

Membangun Search Engine dengan namazu

Saat ini, kita mengenal berbagai jenis *search engine*. Bagaimana kalau kita juga mengimplementasikan sebuah search engine sendiri untuk kebutuhan lokal ataupun intranet? Tidak perlu memiliki kemampuan pemrograman. Kita gunakan saja namazu. Dengan namazu, search engine menjadi begitu mudah dan cepat untuk diimplementasikan.

Manusia pada dasarnya adalah makhluk tukang simpan. Apa saja disimpan. Mulai dari baju-baju yang bahkan sudah tidak terpakai sampai dengan e-mail-e-mail sampah yang pasti tidak akan pernah dibaca lagi. Di dunia komputer, tak heran apabila kapasitas harddisk tidak pernah cukup. Pada saat membeli harddisk berukuran 80 GB mungkin akan terasa besar, namun seiring dengan penggunaannya, besar kemungkinan untuk menambah harddisk.

Masalahnya, selain keterbatasan ruang atau tempat, manusia juga suka sekali lupa. Tidak semua memang, namun jumlah manusia pelupa kini semakin banyak. Apabila budaya menumpuk barang dikombinasikan dengan budaya lupa, maka banyak hal akan menjadi semakin menarik, termasuk tulisan ini.

Pembahasan di atas hanyalah salah satu alasan dari pentingnya fasilitas pencari. Berhubungan dengan masalah fisik saja, mencari barang yang kecil dan telah dimiliki

selama tahunan akan susah. Begitupun dengan topik yang disimpan di antara ribuan file yang juga telah disimpan tahunan.

Hal lain yang mendukung perlunya pencarian adalah waktu. Tidak mungkin bagi kita untuk selalu mencari sesuatu secara manual. Dalam kehidupan sehari-hari, kita mungkin akan membuat daftar berisikan barang yang kita miliki dan tempat penyimpanannya, sehingga, apabila ingin mencari sesuatu, tinggal melihat ke daftar. Tidak perlu mengobrak-abrik seisi rumah (dengan asumsi, kita selalu meletakkan barang di tempat penyimpanan yang seharusnya, tidak mengambil dari gudang namun malas menyimpan kembali ke gudang sehingga dibiarkan begitu saja di kamar kerja misalnya).

Apabila dikembangkan lebih luas, salah satu pentingnya pencarian adalah masalah lokasi.

Terutama di penggunaan komputer. Ada kalanya, kita perlu mencari informasi yang tidak disimpan di lokal, namun tersimpan

di remote entah di komputer mana. Apabila ada fasilitas pencari lintas lokasi, maka masalah keterbatasan lokasi bisa dipecahkan.

Seiring dengan perkembangan teknologi, *search engine* menjadi populer. Dengan adanya *search engine*, kita bisa mencari tanpa terbatas lokasi, waktu, dan lain sebagainya. Masukkan saja apa yang ingin dicari, dan *search engine* akan menampilkan informasi yang didapatkannya.

Untuk menurunkan kebiasaan search engine di internet, Google bahkan merilis pencarian versi desktop sehingga user dapat semakin mudah mencari apa yang tersimpan di dalam harddisk komputer. Pencariannya begitu hebat sehingga berbagai dokumen juga dapat ditelusuri dengan mudah (Google dipenuhi dengan begitu banyak tenaga kerja superunik dan hebat, dalam budaya kerja yang 70% menyenangkan dan mengembirakan, dan sisanya adalah darah, keringat dan air mata. Menurut salah satu pejabat Google).

```

# Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help

2632/19791 - user/share/doc/packages/emacs/INSTALL.intrd (text/plain)
2633/19791 - user/share/doc/packages/emacs/PLAIN_IDS (text/plain)
2634/19791 - user/share/doc/packages/emacs/README (text/plain)
2635/19791 - user/share/doc/packages/emacs/TECHINTRODUCTION (text/plain)
2636/19791 - user/share/doc/packages/emacs/c11/scripts/assign_segment (text/plain)
2637/19791 - user/share/doc/packages/emacs/c11/scripts/create_bbr_feature (text/plain)
2638/19791 - user/share/doc/packages/emacs/c11/scripts/create_bbr_segment (text/plain)
2639/19791 - user/share/doc/packages/emacs/c11/scripts/create_driewlink (text/plain)
2640/19791 - user/share/doc/packages/emacs/c11/scripts/create_filesystem (text/plain)
2641/19791 - user/share/doc/packages/emacs/c11/scripts/create_lun_container (text/plain)
2642/19791 - user/share/doc/packages/emacs/c11/scripts/create_lun_region (text/plain)
2643/19791 - user/share/doc/packages/emacs/c11/scripts/create_region (text/plain)
2644/19791 - user/share/doc/packages/emacs/c11/scripts/create_segment (text/plain)
2645/19791 - user/share/doc/packages/emacs/c11/scripts/create_snapshot (text/plain)
2646/19791 - user/share/doc/packages/emacs/c11/scripts/create_volume (text/plain)
2647/19791 - user/share/doc/packages/emacs/c11/scripts/delete (text/plain)
2648/19791 - user/share/doc/packages/emacs/c11/scripts/list_containers (text/plain)
2649/19791 - user/share/doc/packages/emacs/c11/scripts/list_objects (text/plain)
2650/19791 - user/share/doc/packages/emacs/c11/scripts/list_volumes (text/plain)
2651/19791 - user/share/doc/packages/emacs/c11/scripts/remove_filesystem (text/plain)
2652/19791 - user/share/doc/packages/emacs/c11/scripts/remove_segment (text/plain)
2653/19791 - user/share/doc/packages/emacs/c11/scripts/reset_snapshot (text/plain)
2654/19791 - user/share/doc/packages/emacs/c11/scripts/rollback_snapshot (text/plain)
2655/19798 - user/share/doc/packages/expect/expect.1.gz (application/pdf) skipped.
2656/19798 - user/share/doc/packages/expect/CDPIING (text/plain)
2657/19798 - user/share/doc/packages/expect/ChangeLog (text/plain)
2658/19798 - user/share/doc/packages/expect/README (text/plain)
2659/19798 - user/share/doc/packages/expect/examples/elements.c (text/plain)
2660/19798 - user/share/doc/packages/expect/examples/outline.c (text/plain)
2661/19798 - user/share/doc/packages/expect/expectfaq.html (text/html)
2662/19798 - user/share/doc/packages/expect/expectref.html (text/html)
2663/19798 - user/share/doc/packages/expect/style.css (text/plain)
2664/19798 - user/share/doc/packages/expect/expect.html (text/html)
2665/19798 - user/share/doc/packages/expect/ChangeLog (text/plain)
2666/19798 - user/share/doc/packages/expect/FAQ (message/rfc822)
2667/19798 - user/share/doc/packages/expect/HISTORY (text/plain)
# Shell No. 2 - Konsole
# Shell No. 2 - Konsole

