

Menampilkan Kontributor Linux dengan Menarik

Bagian 1 dari 2 Artikel

Paket *source code* kernel datang bersama file CREDITS, yang di dalamnya berisikan nama-nama pembuat Linux beserta keterangannya dalam format yang siap di-parse oleh *script*. Di dalam artikel ini dan sambungannya, kita akan membahas beberapa cara untuk memformat dan menampilkan nama para *hacker* tersebut dalam tampilan yang lebih menarik.

File CREDITS di dalam *tree source code* kernel memang cukup menarik untuk diamati. Dengan membaca file tersebut, kita dapat melihat siapa-siapa saja yang menyumbangkan tenaga dan waktunya untuk menjadikan Linux seperti saat ini. Walau, sayangnya, tidaklah nyaman membaca file tersebut karena format file-nya. Format file CREDITS tersebut memang dirancang agar mudah di-parse dan diformat agar dapat ditampilkan lebih indah.

Berikut ini adalah 20 baris pertama file tersebut:

```
This is at least a partial
credits-file of people that
have
contributed to the Linux
project. It is sorted by
name and
formatted to allow easy
grepping and beautification
by
scripts. The fields are:
name (N), email (E), web-
address
(W), PGP key ID and
fingerprint (P), description
(D), and
snail-mail address (S).
Thanks,

      Linus
-----
```

```
N:Matti Aarnio
E: mea@nic.funet.fi
D: Alpha systems hacking, IPv6
  and other network related
  stuff
D: One of assisting postmasters
  for vger.kernel.org's lists
S: (ask for current address)
S: Finland

N: Dragos Acostachioaie
E: dragos@iname.com
...
...
...
...
```

Di tulisan kali ini, kita akan mempergunakan bahasa PHP untuk dapat menampilkan isi file tersebut dalam format plain untuk digunakan di *command line*, dalam format HTML untuk dapat ditampilkan di *web browser*, serta ditampilkan dalam aplikasi GUI sederhana.

Distro yang digunakan dalam tulisan kali ini adalah SUSE Pro 9.1. PHP yang digunakan adalah PHP 4.3.4 dan PHP-GTK (<http://gtk.php.net>, digunakan pada edisi berikutnya) yang digunakan adalah PHP-GTK 1.0.1. Apache yang digunakan adalah Apache HTTPD 2.0.49. File CREDITS yang digunakan adalah kredit dari source code kernel 2.6.11.4 milik distro SUSE Pro 9.3.

Format file CREDITS

Melihat penggalan isi file CREDITS sebelumnya, kita dapat membagi isi file ke dalam dua bagian besar: isi kredit dan komentar.

Untuk isi kredit, yaitu keterangan kontributor Linux, dibagi pula dalam kumpulan keterangan yang mana setiap kumpulan dipisahkan oleh sebuah baris kosong. Berikut ini adalah ringkasan keterangan per kontributor dan keterangannya.

```
N: Nama
E: Email
W: Website
P: PGP
D: Deskripsi tugas
S: Alamat surat
```

Masing-masing keterangan yang dapat berulang (penulis mengasumsikan semua dapat berulang, kecuali nama, dengan asumsi kontributor tersebut hanya memiliki satu nama saja) dapat dituliskan dengan kode yang sama sehingga keterangan seorang kontributor dapat berupa:

```
N: Nama
E: Email 1
E: Email 2
W: Website 1
W: Website 2
P: PGP
D: Deskripsi tugas 1
D: Deskripsi tugas 2
D: Deskripsi tugas 3
S: Alamat surat
```

Salah satu atau beberapa keterangan (misal e-mail, web, pgp, alamat surat) bisa saja tidak ditampilkan (diasumsikan semua boleh tidak ditampilkan kecuali nama) sehingga keterangan dapat berupa:

N: Nama
D: Deskripsi tugas

Setiap baris dari kumpulan keterangan tersebut dipisahkan oleh sebuah tanda titik dua (:) diikuti oleh sebuah spasi. Dan sekali lagi, seperti disebutkan sebelumnya, keterangan per kontributor selesai apabila baris kosong ditemukan. Selanjutnya, giliran kontributor lainnya.

