter dari format yang Anda miliki ke salah satu dari format tersebut.

Apabila Anda membangun solusi sen-

diri berdasarkan berbagai program OCR yang ada, Anda selalu bisa memanfaatkan tool-tool konversi yang akan kita bahas, agar program Anda 'terlihat' mendukung banyak format. Pada dasarnya, Anda hanya melakukan konversi memanfaatkan program *converter* yang ada, lalu melempar hasil konversi ke program OCR.

Di bagian ini, kita akan membahas dua *suite* program yang bisa membantu: netpbm dan imagemagick.

## Netpbm

Netpbm adalah paket program untuk pengolahan gambar, sekaligus berfungsi sebagai pustaka pengembangan program. Di dalam paket program ini, terdapat lebih dari 220 program, di mana nama program-program tersebut umumnya mengandung pbm, pgm, ppm, pam atau pnm.

Netpbm sendiri dibangun berdasarkan program pbmplus yang dibangun oleh Jef Poskanzer, yang didistribusikan kali pertama pada 1988, dan di-*maintaince* sampai 1991. Bryan Henderson kemudian *maintaince* paket ini sejak 1991. Selain itu, berbagai developer di seluruh dunia ikut mengembangkan netpbm.

Hampir semua distribusi Linux *general purpose* menyertakan netpbm di dalam media distribusi.

Berikut ini adalah beberapa contoh program yang datang bersama netpbm, yang umum digunakan:

- giftopnm:

```
$ file sample_a.gif
sample_a.gif: GIF image data,
version 89a, 640 x 480
```

```
$ giftopnm sample_a.gif > sample_
a.pgm
giftopnm: Reading Image Sequence 0
```

```
$ file sample_a.pgm
sample_a.pgm: Netpbm PGM
"rawbits" image data
```

- jpegtopnm:

```
$ jpegtopnm sample_a.jpg >
sample_a.pgm
jpegtopnm: WRITING PGM FILE
```

- pngtopnm
- tifftopnm
- bmptopnm
- sgitopnm
- rletopnm
- pnmtjpeg
- pnmtotiff
- pnmtopng
- pnmtorle
- pnmtosgi
- pgmtoipbm
- dan masih banyak lagi

Informasi selengkapnya tentang netpbm bisa didapatkan di website-nya: <http://netpbm.sourceforge.net/>.

## Imagemagick

Imagemagick merupakan kumpulan tool untuk memanipulasi dan menampilkan gambar. Imagemagick dapat melakukan sangat banyak tugas, termasuk:

- *merge* image.
- *transform* image.
- menangkap layar (*screen shot*).
- membuat animasi.
- melakukan konversi format gambar.
- dan lain sebagainya.

Nama	: NOP
Alamat	: JL. RAYA GG. SATU NO. 3
T/TL	: KOTA, TANGGAL
Ket.	: KET 1 KET 2 KET 3

Formulir form1.

ImageMagick dikembangkan oleh ImageMagick Studio LLC

Dalam hal ini, kita akan membahas program *convert*, yang dapat digunakan untuk melakukan konversi berbagai format gambar. Contoh:

```
$ convert sample_a.gif sample_a.pgm
$ convert sample_a.jpg sample_a.pgm
$ convert sample_a.tiff sample_a.pgm
$ convert sample_a.jpg sample_a.png
$ convert sample_a.png sample_a.bmp
$ convert sample_a.bmp sample_a.gif
```

Program *convert*, selain melakukan konversi, juga dapat melakukan tindakan tertentu sambil melakukan konversi, seperti memberikan efek *blur*, meng-*emboss*, memperbaiki image, dan lain sebagainya.

Informasi selengkapnya tentang ImageMagick bisa didapatkan di website-nya: <http://www.imagemagick.org/>.

## DUKUNGAN SCANNER DI LINUX

Seperti kita ketahui, OCR kerap melibatkan proses scan, karena sumber gambar umumnya adalah buku atau material cetak lainnya.

Untuk scan di Linux, kunjungilah situs web SANE project di <http://www.sane-project.org/>, di mana di website tersebut Anda bisa mendapatkan:

- Source code untuk *sane-backends* (*driver scanner, command-line front end, network scanning daemon*, dan dokumentasi API), dan *sane-frontends* (GUI front end, command-line front end, dan lainnya).
- Informasi tentang dukungan merek dan tipe scanner.
- Daftar aplikasi yang bisa digunakan.
- Dokumentasi lengkap.
- Fasilitas pelaporan *bug*.
- dan lain sebagainya.