```

Pembuatan index

The screenshot shows a terminal window titled "Shell No. 2 - Konsole". The user has executed a command to list the contents of the directory `/usr/share/doc/packages/zsh/html`. The output shows a list of files with their sizes and permissions. Below the directory listing, there is a summary of the file size and index statistics.

```

19419/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_37.html [text/html]
19420/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_38.html [text/html]
19421/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_39.html [text/html]
19422/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_40.html [text/html]
19423/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_41.html [text/html]
19424/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_42.html [text/html]
19425/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_43.html [text/html]
19426/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_44.html [text/html]
19427/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_45.html [text/html]
19428/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_46.html [text/html]
19429/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_47.html [text/html]
19430/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_48.html [text/html]
19431/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_49.html [text/html]
19432/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_50.html [text/html]
19433/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_51.html [text/html]
19434/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_52.html [text/html]
19435/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_53.html [text/html]
19436/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_54.html [text/html]
19437/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_55.html [text/html]
19438/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_56.html [text/html]
19439/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_57.html [text/html]
19440/19440 -rusr/share/doc/packages/zsh/html/doc/zsh_58.html [text/html]
Writing index files...
Done!
Date:          Sat Dec 18 14:56:33 2004
Added Documents: 19,440
Size (bytes):   185,389,002
Total Documents: 19,440
Added Requests: 1,478,031
Total Requests: 1,478,031
Mkafk!:        nofile,kakasi -leuc -oeuc -u
Time (sec):    2.161
File/Sec:      3.77
System:        linux
Perl:          5.008003
Namszu:        2.0.12

```

The terminal prompt is `root@linux:/home/MTM/MTM/home/temp/namszu`.

Pembuatan index (selesai)

Sebenarnya, jauh sebelum Google merilis pencarian versi desktop, kita juga telah mengenal fasilitas pembuatan search engine yang dapat diterapkan secara lokal ataupun intranet, juga Internet. Walau, barang kali lebih banyak yang menerapkan untuk lokal ataupun intranet saja. Berbagai tool bisa kita gunakan. Dan, salah satunya adalah dengan mempergunakan namazu.

Namazu adalah search engine *full-text* yang ditujukan untuk kemudahan penggunaan. Namazu dapat dipergunakan untuk sistem pencarian mulai dari harddisk lokal sampai web berukuran medium. Pada awalnya, namazu hanya dikembangkan untuk bahasa Jepang, namun untuk saat ini, juga telah dapat menangani bahasa Inggris dan bahasa latin lainnya. Namazu pada awalnya dikembangkan oleh Satoru Takabayashi dan saat ini telah dikembangkan oleh namazu project (www.namazu.org). Sebagai catatan soal nama, dalam bahasa Inggris, kata jepang namazu berarti catfish.

Indexing

Umumnya, sebelum pencarian dapat dilakukan, kita akan melakukan proses indexing terlebih dahulu, termasuk penggunaan namazu. Contoh yang paling baik adalah dalam kehidupan sehari-hari. Katakanlah kita memiliki banyak barang yang tersimpan di dalam rumah. Terkadang, untuk mencari sebuah obeng saja, kita mungkin harus menelusuri gudang, kamar kerja, garasi dan lain sebagainya. Untuk mempermudah pencarian, sebuah index pun dibuat. Proses pembuatan index disebut dengan indexing. Dalam melakukan indexing inventori

peralatan di rumah tersebut, kita melakukan pencatatan barang-barang kunci dan lokasinya. Sebagai contoh:

| Barang | Lokasi |
|-----------------|--------|
| Obeng | Gudang |
| Kantong Plastik | Dapur |
| Colokan T | Gudang |
| ... | ... |

Daftar tersebut kita gunakan sebagai index untuk mempermudah pencarian. Apabila ingin mencari obeng, maka kita bisa menuju ke gudang. Hanya satu masalah seperti yang telah disebutkan, kita mungkin lupa untuk meletakkan kembali obeng ke gudang setelah digunakan. Dalam hal ini, index harus diperbaiki secara berkala.

