



oleh Gerrit Renker  
<gerrit.renker(at)gmx.de>

*Tentang Penulis:*  
Gerrit tidak menyukai komputer sama sekali sampai dia mencoba C dan Linux.

*Diterjemahkan Ke Indonesia*  
*Oleh:*  
Muhammad Rachmadi  
([homepage](#))

## Motivasi

Sebuah pribahasa lama mengatakan bahwa sebuah buku jangan dinilai dari sampulnya. Hal yang serupa terjadi dengan kode sumber terbuka (*open source*). Bagaimanapun juga, sumber terbuka tidak sama dengan dokumentasi terbuka, dan proses membaca menjadi semakin sulit seiring dengan berkas-berkas yang semakin banyak dan panjang. Saya baru-baru ini harus memrogram dengan sebuah perangkat lunak yang memiliki dokumentasi setengah halaman html dibandingkan dengan lebih dari 384.000 baris kode Java sumber terbuka yang tersebar di lebih dari 2060 berkas (lihat gambar). Dipandang dari sudut dimensi yang demikian, alat pengarah elektronik, rekayasa balik dan analisis menjadi sangat dibutuhkan, seperti navigator kode sumber Red Hat yang disajikan dalam artikel ini.

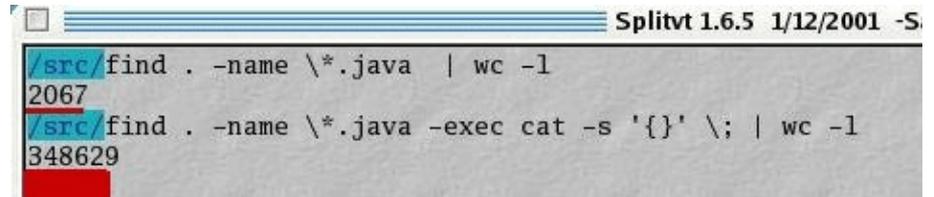
## Instalasi pada debian

Pada debian anda dapat memperoleh semuanya melalui satu baris perintah

```
apt-get install sourcnav sourcnav-doc
```

Perintah ini pada saat yang sama juga mengambil dokumentasinya. Source navigator akan berada di `/usr/lib/sourcnav/`, anda dapat memanggil program utamanya melalui `/usr/lib/sourcnav/bin/snavigator` (lihat tip tentang symlinks di bawah ini). Dokumentasi dapat ditemukan di `/usr/share/doc/sourcnav/html/`.

## Menjelajah dengan snavigator



```
Splitvt 1.6.5 1/12/2001 -S
/src/ find . -name \*.java | wc -l
2067
/src/ find . -name \*.java -exec cat -s '{}' \; | wc -l
348629
```

*Abstrak:*

Artikel ini menyajikan snavigator, sebuah alat analisis kode, lintas-referensi (*cross-referencing*) dan rekayasa ulang yang tangguh yang sangat diperlukan untuk menangani kompleksitas pemeliharaan perangkat lunak yang lebih besar dan paket-paket dengan cara yang efektif.

## Instalasi dari sumber

URL dari laman source navigator adalah <http://sourcnav.sourceforge.net/>, unduhan yang sebenarnya dari [sini \(sourceforge.net/project/showfiles.php?group\\_id=51180\)](http://sourceforge.net/project/showfiles.php?group_id=51180). Dapatkan tarball terbaru `sourcnav-xx.xx.tar.gz`. Selagi mengunduh, coba lakukan hal lain – ukuran sumbernya mencapai 55 Megabita. Ini cukup beralasan, semua paket termasuk di dalamnya. Meskipun perangkat lunak ini menggunakan pustaka-pustaka lain seperti Tcl/Tk, Tix dan Berkeley DB, versi yang sesuai dari paket-paket ini semua sudah termasuk di dalamnya. Untuk menghindari bentrokan dengan versi lain dari Tcl/Tk dsb. pada sistem anda, sebaiknya instal pada direktori yang terpisah, misalnya `/opt/sourcnav`. Selanjutnya dianjurkan untuk menggunakan *build directory* yang terpisah, yang bekerja sebagai berikut. Setelah meng-unzip, dalam direktori yang berisi sumber-sumber yang sudah diekstrak jalankan perintah berikut:

```
mkdir snbuild; cd snbuild
../sourcnav-*/configure --prefix=/opt/sourcnav
make                               ## perlu waktu sejenak ...
make install                       ## mungkin harus jadi root dulu
```

