

## **SURVEY METHODOLOGY**



---

# **SURVEY METHODOLOGY**

## **This is the Subtitle**

---

**Robert M. Groves**

Universitat de les Illes Balears

**Floyd J. Fowler, Jr.**

University of New Mexico



**A JOHN WILEY & SONS, INC., PUBLICATION**

Copyright ©2007 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.  
Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the web at [www.copyright.com](http://www.copyright.com). Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008.

**Limit of Liability/Disclaimer of Warranty:** While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department with the U.S. at 877-762-2974, outside the U.S. at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic format.

***Library of Congress Cataloging-in-Publication Data:***

Survey Methodology / Robert M. Groves . . . [et al.].  
p. cm.—(Wiley series in survey methodology)  
“Wiley-Interscience.”  
Includes bibliographical references and index.  
ISBN 0-471-48348-6 (pbk.)  
1. Surveys—Methodology. 2. Social sciences—Research—Statistical methods. I. Groves, Robert M. II. Series.

HA31.2.S873 2007  
001.4'33—dc22 2004044064  
Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

*To my parents*

## CONTRIBUTORS

---

MASAYKI ABE, Fujitsu Laboratories Ltd., Fujitsu Limited, Atsugi, Japan

L. A. AKERS, Center for Solid State Electronics Research, Arizona State University,  
Tempe, Arizona

G. H. BERNSTEIN, Department of Electrical and Computer Engineering, University  
of Notre Dame, Notre Dame, South Bend, Indiana; formerly of Center for Solid  
State Electronics Research, Arizona State University, Tempe, Arizona



# CONTENTS IN BRIEF

---

<b>1</b>	<b>Installation</b>	<b>1</b>
<b>2</b>	<b>Your First Document</b>	<b>17</b>
<b>3</b>	<b>Structuring Your Document (Section and Paragraph)</b>	<b>33</b>
<b>4</b>	<b>Packages Explained</b>	<b>35</b>
<b>5</b>	<b>Typesetting Math in Latex</b>	<b>37</b>
<b>6</b>	<b>Adding a Picture</b>	<b>39</b>
<b>7</b>	<b>Generate a Table of Contents</b>	<b>51</b>
<b>8</b>	<b>Adding Bibliography</b>	<b>53</b>
<b>9</b>	<b>Adding Footnotes</b>	<b>55</b>
<b>10</b>	<b>Create Tables with Latex</b>	<b>57</b>
<b>11</b>	<b>Using Tables the Smart Way</b>	<b>59</b>
<b>12</b>	<b>Plots Visualizing Your Data With Pgfplots</b>	<b>61</b>
<b>13</b>	<b>Electric Circuit With Circuitikz</b>	<b>63</b>
<b>14</b>	<b>Source Code Hightlighting in Latex using the Listing Package (Listing)</b>	<b>65</b>





# CONTENTS

---

List of Figures	xi
List of Tables	xiii
Foreword	xv
Preface	xvii
Acknowledgments	xix
Acronyms	xxi
Glossary	xxiii
List of Symbols	xxv
Introduction	xxvii
<i>Catherine Clark, PhD.</i>	
References	xxvii
<b>1 Installation</b>	<b>1</b>
<b>2 Your First Document</b>	<b>17</b>
	<b>ix</b>

<b>3</b>	<b>Structuring Your Document (Section and Paragraph)</b>	<b>33</b>
<b>4</b>	<b>Packages Explained</b>	<b>35</b>
<b>5</b>	<b>Typesetting Math in Latex</b>	<b>37</b>
<b>6</b>	<b>Adding a Picture</b>	<b>39</b>
<b>7</b>	<b>Generate a Table of Contents</b>	<b>51</b>
<b>8</b>	<b>Adding Bibliography</b>	<b>53</b>
<b>9</b>	<b>Adding Footnotes</b>	<b>55</b>
<b>10</b>	<b>Create Tables with Latex</b>	<b>57</b>
<b>11</b>	<b>Using Tables the Smart Way</b>	<b>59</b>
<b>12</b>	<b>Plots Visualizing Your Data With Pgfplots</b>	<b>61</b>
<b>13</b>	<b>Electric Circuit With Circuitikz</b>	<b>63</b>
<b>14</b>	<b>Source Code Hightlighting in Latex using the Listing Package (Listing)</b>	<b>65</b>
	References	67
	References	69

## LIST OF FIGURES

---



## LIST OF TABLES

---



# FOREWORD

---

This is the foreword to the book.





