

# Tutorial Emacs

@sopier\*

[https://github.com/sopier/emacs\\_doc](https://github.com/sopier/emacs_doc)

February 4, 2015

## Contents

<b>1</b>	<b>Pendahuluan</b>	<b>2</b>
1.1	Tentang Emacs . . . . .	2
<b>2</b>	<b>Bekerja dengan Berkas</b>	<b>3</b>
<b>3</b>	<b>Dasar-dasar Navigasi</b>	<b>3</b>
3.1	Karakter Control dan Meta . . . . .	3
3.2	Navigasi . . . . .	4
3.3	Numeric argument . . . . .	5
<b>4</b>	<b>Dasar-dasar Penyuntingan Teks</b>	<b>5</b>
4.1	Menyorot Teks . . . . .	5
4.2	Menghapus Teks . . . . .	6
4.3	Killing and Yanking . . . . .	6
4.4	Salin dan Tempel . . . . .	7
4.5	Salin Tempel Seleksi . . . . .	7
4.6	Memotong (Cut) Teks . . . . .	7
<b>5</b>	<b>Mengakses Bantuan (Help)</b>	<b>7</b>
<b>6</b>	<b>Buffers</b>	<b>8</b>
6.1	ido-mode . . . . .	9
6.2	transient-mark-mode . . . . .	9
6.3	*scratch* buffers . . . . .	10
<b>7</b>	<b>Bookmarks</b>	<b>10</b>
<b>8</b>	<b>Registers</b>	<b>11</b>

---

\*<http://twitter.com/sopier>

<b>9</b>	<b>Search and Replace</b>	<b>11</b>
9.1	Replace String . . . . .	12
9.2	Query Replace . . . . .	12
<b>10</b>	<b>Modes</b>	<b>13</b>
10.1	Dired-Mode . . . . .	13
10.2	org-mode . . . . .	14
10.2.1	Instalasi . . . . .	14
10.2.2	Aktifkan org-mode . . . . .	14
10.2.3	Navigasi Cepat . . . . .	14
10.2.4	Format Standar Dokumen Org . . . . .	14
10.2.5	Promote / Demote Headline . . . . .	15
10.2.6	Ekspor Dokumen . . . . .	15
10.2.7	Expand and Collapse Tree . . . . .	15
10.2.8	Format Font . . . . .	16
10.2.9	Comment . . . . .	16
10.2.10	Lists . . . . .	16
10.2.11	Partial Export . . . . .	17
10.2.12	Todo Items . . . . .	17
10.2.13	TODO and DONE items . . . . .	17
10.2.14	Date and Time . . . . .	18
10.2.15	Links . . . . .	18
10.2.16	Tagging . . . . .	19
10.2.17	Membuat Tabel . . . . .	19
10.2.18	Image . . . . .	19
10.2.19	Source Code . . . . .	20
10.2.20	Paket fancyhdr . . . . .	21
10.2.21	Syntax Highlighting . . . . .	21
10.2.22	Mengganti ukuran font pada table . . . . .	21
10.2.23	Contoh Berkas Org-Mode untuk Catatan Harian . . . . .	22
10.3	DocView-Mode . . . . .	22
10.4	viper-mode . . . . .	23
<b>11</b>	<b>Menjalankan Shell dari emacs</b>	<b>23</b>
<b>12</b>	<b>Tips dan Tricks</b>	<b>24</b>
12.1	Multiple Windows . . . . .	24
12.2	Multiple Frame . . . . .	24
12.3	Menggagalkan Perintah . . . . .	25
12.4	Memunculkan Menu Bantuan (Help) . . . . .	25
12.5	Suspend Mode . . . . .	26
12.6	Undo dan Redo . . . . .	26
12.7	Transpose . . . . .	26
12.8	Rectangles / Multi-Cursor . . . . .	27
12.9	Melakukan Perulangan Angka . . . . .	29
12.10	Dynamic Abbrev . . . . .	29

12.11	Hippie-expand . . . . .	29
12.12	FTP atau SCP Berkas pada Dired Mode . . . . .	30
12.13	Menuju ke Baris Tertentu . . . . .	30
12.14	Membuat Tabel . . . . .	30
12.15	Membuat Tabel Latex . . . . .	31
12.16	Keyboard Macro . . . . .	31
12.17	Lowercase, Uppercase dan Capitalize . . . . .	33
12.18	Mengaktifkan Mode Baca (Read-Only) . . . . .	33
12.19	Memahami Mode Layar pada emacs . . . . .	33
12.20	Menjalankan IRC Client . . . . .	34
12.21	Dari org ke tex dan PDF . . . . .	34
12.22	Memanggil Menubar pada emacs-nox . . . . .	35
12.23	Import Tabel HTML kedalam Org-Mode . . . . .	35
12.24	Access Command Line Manual (man) . . . . .	35
12.25	Access Info Mode . . . . .	35
12.26	Melakukan Kustomisasi . . . . .	36
12.27	Merapikan Baris . . . . .	36
12.28	Lain-lain . . . . .	36
<b>13</b>	<b>Version History</b>	<b>37</b>

# 1 Pendahuluan

Dari sekian banyak aplikasi penyunting teks 'jadul' di keluarga Unix adalah **emacs**. Sama halnya dengan **Vim**, penyunting teks satu ini juga memiliki penggemar tersendiri, yang kadang karena kefanatikannya sampai terjadi *war* antara **emacs** vs **Vim**. Pertanyaan yang sering muncul adalah "apakah **emacs** dapat melakukan ini, apakah **Vim** memiliki fitur ini?" Dalam artikel ini, penulis tidak mau terjebak dalam *war* tersebut, melainkan sekedar menuliskan pengalaman menggunakan **emacs**, dari kacamata seorang awam.

Salah satu alasan saya belajar **emacs** adalah kenyataan bahwa beberapa *key bindings* pada program di \*Nix menggunakan **emacs**, sebagai contoh **Bash**. Untuk melakukan navigasi perintah pada **Bash**, *default key bindings* yang digunakan adalah **emacs**.

Sebagai contoh, Anda memiliki perintah seperti berikut:

```
$ ssh -i key.pem -D 9999 ubuntu@someserver.com
```

Anda dapat menyunting perintah di atas dengan menggunakan navigasi seperti pada **emacs**, misal:

Perintah	Keterangan
M-f	Menuju ke kata selanjutnya
M-4 M-f	Maju 4 kata
C-a	Menuju ke awal baris
C-e	Menuju ke akhir baris

Atau perintah **C-r** untuk melihat *history* perintah pada **Bash**, di mana perintah ini kalau di **emacs** digunakan untuk pencarian ke belakang (*backward-search*), hampir sama bukan?

Itu hanya contoh kecil saja, karena saya sering bekerja dengan **terminal**, maka menguasai dasar-dasar navigasi dalam **emacs** berguna untuk menyunting perintah secara cepat.

Sekali lagi, tulisan ini bukanlah sebuah tutorial lengkap program **emacs**, melainkan sekedar catatan pribadi penulis yang coba dituangkan kedalam sebuah berkas elektronik dengan tujuan agar lebih mudah disebarluaskan kepada siapa saja yang membutuhkan. Semoga bermanfaat...

## 1.1 Tentang Emacs

*GNU Emacs is an extensible, customizable text editor - and more.*<sup>1</sup>

**Emacs** termasuk dalam keluarga aplikasi penyunting teks, namun memiliki kelebihan dari sisi *extensibility*. Tidaklah heran beberapa kalangan menyebut **emacs** sebagai sebuah sistem operasi, karena selain menyunting teks, dengan **emacs**, Anda dapat membaca *email*, menjelajah *website*, dan memainkan beberapa aplikasi permainan sederhana, *FTP client*, *news reader*, dan masih banyak

<sup>1</sup><http://www.gnu.org/software/emacs/>

lagi. Anda pun dapat membuat versi **emacs** Anda sendiri, selama Anda memahami bagaimana cara melakukan pengembangan pada **emacs**.

Pengembangan aplikasi ini dimulai pada pertengahan 1970-an, dan masih aktif sampai saat tutorial ini ditulis (2013). **Emacs** memiliki lebih dari 2.000 perintah *built-in* yang dapat digunakan untuk membuat *macro* untuk membuat pekerjaan menjadi otomatis.

**Emacs** dibuat menggunakan **emacs lisp**, sebuah *variant* dari bahasa **Lisp**.

Versi pertama dari **emacs** ditulis oleh Richard Stallman dan Guy L. Steele pada tahun 1976.

**Emacs** sendiri memiliki banyak *variant*, dan saat ini *variant* yang paling banyak dipakai adalah GNU **Emacs** yang dibuat oleh Stallman untuk *GNU Project*.

## 2 Bekerja dengan Berkas

Berikut perintah ketika bekerja dengan berkas:

Perintah	Keterangan
C-x C-f	Membuka berkas
C-x C-s	Menyimpan berkas
C-x s	Menyimpan semua berkas yang sedang disunting
C-x i	Memasukkan teks dari berkas lain ke dalam berkas yang sedang disunting
C-x C-v	Mengganti berkas yang sedang disunting dengan berkas lain
C-x C-w	Menyimpan buffer ke dalam berkas lain
C-x C-q	Mengubah ke mode read-only
C-x C-k	Menutup berkas

## 3 Dasar-dasar Navigasi

Sebelum masuk ke pembahasan mengenai navigasi, mari kita pelajari terlebih dahulu apa yang dimaksud dengan karakter **Control** dan karakter **Meta** pada **emacs**.