Opsi `--prefix` adalah untuk menentukan direktori instalasi. Selagi konfigurasi berjalan, anda telah dapat mengetahui tentang bahasa-bahasa yang dapat ditangani oleh snavigator. Juga dimungkinkan untuk menambahkan pengurai (*parser*) bagi bahasa yang anda pilih. Segera setelah instalasi diselesaikan melalui `make install`, snavigator siap untuk bekerja dan dijalankan sebagai `/opt/sourcnav/bin/snavigator`. Dari pada menggunakan lingkungan shell PATH, saya menganjurkan untuk menggunakan *symlink* sebagai gantinya.

```
ln -s /opt/sourcnav/bin/snavigator /usr/local/bin
```

Berkas utama yang dapat dieksekusi adalah sebuah skrip shell yang perlu mengetahui direktorinya. Maka dari itu berkas tersebut akan bingung jika dipanggil melalui symlink. Ini dapat diatasi dengan mengubah baris-baris berikut dalam `/opt/sourcnav/bin/snavigator`; sebagai ganti

```
snbindir=`dirname $0`
```

gunakan

```
prog=`readlink -f $0`
snbindir=`dirname $prog`
```

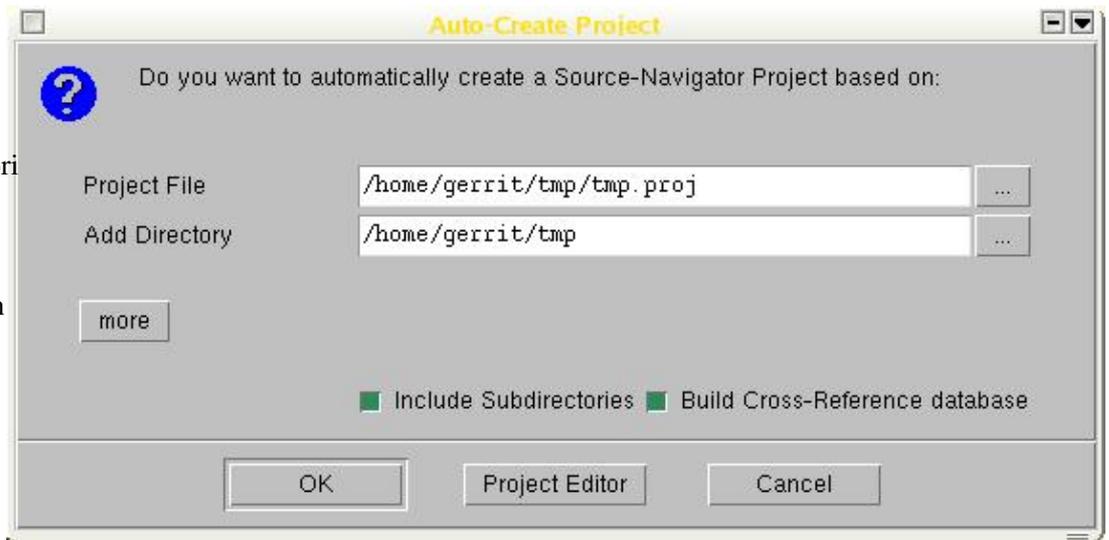
Opsi `-f` pada **readlink(1)** membuat representasi *canonical pathname*, ini berarti bahwa perintah tersebut tetap bekerja jika berkas tersebut diakses melalui symlink tersarang (*nested*) yang sangat panjang.

## Penggunaan snavigator

Pertama kali navigator dipanggil, ia menanyakan direktori–direktori yang berisi berkas–berkas sumber, seperti yang ditunjukkan tangkapan layar berikut.