# PREFACE

---

This is an example preface. This is an example preface. This is an example preface.  
This is an example preface.

R. K. WATTS

*Durham, North Carolina*  
*September, 2007*



## ACKNOWLEDGMENTS

---

From Dr. Jay Young, consultant from Silver Spring, Maryland, I received the initial push to even consider writing this book. Jay was a constant “peer reader” and very welcome advisor during this year-long process.

To all these wonderful people I owe a deep sense of gratitude especially now that this project has been completed.

G. T. S.



## ACRONYMS

---

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association



## GLOSSARY

---

NormGibbs	Draw a sample from a posterior distribution of data with an unknown mean and variance using Gibbs sampling.
pNull	Test a one sided hypothesis from a numerically specified posterior CDF or from a sample from the posterior
sintegral	A numerical integration using Simpson's rule





# SYMBOLS

---

- $A$  Amplitude
- $\&$  Propositional logic symbol
- $a$  Filter Coefficient
- $\mathcal{B}$  Number of Beats



# INTRODUCTION

---

CATHERINE CLARK, PHD.

Harvard School of Public Health  
Boston, MA, USA

The era of modern began in 1958 with the invention of the integrated circuit by J. S. Kilby of Texas Instruments [1]. His first chip is shown in Fig. I. For comparison, Fig. I.2 shows a modern microprocessor chip, [4].

This is the introduction. This is the introduction. This is the introduction. This is the introduction. This is the introduction. This is the introduction.

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$

## REFERENCES

1. J. S. Kilby, "Invention of the Integrated Circuit," *IEEE Trans. Electron Devices*, **ED-23**, 648 (1976).
2. R. W. Hamming, *Numerical Methods for Scientists and Engineers*, Chapter N-1, McGraw-Hill, New York, 1962.
3. J. Lee, K. Mayaram, and C. Hu, "A Theoretical Study of Gate/Drain Offset in LDD MOSFETs" *IEEE Electron Device Lett.*, **EDL-7**(3). 152 (1986).



# CHAPTER 1

---

## INSTALLATION

---

Latex merupakan system pengaturan cara penyetikan dokumen. Latex adalah perangkat lunak yang dapat di download tanpa harus membayar. Sejarahnya, Donald Knuth (Stanford University) mulai mengembangkan sistem pemrosesan dokumen yang disebut "Tex dan Metafont" pada tahun 1977. Knut mengungkapkan bahwa Tex adalah sistem penyetikan dengan tujuan untuk membuat buku yang "cantik", khususnya yang terdiri atas formula-formula matematis. Hingga sekarang, Latex telah berkembang dan project saat ini adalah Latex3.

Biasanya Latex digunakan untuk mengetik dokumen yang berisi formula-formula matematis. Jenis dokumen yang dihasilkan dapat berupa artikel, buku, tesis, disertasi, hingga surat bisnis maupun surat pribadi. Berbeda dengan MS Word atau Libre Office, dokumen hasil dari Latex dapat berbeda antara input dengan output. Misalnya di MS Word kita mengetik "\$a\$" maka yang muncul di dokumen hasil adalah "\$a\$", sedangkan dengan input yang sama, Latex memberikan output a. Karena ide dari Latex adalah membiarkan penulis menulis dokumen dan menyerahkan desain dokumen ke "document designer", maka hasil dari dokumen Latex lebih "cantik".

### Instalasi LaTeX

Ada banyak editor untuk latex dan tidak semua editor sesuai dengan setiap pengguna. Semua itu tergantung pada selera masing-masing. Karena alasan ini, akan saya tunjukkan bagaimana dasar-dasar latex berjalan. Banyak pengguna memilih MikTeX untuk Windows karena MikTeX sudah mempunyai semuanya yang diperlukan untuk menjalankan program latex.