Penting juga untuk diketahui penggunaan istilah *point* yang tidak lain adalah *cursor* (kursor) atau posisi letak *mouse* Anda dalam dokumen.

### 3.1 Karakter Control dan Meta

Program penyunting teks **emacs** akan sering sekali melibatkan dua tombol pada *keyboard*, yakni tombol **Ctrl** dan tombol **Meta**. Untuk tombol **Ctrl** (sering disingkat dengan **C**), saya yakin semua sudah mengetahui, lalu di mana letak tombol **Meta** (**M**)?

Saya menggunakan sistem operasi **Ubuntu** dan tombol **Meta** secara otomatis di-*assign* pada tombol **Alt**.

Ketika Anda bekerja dengan `emacs` dalam mode *GUI*, penggunaan tombol `Alt` sebagai karakter *Meta* tidak menjadi masalah. Namun ketika Anda bekerja dengan mode *command line* (`emacs -nw`) menggunakan program `gnome-terminal`, hal ini akan menjadi masalah, karena begitu Anda tekan tombol `Alt`, otomatis `gnome-terminal` akan mengaktifkan *menu access*.

Solusinya adalah dengan mengubah pengaturan pada `gnome-terminal` dengan cara:

Edit->Keyboard Shortcuts->Hilangkan centang pada "Enable menu access keys"

## 3.2 Navigasi

Sama seperti `Vim`, pengguna `emacs` akan merasakan manfaat yang lebih besar ketika meminimalkan penggunaan perangkat *mouse* mereka. Untuk itu mari kita belajar melakukan navigasi pada `emacs` menggunakan perangkat *keyboard*.

Berikut ini beberapa perintah navigasi dasar pada `emacs`:

Perintah	Keterangan
C-f	Bergerak maju 1 karakter
C-b	Bergerak mundur 1 karakter
M-f	Bergerak maju 1 kata
M-b	Bergerak mundur 1 kata
C-a	Bergerak ke awal baris
C-e	Bergerak ke akhir baris
C-n	Bergerak ke baris selanjutnya (bawah)
C-p	Bergerak ke baris sebelumnya (atas)
C-v	Menggulung layar ke bawah
M-v	Menggulung layar ke atas
C-l	Menempatkan point tepat di tengah layar
M-<	Menuju ke awal (top) buffer
M->	Menuju ke akhir (bottom) buffer
M-a	Menuju ke awal kalimat
M-e	Menuju ke akhir kalimat
C-m	Membuat baris baru tepat di bawah posisi point
C-o	Membuat baris baru tepat di atas posisi point
C-j	Membuat baris baru dan indent

Selain, perintah di atas, salah satu favorit saya untuk melakukan navigasi adalah dengan memanfaatkan *search*. `Emacs` memiliki dua macam *search*, yakni *incremental search* dan *non-incremental search*. Masing-masing terbagi lagi menjadi dua, yakni *forward-search* dan *backward-search*.

Berikut perintah untuk melakukan *search*:

Perintah	Keterangan
C-s <character/word> <RETURN>	forward incremental search
C-r <character/word> <RETURN>	backward incremental search
C-s <RETURN> <character/word>	forward non-incremental search
C-r <RETURN> <character/word>	backward non-incremental search

Berikut contoh untuk melakukan navigasi secara cepat menggunakan fitur *incremental search*:

`emacs` adalah program penyunting teks yang handal.

Letakkan *point* pada awal kalimat di atas dengan menekan `C-a`, kemudian tekan `C-s`, untuk menuju ke huruf `t` pada kata `teks`, gunakan perintah `C-s t C-s` kemudian tekan `enter`. Untuk menuju ke karakter `t` berikutnya, cukup tekan `C-s` berulang-ulang.

Dan berikut contoh untuk *backward incremental search*, mari kita asumsikan *point* berada pada akhir baris, untuk menuju ke huruf `e` pada kata `teks`, cukup tekan `C-r e`, kemudian tekan `enter`.

Dengan menguasai hal ini saja, penulis yakin kecepatan Anda dalam menyunting teks akan meningkat drastis (dibandingkan dengan harus menggunakan *mouse*).

### 3.3 Numeric argument

*Numeric argument* pada `emacs` merupakan salah satu fitur untuk meningkatkan efisiensi pekerjaan. Dengan fitur ini, kita dapat mengulang seberapa banyak sebuah perintah untuk dijalankan. Berikut beberapa contoh penggunaan *numeric argument* pada `emacs`.

Perintah	Keterangan
<code>C-u 8 C-f</code>	Bergerak 8 karakter ke kanan
<code>C-u 8 C-n</code>	Bergerak 8 baris ke bawah
<code>C-u 8 *</code>	Mengetik tanda bintang sebanyak 8 kali

## 4 Dasar-dasar Penyuntingan Teks

Untuk memulai pengetikan, Anda tinggal menekan tombol-tombol yang ada pada *keyboard* Anda. Satu hal yang perlu ditekankan di sini, apabila Anda ingin mengetik secara efektif dan efisien, ada 2 syarat penting, yakni pertama tinggalkan penggunaan perangkat *mouse* Anda, dan sebisa mungkin hindari penggunaan tombol panah untuk melakukan navigasi.

### 4.1 Menyorot Teks

Melakukan seleksi (sorot) pada `emacs` cukup dengan menekan tombol:

Perintah	Keterangan
<code>C-&lt;Space&gt;</code>	Mark Set (mengaktifkan mode sorot)

Kemudian diikuti dengan *motion* untuk menandai daerah yang akan disorot. Misal untuk menyorot sebuah baris:

Perintah	Keterangan
C-a	Menuju ke awal baris
C-<Space>	Mulai sorot
C-e	Menuju ke akhir baris

## 4.2 Menghapus Teks

Berikut kombinasi perintah untuk melakukan penghapusan teks dalam **emacs**:

Perintah	Keterangan
<Backspace>	Menghapus 1 karakter sebelum point
C-d	Menghapus 1 karakter di atas point
M-<Backspace>	Menghapus 1 kata sebelum point
M-d	Menghapus 1 kata setelah point
C-k	Menghapus dari posisi point sampai akhir baris
M-k	Menghapus dari posisi point sampai akhir kalimat

Anda juga dapat menggabungkan perintah penghapusan ini dengan fitur *numeric argument*, misal untuk menghapus 2 baris, gunakan perintah berikut:

Perintah	Keterangan
C-a	Menuju ke awal baris
C-u 2 C-k	Menghapus 2 baris ke bawah

## 4.3 Killing and Yanking

**Emacs** menggunakan istilah *killing* untuk *cutting*, dan *yanking* untuk *pasting*. *Killing* memindahkan teks dari *buffer* ke bagian paling atas dari *kill ring* (*clipboard*). *kill ring* ini sendiri mampu menampung sampai 60 buah *killed items*.

Anda dapat memanggil *kill ring* ini menggunakan perintah M-y, tekan lagi untuk memanggil *kill items* sebelumnya, dan seterusnya.

Berikut cara untuk berinteraksi dengan *kill ring*:

Perintah	Keterangan
C-w	Killing
C-y	Yanking
M-y	Memanggil yank sebelumnya
M-y	Memanggil yank sebelumnya, dst...

Membingungkan? Tenang ... Itu hanya karena belum terbiasa saja ... :)



## 4.4 Salin dan Tempel

Perintah untuk menyalin teks dalam **emacs**:

Perintah	Keterangan
M-w	Menyalin (copy) teks
C-y	Paste (yank) teks

Berikut ini contoh perintah untuk salin baris:

Perintah	Keterangan
C-a	Menuju awal baris
C-k	Hapus baris
C-y	Tempel baris
C-m	Membuat baris baru
C-y	Tempel lagi

## 4.5 Salin Tempel Seleksi

Perintah	Keterangan
C-<Space>	Start the selection
C-p / C-n / C-f / C-b	Memilih area
C-w	Cut
M-w	Copy
C-y	Paste

## 4.6 Memotong (Cut) Teks

Perintah untuk memotong teks dalam **emacs**:

Perintah	Keterangan
C-w	Memotong (cut) teks
C-y	Menempel (paste) (yank)

## 5 Mengakses Bantuan (Help)

Ini adalah fitur yang **wajib** dikuasai oleh setiap pengguna **Emasc**, kenapa? Karena otak kita tidak mungkin untuk mengingat semua kombinasi tombol perintah yang ada pada **Emacs**.

Berikut ini beberapa tips yang penulis pakai untuk mengakses menu bantuan (*help*) untuk menemukan informasi yang kita inginkan.

Tekan **C-h a** (a untuk *apropos*) untuk memunculkan menu bantuan, kemudian ketik *keyword* yang ingin kita cari, misalkan saya ingin mengetahui bagaimana cara membuka daftar *buffers* yang aktif, saya ketikkan saja *buffers list*, kemudian **emacs** merespon dengan informasi yang [mungkin] terkait dengan *keyword buffers list*. Selanjutnya tinggal kita pilih dan buka informasi mana yang mungkin relevan.

