

Belajar L^AT_EX 2_ε Dari Awal

Isro' Hidayatulloh

06 Juli 2015

Daftar Isi

I	Tahap Pengenalan	2
1	Apa itu L^AT_EX?	3
1.1	Instalasi	3
1.1.1	Instalasi di Linux	3
2	Dasar	5
2.1	Sintak L ^A T _E X	5
2.1.1	Spasi	5
2.1.2	Karakter Khusus	6
2.1.3	Group	7
2.1.4	Environment	7
2.1.5	Commands	7
2.1.6	Komentar	8
2.2	Membuat Dokumen Pertama	8
2.2.1	Apa Arti dari Kode Sumber?	9
2.3	File	9
2.3.1	Membuat Nama File yang Baik	9
2.3.2	File Tambahan	10
II	Elemen-elemen Umum	11
3	Struktur Dokumen	12
3.1	Struktur Global	12
3.2	Preamble	13
3.2.1	Document class	13
3.2.2	Package	14
3.3	Lingkungan <i>Document</i>	16
3.3.1	Top matter	16

Bagian I

Tahap Pengenalan

BAB 1

Apa itu L^AT_EX?

T_EX adalah bahasa pemrograman yang dibuat oleh Donald Knuth¹ fungsinya untuk membuat dokumen yang baik dengan mudah. Lalu apa hubungannya dengan L^AT_EX, L^AT_EX adalah pengembangan dari T_EX dengan penambahan package atau style. Sehingga memberikan kemudahan untuk pembuatan dokumen yang kompleks, seperti untuk mengubah font, penambahan gambar, dan masih banyak lagi.

1.1 Instalasi

Instalasi L^AT_EX menggunakan sistem distribusi, distribusi L^AT_EX ini terdiri dari sekumpulan paket dan program. Berikut adalah distribusi L^AT_EX untuk kebanyakan sistem operasi:

- TeX Live² adalah kebanyakan distribusi T_EX untuk BSD, GNU/Linux, Mac OS X, dan Windows.
- MiKTeX³ adalah distribusi khusus Windows.
- MacTeX⁴ distribusi khusus Mac OS turunan dari TeX Live.

Paket distribusi ini tidak mengikutsertakan editor. Kita bisa menggunakan text editor biasa untuk membuat kode sumber L^AT_EX.

1.1.1 Instalasi di Linux

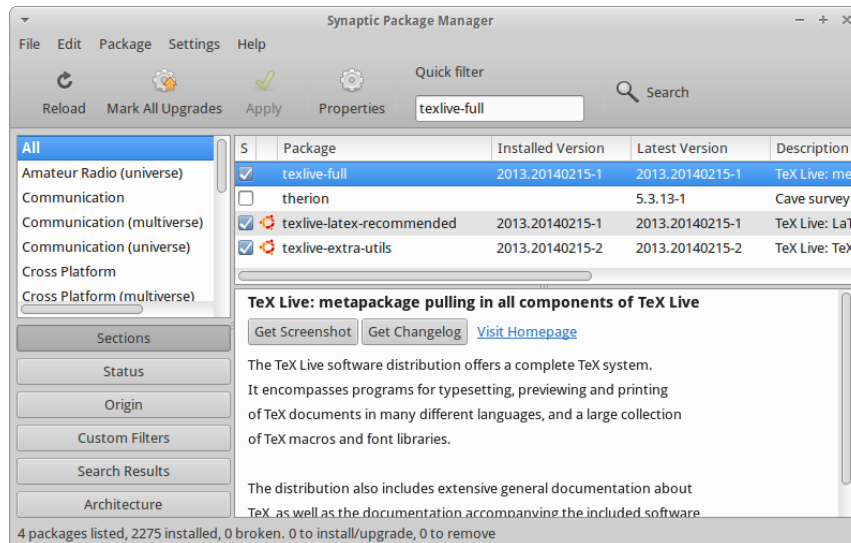
Linux terdiri dari banyak distro/distribusi, ada Ubuntu, Debian, Fedora, dsb. Kebanyakan distro telah mengikutsertakan distribusi TeX Live pada repositorinya. Pada Ubuntu kita bisa menggunakan aplikasi Synaptic Package Manager.