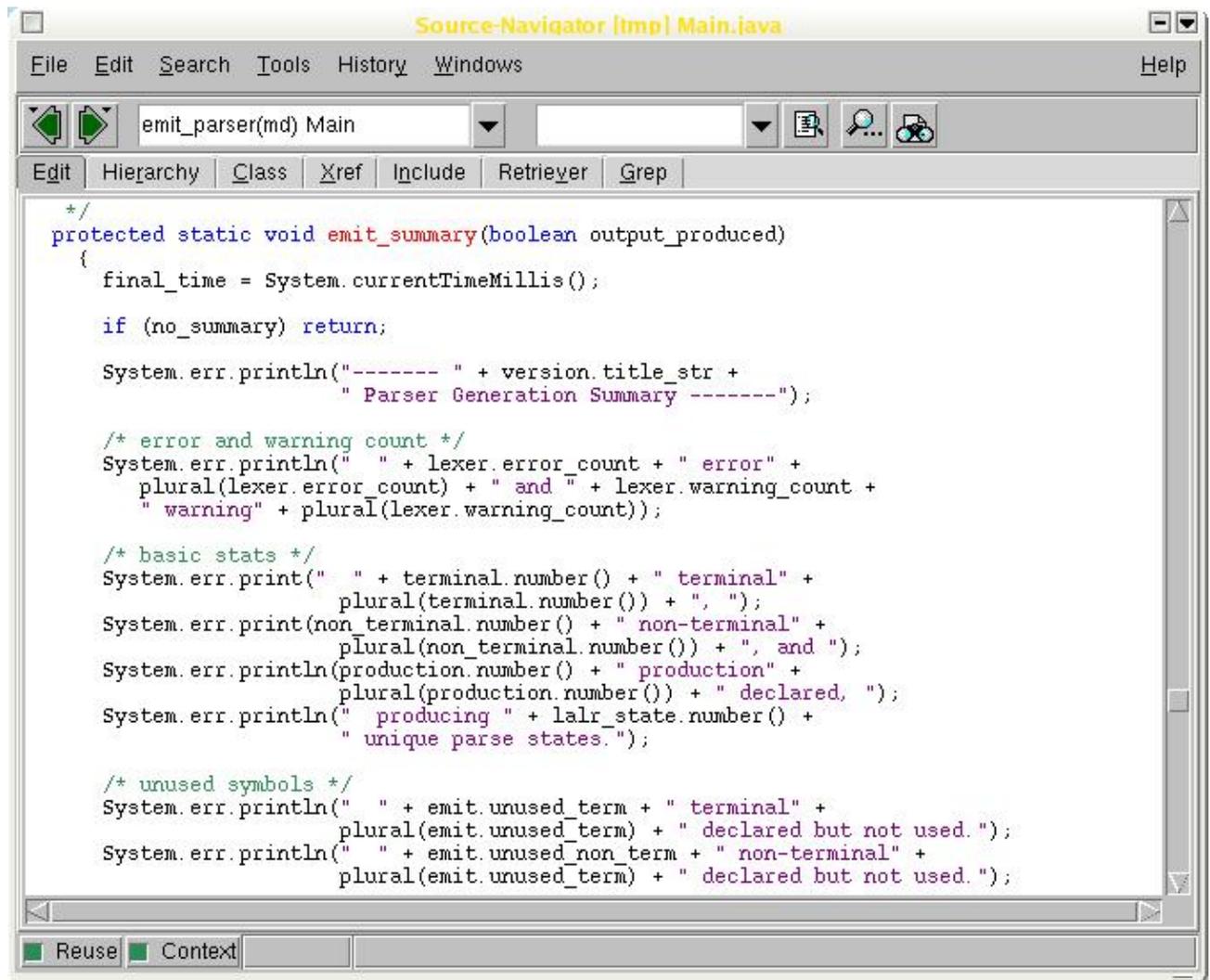
Bahasa–bahasa yang didukung mencakup, tapi tidak terbatas pada, Java, C,

C++, Tcl, Fortran, COBOL, dan program–program assembly. Setelah diberi rincian dari lokasi–lokasi kode, ia secara mandiri membangun sebuah basis data proyek yang mencakup informasi referensi, hierarki kelas, saling ketergantungan berkas ... dan banyak lagi. Waktu yang dibutuhkan untuk membangun tergantung pada ukuran proyek anda. Setelah selesai, basis data dapat di–query dan informasi tambahan tentang kode akan ditanyakan. Berikut adalah sorotan tentang beberapa fitur untuk memberi anda penjelasan. Sebuah *user guide* bergambar, dan juga *reference manual*, dimasukkan dalam direktori `html` dari direktori instalasi.