Di komputer pun sama. Kita umumnya menjalankan program peng-index yang akan mencatat berbagai hal di dalam sistem sesuai kemampuannya. Tentu saja index tersebut tidak sekedar pasangan kata kunci dan lokasi, melainkan harusnya dalam format yang telah dioptimasi untuk pencarian. Sebagai contoh adalah berbagai format binary yang di dalamnya berisikan tree.

Bisa kita bayangkan betapa pentingnya index. Misal, kita memiliki 100 buah file teks. Kita ingin mencari satu kata, misalnya Linux ke dalam kumpulan file tersebut. Kita sebenarnya bisa memanfaatkan program find, grep, cat, dan lain sebagainya untuk mencari. Jadi, tidak mempergunakan search engine. Tapi, bayangkan waktu dan usaha yang diperlukan. Apabila mempergunakan sistem pencarian berbasis index yang sederhana saja, tentunya waktu yang diperlukan akan lebih sedikit karena pencari hanya per-

lu mencari ke daftar dan segera menemukan file mana yang diperlukan.

Masalah klasik dari index seperti manusia yang suka mengembalikan barang ke tempat semula juga terjadi di komputer. Tentu saja, pengguna file tidak harus selalu memindahkan file setelah digunakan. Namun, bisa saja file telah dihapus atau diubah dan kata kunci menjadi tidak relevan lagi.

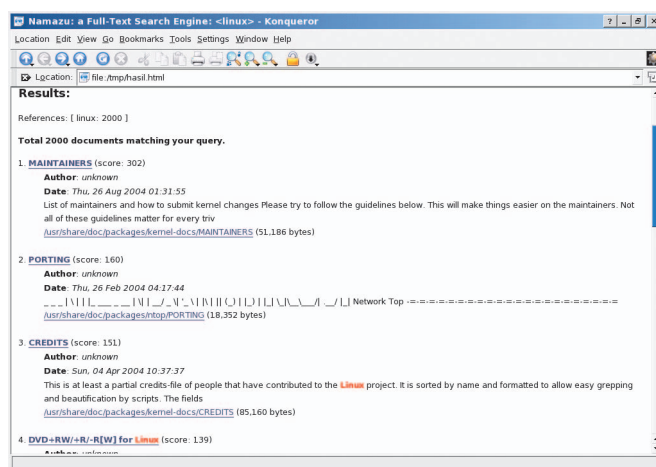
Google juga mengalami masalah yang sama, namun menanganinya dengan pintar. Ada kalanya suatu website telah berubah, namun Google tetap menyimpannya. Google akan menyimpan ini sebagai cache dan pengguna masih bisa mendapatkan informasi yang diinginkan walaupun sumber informasi sudah tidak menampilkannya lagi.

Indexing adalah hal yang penting, dan juga digunakan oleh namazu.

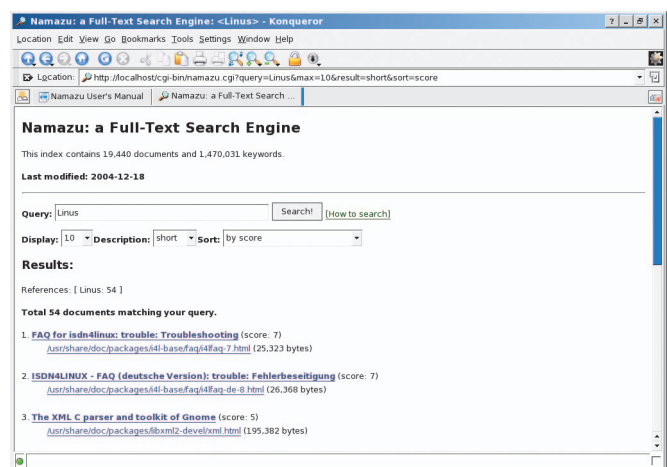
Namazu dan komponen namazu

Namazu datang dengan seperangkat tool dan komponen terpisah yang dapat dipergunakan sesuai dengan kebutuhan. Namun, pada umumnya, kita bisa membedakan tiga komponen utama namazu:

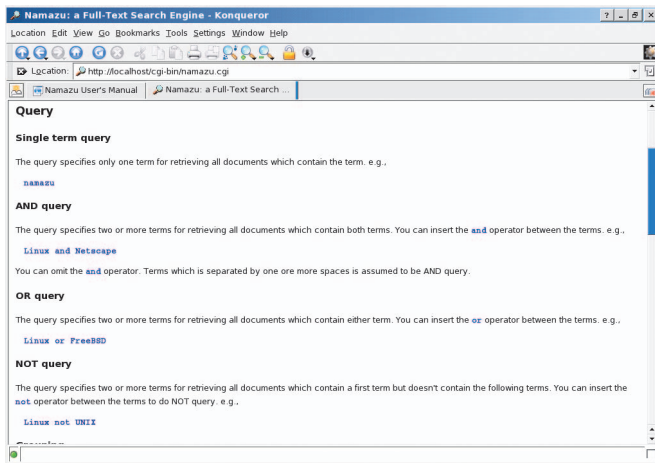
- mknmz, adalah sebuah program Perl yang berguna untuk membuat index sebelum pencarian dilakukan.
- namazu, adalah sebuah program C yang berguna untuk melakukan pencarian secara *command line*, berdasarkan index yang dibuat.
- namazu.cgi, adalah sebuah program C untuk keperluan CGI, yang akan melakukan pencarian berdasarkan index yang dibuat.