Dengan kondisi demikian, berikut adalah salah satu algoritma yang dapat digunakan untuk melakukan parsing file tersebut:

- Baca setiap baris file CREDITS tersebut.
- Untuk setiap baris yang dibaca, periksa apakah baris kosong atau bukan.
 - Apabila baris kosong, maka saatnya untuk me-reset semua keterangan (dipersiapkan untuk kontributor berikutnya). Di sini bisa pula dilakukan aksi seperti menampilkan kode HTML.
 - Apabila bukan baris kosong, maka periksa dua karakter pertamanya apakah merupakan salah satu kode dari NEWPDS diikuti titik dua (:) tersebut.
 - ✓ Apabila bukan, maka dianggap sebagai komentar. Perulangan diteruskan.
 - ✓ Apabila ya, maka: Untuk setiap baris yang ditemukan, pisahkanlah kode dan keterangan dengan salah satu kode dari NEWPDS diikuti titik dua (:) tersebut. Variabel baru bisa diassign untuk keterangan tertentu yang diinginkan.
- Selesai.

Selanjutnya, kita bisa menuangkan algoritma tersebut ke dalam bahasa PHP dengan tujuan untuk menghitung berapa nama yang ditemukan, serta mencetak nama kontributor tersebut ke layar (stdout). Berikut ini adalah source code-nya:

```
<?
$file = fopen("CREDITS","r");
$tokens = array("N:", "E:", "W:",
```

```
"P:", "D:", "S:");
$count = 0;

while (!feof($file))
{
    $buff = fgets($file);
    if (strlen($buff) == 1)
    {
        $n = "";
        $e = "";
        $w = "";
        $p = "";
        $d = "";
        $s = "";
    } else
    {
        $tmp = split(":", $buff);
        if (!in_array(substr(
            $buff, 0, 2), $tokens))
            continue;
        switch ($tmp[0])
        {
            case "N":
                $n = $tmp[1];
                $n = stripslashes(
                    ereg_replace("\n", "", $n));
                $count++;
                echo "\nfound hacker
                $count ($n)";
                break;
            case "E":
                $e .= $tmp[1] . "\n";
                break;
            case "W":
                $w .= $tmp[1] . "\n";
                break;
            case "P":
```

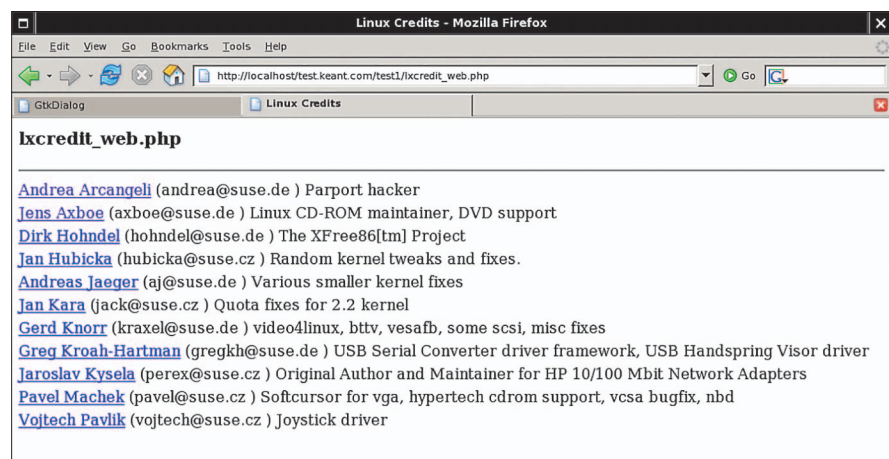
```
$p .= $tmp[1] . "\n";
                break;
            case "D":
                $d .= $tmp[1] . "\n";
                break;
            case "S":
                $s .= $tmp[1] . "\n";
                break;
        }
    }
}

fclose($file);

echo "\nDone, $count hackers
found.\n";
?>
```

Keterangan source code:

- Pertama-tama, kita membaca membuka file CREDITS untuk dibaca dengan fungsi fopen().
- Setelah itu, kita mendefinisikan kode-kode yang digunakan dalam file ini dan menyimpannya dalam array \$tokens.
- Kita mengatur variabel \$count menjadi 0. Variabel ini akan digunakan sebagai counter.
- Dalam perulangan sampai akhir file, kita akan membaca baris-per baris dan menerapkan algoritma sebelumnya.
- Kita menggunakan split() untuk memecah baris dengan kode titik dua diikuti spasi.
- Kita mempergunakan ereg_replace() untuk mengganti karakter \n dengan "".
- Karena kode-kode EWPDS boleh beru-



Linux-credit versi web.

lang, maka kita kita selalu menambahkan keterangan EWPDS tersebut (misal: \$d := \$tmp[1] . "\n");).