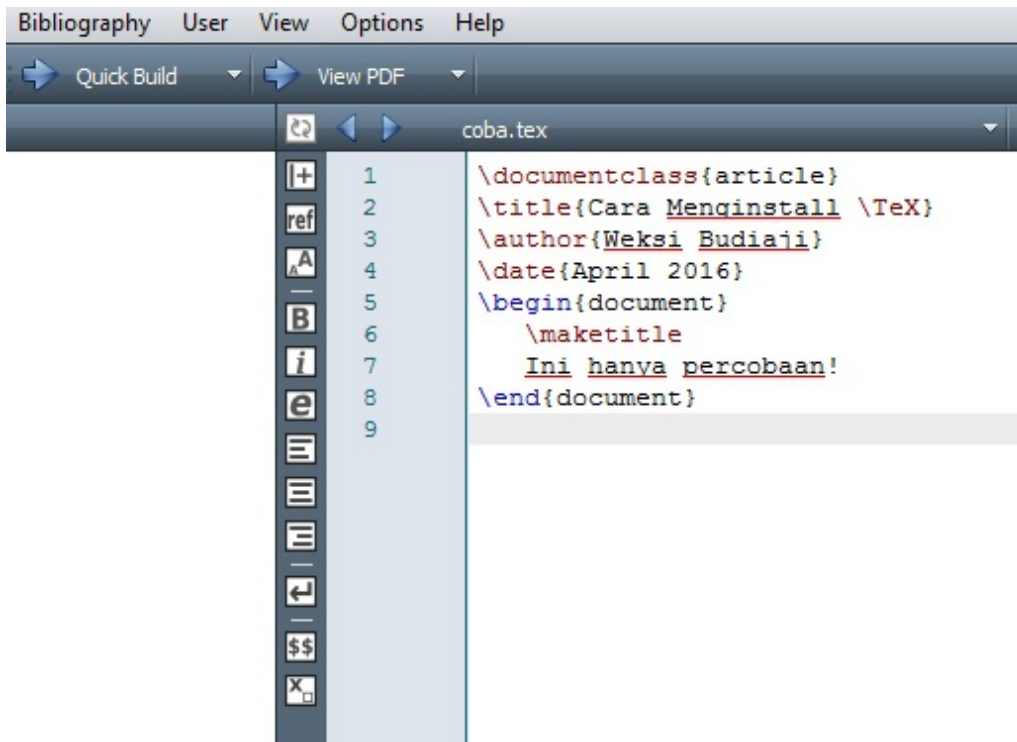
### Linux

Jika menggunakan linux, maka Anda bisa menggunakan paket texlive di beberapa repository. Setelah itu Anda bisa menggunakan berbagai macam text editor dan menjalankan file dengan format .tex dengan perintah pdflatex.

### Windows

Langkah pertama untuk pengguna windows yaitu :

1. Install tex compiler. Tex compiler yang sering digunakan yaitu Miktex dan Texlive. Kedua perangkat lunak ini merupakan software yang paling banyak digunakan meskipun banyak perangkat lunak yang lain. Miktex yang diinstall pertama kali hanya memiliki paket dasar saja. Paket-paket tambahan lain dapat diinstall kemudian sesuai kebutuhan. Berbeda dengan Texlive, saat menginstall Texlive maka semua paket yang ada akan diinstal meskipun belum tentu kita pakai. Sehingga instalasi Texlive lebih lama (kurang lebih 20 menit pada komputer dengan processor pentium i3 dan RAM 6 GB) daripada Miktex. Pilihan Miktex atau Texlive tergantung dari ruang hardisk yang ada, jika ruang hardisk masih banyak tidak ada salahnya menginstall Texlive. Saat ini saya menggunakan Miktex (OS Windows) dan Texlive (Unix). Keduanya dapat didownload di Miktex website dan Texlive website (pilih salah satu).
2. Install Tex Editor. Tex editor yang tidak berbayar juga banyak. Baik di windows maupun unix, bisa menggunakan texmaker sebagai tex editor. Perangkat lunak tersebut bisa di download melalui websitenya. Didalam texmaker inilah nantinya untuk menulis dokumen latex.
3. Langkah terakhir yaitu memulai menggunakan latex. Buka texmaker, pilih file kemudian pilih new, kemudian tulis seperti contoh dibawah ini:



4. Setelah menulis file latex seperti diatas, maka save file. Misalkan save dengan nama 'coba'. Kemudian klik tanda panah "quick build" atau bisa dengan tekan tombol F6. Untuk melihat hasil pdf nya bisa dengan menekan tombol F7.