Contoh kedua adalah mencari tahu kombinasi tombol pada **emacs** untuk melakukan suatu pekerjaan. Misalkan saya ingin mencari kombinasi tombol untuk menggulung layar pada *window* lain. Ketikkan saja **C-h b**, **b** di sini singkatan dari *bindings*, ketik *keyword scroll other*, tekan **Enter**, dan muncullah informasi yang mungkin terkait dengan apa yang kita cari, untuk lebih cepat menemukan tinggal kita gunakan saja fungsi *search* (**C-s**) diikuti dengan kata yang ingin dicari.

Dan tentunya masih banyak lagi contoh lain, yang jelas fitur bantuan yang sangat lengkap ini tentu saja memudahkan bagi siapa saja (baik yang sudah mahir atau pun masih pemula) untuk mengatasi permasalahan yang dihadapi ketika bekerja dengan **Emacs**, tanpa harus melakukan jelajah ke dunia maya.

## 6 Buffers

**Emacs** memiliki 2 jenis *buffer*, yang terkait dengan berkas dan yang tidak terkait dengan berkas.

Perintah	Keterangan
<b>C-x C-f</b>	Membuka buffer berkas
<b>C-x C-c</b>	Menutup emacs
<b>C-x k</b>	Menutup (kill) berkas (buffer)
<b>C-x C-s</b>	Menyimpan berkas
<b>C-x C-w</b>	Menyimpan buffer sebagai berkas baru (save as)

Berikut ini beberapa perintah untuk bekerja dengan *buffer* pada **emacs**:

Perintah	Keterangan
<b>C-x &lt;Panah Kanan&gt;</b>	Berpindah ke buffer berikutnya
<b>C-x &lt;Panah Kiri&gt;</b>	Berpindah ke buffer sebelumnya
<b>C-x b</b>	Berpindah ke named buffer
<b>C-x C-b</b>	Membuka named buffer dalam mode horizontal split
<b>C-x C-b</b>	Daftar buffer yang aktif
<b>C-x 1</b>	Kembali ke buffer aktif
<b>C-x s</b>	Menyimpan perubahan di semua buffer

Ada dua perintah untuk menyimpan berkas, yang pertama menggunakan **C-x C-s**, di mana perintah ini akan menyimpan perubahan hanya pada *buffer* yang aktif, dan perintah kedua adalah **C-x s**, yang mana akan menyimpan semua perubahan yang ada pada semua *buffer* berkas yang sedang kita sunting.

Perintah kedua ini akan diikuti dengan *confirmation* apakah kita akan menyimpan perubahan tersebut pada setiap *buffer*-nya.

## 6.1 ido-mode

Selain menggunakan *default buffer*, **emacs** juga menyediakan beberapa *mode* untuk bekerja dengan *buffer* secara lebih efisien. Keduanya adalah **iswitchb-mode** dan **ido-mode**.

Keduanya memiliki cara kerja yang sama, namun untuk **emacs 22+**, lebih disarankan menggunakan **ido-mode**. Pertanyaan selanjutnya apa beda antara **ido-mode** dengan sistem *buffer* bawaan?

Salah satu hal yang mendasar adalah dengan menekan **C-x b**, pada **ido-mode** Anda dapat melihat semua daftar *buffer* yang tersedia, sedangkan jika menggunakan *buffer* bawaan, menekan **C-x b**, Anda tidak dapat melihat daftar *buffer*.

Berikut perintah untuk **ido-mode**:

Perintah	Keterangan
M-x ido-mode	mengaktifkan ido-mode
C-x b	melihat daftar buffer
C-s	berpindah (rotate) antar buffer forward
C-r	berpindah (rotate) antar buffer backward
C-x k	kill buffer

**ido-mode** (singkatan dari *interactive-do*) juga mendukung *auto-completion* dengan mengetikkan beberapa karakter kemudian tekan **TAB**. Anda dapat membuat *buffer* baru dengan mengetikkan nama *buffer*, kemudian tekan **ENTER** dua kali.

Apabila Anda ingin selalu menggunakan **ido-mode** setiap kali **emacs** dijalankan, tambahkan baris berikut pada berkas `~/.emacs` Anda:

```
(ido-mode t) ; set ido-mode true
```

## 6.2 transient-mark-mode

Mulai dari **emacs 23+**, *transient-mark-mode* diaktifkan secara *default*. Anda dapat menonaktifkan *mode* ini dengan menekan:

Perintah	Keterangan
M-x	transient-mark-mode

Apa keuntungan kita men-*disabled mode* ini? Salah satunya adalah untuk keperluan *bookmarking*, sebagai contoh Anda sekarang berada pada baris ke-200, kemudian Anda tekan **M-g g 100** untuk lompat ke baris 100. Setelah selesai melakukan penyuntingan pada baris ke-100, Anda dapat lompat kembali ke baris ke-200 dengan hanya menekan **C-x C-x**.

Pertanyaan selanjutnya adalah apakah sebaiknya *transient-mark-mode* di-*enable* atau di-*disabled*? Itu semua kembali pada Anda, jika Anda sering melakukan *copying*, *killing*, dan *yanking*, maka sebaiknya *transient-mark-mode* diaktifkan. Namun jika Anda sering lompat-lompat antar baris, sebaiknya di non-aktifkan, tergantung selera ...

### 6.3 \*scratch\* buffers

Setiap kali Anda membuka **emacs**, ada beberapa *buffers* yang secara otomatis dibuka oleh **emacs** (dinamakan juga **emacs buffer**). Kalau Anda membuka *buffer* lagi, maka itu dinamakan sebagai *user buffers*.

Salah satu **emacs buffer** adalah **\*scratch\***, yang sesuai dengan namanya, ini adalah tempat di mana Anda dapat melakukan corat-coret terkait dengan apa yang sedang Anda kerjakan.

Selain itu, penulis sering memanfaatkan *buffer* ini sebagai *command-mode*-nya **emacs**. Kenapa? Karena ketika saya sedang di depan komputer, kadang ada gangguan internal (baca: anak minta ikut coding), sementara kita tidak ingin pekerjaan teracak-acak olehnya. Dengan menggunakan **\*scratch\*** *buffer* ini, kita tidak perlu lagi khawatir pekerjaan jadi kacau balau karenanya.

Anda pun dapat menjadikan **\*scratch\*** *buffer* ini sebagai media untuk belajar bahasa pemrograman **Lisp**, berikut ini contoh operasi aritmatika sederhana:

```
(+ 5 3)
```

Tekan C-j, maka **emacs** akan mengevaluasi baris tersebut dan memberikan hasilnya tepat di garis setelahnya.

```
(+ 5 3)
8
```

## 7 Bookmarks

Sesuai dengan namanya, *bookmarks* memungkinkan kita untuk menyimpan posisi *point* dalam berkas. *Bookmarks* ditandai dengan sebuah nama yang akan disimpan selama *sessions* **emacs** masih berjalan. Bahkan ketika isi dari berkas berubah, *bookmark* masih dapat bekerja dengan baik. Hal ini dikarenakan *bookmarks* tidak hanya menyimpan posisi *point* namun juga menyimpan teks apa saja yang ada di sekitar *point*. Namun jika berkas sudah banyak berubah, mungkin saja *bookmarks* tidak akan bekerja lagi.

Berikut ini perintah-perintah terkait dengan *bookmarks*:

Perintah	Keterangan
C-x r m BOOKMARK <RET>	menyimpan bookmark dengan nama BOOKMARK
C-x r b BOOKMARK <RET>	memanggil bookmark dengan nama BOOKMARK
C-x r l	melihat daftar bookmark
M-x bookmark-save	menyimpan bookmark ke dalam berkas
M-x edit-bookmarks	menyunting bookmark

## 8 Registers

Fitur *register* dapat menyimpan teks yang Anda salin, untuk kemudian Anda tempel di mana pun Anda inginkan (tanpa harus mencari pada *ring*). Caranya adalah dengan menyimpan teks dalam *register* yang dapat Anda panggil setiap Anda butuhkan.

Berikut cara untuk bekerja dengan *registers*

Perintah	Keterangan
C-x r s k	copy region ke dalam register r (copy-to-register)
C-x r i k	panggil register k (insert-register)
M-x append-to-register <RET> k	tambahkan (append) region ke dalam register k
M-x prepend-to-register <RET> k	prepend region ke dalam register k

## 9 Search and Replace

Salah satu fungsi penting sebuah program penyunting teks adalah fungsi cari dan ganti (*search and replace*). Pada bagian ini kita akan mempelajari bagaimana **emacs** melakukan fungsi tersebut.

Pencarian pada **emacs** dapat dibagi menjadi dua, pencarian maju dan pencarian mundur. Pencarian maju lakukan dengan menekan:

Perintah	Keterangan
C-s	Mengaktifkan mode search maju
C-s	Mencari next occurrence (jika ada)
Enter	Keluar dari search mode

Pencarian mundur:

Perintah	Keterangan
C-r	Mengaktifkan mode search mundur
C-r	Mencari previous occurrence (jika ada)
Enter	Keluar dari search mode

Untuk mencari *previous occurrence* pada pencarian mundur dapat juga dengan menekan tombol **Backspace**.

Anda dapat juga mengubah mode pencarian maju ke mundur (atau sebaliknya) dengan menekan **C-s** atau **C-r**.