- Setelah selesai, jumlah hacker yang ditemukan akan ditampilkan.

- File ditutup dengan fungsi fclose().

Berikan nama lxcredit_basic.php dan jalankanlah dengan perintah berikut:

```
$ php lxcredit_basic.php
```

Berikut ini adalah keluaran dari perintah tersebut:

```
found hacker 1 (Matti Aarnio)
found hacker 2 (Dragos
Acostachioaie)
found hacker 3 (Monalisa
Agrawal)
found hacker 4 (Dave Airlie)
found hacker 5 (Tigran A.
Aivazian)
found hacker 6 (Werner
Almesberger)
...
...
found hacker 465 (Leonard N.
Zubkoff)
found hacker 466 (Alessandro
Zummo)
found hacker 467 (Marc Zyngier)
...
Done, 467 hackers found.
```

lxcredit_plain

Kita akan melengkapi contoh sebelumnya dengan menambahkan e-mail dan D pertama yang ditemukan (deskripsi bisa dalam banyak baris, kita akan mengambil yang pertama saja). Pada dasarnya, contoh ini sama saja dengan contoh sebelumnya. Hanya, di sini, kita menambahkan pemformatan sederhana, yang akan berguna untuk contoh-contoh berikutnya.

Berikut ini adalah isi dari source code lxcredit_plain.php:

```
<?
//(c) Nop, 8 Mei 2005, 12:20

$file = fopen("CREDITS", "r");
$tokens = array("N:", "E:", "W:",
"P:", "D:", "S:");
$count = 0;
while (!feof($file))
{
    $buff = fgets($file);
```

```
    if (strlen($buff) == 1)
    {
        if ($n)
        {
            $hacker_str = sprintf(
                "%3d %s (%s) %s",
                $count, $n, $e, $d1);
            echo $hacker_str .
                "\n";
        }
        $n = "";
        $e = "";
        $w = "";
        $p = "";
        $d1 = "";
        $d = "";
        $s = "";
        $hacker_str = "";
    } else
    {
        $tmp = split(":", $buff);
        if (!in_array(substr(
            $buff, 0, 2), $tokens))
            continue;
        switch ($tmp[0])
        {
            case "N":
                $n = $tmp[1];
                $n = stripslashes
                    (ereg_replace("\n",
                        "", $n));
                $count++;
                break;
            case "E":
                $e = $tmp[1];
                $e = ereg_replace(
                    "\n", "", $e);
                $e .= " ";
                break;
            case "W":
                $w .= $tmp[1] . "\n";
                break;
            case "P":
                $p .= $tmp[1] . "\n";
                break;
            case "D":
                if (!$d1)
                {
                    $d1 = $tmp[1];
                    $d1 = ereg_
                        replace("\n", "",
                            $d1);
                }
                $d .= $tmp[1] . "\n";
```

```
                break;
            case "S":
                $s .= $tmp[1] . "\n";
                break;
        }
    }
}
fclose($file);
echo "\nDone, $count hackers
found.\n";
?>
```

Keterangan source code:

- Kita menambahkan variabel \$d1 yang berfungsi untuk menampung deskripsi pertama yang ditemukan. Pada saat memproses kode D, kita memeriksa apakah \$d1 telah berisi atau tidak. Apabila tidak, kita mengisikan D yang ditemukan.
- Selanjutnya, pada blok untuk me-reset semua variabel, kita menampilkan satu baris informasi ke layar untuk setiap kontributor yang ditemukan (variabel \$hacker_str). Tentu saja, ini dilakukan sebelum kita me-reset semua variabel (termasuk \$d1 dan \$hacker_str). Apabila diinginkan, Anda bisa menambahkan aksi apa saja pada saat kita menampilkan informasi kontributor tersebut.
- Kita menggunakan sprintf() untuk memformat informasi sebelum ditampilkan.