## Manajemen proyek

Bagian dari perangkat ini adalah *editor* dengan sorotan sintaks, ini juga dapat digunakan untuk berkas–berkas yang dicetak indah. Tangkapan layar berikut menggambarkan jendela utama *editor*. Ini hampir menyerupai sebuah lingkungan pengembangan yang hadir dengan fasilitas pengawakutuan(*debugging*), perintah membangun proyek, kendali versi dan sebagainya.



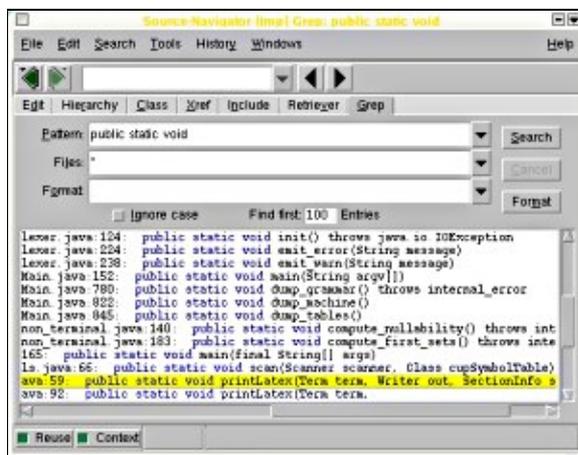
Secara khusus, panah-panah hijau besar pada menu bekerja seperti pada peramban (*browser*) web. *Editor* proyek memungkinkan anda untuk mengendalikan informasi basis data, misalnya jika sebuah berkas telah diperbaharui; untuk menambah atau menghapus berkas dari daftar dan tugas-tugas manajemen lainnya. Semua berkas diperlakukan sebagai satu proyek besar. Maka jika perubahan dilakukan, anda dapat memperbaharui informasi basis data melalui *Refresh Project* atau *Reparse Project*. Ketika berada dalam jendela *editor*, menyorot sesuatu, misalnya nama fungsi, wilayah yang disorot akan berwarna uning. Kemudian klik-kanan dengan tetikus(*mouse*) – ada pilihan untuk mencari (a) *declaration* dari apa yang anda baru saja sorot (misalnya berkas *header*) atau (b) *implementation* dari simbol yang disorot (misalnya berkas *.cpp*), ditambah pilihan-pilihan berguna lainnya.



## Peramban Simbol

Ini adalah jendela pertama yang terbuka setelah menghasilkan basis data proyek. Biasanya peramban ini berisi nama berkas, tapi ia juga menampilkan metode kelas, simbol–simbol fungsi dan sejenisnya. Ketika sebuah nama berkas diklik, berkas akan dibuka pada editor.