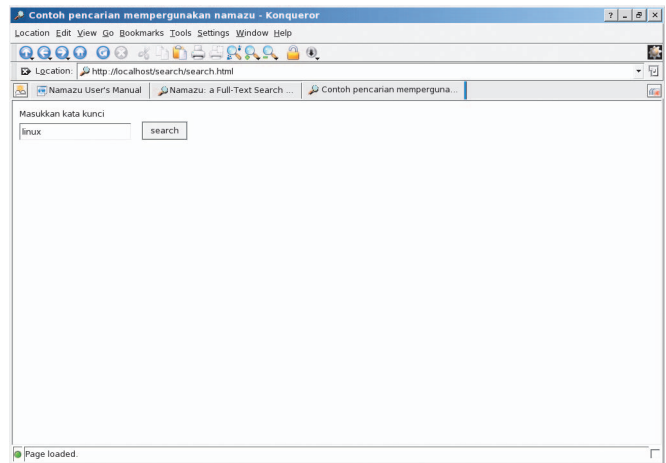
HTML yang dihasilkan namazu.



Penggunaan namazu.cgi.



Bantuan pencarian.



Form html untuk pencarian.

Selain datang dengan program-program utama, namazu juga datang dengan berbagai tool (akan dibahas belakangan) dan filter. Namazu dibangun dengan modular dan mempergunakan filter-filter untuk menangani indexing dokumen. Berikut ini adalah filter standar yang datang dengan namazu:

- `bzip2.pl`, untuk menangani file `bzip2` (membutuhkan program `bzip2`).
- `compress.pl`, untuk menangani file hasil kompresi program `compress`.
- `deb.pl`, untuk menangani paket `debian` (membutuhkan program `dpkg`).
- `dvi.pl`, untuk menangani file `dvi` (membutuhkan `dvi2tty`).
- `excel.pl`, untuk menangani format MS Excel (membutuhkan `xlhtml`).
- `gzip.pl`, untuk menangani file `gzip` (membutuhkan program `gzip` atau modul Perl `Compress::Zlib`).
- `hhtml.pl`, untuk menangani file `hhtml`.
- `hnf.pl`, untuk menangani file Hyper NIKKI System.
- `html.pl`, untuk menangani file `html`.
- `mailnews.pl`, untuk menangani Mail/News.
- `man.pl`, untuk menangani halaman manual.
- `mhonarc.pl`, untuk menangani file Mho-Arc (www.mhonarc.org).
- `mword.pl`, untuk menangani format MS Word (membutuhkan `wvWare`).
- `pdf.pl`, untuk menangani format PDF (membutuhkan `pdftotext`, bagian dari `xpdf`).
- `postscript.pl`, untuk menangani file postscript (membutuhkan `ps2ascii`).
- `powerpoint.pl`, untuk menangani format

MS Powerpoint (membutuhkan `ppt-HTML`, bagian dari `xlhtml`).

- `rfc.pl`, untuk menangani file RFC.
- `rpm.pl`, untuk menangani paket RPM (membutuhkan program `rpm`).
- `taro.pl`, untuk menangani format `ichitaro`, pengolah kata untuk bahasa Jepang.
- `tex.pl`, untuk menangani format `tex` (membutuhkan `detex`).

Program namazu yang dikompilasi untuk Windows memiliki beberapa filter khusus seperti berikut:

- `ichitaro456.pl`, untuk menangani format `ichitaro`, pengolah kata untuk bahasa Jepang (membutuhkan `JSTXT`. tool untuk MSDOS).
- `oleexcel.pl`, untuk menangani format MS Excel (membutuhkan Excel).
- `olemsword.pl`, untuk menangani format MS Word (membutuhkan Word).
- `olepowerpoint.pl`, untuk menangani format MS powerpoint (membutuhkan PowerPoint).
- `oletaro.pl`, untuk menangani format `ichitaro`.
- `olertf.pl`, untuk menangani format `rtf` (membutuhkan Word).

Untuk bekerja dengan namazu, seperti telah disinggung, kita perlu membuat index terlebih dahulu dengan program `mknmz` (yang bekerja dengan filter-filter yang baru saja dibahas). Setelah itu, kita bisa mempergunakan program `namazu` atau `namazu.cgi` untuk pencarian.

Sebelum kita melanjutkan lebih jauh, kita perlu memahami apa yang bisa dan

tidak bisa dilakukan oleh namazu, sebagai referensi agar Anda bisa menyesuaikan kebutuhan Anda. Namazu dapat dimanfaatkan untuk hal-hal berikut ini:

- Mencari file yang tersimpan di komputer.
- Mencari kata yang tersimpan dalam file.
- Mencari dengan berbagai kriteria dan dapat bekerja dengan *regular expression*.
- Dapat mencari berdasarkan command line dan web.

Namazu tidak dapat dimanfaatkan untuk:

- Mencari file di komputer lain.
- Pencarian web tidak bisa diandalkan untuk server yang sangat sibuk.

Harap diperhatikan bahwa penggunaan komputer yang cukup cepat, dengan RAM yang cukup besar dan harddisk yang cukup cepat akan mampu meningkatkan kinerja namazu.