Berikan perintah berikut ini untuk menjalankan lxcredit_plain.php:

```
$ php lxcredit_plain.php
```

Berikut adalah contoh keluaran dari perintah tersebut:

```
1 Matti Aarnio (mea@nic.funet.
fi ) Alpha systems hacking, IPv6
and other network related stuff
2 Dragos Acostachioaie
(dragos@iname.com ) /proc/
sysvipc
3 Monalisa Agrawal
(magrawal@nortelnetworks.com )
Basic Interphase 5575 driver
with UBR and ABR support.
...
...
459 Eric Youngdale
(eric@andante.org ) General
```

```
kernel hacker
460 Niibe Yutaka (gniibe@mri.
co.jp ) PLIP driver
461 James R. Van Zandt
(jrv@vanzandt.mv.com ) Author
and maintainer of the Double
Talk speech synthesizer driver
462 Orest Zborowski
(orestz@eskimo.com ) XFree86 and
kernel development
463 Richard Zidlicky (rz@linux-
m68k.org, rdzidlic@geocities.
com ) Q40 port - see arch/m68k/
q40/README
464 Werner Zimmermann (Werner.
Zimmermann@fht-esslingen.de )
CDROM driver "aztcd" (Aztech/
Okano/Orchid/Wearnes)
465 Leonard N. Zubkoff ( )
BusLogic SCSI driver
466 Alessandro Zummo
(azummo@ita.flashnet.it )
CMI8330 support is sb_card.c
467 Marc Zyngier (maz@wild-wind.
fr.eu.org ) MD driver

Done, 467 hackers found.
```

lxcredit_web.php

Setelah selesai dengan contoh sebelumnya, kita akan melanjutkan ke aplikasi web. Di contoh berikut, kita menambahkan fasilitas pencarian, sehingga user dapat mencari berdasarkan nama, e-mail, web, pgp, deskripsi dan alamat surat. Pencarian dapat pula dikombinasikan untuk hasil yang lebih akurat. Secara *default*, apabila user tidak memasukkan kriteria pencarian apapun, maka keseluruhan hacker akan ditampilkan.

Untuk setiap hasil pencarian, kita akan menambahkan *hyperlink* pada nama sehingga pengguna dapat melihat informasi lain yang lebih detil (sama seperti pada contoh sebelumnya, pada awalnya kita hanya menampilkan informasi nama, e-mail dan deskripsi pertama).

Kita akan membuat aplikasi ini *se-com-pact* mungkin sehingga tidak ada file HTML yang digunakan. Cukup dengan satu file php saja, yaitu `lxcredit_web.php`. Di dalam file ini, kita membuat form dan processor-nya sekaligus.

Pada hakikatnya, selain hal-hal yang berurusan dengan HTML, kode-kode yang

digunakan untuk melakukan iterasi setiap kontributor sebenarnya sama saja. Yang berbeda adalah ketika harus melakukan pencarian. Penjelasan lebih lengkap dapat dilihat pada penjelasan source code setelah ini:

isi file `lxcredit_web.php`:

```
<?
//(c) Nop, 8 Mei 2005, 14:35

echo "
<html>
<head>
<title>Linux Credits</title>
</head>
<body>
<h3>lxcredit_web.php</h3>
<hr noshade>
";

if (!$_POST && $_REQUEST
['task'] == "")
{
echo "

<form action='{$_SERVER
['PHP_SELF']}' method=
post>
<table border=0>
<tr><td>Name</td><td>
<input type=text name=
'N'></td></tr>
<tr><td>Email</td><td>
<input type=text name=
'E'></td></tr>
<tr><td>Web</td><td>
<input type=text name=
'W'></td></tr>
<tr><td>PGP</td><td>
<input type=text name=
'P'></td></tr>
<tr><td>Desc</td><td>
<input type=text name=
'D'></td></tr>
<tr><td>Snail Mail</td>
<td><input type=text
name='S'></td></tr>
<tr><td>&nbsp;</td>
<td><input type=submit
name='search' value=
'search'></td></tr>
</table>
</form>
```

```
";
die();
};

$pn = trim($_POST['N']);
$pe = trim($_POST['E']);
$pw = trim($_POST['W']);
$pp = trim($_POST['P']);
$pd = trim($_POST['D']);
$ps = trim($_POST['S']);

$file = fopen("CREDITS","r");
$tokens = array("N:", "E:", "W:",
"P:", "D:", "S:");
$count = 0;
while (!feof($file))
{
$buff = fgets($file);
if (strlen($buff) == 1)
{
if ($n)
{
if ($_REQUEST['task']
== "")
{
if ($pn != "") if
(!ereg($pn, $n))
{$dl = ""; continue; }
if ($pe != "") if
(!ereg($pe, $e))
{$dl = ""; continue; }
if ($pw != "") if
(!ereg($pw, $w))
{$dl = ""; continue; }
if ($pp != "") if
(!ereg($pp, $p))
{$dl = ""; continue; }
if ($pd != "") if
(!ereg($pd, $d))
{$dl = ""; continue; }
if ($ps != "") if
(!ereg($ps, $s))
{$dl = ""; continue; }
$hacker_str =
sprintf("<a href='
{$_SERVER['PHP_
SELF']}'?task=
detail&hid=$count'
>%s</a> (%s) %s",
$n, $e, $dl);
echo $hacker_str .
"<br>";
}
}
```

```

        else
        if ($_REQUEST['task']
        == "detail")
        {
            $s_count = $_
            REQUEST['hid'];
            if ($s_count ==
            $count)
            {
                echo "Detail
for $n&nbsp;&nbsp;&nbsp;<a href=
javascript:history.go(-1)>back</
a><br>";
                echo "<table
border=1>";
                echo "<tr><td>
Name</td><td>$n</td></tr>";
                echo "<tr><td>
Email</td><td>". nl2br($e) .
"</td></tr>";
                echo "<tr><td>
Web]</td><td>". nl2br($w) .
"</td></tr>";
                echo "<tr><td>
PGP</td><td>". nl2br($p) .
"</td></tr>";
                echo "<tr><td>
Desc</td><td>". nl2br($d) .
"</td></tr>";
                echo "<tr><td>
Snail Mail</td><td>". nl2br($s)
. "</td></tr>";
                echo "</table>";
            }
        }
        $n = "";
        $e = "";
        $w = "";
        $p = "";
        $d1= "";
        $d = "";
        $s = "";
        $hacker_str = "";
    } else
    {
        $tmp = split(":", $buff);
        if (!in_array(substr
($buff,0,2),$tokens))
        continue;
        switch ($tmp[0])
        {
            case "N":