¹<http://en.wikipedia.org/wiki/Donald%20Knuth>

²<http://www.tug.org/texlive/>

³<http://www.miktex.org/>

⁴<http://www.tug.org/mactex/>



Gambar 1.1: Install Texlive-full Menggunakan Synaptic

BAB 2

Dasar

Pada bab ini kita akan mulai belajar dasar-dasar LaTeX, jadi anda harus sudah menginstall aplikasi LaTeX pada komputer. Kami menyarankan menggunakan Texmaker sebagai editor LaTeX. Karena kelengkapan dan juga kemudahan penggunaannya. Sedangkan pada sistem operasi Windows teman-teman bisa menggunakan Texwork.

2.1 Sintak \LaTeX

\LaTeX mengkonversi teks sumber dikombinasikan dengan markup kedalam dokumen dengan kualitas tinggi. File input untuk \LaTeX adalah plain text¹. Bentuk umum paling sederhana adalah sebagai berikut:

```
1 \documentclass{article}
2
3 \begin{document}
4 Hello world!
5 \end{document}
```

2.1.1 Spasi

Kompiler LaTeX menormalisasi spasi, baik itu berupa spasi atau tab. Meski kita membuat spasi sebanyak apapun LaTeX akan menganggapnya sebagai spasi tunggal. Begitu juga dengan ganti baris tunggal juga dianggap sebagai spasi, jika ganti baris ini dilakukan dua kali atau membuat baris kosong antar dua baris. Hal ini baru menjadikan Paragraf baru pada output.

Berikut contoh yang akan membantu anda memahamai bagaimana kerja spasi di LaTeX:

¹<http://en.wikipedia.org/wiki/plain%20text>

```

1 \documentclass[12pt]{article}
2
3 \begin{document}
4 Tidak peduli berapa pun anda
5 memasukkan satu atau beberapa      spasi
6 setelah sebuah kata.
7
8 Sebuah baris kosong memulai sebuah
9 baris baru.
10
11 \end{document}

```

Listing 2.1: spasi.tex

Sehingga output yang didapat dari kode sumber diatas adalah:

Tidak peduli berapa pun anda memasukkan satu atau beberapa spasi
 setelah sebuah kata.
 Sebuah baris kosong memulai sebuah baris baru.

Gambar 2.1: Spasi Pada LaTeX

Untuk selanjutnya silahkan anda beruji coba sendiri yah.. :-). Selanjutnya adalah

2.1.2 Karakter Khusus

Karakter-karakter ini adalah karakter/huruf yang memiliki suatu arti bagi kompiler LaTeX. Jika kita tuliskan karakter-karakter ini secara langsung pada kode, maka akan menghasilkan hal yang tidak sesuai keinginan. Karakter-karakter ini adalah :

\$ % ^ & _ { } ~ \

```

1 \# \$ \% \^{} \& \_ \{ \} \~{} \textbackslash{}

```

Listing 2.2: karakterkhusus.tex

Karakter backslash(\) tidak bisa ditambahkan dengan cara menambahkan backslash lagi setelahnya (\\). Cara ini digunakan untuk mengakhiri baris. Sedangkan pada karakter ^ dan ~ harus ditambahkan kurung kurawal, karena kalau tidak perintah ini \n akan menghasilkan ñ.

2.1.3 Group

Sebuah *group* pada dasarnya didefinisikan dengan sepasang kurung kurawal `{}`. Dan perintah-perintah yang ada didalamnya hanya berkerja pada group tersebut. Perintah `\begin{group}` dan `\end{group}` merupakan perintah alternatif dari kurung kurawal. Contoh:

```
1 {  
2 \bf Yang ini BOLD  
3 }  
4 di sini tidak lagi
```

Yang ini BOLD di sini tidak lagi

group berguna ketika kita menginginkan suatu perintah hanya berpengaruh pada beberapa baris atau kata saja.