### Macam- macam Editor Latex

#### 1. Emacs dengan AUCTeX

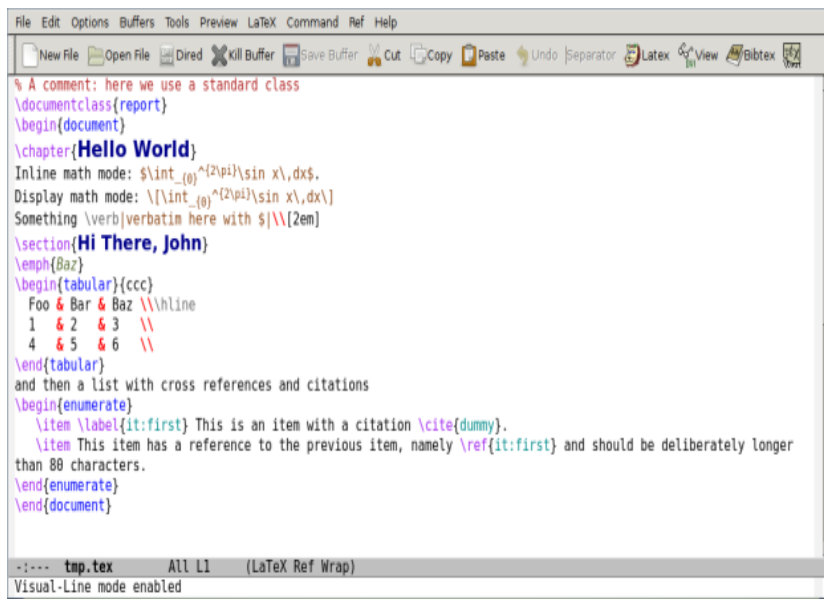
- OS: Windows, Mac (termasuk fork Aquamacs), Unix
- Lisensi: Free software (GPL)
- Bahasa: de, dk, fr, is, it, jp, nl, pl, se, sk didukung oleh AUCTeX
- Unicode: Ya, sejak Emacs 23
- RTL/bidirectional support: sejak Emacs 24, melalui bidi-mode
- % !TeX directives: Tidak, tetapi Emacs memiliki beberapa realisasi untuk file local variables
- Syntax highlighting: Ya, bisa diatur lewat customize and Elisp
- Code completion: Ya, via Emacs Predictive Completion, yang mendukung AUCTeX tanpa konfigurasi lebih lanjut
- Code folding: Ya



## 4 INSTALLATION

- Spell checking: Ya
- SyncTeX: Ya
- Built-in output viewer: Ya
- Project management: org-mode, reftex-mode

Emacs adalah salah satu editor tertua, yang mendukung mode penyuntingan LaTeX, ConTeXt, dan Plain TeX, AUCTeX dan paket untuk mengelola kode-kode sumber, RefTeX.



The screenshot shows the Emacs editor interface with a menu bar (File, Edit, Options, Buffers, Tools, Preview, LaTeX, Command, Ref, Help) and a toolbar. The main text area contains LaTeX source code for a document. The code includes comments and various LaTeX commands such as `\documentclass{report}`, `\begin{document}`, `\chapter{Hello World}`, `\section{Hi There, John}`, `\emph{Baz}`, `\begin{tabular}{ccc}`, `\end{tabular}`, `\begin{enumerate}`, `\item`, `\cite{dummy}`, `\ref{it:first}`, `\end{enumerate}`, and `\end{document}`. The status bar at the bottom indicates the file is `tmp.tex`, the mode is `ALL L1 (LaTeX Ref Wrap)`, and `Visual-Line mode enabled`.

```
% A comment: here we use a standard class
\documentclass{report}
\begin{document}
\chapter{Hello World}
Inline math mode:  $\int_0^{2\pi} \sin x, dx$ .
Display math mode: 
$$\int_0^{2\pi} \sin x, dx$$

Something \verb|verbatim| here with  $\|2em$ 
\section{Hi There, John}
\emph{Baz}
\begin{tabular}{ccc}
Foo & Bar & Baz \\
1 & 2 & 3 \\
4 & 5 & 6 
\end{tabular}
and then a list with cross references and citations
\begin{enumerate}
\item \label{it:first} This is an item with a citation \cite{dummy}.
\item This item has a reference to the previous item, namely \ref{it:first} and should be deliberately longer
than 80 characters.
\end{enumerate}
\end{document}
```