## Jendela grep

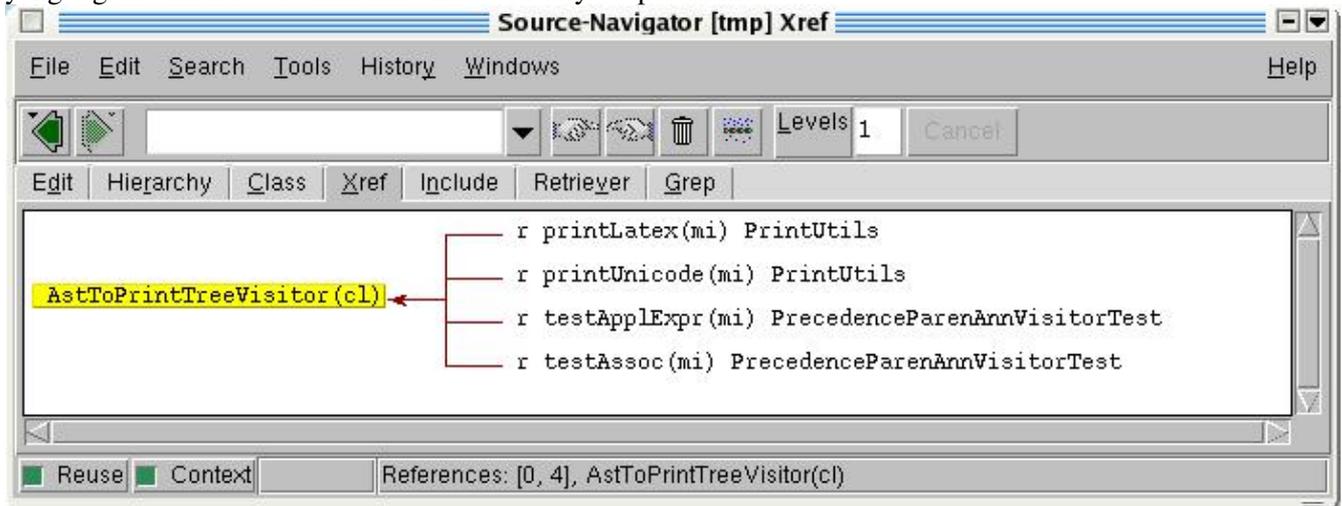


Jendela ini berlaku sesuai namanya, ia menyediakan antarmuka grafis yang nyaman untuk melakukan grep melalui semua berkas kode sumber yang terlibat. Entri–entri yang bersesuaian akan disorot dan dihipertaut (*hyper-linked*), sehingga kode sumber dapat dirambah seolah–olah ia adalah sekumpulan halaman web. Sebagaimana yang ditampilkan tangkapan layar, masing–masing berkas dan lokasinya dapat dipilih dan hanya dengan mengklik entri tersebut anda bisa membuka editor pada posisi yang tepat. (Cara pencarian khusus ini memberikan hasil positif pada banyak berkas Java :)

## Jendela Xref

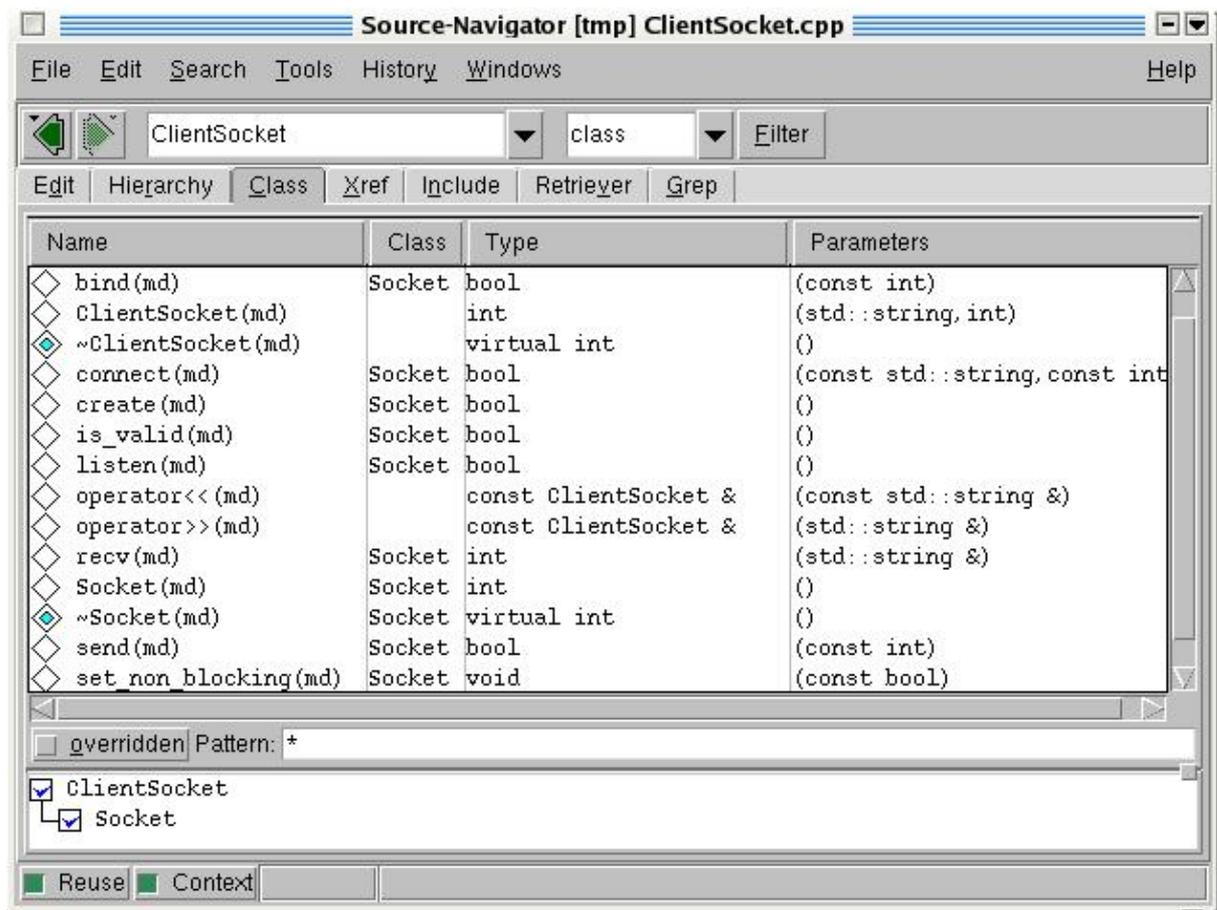
Di sini kita memiliki senarai lintas–referensi dari semua simbol, secara khusus kita dapat melihat metode mana yang melakukan read (r), write (w), ... pada data yang mana dan melihat hubungan di antara simbol