## 9.1 Replace String

Cara pertama adalah menggunakan *eXtended Command* yang dapat diakses menggunakan M-x. Sebagai contoh kita memiliki kalimat berikut:

Ini ibu budi

Untuk mengganti kata ibu menjadi bapak, berikut perintah yang harus dilakukan:

Perintah	Keterangan
C-a	Menuju ke awal baris
C-<SPC>	Mark Set
C-e	Sorot sampai akhir baris
M-x repl<Tab>s<Tab><Enter>ibu<Enter>bapak<Enter>	

Cara kedua adalah dengan menggunakan perintah **replace-string**. Caranya adalah dengan menyorot daerah yang mengandung kata yang akan diganti kemudian tekan M-x **replace-string** tekan **enter**, masukkan kata yang ingin diganti, tekan **enter** lagi, kemudian masukkan kata yang pengganti.

Perintah	Keterangan
C-space	Mulai menyorot
Motion	Tandai daerah yang mengandung kata
M-x replace-string	
Enter	
Masukkan kata yang dicari	
Enter	
Masukkan kata pengganti	
Enter	

## 9.2 Query Replace

Fitur ini akan mengganti setiap perulangan kata yang dicari pada seluruh dokumen namun dengan proses konfirmasi sebelumnya. Berikut ini perintahnya:

Perintah	Keterangan
M-x query-replace <teks awal> <Enter> <teks pengganti> <Enter>	
atau	
M-% <teks awal> <Enter> <teks pengganti> <Enter>	

Misal:

Saya seorang programmer

Anda ingin mengganti kata **programmer** menjadi **coder**:

Perintah
C-a
M-x query-replace <Enter> <programmer> <Enter> <coder> <Enter>

## 10 Modes

**emacs** memiliki banyak *mode*, berikut ini beberapa *mode* yang sering penulis pakai:

### 10.1 Dired-Mode

Untuk melakukan pekerjaan manipulasi berkas atau direktori, **emacs** menyediakan sebuah *mode* bernama **Dired mode**. Sesuai dengan namanya, *Dired* (*directory editor*) mempermudah kita dalam melakukan manipulasi berkas atau direktori dengan mudahnya.

Untuk berpindah ke **dired mode**, kita dapat mengetik:

Perintah	Keterangan
C-x d	aktifkan dired mode, atau
M-x dired	aktifkan dired mode

Dari sini, Anda dapat melakukan manipulasi berkas dan direktori, seperti misalnya untuk menghapus sebuah berkas, tekan D dan seterusnya ...

Berikut beberapa perintah terkait dengan *dired mode*

Perintah	Keterangan
<RETURN>	membuka berkas
q	menutup direktori
C	menyalin berkas
R	mengubah (move) nama berkas
D	menghapus berkas
+	membuat direktori baru
Z	compress / decompress berkas menggunakan gzip
m	memberi tanda pada berkas
u	menghilangkan tanda pada berkas
U	menghilangkan tanda pada semua berkas yang sudah ditandai
% m	memberi tanda menggunakan regex
g	refresh direktori
^	kembali ke direktori atasnya
w	menyalin nama berkas
ow	menyalin full path dan nama berkas

Mungkin ada yang bingung bagaimana cara memindahkan berkas / direktori pada *dired-mode*? Gampang kok tinggal pilih satu [beberapa] atau direktori yang ingin Anda pindahkan, kemudian tekan R.

## 10.2 org-mode

Apa itu *org-mode*?

*Org mode is for keeping notes, maintaining TODO lists, planning projects, and authoring documents with a fast and effective plain-text system.*<sup>2</sup>

*org-mode* merupakan salah satu fitur andalan **emacs**. Seperti apakah *org-mode* itu? Mari kita langsung ke contoh saja ...

### 10.2.1 Instalasi

Pertama kali mari kita pasang **org-mode**, pada **Ubuntu**, ketik:

```
sudo apt-get install org-mode
```

### 10.2.2 Aktifkan org-mode

Buka berkas baru, kemudian aktifkan *org-mode* dengan:

-----+-----
Perintah   Keterangan
-----+-----
M-x org-mode   Aktifkan org-mode
-----+-----

Jika Anda ingin *org-mode* terpanggil secara otomatis setiap kali **emacs** dinyalakan, silakan sunting berkas **.emacs** Anda dan isikan baris berikut:

```
;; Must have org-mode loaded before we can configure org-babel
(require 'org-install)
```

### 10.2.3 Navigasi Cepat

Untuk berpindah secara cepat antar headline, berikut tombol kombinasinya:

-----+-----
Perintah   Keterangan
-----+-----
C-c C-n   Berpindah ke headline berikutnya
C-c C-p   Berpindah ke headline sebelumnya
-----+-----

### 10.2.4 Format Standar Dokumen Org

Format dasar dokumen **org** adalah:

-----+-----
Kode   Keterangan
-----+-----
*   bagian utama
**   subbagian tingkat dua
***   subbagian tingkat tiga

---

<sup>2</sup><http://www.orgmode.org>



-	list	
/i/	cetak miring	
*b/	cetak tebal	
_u_	garis bawah	
=c=	true type font (code)	
-----+-----		

Untuk mengetahui format lain, tekan C-c C-x C-f.

### 10.2.5 Promote / Demote Headline

Fitur ini berguna untuk mengubah secara cepat sebuah *headline* yang tadinya bintang satu, menjadi bintang dua dan seterusnya.

-----+-----	
Perintah	Keterangan
-----+-----	
M-S-Panah Kanan / Kiri	Promote / Demote Headline
-----+-----	

### 10.2.6 Ekspor Dokumen

Dengan menggunakan *org-mode*, Anda dapat dengan mudah meng-*export* dokumen Anda ke berbagai format, misal HTML, LaTeX maupun format-format lain.

Berikut contoh berkas yang menggunakan *org-mode*

```
* Judul
  Isi dari judul
** Sub Judul 1
*** Sub Judul 2
```

Mari kita *export* dokumen di atas ke dalam format HTML dengan cara:

-----+-----	
Perintah	Keterangan
-----+-----	
C-c C-e	Aktifkan export mode
b	Pilih b untuk export ke html
<RETURN>	Tekan Enter
-----+-----	

Silakan ditunggu sebentar, berikutnya **emacs** akan menampilkan hasil *export* tadi dalam aplikasi *browser*.

Perlu diketahui bahwa *option* **b** di atas hanyalah salah satu dari sekian banyak pilihan format dokumen yang akan di-*export*. Untuk format-format lain, silakan di-*explore* lebih lanjut.

### 10.2.7 Expand and Collapse Tree

Jika Anda sudah pernah menggunakan konsep *folding*, tentu tidak akan bingung, *org-mode* mendukung hal tersebut dengan cara menekan:

Perintah	Keterangan
<TAB>	rotate current subtree
atau	
S-<TAB>	rotate entire buffer

S adalah tombol Shift.

### 10.2.8 Format Font

Untuk memformat huruf dalam lingkungan *org-mode*, cukup tambahkan karakter-karakter khusus berikut di awal dan di akhir kata yang ingin Anda format.

Misalnya, untuk mencetak huruf tebal, tambahkan tanda **\*** di awal dan di akhir kata, **\*ini akan dicetak tebal\***.

Berikut daftar lengkap format huruf pada *org-mode*:

Perintah	Keterangan
<b>*bold*</b>	cetak tebal
<i>/italics/</i>	cetak miring
<u>_underline_</u>	garis bawah
<code>=code=</code>	format untuk kode
<code>~verbatim~</code>	format font mesin ketik

### 10.2.9 Comment

Untuk melakukan *comment*, gunakan tanda **#** di awal baris, atau tambahkan kata **COMMENT** jika Anda ingin memberi *comment* pada sebuah *entry*, sebagai contoh:

```
# Ini tidak akan di export
** COMMENT subbagian ini juga tidak akan di export
isi dari subbagian ini juga tidak akan di export
```

### 10.2.10 Lists

Berikut ini adalah cara membuat *lists* pada *org-mode*:

```
- TODO
- Mandi
- Sarapan
- Coding
```

### Merubah Jenis List

Ada beberapa macam list dalam *org-mode*, seperti angka, tanda kurang, angka dengan kurung, tanda tambah, dsb. Anda dapat merubahnya secara otomatis menggunakan tombol:

Perintah	Keterangan
S-Panah Kanan / Kiri	Merubah jenis list numbering

### 10.2.11 Partial Export

Selain meng-*export* seluruh dokumen, Anda pun dapat melakukan *export* hanya pada bagian tertentu, dengan cara menyorot bagian yang ingin Anda sorot, kemudian tekan C-c C-e b untuk *export* ke format HTML.

### 10.2.12 Todo Items

Untuk membuat daftar *todo items*, lihat contoh berikut:

- [X] emacs
- [ ] vim
- [ ] others

Untuk memberi (menghilangkan) centang, gunakan perintah:

Perintah	Keterangan
C-c C-c	Memberi / menghilangkan tanda centang

### 10.2.13 TODO and DONE items

Selain menggunakan sistem *checklist* di atas, Anda pun dapat membuat *TODO* *DONE* menggunakan format sebagai berikut:

- \* DONE Go to school
- \* TODO Doing homework

Berikut cara membuatnya:

- \* <SPACE> C-c C-t <Go to school>

Atau:

- \* S-<RightArrow>

Untuk mengubah dari *TODO* menjadi *DONE*, cukup tekan C-c C-t sekali lagi.