Pembuatan index

Untuk membuat index, kita akan mempergunakan program `mknmz`. Berikut ini adalah cara penggunaannya:

```
mknmz [options] <target>...
```

Untuk `<target>`, kita memberikan direktori yang ingin di-index. Berikut ini adalah beberapa opsi yang mungkin berguna:

- `-a`, menargetkan semua file (umumnya diset).
- `-t`, menspesifikan tipe media.
- `-e`, tidak memasukkan pencarian file HTML yang berisikan `<meta name="ROBOTS" content="NOINDEX">`.
- `-U`, tidak mengencode URI.

- -X, tidak membuat summary pada heading.
- -C, menampilkan konfigurasi.
- -f, mempergunakan file konfigurasi lain.
- -q, melakukan indexing dengan tidak banyak menampilkan output.
- -V, bertindak verbose.

Berikut ini adalah contoh penggunaan mknmz untuk mengamati konfigurasi aktif:

```
$ mknmz -C

Loaded rcfile: /etc/namazu/
mknmzrc
System: linux
Namazu: 2.0.12
Perl: 5.008003
NKF: module_nkf
KAKASI: module_kakasi -ieuc -
oeuc -w
ChaSen: module_chasen -j -F '%m '
Wakati: module_kakasi -ieuc
-oeuc -w
```

```
Lang_Msg: en_US.UTF-8
Lang: en_US.UTF-8
Coding System: euc
CONFDIR: /etc/namazu
LIBDIR: /usr/share/namazu/pl
FILTERDIR: /usr/share/namazu/
filter
TEMPLATEDIR: /usr/share/namazu/
template
Supported media types:
application/postscript
application/x-bzip2
application/x-compress
application/x-gzip
application/x-rpm
message/news
message/rfc822
text/hnf
text/html
text/html; x-type=mhonarc
text/plain
text/plain; x-type=rfc
text/x-hdml
text/x-roff
```

Dengan mengamati apa yang tertera pada konfigurasi aktif ini, kita bisa menentukan paling tidak media tipe mana yang dapat menjadi target.

Pada contoh berikut, penulis melakukan indexing pada /usr/share/doc/packages dan mknmz mendapatkan sekitar hampir 20.000 file yang diindex. Keluarannya sangatlah panjang. Berikut ini baris-baris terakhir perintah mknmz di komputer penulis:

```
$ mknmz -a /usr/share/doc/
packages

...
19436/19440 - /usr/share/doc/
packages/zvbi/ChangeLog [text/
plain]
19437/19440 - /usr/share/doc/
packages/zvbi/INSTALL [text/
plain]
19438/19440 - /usr/share/doc/
packages/zvbi/NEWS [text/plain]
19439/19440 - /usr/share/doc/
packages/zvbi/README [text/
```

WORKSHOP SEHARI

UNTUK ANDA SEORANG.....
Eksekutif TI, Programmer, Administrator
Sistem Linux, atau Mahasiswa.

Meningkatkan Keamanan Kernel Linux dgn LIDS "Linux Intrusion Detection System"

Persyaratan

- Pernah menggunakan Linux
- Pernah melakukan kompilasi kernel (Tidak harus mahir)

Tempat & waktu

Depok: Sabtu, 9 April 2005
Jakarta: Sabtu, 16 April 2005
Bekasi: Sabtu, 2 & 30 April 2005
Jam: 08.30 s.d. 15.30 WIB.

Materi

- ☐ Pengantar Keamanan (Security) Sistem
- ☐ Pengantar Kernel Linux
- ☐ Kompilasi Kernel



PostgreSQL Database Server Open Source Yang Handal

Persyaratan

- Memahami konsep dasar database relasional, termasuk memahami SQL (Structured Query Language)
- Peserta Juga diharapkan pernah menggunakan Linux

Tempat & waktu

Depok: Jum'at, 8 April 2005
Jakarta: Jum'at, 15 April 2005
Bekasi: Jum'at, 29 April 2005
Jam : 08.30 s.d. 15.30 WIB.

Materi

- ☐ Pengantar & Instalasi PostgreSQL
- ☐ Administrasi User & Database
- ☐ Membuat Rule, Trigger, & Membuat Store Procedure dengan PL/PGSQL
- ☐ Membuat Aplikasi WEB dgn PHP yang Mengakses Database PostgreSQL
- ☐ Backup & Restore, Replikasi Database
- ☐ Menggunakan Schema

FASILITAS

- ☺ Ruang kuliah full AC
- ☺ 1 peserta 1 Komputer
- ☺ Modul workshop
- ☺ CD Software
- ☺ Block notes
- ☺ Sertifikat
- ☺ Disket
- ☺ Makan siang
- ☺ Coffee break



LEMBAGA PENDIDIKAN KOMPUTER NURUL FIKRI

- Jl. Margonda Raya No. 522 - Depok 16424 ☎ (021) 7874223-24, 77206991
- Jl. Mampang Prapatan X/4 Jakarta 12790 ☎ (021) 7947115, 7975235
- Jl. A. Yani - Sentra Niaga B.I/12, Bekasi ☎ (021) 78853537

LPKNF <http://www.nurulfikri.com> | info@nurulfikri.com

```
plain]
19440/19440 - /usr/share/doc/
packages/zvbi/TOD0 [text/plain]
Writing index files...
[Base]
Date: Sat Dec 18
14:56:33 2004
Added Documents: 19,440
Size (bytes): 185,389,802
Total Documents: 19,440
Added Keywords: 1,470,031
Total Keywords: 1,470,031
Wakati: module_
kakasi -ieuc -oeuc -w
Time (sec): 5,161
File/Sec: 3.77
System: linux
Perl: 5.008003
Namazu: 2.0.12
```

Bisa kita lihat bahwa pembuatan index tersebut menambahkan *keyword* pencarian sebanyak 1.470.031 buah. Harap diperhatikan juga, bahwa indexing sebagian besar file teks tersebut memakan waktu sekitar 5000

detik (kurang lebih 1,5 jam; penulis membuatnya di hari libur :) di mesin celeron di atas 2 GHz dan RAM 256. Cukup memakan waktu. Indexing ratusan MB file RPM/GZIP/BZIP2 mungkin akan lebih lama lagi.