RefTeX membuat seluruh referensi Anda mudah ditemukan layaknya C-c <key>, baik untuk BibTeX maupun biblatex, dan ia memiliki pintasan (shortcut key) pula untuk bernavigasi di antara bagian-bagian dokumen dengan menggunakan C-c =: secara default.

```

TABLE-OF-CONTENTS on ~/research/references/sequences/sequences.tex
SPC=view TAB=goto RET=goto+hide [q]uit [r]escan [l]abels [f]ollow [x]r [?]Help
-----
1 Chloroplast
  1.1 Coding regions
    1.1.0.1 \emph{rbcl}
    1.1.0.2 \emph{atpB}
    1.1.0.3 \emph{matK}
    1.1.0.4 \emph{ndhF}
    1.1.0.5 \emph{ndhF-rpl32-trnL}
    1.1.0.6 16S rDNA
    1.1.0.7 \emph{rps2}
    1.1.0.8 \emph{rps4}
---- *toc* Wed Feb 20 09:17 0.40 Mail (TOC) L< I< T<ALL> -----

\section{Chloroplast}

From \citet{soltis_soltis_1998}: a circular molecule, divided by two
inverted repeat segments, separating a large and a small region of
single-copy genes. 120-200kb. Relatively conservative, but with
inter-region variability in evolutionary rates. Maybe too conservative
for lower-level studies, although with some contrary evidence (and more
since this was published). Another problem is potential chloroplast
transfer between lineages, so-called chloroplast capture.

U:--- sequences.tex 3% L20 (LaTeX/P Ref Fly hl-p yas Wrap Fill) Wed Feb 20 09:17 0.40 Mail

```

(Tema warna dapat dikonfigurasi sebebas mungkin)

AUCTeX mendukung multi-file parsing, sehingga dokumen-dokumen besar dengan perintah `\input` atau `\include` mudah dikompilasikan dengan `C-c C-c` pada berkas yang bersangkutan. Tidak perlu lagi kembali ke master file hanya untuk mengompilasi.

Fitur AUCTeX *preview-latex* adalah pratyayang WYSIWYG untuk rumus-rumus. Fitur-fitur terkemuka Emacs:

- Menggunakan table-insert bersama dengan fungsi table-generate-source dan table-recognize-\* untuk membuat tabel-tabel dengan mudah.
- Banyak sekali shortcut key tersedia
- Terdokumentasi dengan baik, baik Emacs itu sendiri melalui manual Emacs dan manual AUCTeX Texinfo, maupun melalui banyak buku dalam beberapa bahasa.

## 2. Vim dengan LaTeX-suite

- OS: Windows, Mac, Linux, BSD, dan lain-lain
- Lisensi: Open Source Charityware
- Bahasa: ?
- Unicode: Ya