```

```

            $n = $tmp[1];
            $n = stripslashes(
ereg_replace("\n",
""),$n));
            $count++;
            break;
            case "E":
                $e = $tmp[1];
                $e = ereg_replace("\
n","", $e);
                $e .= " ";
                break;
            case "W":
                $w .= $tmp[1] . "\n";
                break;
            case "P":
                $p .= $tmp[1] . "\n";
                break;
            case "D":
                if (!$d1)
                {
                    $d1 = $tmp[1];
                    $d1 = ereg_replace
("\n","", $d1);
                }
                $d .= $tmp[1] . "\n";
                break;
            case "S":
                $s .= $tmp[1] . "\n";
                break;
        }
    }
    fclose($file);

    echo "
</body>
</html>
";
?>

```


Penjelasan source code:

- Ketika user menjalankan aplikasi ini, maka dilakukan pemeriksaan apakah ada superglobal \$_POST yang aktif. Apabila superglobal \$_POST tidak ada dan \$_REQUEST['task'] bernilai "", maka form pencarian akan ditampilkan, yang segera diikuti dengan terminasi script menggunakan fungsi die(). Guna dari pemeriksaan ini adalah semata-mata agar dapat menampilkan form pencarian ketika user

memanggil aplikasi ini tanpa ada permintaan khusus apapun (seperti melihat detail kontributor dan lain sebagainya).

- Untuk mempersingkat penulisan, kita menulis ulang \$_POST['N'] menjadi pn, dengan p untuk \$_POST dan n untuk N, dan seterusnya.
- Rutin iterasi untuk setiap kontributor sama saja dengan contoh sebelumnya. Bedanya, pada saat blok untuk me-reset variabel.
- Kita tidak memiliki fasilitas pencarian pada SQL dengan keyword like %, oleh karena itu, kita harus mensimulasikannya. Pencarian tersebut bisa dilakukan dengan algoritma berikut:
 - Untuk setiap kriteria apabila kriteria diisi, dan pencarian dalam satu field tidak ketemu, maka kita harus loncat ke record berikutnya. Dalam konteks aplikasi ini, field adalah setiap NEWPDS dan record adalah kontributor.
- Untuk setiap hasil yang ditemukan, kita menampilkan informasi singkat mirip dengan contoh sebelumnya, hanya, kita meniadakan nomor dan memberikan hyperlink. Berikut ini adalah format hyperlink yang akan kita bangun: lxcredit_web.php?task=detail&hid=<NOMOR_URUT>.
- Ketika user memberikan alamat seperti lxcredit_web.php?task=detail&hid=6, maka informasi detil tentang Werner Almesberger (di file yang penulis gunakan, nomor 6 adalah Werner Almesberger) akan ditampilkan dalam sebuah tabel.