Di direktori aktif tempat menjalankan mknmz, akan terdapat file dengan nama file diawali dengan NMZ*. Bagi Anda yang tertarik, dapat melihat tabel 1 tentang kegunaan masing-masing file berawalan NMZ. Bagi Anda yang sekedar ingin mengimplemantasikan search engine, Anda bisa melanjutkan ke pembahasan berikutnya.

Pencarian berbasis command line

Setelah kita memiliki index, maka kita dapat memulai pencarian dengan pencari berbasis command line: namazu. Berikut ini adalah cara penggunaan namazu:

```
namazu [options] <query>
[index]...
```

Berikut ini adalah beberapa opsi yang mungkin berguna:

- -n, jumlah dokumen yang ditampilkan.

- -w, nomor dokumen yang pertama ditampilkan.
- -l, menampilkan hasil pencarian dengan menampilkan format.
- -s, menampilkan hasil pencarian singkat.
- --late, mengurutkan dokumen dalam late order.
- --early, mengurutkan dokumen dalam early order.
- --sort-METHOD, mengurutkan dokumen dengan metode tertentu.
- --ascending, mengurutkan dokumen dalam modus ascending (default).
- --all, menampilkan semua hasil pencarian.
- -c, menampilkan jumlah hasil pencarian.
- -h, menampilkan hasil pencarian dalam format HTML.
- -o, menyimpan hasil pencarian dalam file tertentu.
- -q, tidak menampilkan informasi ekstra.

Berikut ini adalah contoh penggunaan namazu:

```
$ namazu -s -n 10 -w 0 linux
/home/nop/temp/namazu/
Results:
References: [ linux: 2000 ]
Total 2000 documents matching
your query.

1. MAINTAINERS (score: 302)
/usr/share/doc/packages/kernel-
docs/MAINTAINERS (51,186 bytes)

2. PORTING (score: 160)
/usr/share/doc/packages/ntop/
PORTING (18,352 bytes)

3. CREDITS (score: 151)
/usr/share/doc/packages/kernel-
docs/CREDITS (85,160 bytes)

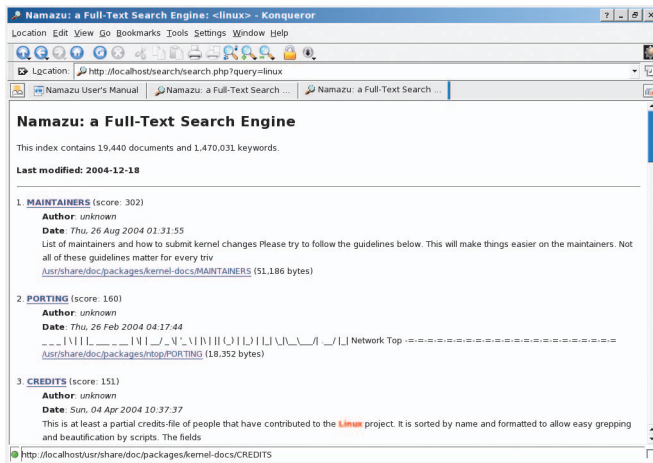
4. DVD+RW/+R/-R[W] for Linux
(score: 139)
/usr/share/doc/packages/dvd+rw-
tools/index.html (59,652 bytes)

5. Changelog (score: 136)
/usr/share/doc/packages/cdrecord
/Changelog (206,483 bytes)

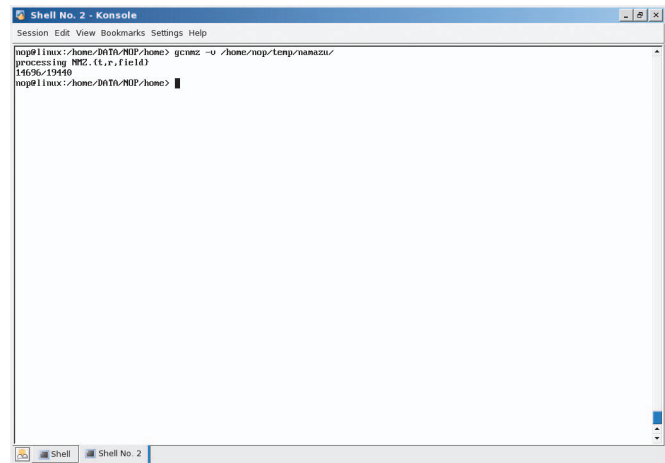
6. HISTORY (score: 132)
/usr/share/doc/packages/util-
linux/HISTORY (54,742 bytes)
```

Tabel 1. Kegunaan masing-masing file berawalan NMZ.