### 10.2.14 Date and Time

Untuk memasukkan *date and time stamp*, gunakan perintah berikut:

Perintah	Keterangan
C-c .	memasukkan timestampe
C-u C-c .	memasukkan timestamp dengan jam

Begitu Anda tekan **RETURN**, maka akan muncul kalender di bagian bawah layar **emacs** Anda. Selanjutnya Anda dapat melakukan navigasi dengan menggunakan *mouse* Anda.

```
<2014-05-15 Thu>  
<2013-01-14 Mon 08:57>
```

Anda dapat menyunting tanggal di atas dengan mengarahkan *point* Anda ke angka tahun, kemudian tekan **S-Up** atau **S-Down** untuk menambah atau mengurangi angka tahun, begitu pula dengan bulan, tanggal, hari dapat Anda ubah seperti cara di atas.

Khusus untuk mengganti hari (tanggal), Anda dapat menggunakan tombol:

Perintah	Keterangan
S-LeftArrow	previous date / day
S-RightArrow	next date / day

### 10.2.15 Links

Berikut ini format untuk membuat *links* dalam *org-mode*:

```
http://www.infotiket.com
```

atau:

```
[[http://www.elapak.com][Situs Jual Beli Indonesia]]
```

Begitu Anda tutup tanda *square bracket* yang terakhir, format *URL* di atas otomatis akan berubah menjadi sebuah *link* dengan *anchor*:

```
Situs Jual Beli Indonesia
```

Untuk menyunting *link* di atas, tempatkan *point* pada *URL*, kemudian tekan:

Perintah	Keterangan
C-c C-l	Sunting link (url)

### 10.2.16 Tagging

Berikut cara membuat *tagging* dalam **emacs**:

Arahkan *mouse* ke salah satu bagian atau subbagian, kemudian tekan:

```
|-----+-----|
| Perintah | Keterangan |
|-----+-----|
| C-c C-c | Menambahkan tagging |
|-----+-----|
```

Nanti akan muncul *prompt* di bagian bawah layar **emacs**, selanjutnya isikan *tags* yang Anda inginkan. Untuk menyunting *tags*, tinggal ulangi perintah C-c C-c di atas.

### 10.2.17 Membuat Tabel

Berikut cara membuat tabel dalam *org-mode*:

```
| kolom 1 | kolom 2 |
|---<TAB>
some data<TAB>123
<ENTER>
<ENTER>
<ENTER>
<TAB>
<TAB>
```

Dan berikut hasilnya:

```
| kolom 1 | kolom 2 |
|-----+-----|
| some data | 123 |
| | |
| | |
| | |
| | |
| | |
```

Anda dapat melakukan operasi matematika dengan tabel tersebut, silakan di-*explore* sendiri.

### 10.2.18 Image

Untuk menampilkan gambar di *org-mode* yang latex friendly, gunakan kode berikut:

```
#+CAPTION: Plot TLKM.JK dan SMA20
#+NAME: fig:tlkm
#+ATTR_LaTeX: placement=[H]
[[./images/tlkm.png]]
```

### 10.2.19 Source Code

Sebelum kita dapat menjalankan kode dalam *org-mode*, pastikan kita sudah melakukan pengaturan *org-babel* pada berkas `~/.emacs`. Isi berkas tersebut dengan baris berikut:

```
; Must have org-mode loaded before we can configure org-babel
(require 'org-install)

;Some initial languages we want org-babel to support
(org-babel-do-load-languages
 'org-babel-load-languages
 '(
  (sh . t)
  (python . t)
  (R . t)
  (ruby . t)
  (ditaa . t)
  (dot . t)
  (octave . t)
  (sqlite . t)
  (perl . t)
  ))
```

Selanjutnya silakan *restart emacs*.

Untuk menuliskan *source code*, kita dapat menggunakan *tag* berikut:

```
#+BEGIN_SRC sh :exports both
echo "Hello World!" # C-c C-c
#+END_SRC
```

Tempatkan *point* pada kode, kemudian tekan `C-c C-c`, dan hasilnya:

```
#+BEGIN_SRC sh :exports both
echo "Hello World!" # C-c C-c
#+END_SRC
```

```
#+results:
: Hello World!
```

Contoh lain untuk kode `python`:

```
#+BEGIN_SRC python :exports both
return "Hello Python!" # C-c C-c
#+END_SRC
```

```
#+results:
: Hello Python!
```

Mungkin bagi Anda yang terbiasa dengan bahasa pemrograman `python` bertanya-tanya, kenapa menggunakan `return`, dan tidak menggunakan `print`. Saya sendiri sampai saat ini belum apa alasan `emacs` menggunakan *keyword* `return`, yang jelas jika Anda coba mengganti dengan `print`, maka *result*-nya menjadi `none`.

*Note:* *org-babel* ini hanya akan jalan jika Anda sudah memasang paket `texlive-latex-extra` pada sistem operasi Anda. Jika Anda pengguna Ubuntu, tentunya gunakan:

```
sudo apt-get install texlive-latex-extra
```

### 10.2.20 Paket fancyhdr

Org mode mendukung paket fancyhdr seperti ketika membuat dokumen Latex pada umumnya, berikut caranya:

```
#+LATEX_HEADER: \usepackage{fancyhdr}
#+LATEX_HEADER: \pagestyle{fancy}
#+LATEX_HEADER: \cfoot{}
#+LATEX_HEADER: \rfoot{\thepage}
```

### 10.2.21 Syntax Highlighting

Apabila Anda ingin fitur /syntax highlighting/ aktif pada berkas PDF, caranya adalah dengan mengaktifkan paket `minted`.

Pertama, pasang pustaka pygments, dengan cara:

```
$ sudo pip install pygments
```

Setelah itu, sunting berkas `.emacs` Anda:

```
(require 'org-latex)
(setq org-export-latex-listings 'minted)
(add-to-list 'org-export-latex-packages-alist '("" "minted"))
```

Terakhir, masukkan kode berikut di *header* berkas Org Anda:

```
#+LaTeX_HEADER: \usepackage{minted}
```

*Voila*, semua kode sekarang menjadi berwarna!

### 10.2.22 Mengganti ukuran font pada table

Berikut ini contoh mengganti ukuran font tabel menjadi footnotesize

```
#+LATEX: {\footnotesize
#+ATTR_LaTeX: align=cl placement=[H]
|-----+-----|
| H0 | Korelasi antara harga saham tinggi sehingga harga |
|     | saham yang akan datang dapat diramalkan dengan   |
|     | menggunakan harga di masa lalu                     |
|-----+-----|
| H1 | Korelasi antara harga saham rendah sehingga harga  |
|     | saham di masa depan tidak dapat diprediksi menggunakan |
|     | data harga masa lalu                                 |
|-----+-----|
```

### 10.2.23 Contoh Berkas Org-Mode untuk Catatan Harian

Berikut ini contoh penggunaan berkas *org-mode* untuk *daily log*:

```
* Jan 14 in Yogyakarta <2013-01-14 Mon>:day:

** Writing emacs documentation

    Today we write about org-mode / org-babel
```

Salah satu hal yang membuat saya suka dengan **emacs** adalah komunitas pengguna **emacs** sangat mendorong kita sebagai pengguna **emacs** untuk membuat catatan harian seperti di atas, hal ini berlaku tidak hanya bagi peneliti namun juga *programmer* pengguna **emacs**.

## 10.3 DocView-Mode

Tahukah Anda, bahwa Anda dapat membuka berkas PDF di **emacs**? Emacs menyediakan *mode* khusus untuk ini, yakni *docview-mode*.

Caranya sama dengan ketika Anda membuka berkas baru, yakni dengan menekan **C-x C-f** kemudian pilih berkas PDF Anda, kemudian tekan **Enter**.

Berikut ini adalah perintah untuk berinteraksi ketika bekerja dengan *DocView-mode*:

Perintah	Keterangan
<SPC>	Menuju ke halaman berikutnya
<Backspace>	Menuju ke halaman sebelumnya
+	Memperbesar halaman (Zoom in)
-	Memperkecil halaman (Zoom out)
M-<	Menuju ke awal dokumen
M->	Menuju ke akhir dokumen
M-g g <num> atau M-g M-g <num>	Menuju ke halaman tertentu
C-s	forward search
C-r	backward search
C-u C-s	memulai pencarian baru (forward)
C-u C-r	memulai pencarian baru (backward)
s s	slicing halaman
s m	slicing dengan mouse
s r	reset ke default
q	keluar dari mode docview

Terkait dengan fitur *search*, *docview-mode* memiliki beberapa perintah khusus, misalnya ketika Anda sudah selesai melakukan pencarian, kemudian ingin melanjutkan ke pencarian berikutnya, maka Anda harus menekan **C-u C-s** atau **C-u C-r**. Setelah menekan tombol **Enter**, maka hasil pencarian pada dokumen tidak akan di-*highlight*, melainkan akan ditampilkan dalam bentuk *tooltip*, yang akan muncul ketika Anda melakukan *hover* perangkat *mouse* Anda pada halaman tersebut.

Berkas lain yang didukung oleh *docview-mode* ini adalah DVI, PostScript (PS), OpenDocument, dan Microsoft Office documents.