- RTL/bidi support: sebagian
- % !TEX directives: Tidak, tetapi memiliki modelines
- Syntax Highlighting: Ya, bisa dikustomisasi
- Code Completion: Ya (menggunakan Omni Completion, bisa diperluas dengan plugin SnipMate)
- Code Folding: Ya
- Spell Checking: Ya
- SyncTeX: Ya, lihat pertanyaan ini
- Built-in Output Viewer: Tidak
- Project Management: ?
- Jika Anda benar-benar kelas berat, Anda akan selalu menggunakan Vim. Ada banyak macro yang dibuat untuk Vim untuk membantu menyunting berkas LaTeX.

```

% A comment: here we use a standard class
\documentclass{report}
\begin{document}
\chapter{Hello World}
Inline math mode:  $\int_0^{2\pi} \sin x \, dx$ .
Display math mode: 
$$\int_0^{2\pi} \sin x \, dx$$

Something \verb|verbatim| here with  $\|$ 
\section{Hi There, John}
\emph{Baz}
\begin{tabular}{ccc}
Foo & Bar & Baz \\ \hline
1 & 2 & 3 \\
4 & 5 & 6 
\end{tabular}
and then a list with cross references and citations
\begin{enumerate}
\item \label{it:first} This is an item with a citation \cite{dummy}.
\item This item has a reference to the previous item, namely \ref{it:first} and should be deliberately longer than 80 characters.
\end{enumerate}
\end{document}

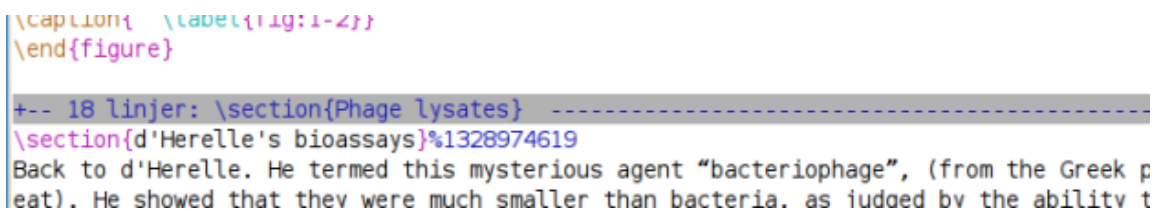
```

Anda dapat melakukan word/command completion melalui <C-P> dan <C-N>, untuk memilih saran sebelumnya atau sesudahnya.

Ada versi Vim dengan menu-menu grafis, yang bernama gVim. Jika ia digunakan dengan Latex-suite, maka banyak perintah TeX ditampilkan di menubar untuk mempercepat penyuntingan.

### Fitur-Fitur

Vim juga memiliki fitur code-folding, karena paket vim-late menawarkan code-folding otomatis. Folding juga bisa dilakukan secara manual berdasarkan kunci (misalnya {{{ dan }}}) untuk membuka dan menutup fold otomatis. Contoh folds bisa dilihat pada gambar berikut:



```

\caption{ \label{fig:1-4}}
\end{figure}

+-- 18 linjer: \section{Phage lysates} -----
\section{d'Herelle's bioassays}%1328974619
Back to d'Herelle. He termed this mysterious agent "bacteriophage", (from the Greek p
eat). He showed that they were much smaller than bacteria. as iudged by the abilitv t

```

Fitur Vim masih sangat banyak. Namun di dalam tulisan ini, yang bisa disebutkan adalah:

#### VIM

##### Regex

- Perintah dan pintasan kibor yang powerful
- Sangat bisa dikustomisasi
- Smart Indenting

#### LaTeX-Suite

- Panggil cepat kompilator dengan \ll; tayangkan hasil dengan \lv
- Environments dapat diakses dengan tiga huruf dalam insert mode:

EEQ = environment persamaan

EFI = environment gambar (figure)

- Place-holders (<+text+>) dapat dilompati dengan Ctrl-J tanpa meninggalkan insert mode
- Inverse searching: klik ganda penampil PDF dan Anda lompat ke baris kode sumber tex yang bersesuaian.

### 3. Texmaker - texmaker

- Platforms: Windows XP/Vista/7/8, OS X 10.5+, Linux

- License: GPL, gratis
- Languages: cs, de, el, en, es, fa, fr, gl, hu, it, nl, pl, pt, pt (bra), ru, se, sr, zh (cn), zh (tw)
- Unicode: Ya
- RTL/bidi: ?
- % !TEX directives: Tidak
- Syntax Highlighting: Ya, bisa dikustomisasi
- Code Completion: Ya, bisa dikustomisasi
- Code Folding: Ya
- Spell Checking: Ya
- SyncTeX: Ya
- Built-in Output Viewer: Ya, mendukung PDF