Untuk menjalankan aplikasi ini, Anda akan membutuhkan web server yang telah dikonfigurasi untuk menjalankan PHP, dan sebuah web browser.

Sampai di sini dulu pembahasan kita di edisi ini. Kita telah membahas bagaimana menampilkan dalam interface teks dan web. Tentu saja Anda bisa membangun versi ncurses baik dengan libncurses ataupun memanfaatkan program dialog. Di edisi depan, kita akan melihat satu contoh lagi untuk melihat bagaimana menampilkan nama-nama hacker tersebut dalam interface GUI, dengan memanfaatkan PHP-GTK.  **Noprianto** (noprianto@infolinux.co.id)

Menampilkan Kontributor Linux dengan Menarik

Bagian 2 dari 2 Artikel

Di edisi lalu, kita telah melihat bagaimana menyajikan kredit kontributor Linux di *interface* teks dan web. Di edisi ini, kita akan melihat bagaimana menyajikan dalam GUI.

Di artikel ini, kita akan menambahkan `lxcredit_gtk.php`, script PHP untuk menampilkan kredit Linux di dalam GUI yang sederhana memanfaatkan PHP-GTK.

lxcredit_gtk.php

Aplikasi ini memiliki satu window, yang terdiri dari enam entry tempat memasukkan kriteria pencarian pada bagian atas layar. Pada bagian bawah layar, aplikasi ini memiliki sebuah *list-box* yang berguna untuk menampilkan hasil pencarian. Dalam *listbox* tersebut, kita hanya akan menampilkan tiga kolom data yaitu nama, email dan deskripsi pertama. Untuk versi GTK, kita tidak menampilkan informasi detail seperti pada aplikasi web. Berikut ini adalah source code aplikasi:

```
<?php
//(c) nop, 8 Mei 2005, 19:45
dl("php_gtk.so");

function close()
{
    gtk::main_quit();
}

function search_action($button,
    $entry_name, $entry_email,
    $entry_web, $entry_pgp, $entry_
    desc, $entry_snail, $clist_data,
    $statusbar)
{
    $statusbar -> push(1,"Please
    wait...");

    $pn = $entry_name -> get_
    text();
```

```
$pe = $entry_email -> get_
text();
$pw = $entry_web -> get_
text();
$pp = $entry_pgp -> get_
text();
$pd = $entry_desc -> get_
text();
$ps = $entry_snail -> get_
text();

$clist_data -> clear();

$file = fopen("CREDITS","r");
$tokens = array("N:", "E:", "W:",
    "P:", "D:", "S:");
$count = 0;
while (!feof($file))
{
    $buff = fgets($file);
    if (strlen($buff) == 1)
    {
        if ($n)
        {
            if ($pn != "") if
            (!ereg($pn, $n)) {$d1
            = ""; continue; }
            if ($pe != "") if
            (!ereg($pe, $e)) {$d1
            = ""; continue; }
            if ($pw != "") if
            (!ereg($pw, $w)) {$d1
            = ""; continue; }
            if ($pp != "") if
            (!ereg($pp, $p)) {$d1
            = ""; continue; }
            if ($pd != "") if
            (!ereg($pd, $d)) {$d1
```

```
= ""; continue; }
            if ($ps != "") if
            (!ereg($ps, $s)) {$d1
            = ""; continue; }

            $temp = array($n, $e,
            $d1);
            $clist_data ->
            append($temp);
        }
        $n = "";
        $e = "";
        $w = "";
        $p = "";
        $d1 = "";
        $d = "";
        $s = "";
        $hacker_str = "";

    } else
    {
        $tmp = split(":", $buff);
        if (!in_array(substr(
            $buff, 0, 2), $tokens))
            continue;
        switch ($tmp[0])
        {
            case "N":
                $n = $tmp[1];
                $n = stripslashes
                (ereg_replace("\
                n", "", $n));
                $count++;
                break;
            case "E":
                $e = $tmp[1];
                $e = ereg_replace("\
                n", "", $e);
```

```

        $e .= " ";
        break;
        case "W":
            $w .= $tmp[1] . "\n";
            break;
        case "P":
            $p .= $tmp[1] . "\n";
            break;
        case "D":
            if (!$d1)
            {
                $d1 = $tmp[1];
                $d1 = ereg_
                    replace("\
n", "", $d1);
            }
            $d .= $tmp[1] . "\n";
            break;
        case "S":
            $s .= $tmp[1] . "\n";
            break;
    }

}

fclose($file);
$statusbar -> push(1,"Ready");
}

$window = &new GtkWindow();
$window->set_title("Linux
Credits");
$window->set_usize(640, 480);
$window->set_position(GTK_WIN_
POS_CENTER);
$window->set_policy(false,
false, false);
$window->connect("destroy",
"close");

$main_table = &new
GtkTable(16,12);
$window->add($main_table);

$entry_name = &new GtkEntry();
$label_name = &new
GtkLabel("name");
$entry_email = &new GtkEntry();
$label_email = &new
GtkLabel("email");
$entry_web = &new GtkEntry();
$label_web = &new
GtkLabel("web");
$entry_pgp = &new GtkEntry();