## 10.4 viper-mode

Bagi Anda yang suka dengan model navigasi Vim, Anda dapat mengaktifkan *viper-mode*. Mode ini mendukung semua perintah navigasi seperti ketika bekerja dengan Vim, misal **w** untuk menggerakkan *point* ke kata selanjutnya, **G** untuk bergerak ke akhir *buffer*, dan perintah-perintah navigasi lainnya.

Apabila Anda tertarik dengan Vim, penulis juga menyediakan tutorial Vim dalam format PDF, dan bisa diunduh di alamat berikut:

[http://dl.dropbox.com/u/5052616/vim\\_docs.pdf](http://dl.dropbox.com/u/5052616/vim_docs.pdf)

## 11 Menjalankan Shell dari emacs

Emacs memiliki berbagai macam perintah untuk berinteraksi dengan *shell*:

Perintah	Keterangan
M-! cmd <RET>	menjalankan perintah shell dan menampilkan hasil
M-/ cmd <RET>	menjalankan perintah shell dengan region sebagai input.
M-& cmd <RET>	menjalankan perintah shell asynchronously, dan menampilkan hasil.
M-x shell	menjalankan subshell dengan input dan output melalui buffer emacs. Beberapa command tidak dapat berjalan pada mode ini.
C-u M-x <RET>	Menjalankan shell kedua (multiple shell).
M-x term	menjalankan subshell dengan input dan output melalui buffer emacs. Mode ini mendukung semua command seperti kita membuka aplikasi terminal.
M-x rename-buffer	Mengganti nama buffer.

Shell tersebut tidak ubahnya seperti *buffer*, sehingga Anda dapat memanggilnya (menutupnya) seperti ketika kita bekerja dengan *buffer*.

Untuk mengulangi perintah sebelumnya pada *shell*, tekan:

C-<Panah atas>

Penulis sendiri tidak mengetahui kenapa *shell mode* tidak mendukung semua perintah yang ada pada *Shell*, misal perintah `mkvirtualenv` yang sering penulis pakai ketika bekerja dengan *Python* tidak dapat bekerja pada *shell mode*, namun dapat bekerja pada *term mode*.

Anda dapat menjalankan lebih dari satu *shell mode* dengan cara menekan C-u M-x `shell` diikuti dengan nama *shell* yang Anda inginkan, atau cukup tekan **Enter** untuk memberi nama secara *default* (`*shell*<2>`). Untuk mengganti nama *buffer* Anda dapat menggunakan perintah M-x `rename-buffer`.

Khusus untuk mode *term*, jangan bingung ketika menggunakannya, ada beberapa perbedaan perintah, semua perintah `emacs` yang diawali dengan C-x berubah menjadi C-c, jadi semisal Anda ingin berpindah *buffer*, tekan C-c b. Penulis sendiri sebisa mungkin menghindari mode *term*, karena terkadang lupa

harus mengganti perintah C-x menjadi C-c, namun untuk hal-hal yang memang harus dijalankan melalui *terminal*, mau tidak mau penulis menggunakan mode *term*.

## 12 Tips dan Tricks

### 12.1 Multiple Windows

Untuk meningkatkan efisiensi dalam bekerja, Anda dapat menggunakan fitur *multiple windows*.

Perintah	Keterangan
C-x 3	Split Vertical
C-x 2	Split Horizontal
C-x 1	Menutup semua window kecuali window di mana point aktif
C-x 0	Menutup window aktif (tempat point berada)
C-x o	Berpindah antar window
C-M-v	Menggulung layar window dibawahnya
C-x 4 C-f	Membuka berkas (buffer) pada window sebelahnya
C-u 6 C-x ^	Memperbesar ukuran window sebanyak 6 baris
C-u - 6 C-x ^	Memperkecil ukuran window sebanyak 6 baris

Perintah C-M-v berguna ketika kita ingin menjadikan *window* lain sebagai referensi terhadap pekerjaan kita, tanpa harus berpindah *window*, Anda dapat menggulung layar referensi tersebut, dan *point* tetap berada pada dokumen (berkas) yang sedang kita kerjakan.

Perintah C-x 4 C-f berguna untuk membuka *buffer* berkas pada *window* sebelahnya.

### 12.2 Multiple Frame

Apabila Anda bekerja dengan *GUI*, maka membuat *frame* baru adalah seperti ketika Anda menjalankan *emacs* pertama kali. Fitur ini tidak berguna ketika Anda bekerja dengan *emacs* dalam *mode terminal* (*emacs -nw*).

Berikut ini beberapa perintah untuk bekerja dengan *multi-frame*:

Perintah	Keterangan
C-x 5 f <some_file>	Membuka berkas dalam frame baru
C-x 5 o	Berpindah antar frame
C-x 5 b	Berpindah ke buffer yang ada pada frame lain
C-x 5 0	Menutup frame

Jika diperhatikan perintah-perintah di atas sama dengan perintah ketika kita bekerja dalam *single-frame*, yang membedakan adalah kita menambahkan angka 5 pada setiap perintah yang kita lakukan untuk bekerja dengan *multi-frame*.

## 12.3 Menggagalkan Perintah

Jika Anda membuka `emacs` pertama kali, maka pada baris ke-13 Anda akan temukan kalimat berikut:

To quit a partially entered command, type Control-g.

Perintah	Keterangan
C-g	Menggagalkan perintah

Biar lebih jelas, silakan perhatikan contoh berikut:

Anda ingin membuka berkas dengan menekan tombol `C-x C-f`, namun tiba-tiba Anda ingin menggagalkan perintah tersebut, caranya cukup hanya dengan menekan `C-g`, dengan begitu Anda dapat memberikan perintah baru ke `emacs`.

## 12.4 Memunculkan Menu Bantuan (Help)

Anda dapat menampilkan menu bantuan (*help*) dengan menekan tombol berikut:

Perintah	Keterangan
F1	
M-x help <RET>	
C-h <Kombinasi Perintah>	
C-h ?	
C-h c <Kombinasi Perintah>	help singkat, misal: C-h c C-a
C-h k <Kombinasi Perintah>	help panjang, misal: C-h k C-a
C-h f <Deskripsi Fungsi>	help berdasar fungsi, misal: C-h previous line
C-h m	help untuk mode yang sedang digunakan
C-h v	dokumentasi variabel
C-h a <keyword>	help berdasar keyword, misal C-h a file
C-h i	emacs manual, a.k.a Info

Sebagai contoh, Anda ingin mengetahui untuk apakah kombinasi perintah `M-a`:

`C-h k M-a`

Maka `emacs` akan memunculkan *window* baru dalam *mode split* horisontal. Untuk kembali melakukan penyuntingan, cukup tutup *window* tersebut dengan perintah yang sama dengan ketika kita ingin menutup *split window*, yakni:

Perintah	Keterangan
C-x 1	Hanya menampilkan 1 window dimana point berada

Perintah `C-h` ini juga dapat kita ganti dengan tombol `F1`, jadi kalau mau melihat dokumentasi tentang *mode* dokumen yang sedang digunakan, cukup tekan `F1 m`, yang *equal* dengan `C-h m`.

## 12.5 Suspend Mode

Fitur *suspend mode* ini berguna ketika Anda tidak sedang dalam *mode* GUI, yakni untuk beralih dengan cepat ke *Shell Interpreter* kemudian menjalankan *Shell command* dari sana.

Caranya adalah:

Perintah	Keterangan
C-z	Suspend emacs

Ketika Anda sedang dalam *mode* GUI, dan menekan C-z, maka layar **emacs** akan ter-*minimize* (*iconify*).

Begitu sudah selesai, Anda dapat kembali ke **emacs** dengan mengetik **fg** pada *Shell*.

Perintah	Keterangan
fg	kembali ke emacs

## 12.6 Undo dan Redo

Untuk melakukan *undo* dan *redo* pada **emacs**:

Perintah	Keterangan
C-x u	Undo
C-_	Undo
C-g C-_	Redo, untuk multiple redo, tekan C-_ lagi sampai menemukan yang dicari.

Konsep *redo* pada **emacs** agak membingungkan, saran saya banyak banyaklah berlatih dan membiasaka diri dengan *redo* pada **emacs**.

## 12.7 Transpose

Sesuai dengan namanya, fungsi perintah ini adalah untuk mengubah urutan karakter / kata, sebagai contoh:

```
thisi s
```

Letakkan *point* pada spasi, kemudian tekan:

```
C-t
```

Dan hasilnya:

```
this is
```

Sekarang menggunakan kalimat `this is` di atas, mari kita rubah menjadi `is this`:

Letakkan *point* pada kata `this`, kemudian tekan:

```
M-t
```

Hasilnya:

```
is this
```

Anda juga dapat melakukan *transpose* pada baris di mana *point* berada dengan baris di atasnya:

```
C-x C-t
: transpose baris dengan baris atasnya
```

Dan berikut adalah rangkuman perintah untuk *transpose*:

Perintah	Keterangan
C-t	transpose point dengan point sebelumnya
M-t	transpose kata dengan kata sebelumnya
C-x C-t	transpose baris dengan baris sebelumnya

## 12.8 Rectangles / Multi-Cursor

Fitur ini berfungsi untuk menyunting kolom pada teks. Sebagai contoh Anda memiliki kode javascript berikut:

```
var a = 'me';
var b = 'you';
var c = 'they';
```

Anda ingin mengubah kata `var` menjadi `this`.. Dengan menggunakan fitur `rectangles`, hal ini mudah untuk dilakukan. Berikut caranya:

Perintah	Keterangan
C-n	Turun satu baris
C-n	Turun satu baris
M-f	Menuju ke kata selanjutnya
M-f	Menuju ke kata selanjutnya
C-b	Menuju ke 1 karakter sebelumnya
C-x r t this.	Ganti rectangle content dengan kata <code>this.</code>
Enter	Enter

Perintah `rectangle` yang lain:

Perintah	Keterangan
C-x r k	hapus (kill) teks dalam region yang disorot
C-x r d	hapus (delete) teks dalam region yang disorot
C-x r y	tempel (yank) rectangle terakhir yang dihapus (kill)
C-x r o	insert blank space ke dalam region yang disorot
C-x r N	memasukkan nomor baris pada region yang disorot
C-x r c	mengosongkan (clear) region yang disorot (ganti dengan)
M-x delete-whitespace-rectangle	spasi menghapus (delete) semua whitespace dalam region
C-x r t string <RET>	replace region yang disorot dengan string teks
M-x string-insert-rectangle<RET>string<RET>	memasukkan teks string dalam setiap baris pada rectangle

Perbedaan antara *kill* dan *delete* adalah kalau perintah *kill*, Anda menyimpan teks dalam *rectangle ring (clipboard)*, sedangkan perintah *delete* tidak menyimpan.