2.1.4 Environment

Environment di LaTeX punya aturan yang mirip dengan perintah/command, biasanya environment punya pengaruh lebih luas. Sintaks umumnya:

```
1 \begin{namaenvironment}  
2 di sini teks yang terpengaruh oleh environment  
3 \end{namaenvironment}
```

Diantara `\begin` dan `\end` kita dapat menambahkan perintah lain dan juga bisa nested environment(environment bertumpuk).

2.1.5 Commands

(*Commands*) bersifat case sensitive dan mengikuti dua aturan berikut:

- diawali dengan backslash `\` dan selanjutnya nama command yang hanya berisi hanya huruf. Command diterminasi dengan sebuah spasi, sebuah angka atau apapun selain huruf.
- Berisi sebuah backslash `\` dan tentunya sebuah non-huruf.

Beberapa commands butuh argumen yang diletakkan di dalam kurung kurawal `{}` setelah nama command. Juga terkadang butuh opsi tambahan yang diletakkan di dalam kurung siku `[]`. Sintaks sebagai berikut:

```
1 \namacommand[opsi1,opsi2,...]{argumen1}argumen2}...
```


Kebanyakan perintah standar LaTeX punya *switch*. Switch ini adalah sebuah kesamaan dari command/perintah hanya saja tidak punya argumen dan berada di dalam kurung kurawal. Jangan memanggилnya diluar kurung karena akan berpengaruh ke seluruh dokumen.

Contoh:

```

1 % \emph adalah command/baris perintah, \em adalah switch
2 \emph{text tercetak miring}, bagian ini normal % Format Benar
3 {\em text tercetak miring}, bagian ini normal % Format benar
4
5 \em text tercetak miring, bagian normal % Format salah
6 \emph{text tercetak miring}, bagian normal % Format salah

```

2.1.6 Komentar

Jika pada input file terdapat karakter % maka baris setelahnya tidak ikut dieksekusi, dan tidak berpengaruh pada hasil output. Biasanya digunakan untuk membuat catatan tentang perintah atau keterangan dari dokumen yang dibuat.

```

1
2 Saya sedang % mangan telo
3 % Jadi ini lah komentar, bukan komentator
4 menulis % huruf apa saja ya..?${}^&
5     menggunakan % Pisau tajam
6 \LaTeXe

```

Listing 2.3: komentar.tex

Saya sedang menulis menggunakan L^AT_EX 2_ε

Sebagai catatan bahwa karakter % bisa digunakan untuk memisahkan kata yang panjang tanpa ada spasi atau ganti baris.

2.2 Membuat Dokumen Pertama

Sekarang saatnya membuat dokumen sederhana untuk langkah awal. Kita bisa membuat kode sumber LaTeX ini menggunakan text editor seperti Notepad++, vim, dan gedit. Namun kami sarankan untuk menggunakan editor LaTeX khusus seperti Texwork untuk Windows sedangkan untuk Linux menggunakan Texmaker.

```

1 % assalam.text - Contoh pertama LaTeX
2 \documentclass{article}
3 \begin{document}
4 Assalamualaikum Warahmatulloh... % Salam Rek...! Biar beda :-)

```

```
5 \end{document}
```

Listing 2.4: assalam.tex

2.2.1 Apa Arti dari Kode Sumber?

Mari kita pelajari tiap baris dari kode sumber di atas:

- `% assalam.text` - Contoh pertama LaTeX
Baris pertama adalah baris komentar, baris ini tidak akan nampak dan tidak menjalankan perintah apapun. Bisa digunakan untuk memberi keterangan mengenai isi kode sumber, dan informasi tentang pengarang, dsb.
- `\documentclass{article}`
Ini adalah baris perintah atau command yang memberitahukan LaTeX untuk menggunakan format dokumen `article`.
- `\begin{document}`
Ini adalah gerbang untuk lingkungan dokumen, Isi dokumen diletakkan setelah perintah ini. Sedangkan perintah-perintah yang berada di atasnya dinamakan *preamble*.
- `Assalamualaikum Warahmatulloh... % Salam Rek..! Biar beda :-)`
Nah ini adalah baris yang akan nampak pada hasil akhir.
- `\end{document}`
Ini adalah baris penutup dari perintah `\begin{document}`. Apa pun setelah perintah ini tidak akan dipedulikan.