```

```

$label_pgp = &new
GtkLabel("pgp");
$entry_desc = &new GtkEntry();
$label_desc = &new
GtkLabel("desc");
$entry_snail = &new GtkEntry();
$label_snail = &new
GtkLabel("snail");
$button_search = &new
GtkButton("search");

$clist_data = &new GtkCList(3,
array("Name","Email","Desc1"));
$clist_data -> set_column_
width(0,150);
$clist_data -> set_column_
width(1,150);

$statusbar = &new
GtkStatusBar();
$statusbar -> push(1,"Ready");

$scrolled_win = &new
GtkScrolledWindow();
$scrolled_win -> set_policy(GTK_
POLICY_AUTOMATIC, GTK_POLICY_
AUTOMATIC);
$scrolled_win -> add($clist_
data);

$button_search -> connect(
"clicked","search_action",
$entry_name, $entry_email,
$entry_web, $entry_pgp, $entry_
desc, $entry_snail, $clist_data,
$statusbar);

$main_table -> attach($label_
name, 0,1,0,1);
$main_table -> attach($entry_
name, 1,2,0,1);
$main_table -> attach($label_
email, 2,3,0,1);
$main_table -> attach($entry_
email, 3,4,0,1);
$main_table -> attach($label_
web, 4,5,0,1);
$main_table -> attach($entry_
web, 5,6,0,1);
$main_table -> attach($label_
pgp, 0,1,1,2);
$main_table -> attach($entry_
pgp, 1,2,1,2);
$main_table -> attach($label_

```

```

desc, 2,3,1,2);
$main_table -> attach($entry_
desc, 3,4,1,2);
$main_table -> attach($label_
snail, 4,5,1,2);
$main_table -> attach($entry_
snail, 5,6,1,2);
$main_table -> attach($button_
search, 5,6,2,3);
$main_table -> attach($scrolled_
win, 0,16,6,11);
$main_table ->
attach($statusbar, 0,16,11,12);

$window->show_all();
gtk::main();
?>

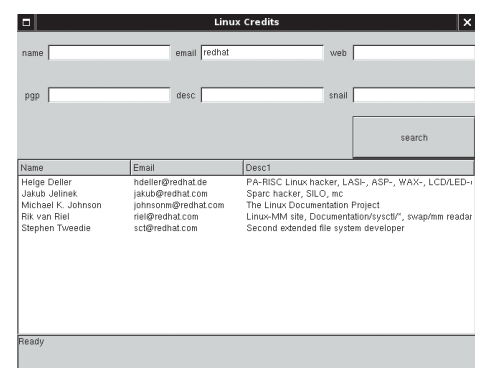
```

Penjelasan source code:

- Kita memiliki dua fungsi: close() dan search_action(). Fungsi pertama berfungsi untuk keluar dari perulangan utama GTK dan fungsi kedua untuk melakukan pencarian. Isi fungsi ini kurang lebih sama dengan rutin pencarian pada contoh-contoh sebelumnya. Bedanya, dalam fungsi pencarian ini, untuk mempersingkat, kita juga langsung mengatur hasil pencarian ke dalam listbox di bagian bawah layar.
- Program ini menggunakan GtkTable untuk mengatur layout.
- Untuk mendapatkan PHP GTK terbaru, kunjungilah gtk.php.net.

Selesai sudah pembahasan mengenai kredit pembuat Linux. Anda bisa juga membuat versi lain, seperti versi flash yang menarik. Sebuah kontribusi di dunia open source. 🙏

Noprianto (noprianto@infolinux.co.id)



Linux-credit versi GUI.