Satu contoh lagi penggunaan C-x r c, misal Anda memiliki teks berikut:

```
+++++++
+++++++
+++++++
+++++++
+++++++
```

Lakukan perintah berikut (*point* berada di pojok kiri atas)

Perintah	Keterangan
C-a	Menuju ke awal baris
C-n	Menuju ke baris ke-2
C-f	Maju 1 karakter
C-Space	Mart set
C-u 2 C-n	Turun 2 baris
C-e	Menuju ke akhir baris
C-b	Mundur 1 karakter
C-x r c	rectangle clear (ganti karakter dengan spasi)

Hasilnya:

```
+++++++
+      +
+      +
+      +
+++++++
```

*Pretty neat, huh?*

## 12.9 Melakukan Perulangan Angka

Pada bagian atas sudah disinggung bagaimana melakukan perulangan pada **emacs**, yakni dengan menggunakan perintah:

```
C-u 5 $
```

Maka hasilnya:

```
$$$$$
```

Namun perulangan ini tidak berlaku untuk angka, misal Anda ingin mengulang angka 8 sebanyak 8 kali, maka perintah **C-u 8 8** tidak akan berhasil. Bagaimana caranya:

```
C-u 8 C-u 8
```

Dan, hasilnya:

```
88888888
```

## 12.10 Dynamic Abbrev

Fitur ini mungkin bagi sebagian orang dinamakan sebagai *auto completion*, sesuai dengan namanya, fitur ini berfungsi untuk *auto complete* dengan cara mencocokkan dengan pola kata yang sudah ada pada dokumen.

Caranya adalah dengan menekan:

-----+-----	
Perintah	Keterangan
-----+-----	
M-/	auto-complete word
C-M-/	menampilkan semua kata yang cocok
	(urut sesuai dengan frekuensi kemunculan
	dalam dokumen
-----+-----	

Misal dalam dokumen Anda sudah terdapat kata **emacs**, maka Anda dapat melakukan *auto-complete* dengan menekan:

```
em M-/
```

Berikutnya akan muncul kata yang diawali dengan kata **em**. Tekan **M-/** lagi akan membawa Anda dengan kata lain yang cocok dengan pola tersebut.

## 12.11 Hippie-expand

Jika pada **dabbrev-expand**, Anda dapat melakukan *auto-completion* pada kata, maka dengan **hippie-expand**, Anda dapat mengulang baris.

Caranya adalah dengan mengetikkan beberapa karakter baris yang ingin Anda lakukan *auto-completion*, kemudian tekan:

Perintah	Keterangan
M-x hippie-expand	Mengaktifkan line-completion

Maka secara otomatis, **emacs** akan melakukan *auto-completion* untuk baris yang memiliki pola yang cocok dengan karakter-karakter tersebut.

## 12.12 FTP atau SCP Berkas pada Dired Mode

Anda dapat melakukan perintah **ftp** atau **scp** pada *dired-mode* dengan cara:

```
Mark the files
! scp * host:/target/dir/ & <RETURN>
```

Di sini **emacs** akan mengganti tanda **\*** dengan berkas yang sudah ditandai dan tanda **&** berarti perintah akan dijalankan secara *async*.

## 12.13 Menuju ke Baris Tertentu

Kita dapat menuju ke baris tertentu pada dokumen dengan menggunakan perintah:

```
M-g g <Number> <RETURN>
```

Sebagai contoh Anda ingin menuju ke baris 1, maka perintahnya:

```
M-g g 1 <RETURN>
```

## 12.14 Membuat Tabel

Secara *default*, **emacs** memiliki fitur untuk membuat tabel dalam format **ASCII**. Bagaimana caranya:

```
M-x table-insert
```

Selanjutnya akan muncul pertanyaan (konfirmasi) mengenai berapa jumlah kolom, baris dan pertanyaan lain terkait dengan tabel yang ingin Anda buat.

Kemudian Anda tinggal masukkan data ke dalam kolom dan baris yang tersedia. Untuk berpindah ke kolom (baris) berikutnya, tekan tombol **Tab**.

Berikut ini contoh tabel **ASCII** dalam **emacs**:

Nama	Alamat	Jenis Kelamin



Untuk mengetahui perintah-perintah lain terkait dengan tabel, ketik:

```
M-x table-<Tab>
```

## 12.15 Membuat Tabel Latex

Bagi Anda yang sering berkulat dengan **LaTeX**, tentu tidak asing lagi dengan *syntax* pembuatan tabel yang terasa membingungkan dan sulit dihafal.

Untunglah **emacs** menyediakan *tools* di mana kita bisa membuat tabel di **emacs** untuk kemudian di-*export* ke *syntax LaTeX*.

Pertama kita buat dulu tabel pada **emacs** dengan cara:

```
M-x table-insert
Masukkan variabel jumlah kolom, baris, dan seterusnya
```

Yang hasilnya kurang lebih seperti berikut ini:

```
+-----+-----+-----+
|      |      |      |
+-----+-----+-----+
|      |      |      |
+-----+-----+-----+
|      |      |      |
+-----+-----+-----+
```

Anda dapat berpindah dari satu kolom ke kolom berikutnya dengan menekan tombol **Tab**. Dengan *point* masih berada pada tabel, mari kita ubah tabel tersebut ke dalam bahasa **LaTeX** dengan cara:

```
C-~
latex <RETURN>
```

Berikutnya akan muncul *window* baru berisikan kode **LaTeX** untuk tabel kita tadi. Anda tinggal menyalin kode tersebut ke dalam dokumen Anda. Selesai!

Berikut ini contoh kode untuk tabel kita di atas:

```
% This LaTeX table template is generated by emacs 23.3.1
\begin{tabular}{|l|l|l|l|}
\hline
Number & Title & Address & \\
\hline
& & & \\
\hline
& & & \\
\hline
\end{tabular}
```

## 12.16 Keyboard Macro

Ini adalah fitur yang mampu meningkatkan efektivitas ketika bekerja dengan teks. Secara *default*, berikut cara untuk bekerja dengan *macro*:

Perintah	Keterangan
F3	emulai macro
F4	engakhiri macro
F4	engulang macro

Selain tombol di atas, kita juga dapat menggunakan tombol berikut:

Perintah	Keterangan
C-x (	memulai macro
C-x )	mengakhiri macro
C-x e	eksekusi macro

Sebagai contoh, ketika bekerja dengan **LaTeX**, Anda akan sering sekali mengetik `\emph{}` untuk mencetak miring sebuah kata.

Untuk menghindari pengetikan berulang-ulang, Anda dapat menyimpan hal tersebut ke dalam sebuah *macro*:

Perintah	Keterangan
F3	memulai macro
<code>\emph{ }</code>	tulis teks yang akan Anda ulang
F4	akhiri macro

Sekarang waktunya untuk mengulang *macro* tersebut, caranya adalah cukup dengan menekan **F4** sebanyak yang Anda suka, dan teks tersebut akan muncul secara otomatis.

Anda juga dapat memberi nama pada *macro* yang sudah Anda buat sebelumnya. Pada contoh di atas, kita beri nama **miring**. Caranya:

Perintah	Keterangan
C-x C-k n (kmacro-name-last-macro)	Memberi nama
miring	nama
<RET>	

Sekarang Anda dapat memanggil *macro* dengan menekan tombol **M-x miring**, dan teks di atas akan terpanggil.

Nama ini akan tersimpan selama session masih aktif, jika Anda ingin permanen, Anda dapat menyimpannya dalam sebuah berkas tersendiri atau cukup tambahkan dalam berkas `/.emacs` Anda.