| File | Kegunaan |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NMZ.i | Index untuk pencarian kata (inverted file) |
| NMZ.ii | Index untuk mencari ke dalam NMZ.i |
| NMZ.w | Daftar kata |
| NMZ.wi | index untuk mencari ke dalam NMZ.w |
| NMZ.r | Daftar file yang terdaftar dalam index |
| NMZ.p | Index untuk pencarian berbasis frase |
| NMZ.pi | Index untuk NMZ.p |
| NMZ.t | Mencatat informasi tentang timestamp dan dokumen yang telah dihapus. Memiliki masalah dengan tahun 2038 (timestamp berbasiskan epoch dihitung setiap detik sejak 1 Januari 1970). |
| NMZ.field.{subject,from,date,message-id,...} | Mencatat informasi field |
| NMZ.field.{subject,from,date,message-id,...}.i | Index untuk mencari ke dalam NMZ.field.{subject,from,date,message-id,...} |
| NMZ.access | Konfigurasi akses control |
| NMZ.status | Mencatat berbagai status untuk kebutuhan update index |
| NMZ.result | Menentukan hasil pencarian |
| NMZ.head | Header hasil pencarian |
| NMZ.footer | Footer hasil pencarian |
| NMZ.body | Deskripsi query |
| NMZ.tips | Tips pencarian dokumen |
| NMZ.log | Logfile untuk upate index |
| NMZ.lock | Lockfile untuk mencegah pencarian |
| NMZ.lock2 | Lockfile untuk mencegah pembuatan index dalam waktu bersamaan |
| NMZ.slog | Logfile untuk kata kunci yang dicari |



Kombinasi PHP dan namazu.



Penggunaan garbage cleaner.

7. FAQ (score: 126)

/usr/share/doc/packages/glibc-devel/FAQ (82,646 bytes)

8. Linux Client with Linux Kernel-Space NFS server (score: 118)

/usr/share/doc/packages/nfs-utils/node24.html (15,226 bytes)

9. Linux Client with Linux Kernel-Space NFS server (score: 118)

/usr/share/doc/packages/nfs-utils/node26.html (15,230 bytes)

10. README (score: 102)

/usr/share/doc/packages/lilo/README (132,914 bytes)

Current List: 1 - 10

docs/CREDITS

/usr/share/doc/packages/dvd+rw-tools/index.html

/usr/share/doc/packages/cdrecord/Changelog

/usr/share/doc/packages/util-linux/HISTORY

/usr/share/doc/packages/glibc-devel/FAQ

/usr/share/doc/packages/nfs-utils/node24.html

/usr/share/doc/packages/nfs-utils/node26.html

/usr/share/doc/packages/lilo/README

Atau, meminta namazu untuk menampilkan hasil pencarian dalam format HTML dan menyimpannya ke dalam file hasil.html.

```
$ namazu -h -n 10 -w 0 linux
/home/nop/temp/namazu/ > hasil.html
```

Anda juga bisa meminta namazu untuk hanya menampilkan jumlah hasil pencarian seperti cara berikut ini:

```
$ namazu -c linux /home/nop/
temp/namazu/
2000
```

query lanjutan

Menggunakan namazu, sebenarnya Anda tidak harus mencari sesuatu sebagai suatu kata tunggal. Anda bisa mempergunakan query yang lebih kompleks seperti:

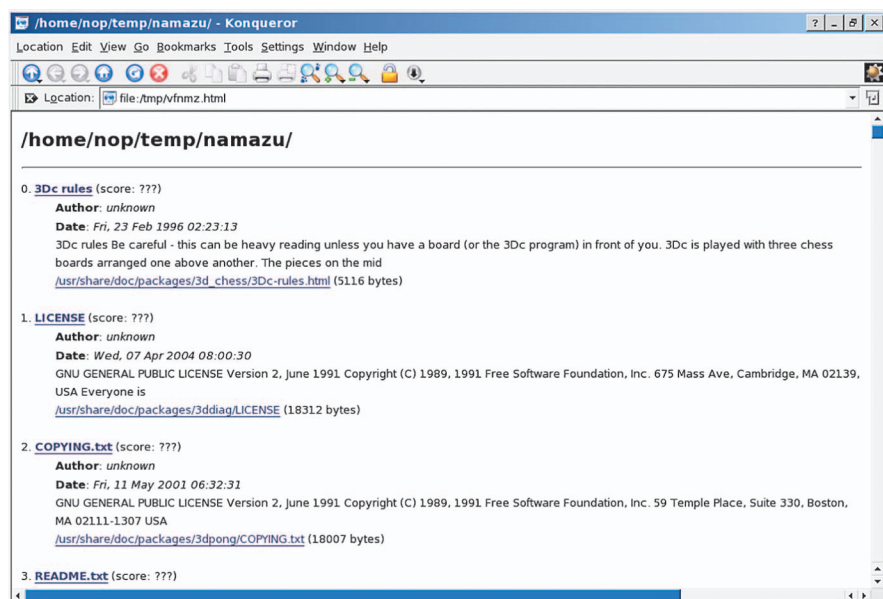
- penggunaan AND. Sebagai contoh: Linux and Linus. Harap diperhatikan bahwa AND adalah default di dalam namazu.

- penggunaan OR. Sebagai contoh: Linux or Windows.
- Penggunaan not. Sebagai contoh: Linux not unix.
- Penggunaan grouping dengan tanda kurung (dipisahkan satu atau lebih spasi). Sebagai contoh (linux and unix) not windows.
- Penggunaan frase, dengan mengapit frase dalam “ “ atau { }. Sebagai contoh: { Linus Torvalds }. Harap diperhatikan bahwa pencarian berbasis frase tidak selalu tepat.
- Penggunaan substring di awal, di tengah atau di akhir. Sebagai contoh *linux, *bsd*, free*.
- Penggunaan regular expression. Sintaks RE yang digunakan adalah gaya Ruby (sama-sama dari Jepang), walaupun gaya Perl juga ditawarkan. Sebagai contoh: /pro(gram|blem)s?/.
- Pencarian field specific (terutama untuk mail/news). Field yang bisa digunakan adalah Subject:, From:, Message-Id:. Sebagai contoh: +subject:Linux.