## SOLUSI CUSTOM MEMANFAATKAN OCR

Bayangkan ketika Anda harus melakukan tugas-tugas berikut ini:

- Memeriksa data yang diisi oleh user, yang diisikan dalam form dengan format baku seperti contoh berikut:

```
Nama : _____
Alamat : _____
TTL : _____
Ket. : _____
```

- Satu per satu formulir tersebut harus Anda baca, dan simpan ke dalam database.
- Memeriksa puluhan sampai ratusan formulir setiap hari, dengan beberapa tenaga kerja

Anda ingin melakukannya secara manual? Bisa-bisa saja tentunya. Namun, apabila Anda memiliki scanner, Anda dapat pula memanfaatkan OCR untuk mengotomatisasi pekerjaan.

Dengan OCR, prosedur kerja Anda menjadi:

- Memastikan formulir di-layout cukup longgar untuk setiap *field* (apabila Anda berhak melakukan ini).
- Memastikan formulir di-scan dengan posisi yang benar, dengan hasil sebaik mungkin.
- Menjalankan *script* yang akan:
  - Membaca setiap *field* pada formulir.
  - Membangun perintah SQL yang diperlukan.
  - Mengeksekusi perintah SQL.

Walaupun cara ini memiliki keterbatasan, diantaranya adalah kualitas OCR (terutama untuk formulir tulisan tangan), cara ini setidaknya bisa dicoba.

Di tulisan ini, kita akan mengabaikan proses scanning. Script yang akan kita bangun mengasumsikan file gambar sudah tersedia.

Untuk OCR, kita akan menyerahkannya kepada program OCRAD. Format formulir yang akan digunakan adalah formulir yang telah disebutkan sebelumnya (memiliki empat *field*: nama, alamat, ttl dan keterangan). Berikut ini adalah source code *nop-ocr-multiple\_field.sh*:

```
#!/bin/sh

#(c) Nop, 2007
# GPL v2
#read form1.pgm

NAMA_REG=235,43,630,80
ALAMAT_REG=235,100,630,200
TTL_REG=235,215,630,255
KET_REG=235,275,630,375

FORM1=form1.pgm
OCRAD=ocrad
```

```
NAMA=`$OCRAD --crop=$NAMA_REG
$FORM1`
ALAMAT=`$OCRAD --crop=$ALAMAT_REG
$FORM1`
TTL=`$OCRAD --crop=$TTL_REG $FORM1`
KET=`$OCRAD --crop=$KET_REG $FORM1`

echo "NAMA : $NAMA"
echo -e "ALAMAT : \n$ALAMAT"
echo "TTL : $TTL"
echo -e "KET : \n$KET"

SQL_CMD="INSERT INTO ms_user (nama,
alamat, ttl, ket) VALUES ('$NAMA', '$
ALAMAT', '$TTL', '$KET');"

echo
echo "SQL Command: "
echo $SQL_CMD
```

Berikanlah hak akses *executable* dengan perintah:

```
$ chmod +x nop-ocr-multiple_field.sh
```

Contoh output (pastikan *form1.pgm* tersedia):

```
./nop-ocr-multiple_field.sh
NAMA : NOP
ALAMAT :
JL. RAYA
GG. SATU
NO. 3
TTL : KOTA, TANGGAL
KET :
KET 1
KET 2
KET 3

SQL Command:
INSERT INTO ms_user (nama, alamat,
ttl, ket) VALUES ('NOP', 'JL. RAYA
GG. SATU NO. 3', 'KOTA, TANGGAL', 'KET
1 KET 2 KET 3');
```

Tentu saja, solusi tersebut bisa Anda kembangkan sesuai keinginan Anda. Termasuk pemeriksaan yang sangat penting, khususnya ketika OCR dilakukan.

Bisa kita lihat, di dunia Linux dan open source terdapat banyak sekali program dan pustaka siap pakai, yang sesungguhnya bisa kita gunakan untuk menyelesaikan berbagai tugas yang kita hadapi. Selama kita mau mencari dan berusaha, maka pastilah ada solusi.

Sampai di sini dulu pembahasan kita. Selamat menggunakan Linux! 🐧