Caranya buka berkas `/.emacs` kemudian tambahkan pada baris paling bawah, dengan menekan:

```
M-x insert-kbd-macro <RET> miring <RET>
```

Sekarang *macro* sudah tersimpan secara permanen, Anda dapat memanggil *macro* ini dengan menekan tombol

M-x miring

## 12.17 Lowercase, Uppercase dan Capitalize

Berikut ini perintah untuk mengubah teks menjadi *uppercase/lowercase* pada **emacs**:

Perintah	Keterangan
C-x C-u	uppercase
C-x C-l	lowercase
M-l	membuat lowercase dari point sampai akhir kata
M-u	membuat uppercase dari point sampai akhir kata
M-c	membuat capitalize daerah yang disorot

## 12.18 Mengaktifkan Mode Baca (Read-Only)

Dengan mengaktifkan status *read-only*, maka kita tidak dapat menyunting berkas yang ada pada **emacs**. Hal ini penting untuk menghindari adanya ketikan-ketikan dari pihak luar yang tidak kita inginkan.

Berikut perintah untuk menyalakan / mematikan status *read-only*:

Perintah	Keterangan
C-x C-q	Menyalakan / mematikan status <i>read-only</i>

Mode *read-only* yang aktif ditandai dengan tanda **%%** di bagian bawah dari layar **emacs** Anda.

## 12.19 Memahami Mode Layar pada emacs

Jika Anda perhatikan pada bagian bawah dari layar kerja **emacs** Anda, ada beberapa tanda yang penting untuk mengetahui status dari pekerjaan yang sedang kita kerjakan.

-UU-:\*\*\*-F1  
-UU-:----F1  
-UU-:%%--F1

Tanda **\*\*** berarti ada pekerjaan yang belum kita simpan, sedangkan tanda **--** berarti pekerjaan belum mengalami perubahan sejak terakhir kali kita melakukan penyimpanan, dan tanda **%%** berarti kita sedang berada pada modus *read-only*.

## 12.20 Menjalankan IRC Client

Anda mungkin masih ingat dengan *irc*? Sebuah layanan *chat* berbasis teks yang dulu sempat berjaya (waktu itu belum ada Facebook, Twitter, dll). Dan tahukah Anda bahwa layanan itu sampai sekarang masih *eksis* meski tidak seramai dulu. Salah satu yang masih aktif menggunakan layanan ini adalah komunitas *geek* yang bertebaran di berbagai *room*, termasuk komunitas pengembangan Python yang dapat Anda temui di `#python`.

Anda tentu juga tahu program `mircc`? Sebuah program *irc chat client* yang ngetrend sekali pada jamannya. Syukurlah pengguna `Emacs`, Anda tidak perlu untuk memasang *client* tersendiri, karena `Emacs` sudah memasukkan program untuk itu, namanya `erc`.

Anda dapat mengaktifkan `erc` dengan cara:

```
M-x erc
```

Selanjutnya tinggal ikuti langkah-langkah seperti Anda menggunakan program *irc client* lainnya. Untuk dapat bergabung dalam komunitas ini Anda harus memiliki *id* yang sudah teregistrasi pada `NickServ`, jika tidak Anda akan di-*redirect* ke ruang `#python-unregistered`.

Selamat mencoba!

## 12.21 Dari org ke tex dan PDF

Pada bahasan di atas sempat disinggung bagaimana mengubah dokumen `Org` ke dalam format `HTML`. Ketika Anda menekan tombol `C-cC-e` kalau Anda perhatikan banyak sekali jenis dokumen yang didukung oleh `emacs`. Salah satu favorit saya adalah format `tex`. Anda dapat langsung memilih format `PDF` atau `tex` saja. Kalau ingin praktis, format `PDF` mungkin jadi pilihan, tapi kalau ingin lebih fleksibel, format `tex` tentu tidak ada tandingan.

Kenapa fleksibel? Karena jika menggunakan format langsung ke `PDF`, semuanya serba otomatis, mulai dari judul dokumen yang menggunakan nama berkas, tanggal, warna halaman isi, pengarang, jenis *font*, dan masih banyak lagi. Berdasarkan pengalaman, hasil konversi ke `PDF` ini kurang begitu bagus.

Oleh karena itu penulis lebih memilih mengubah dokumen `org` ke dalam dokumen `tex`, baru setelah itu kita ubah secara manual menggunakan *engine* `pdflatex`.

Bagaimana caranya? Untuk ekspor dokumen, seperti sudah dibahas pada bagian sebelumnya, tekan `C-cC-e` kemudian tekan `1` untuk format `tex`. Setelah itu silakan sunting seperlunya, kalau penulis biasanya menghilangkan ukuran huruf yang secara *default* ditetapkan `11pt`, kemudian menonaktifkan paket `fontenc` agar tampilan huruf terlihat lebih halus.

```
|-----+-----|
| Dari                      | Menjadi                      |
|-----+-----|
| \documentclass[11pt]{article} | \documentclass{article} |
| \usepackage[T1]{fontenc}      | %\usepackage[T1]{fontenc} |
|-----+-----|
```

Jika sudah, sekarang saatnya mengubah dari format `tex` menjadi PDF dengan menekan tombol `C-cC-c` kemudian tekan *Enter*. Dalam sekejap `emacs` membuat berkas baru dengan ekstensi PDF pada direktori yang sama dengan direktori tempat berkas `tex` berada.

Untuk melihat hasil, Anda dapat menekan tombol `C-cC-c` sekali lagi, dan perhatikan layar `emacs` Anda berubah menjadi *pdf viewer*. Atau Anda dapat juga menggunakan aplikasi eksternal untuk membuka berkas PDF tersebut.

## 12.22 Memanggil Menubar pada emacs-nox

Jika Anda bekerja dengan mode GUI, tentu tidak jadi soal, karena Anda dapat memanggil/membuka menubar menggunakan perangkat tetikus Anda, tapi jika Anda bekerja pada `emacs-nox`, bagaimana cara memanggil menubar?

Ternyata caranya cukup mudah, hanya dengan menekan tombol `F10` saja.

Perintah	Keterangan
F10	Memanggil / Membuka Menubar pada mode nox

## 12.23 Import Tabel HTML kedalam Org-Mode

Untuk mengimport html table ke dalam orgmode, salin dan tempel tabelnya dulu, kemudian tekan `C-c` |

## 12.24 Access Command Line Manual (man)

Misal Anda ingin mengetahui manual untuk perintah `ls`, tanpa harus keluar dari `emacs` Anda dapat membacanya dengan:

Perintah	Keterangan
M-x man	Mengaktifkan menu man
command-line	misal, <code>ls</code> , <code>tr</code> , dll.

## 12.25 Access Info Mode

*Info-mode* tidak lain adalah sumber informasi untuk `emacs` sekaligus beberapa topik yang masih berkaitan, khususnya terkait dengan sistem operasi Unix (CMIW), sangat disarankan untuk dibaca, baik bagi Anda yang baru mencoba sistem operasi `emacs` dan Unix (atau turunannya) maupun Anda yang sehari-hari sudah bergelut dengan `emacs/Unix`.

Perintah	Keterangan
M-x info	Mengakses dokumentasi info

Untuk melakukan navigasi pada mode ini, Anda dapat menekan tombol **F10** kemudian tekan **i** untuk *info*, dari sini Anda dapat memilih navigasi yang Anda inginkan (*up*, *next*, dll.)

atau gunakan *shortcut* berikut:

Perintah	Keterangan
<b>h</b>	memunculkan help
<b>u</b>	up
<b>n</b>	next
<b>p</b>	previous
<b>Ret</b>	jump into link

## 12.26 Melakukan Kustomisasi

Salah satu kelebihan emacs adalah pada ekstensibilitas yang dimilikinya. Pengguna dapat dengan mudah melakukan kustomisasi dengan mengaktifkan mode customization (**M-x customize**) dan melakukan pencarian pada kotak pencarian yang disediakan.

Sebagai contoh kita ingin melakukan kustomisasi terhadap **ido-mode**, maka cari ido-mode, kemudian ubah pengaturan dari sana. Setelah itu kita dapat menyimpan pengaturan dengan memilih tautan state, dan pilih model penyimpanan, apakah hanya untuk sesi sekarang, atau disimpan untuk sesi-sesi yang akan datang, jika pilihan kedua, maka emacs akan secara otomatis menyunting berkas **.emacs** pada direktori home Anda.

## 12.27 Merapikan Baris

Untuk membuat kegiatan sunting-menyunting berkas dengan **emacs** lebih rapi dan enak dilihat, saya biasa membatasi panjang setiap baris maksimal 78 karakter, biar tampilan emacs menjadi rapi dan tidak terlalu lebar.

Caranya mudah, cukup menekan **M-q**, dan otomatis baris yang panjang akan terpotong sesuai dengan pengaturan yang Anda tetapkan (secara *default* maksimal 78 karakter).

## 12.28 Lain-lain

Perintah	Keterangan
<b>M-\</b>	remove whitespace before and after cursor and after cursor
<b>M-^</b>	join the line with previous line and fix white space
<b>M-x delete-trailing-whitespace</b>	removes blanks after last char on all lines

Dalam github ini juga disertakan contoh konfigurasi berkas **.emacs** yang penulis gunakan.

[https://github.com/sopier/emacs\\_doc](https://github.com/sopier/emacs_doc)

## 13 Version History

Bagian ini merupakan *log* dari perubahan-perubahan yang ada pada buku kecil ini.

Tanggal	Versi	Keterangan
17 Januari 2013	0.1	Versi pertama
25 April 2013	0.2	Versi kedua, menambahkan beberapa subsection baru, seperti konversi dari org ke tex kemudian ke pdf
29 Desember 2014	0.3	Versi ketiga, sekarang dokumentasi ini available di github