2.3 File

2.3.1 Membuat Nama File yang Baik

Tidak semua nama file bisa digunakan, ada tata caranya. Jangan menggunakan nama file atau folder menggunakan spasi, kita bisa membuat nama yang panjang tapi hindari spasi. Gunakan huruf (a-z) angka (1-9) dan tanda minus (-) dan sebuah titik untuk pemisah ekstensi file. Jangan mencampur antara huruf besar dan kecil dalam satu nama, karena beberapa sistem operasi tidak memperdulikan namun yang lainnya membedakannya.

2.3.2 File Tambahan

Selain file `.tex` ada beberapa file pendukung lain, file-file ini ada yang bersifat semetara. Berikut ekstensi umum file-file di LaTeX:

Ekstensi	Penjelasan
.aux	File ini digunakan untuk menyimpan informasi yang berhubungan dengan cross-referense
.bbl	File bibliography, merupakan output dari BiBTeX dan digunakan oleh LaTeX
.bib	File database bibliography
.blg	File log dari BiBTeX
.bst	File Tipe/model BiBTeX
.cls	File class yang mendefinisikan seperti apa dokumen anda. Dipilih dengan menggunakan perintah <code>\documentclass</code> .
.dtx	TeX terdokumentasi. Merupakan format distribusi utama dari file style LaTeX.
.ins	File installer untuk file <code>.dtx</code> .
.fd	File yang mendeskripsikan font baru agar dikenali oleh LaTeX.
.dvi	File Device Independent. Ini adalah output utama yang dihasilkan dari kompilasi kode sumber LaTeX menggunakan kompiler <i>latex</i> . Bisa dilihat menggunakan aplikasi DVI previewer atau aplikasi-aplikasi yang mendukungnya.
.pdf	Portable Document Format. Merupakan hasil utama dari kompilasi kode sumber LaTeX menggunakan kompiler <i>pdflatex</i> . Banyak sekali aplikasi untuk menampilkan dokumen dengan ekstensi PDF, salah satunya Adobe Reader, dan Okular.
.log	Memberikan detail laporan selama kompiler berjalan.
.toc	Menyimpan semua header section. Digunakan untuk membuat daftar isi.
.lof	Mirip dengan.toc hanya saja untuk membuat daftar gambar (<i>list of figures</i>).
.lot	Sama juga tapi untuk membuat daftar tabel (<i>list of tables</i>).
.idx	Jika di dalam dokumen terdapat indeks. LaTeX menyimpan kata-kata indeks tersebut di dalam file ini. Diproses dengan menggunakan perintah <code>makeindex</code> .
.ind	File .idx yang telah diproses dan siap untuk dimasukkan ke dalam dokumen pada perputaran kompilasi selanjutnya.
.ilg	File log yang memberitahukan apa yang telah dilakukan oleh <code>makeindex</code> .
.sty	File makro package LaTeX. File ini bisa di-load ke dalam dokumen LaTeX menggunakan perintah <code>\usepackage</code> .
.tex	File Input LaTeX atau TeX.
.out	File paket hyperref, hanya satu digunakan untuk file master.

Bagian II

Elemen-elemen Umum

BAB 3

Struktur Dokumen

Tujuan dari menulis adalah untuk menyampaikan apa yang kita pikirkan, informasi atau pengetahuan kepada pembaca. Pembaca akan mudah memahami jika ide/tulisan kita tertata dengan baik, dalam hal ini hasil tulisan kita terstruktur secara rapi.