Boleh dikatakan, namazu sangat sukses dalam menangani query. Sampai di sini, sebenarnya kita sudah selesai mengimplementasikan search engine. Namun, agar lengkap, kita juga akan membahas konfigurasi namazu, namazu di web (baik menggunakan namazu.cgi ataupun PHP), berbagai tip untuk mempercepat pencarian dan pembahasan berbagai tool lain yang datang bersama namazu).

Konfigurasi namazu

Mknmz dan namazu menyimpan file kon-



Preview dengan vfnmz.

figurasi mereka di `/etc/namazu/mknmzrc` dan `/etc/namazu/namazurc`. Berikut ini, kita akan membahas beberapa opsi penting di kedua file konfigurasi tersebut.

mknmzrc:

Berikut ini adalah beberapa opsi penting dalam file konfigurasi mknmzrc.

- `$HTML_SUFFIX`, menspesifikasi suffix html.
- `$ALLOW_FILE`, menentukan file yang diperbolehkan dalam pembuatan index.
- `$DENY_FILE`, menentukan file yang tidak diperbolehkan dalam pembuatan index.
- `$SEARCH_FIELD`, menentukan field dalam pencarian field-specific.
- `$ON_MEMORY_MAX`, menentukan ukuran maksimal file yang dapat di-load ke dalam memory.
- `$FILE_SIZE_MAX`, menentukan ukuran maksimal file untuk indexing.
- `$TEXT_SIZE_MAX`, menentukan ukuran maksimal file text untuk indexing.
- `$WORD LENG_MAX`, menentukan maksimal panjang kata untuk indexing.

namazurc:

Berikut ini adalah beberapa opsi penting dalam file konfigurasi `namazurc`:

- Index, menentukan direktori default untuk index.
- Template, menentukan direktori default untuk template.

- Replace, menentukan string yang ingin di-replace.
- Logging, menentukan aktif atau tidaknya (On atau Off, default adalah on) fasilitas *keyword logging*.
- Scoring, metode scoring yang digunakan (tfidf atau simple).

Anda bisa mengubah beberapa hal seperti direktori index dan opsi memory (dibahas kemudian). Jangan lupa menghapus tanda komentar pada suatu opsi.

namazu.cgi

Komponen utama ketiga dari namazu adalah namazu.cgi yang dapat berfungsi sebagai aplikasi CGI untuk pencarian data di internet. Penggunaan namazu.cgi begitu mudah untuk dilakukan.

Namun, sebelum mempergunakan `namazu.cgi`, sangat disarankan untuk melakukan konfigurasi terhadap file `namazurc`. Setelah itu, nyalakanlah web server Anda (Anda akan membutuhkan Apache terinstal dan terkonfigurasi dengan baik untuk `cgi`, berbagai distro modern seperti SUSE dan Red Hat telah melakukannya untuk Anda).

Cara penggunaan `namazu.cgi` yang paling mudah adalah dengan memberikan langsung kata kunci pencarian dan berbagai opsi ke dalam URL pemanggilan `namazu.cgi`. Sebagai contoh:

http://localhost/cgi-bin/namazu.

```
cgi?query=linux
```

Pemberian query akan langsung membuat `namazu.cgi` mencari sesuai kata kunci yang dimasukkan. Sementara, dengan memberikan parameter `max`, kita dapat mengubah penampilan entri per halaman `namazu`:

```
http://localhost/cgi-bin/namazu.  
cgi?query=linux&max=10
```

Pemberian parameter whence juga dapat dipergunakan untuk langsung melompat ke nomor tertentu hasil pencarian:

`http://localhost/cgi-bin/namazu.cgi?query=linux&max=10&whence=27`

Untuk pengurutan data, Anda bisa memberikan parameter sort, sebagai contoh:

http://localhost/cgi-bin/namazu.cgi?query=linux&max=10&whence=27&sort=score

Dan, untuk penampilan data, Anda bisa memberikan parameter `result`, sebagai contoh:

```
http://localhost/cgi-bin/namazu.cgi?query=
linux&max=10&result=short&sort=score
```

Namaz di web dengan php

Sebenarnya, kita juga dapat mempergunakan namazu dalam kombinasi dengan script PHP. Namazu selanjutnya kita manfaatkan sebagai *backend* search engine. Pada pengembangan berikutnya, namazu akan sekaligus datang dengan libnmz yang matang sehingga kita bisa langsung mempergunakan pustaka namazu dengan berbagai bahasa pemrograman yang mendukung penggunaan pustaka sistem.

Berikut ini adalah contoh kerjasama antara PHP dan namazu (bukan namazu.cgi) dalam menghasilkan pencarian.

Kita akan memiliki 2 file:

- search.html: form HTML.
- search.php: prosesor yang akan meminta namazu untuk mencari dan mengembalikan hasilnya ke browser.

Isi file search.html:

```
<html><head>  
<title>Contoh pencarian  
mempergunakan namazu</title>  
</head>  
<body>  
Masukkan kata kunci &nbsp;&nbsp;&
```