yang digambarkan secara hierarkis. Entri–entri-nya dapat diklik.



## Jendela kelas

Antarmuka ini mengumpulkan semua informasi yang berguna yang kita ingin ketahui tentang kelas–kelas dalam bahasa berorientasi objek. Secara khusus, *super*– dan *subclass* ditunjukkan, juga atribut dan nama metode bersama dengan parameter– parameternya. Untuk perubahan, jendela di bawah ini menunjukkan kelas C++ `ClientSocket` yang mewarisi dari `Socket` dan mempunyai beberapa metode. Sekali lagi, dengan mengklik salah satu entri anda dapat membuka jendela editor pada posisi yang tepat.



# Alternatif lain

`cscope` adalah peramban kode sumber C berbasis layar-konsol (dapat digunakan untuk C++ juga). Peramban ini memiliki beberapa fungsi snavigator, tangkapan layarnya ada di [sini](#). Kenyataannya, peramban ini jauh lebih tua dan telah digunakan dalam banyak proyek besar. Lamannya adalah <http://cscope.sourceforge.net/>. Tapi anda tidak perlu ke sana, peramban ini tertanam dalam **vim** dan dapat digunakan sebagaimana (g)vim digunakan dengan dikombinasikan bersama `tags`. Cukup ketik

```
:help cscope
```

pada sesi vim anda untuk memeriksa keberadaan opsi tersebut. Ada beberapa turunan dari `cscope`. **Freescope** merupakan klon dari `cscope` yang memiliki beberapa fungsionalitas tambahan seperti pelengkapan simbol. Sekarang ada juga `cscope` dengan sisi depan (*front end*) antarmuka KDE yang disebut `kscope`, peramban ini dapat ditemukan di <http://kscope.sourceforge.net/>.

## Kesimpulan

Bagi siapapun yang terlibat dalam paling tidak sebagian rekayasa ulang atau pepaduan kode sumber, snavigator adalah alat yang sangat berguna dan tangguh. Saya pernah memiliki aplikasi Qt yang lama yang sayangnya tidak dapat bekerja dengan pustaka Qt versi baru. Dengan melihat pesan kesalahan dan menjelajah sedikit dengan snavigator, saya segera menemukan bahwa hanya senarai parameter dari satu fungsi yang perlu diubah. Menggunakan fungsionalitas *click-and-locate*, adalah mungkin untuk memperbaharui paket perangkat lunak hanya dalam beberapa menit.

---

<p><u>Halaman Web Dirawat oleh Team Editor LinuxFocus</u> © Gerrit Renker "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a></p>	<p>Informasi Terjemahan: en --&gt; -- : Gerrit Renker &lt;gerrit.renker(at)gmx.de&gt; en --&gt; id: Muhammad Rachmadi (<a href="#">homepage</a>)</p>
---	--