LaTeX berbeda dengan sistem penulisan yang lainnya. LaTeX menyediakan kemudahan untuk pengguna menata dokumen mereka dengan berbagai hirarki, termasuk bab, subbab, dan paragraf.

3.1 Struktur Global

Ketika LaTeX memproses kode sumber, dia meminta kode tersebut mengikuti aturan tertentu. Oleh karena itu lah kode sumber ini harus berisi perintah

```
1 \documentclass{...}
2
3 \begin{document}
4 ...
5 \end{document}
```

Wilayah antara `\documentclass{...}` dan `\begin{document}` dinamakan *preamble*. Biasanya berisi perintah yang mempengaruhi keseluruhan dokumen.

Setelah preamble adalah teks dokumen, berada diantara perintah yang menandakan dimulai dan berakhirnya suatu teks dokumen.

```
1 \begin{document}
2 ...      % Disini teks yang akan ditampilkan pada output.
3 \end{document}
```

3.2 Preamble

3.2.1 Document class

Ketika memproses file sumber, LaTeX perlu tahu tipe dokumen yang ingin dibuat penulis. Caranya dengan menggunakan perintah `\documentclass`. Seharusnya diletakkan diawal kode input.

```
1 \documentclass[option]{class}
```

`class` menentukan tipe dokumen yang akan dibuat. Misal LaTeX juga mendukung class dokumen surat(letter) dan presentasi(slide). Untuk saat ini kita hanya menggunakan class article sebagai standar. Sedangkan parameter `option` digunakan untuk merubah perilaku dari class dokumen. Misal saja:

```
1 \documentclass[11pt,twoside,a4paper]{article}
```

perintah ini memberitahukan LaTeX untuk menulis dokumen dengan format artikel dengan ukuran font standar 11 point dengan layout dua sisi dan menggunakan kertas A4.

Berikut daftar class dokumen yang didukung LaTeX:

Class	Keterangan
<code>article</code>	Untuk artikel pada jurnal penelitian sains, presentasi, laporan pendek, dokumentasi program, ajakan, ...
<code>IEEEtran</code>	Artikel dengan format IEEE Transactions.
<code>proc</code>	Class untuk proceeding turunan dari class article.
<code>minimal</code>	Untuk dokumen minimalis, hanya terdiri dari ukuran halaman dan font dasar. Terutama digunakan untuk tujuan debugging.
<code>report</code>	Untuk bentuk laporan panjang, berisi beberapa bab seperti buku kecil, tesis,...
<code>book</code>	Untuk membuat dengan format buku secara utuh.
<code>slides</code>	Untuk bentuk slide. Class ini menggunakan font sans serif ukuran besar.
<code>memoir</code>	Untuk mengubah kepantasan output dari dokumen. Turunan dari class buku, tapi kita bisa membuat berbagai macam dokumen dengannya. http://www.ctan.org/tex-archive/macros/latex/contrib/memoir/memman.pdf
<code>letter</code>	Untuk menulis surat.
<code>beamer</code>	Untuk menulis presentasi

Pada umumnya class document punya banyak opsi/option. Beberapa ada yang punya opsi berbeda atau bahkan tidak sama sekali. Untuk class dari pihak ketiga

biasanya telah disertai dokumentasi. Pada tabel di bawah ini adalah opsi-opsi yang paling banyak digunakan dalam class:

Opsi	Penjelasan
10pt, 11pt, 12pt	Merubah ukuran utama font, jika tidak ditentukan maka nilai defaultnya adalah 10pt.
a4paper, letterpaper,...	Menentukan ukuran kertas, ukuran standar adalah letterpaper. Selain itu bisa juga menggunakan ukuran a5paper, b5paper, executivepaper, dan legalpaper.
fleqn	Menampilkan formula rata kiri bukannya diten-gah.
leqno	Menampilkan penomoran formula pada sebelah kiri bukannya dikanan.
titlepage, notitlepage	Menentukan apakah halaman baru harus dibuat setelah judul dokumen atau tidak. Normalnya class article tidak memulai halaman baru, sedan-gkan class book dan report sebaliknya.
twocolumn	Memerintahkan LaTeX untuk mengeset dokumen dalam bentuk dua kolom bukannya satu.
twoside, oneside	Menentukan apakah output untuk halaman dua sisi atau satu sisi. Class article dan report normalnya mengguanan mode satu sisi sedan-gkan class book menggunakan dua sisi.
openright, openany	Membuat bab dimulai dari sebelah kanan hala-man atau pada halaman selanjutnya. Ini tidak berfungsi pada article, karena class article tidak mengenal bab. Pada class report dimulai pada halaman selanjutnya sedangkan pada book pada halaman sebelah kanan.
landscape	Merubah layout dokumen untuk diprint dalam mode landscape.
draft	Membuat LaTeX mengindikasikan masalah per-ataan dan jarak antar akhir kata pada kotak ke-cil di sebelah kanan margin. Dan juga memberi penekanan dalam penyertaan gambar.

3.2.2 Package

Tidak semua bisa diselesaikan oleh LaTeX murni. Jika kita ingin menam-bahkan gambar, teks berwarna, atau sumber kode dari suatu file ke dalam doku-men, kita harus meningkatkan kemampuan LaTeX. Cara meningkatkannya ini lah dengan menggunakan *package*. Package diaktifkan dengan menggunakan perintah

```
1 \usepackage[opsi]{package}
```

dimana package adalah nama paket-nya, dan opsi adalah kata kunci untuk mengakses fitur-fitur pada paket yang dipanggil. Misal kita panggil package `color` untuk merubah warna teks:

```
1 \documentclass[11pt,oneside,a4paper]{report}
2
3 \usepackage{color}
4 \begin{document}
5 ...
6 \end{document}
```

Bisa juga menambahkan beberapa package dalam satu perintah `\usepackage` dengan memisahkan nama package dengan koma (.). Seperti ini:

```
1 \usepackage{package1,package2,package3}
```

namun jika kita memiliki opsi pada package hal ini tidak berlaku. Perintah `\usepackage` harus dipisah dari perintah `\usepackage` lain. Misal kita menggunakan package `geometry`:

```
1 \documentclass[11pt,a4paper,oneside]{report}
2
3 \usepackage{pslatex,palatino,avant,graphicx,color}
4 \usepackage[margin=2cm]{geometry}
5
6 \begin{document}
7 \title{\color{red}Belajar LaTeX}
8 \author{\color{blue}Isro Hidayat}
9 \date{\color{green}Oktober 2015}
10 \maketitle
11
12 \end{document}
```

Listing 3.1: package.tex

Hasilnya adalah sebagai berikut.

Belajar LaTeX

Isro Hidayat

Oktober 2015

Gambar 3.1: package.tex

3.3 Lingkungan *Document*

3.3.1 Top matter

Pada bagian paling awal dokumen ada informasi mengenai dokumen itu sendiri, judul dan tanggal juga informasi mengenai penulis seperti nama, alamat dan email. Informasi ini lah yang dinamakan dengan *top matter*

Contoh sederhana:

```
1 \documentclass[11pt,a4paper]{report}
2
3 \begin{document}
4 \title{Belajar LaTeX}
5 \author{Isro' Hidayat}
6 \date{Oktober 2015}
7 \maketitle
8 \end{document}
```

Perintah `\title` dan `\author` biasanya diharuskan untuk diisi. Perintah `\date` saja yang diketikkan, maka tanggal hari saat itu lah yang ditampilkan. Sedangkan perintah `\maketitle` adalah penutup dari bagian top matter ini.

Ini adalah contoh yang lebih kompleks:

```
1 \title{Cara Membuat Dokumen Menggunakan \LaTeX{}}
2 \author{Isro Hidayat\\
3 Sekolah Rakyat\\
4 Universitas Rakyat Kecil\\
5 Buku\\
6 Indonesia\\
7 RN 98899\\
8 \texttt{orsisam@gmail.com}}
9 \date{\today}
10 \maketitle
```