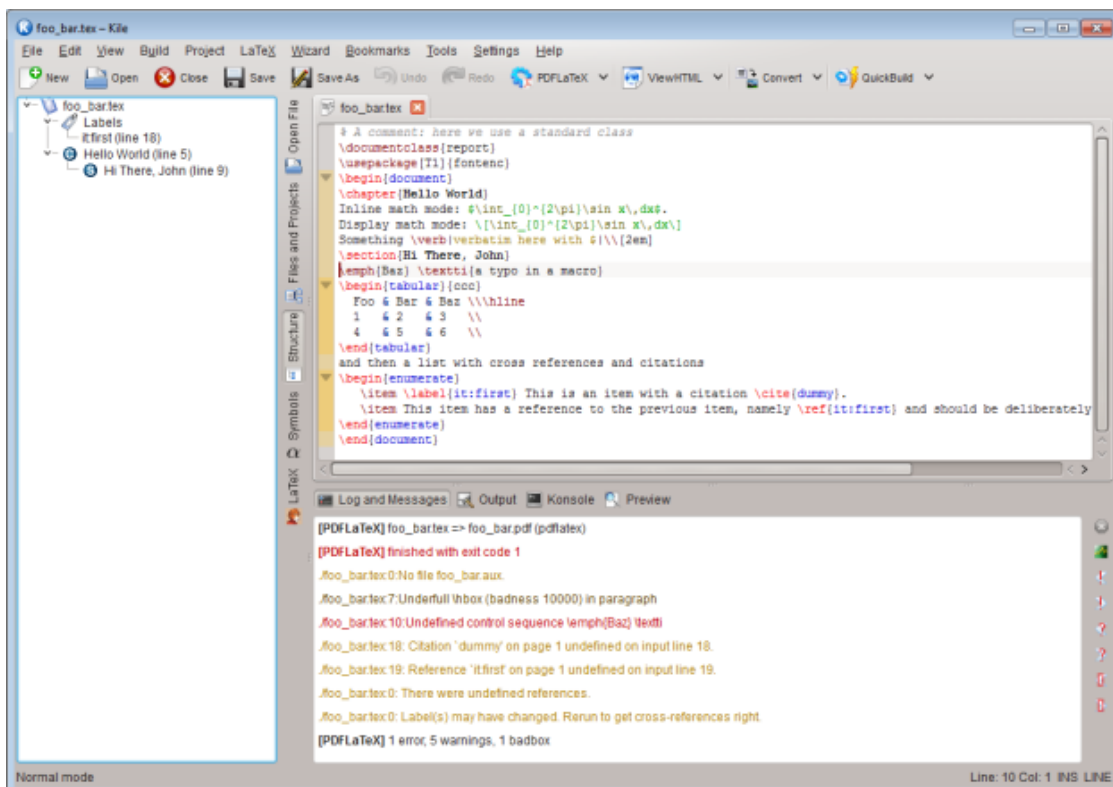
#### 4. TeXworks - texworks

- OS: Windows XP/Vista/7/8, OS X, Linux
- Lisensi: GPL
- Bahasa: en, af, ar, ca, cs, de, fa, fo fr, it, ja, nl, ko, pl, pl, ru, sl, tr zh
- Unicode: Ya
- RTL/bidi: Ya
- % !TEX directives: Ya
- Syntax Highlighting: Ya, regex-based
- Code Completion: Ya, bisa dikustomisasi berdasarkan daftar 'known entry'
- Code Folding: Tidak
- Spell Checking: Ya, tetapi harus diinstal sendiri
- SyncTeX: Ya
- Built-in Output Viewer: Ya, PDF (Poppler-based)
- Project Management: Tidak
- Di Windows dan Linux, saya menggunakan TeXworks, yang menyediakan jendela editor kode dan pratayang. Klik pada pratayang dokumen akan langsung menandai kode LaTeX yang bersesuaian.

#### 5. Kile - kile

- OS: Linux, Windows1 (XP, Vista, 7)
- Lisensi: GNU GPL 2
- Bahasa: bg, bs, ca, cs, da, de, el, en\_GB, eo, es, et, fi, fr, ga, gl, hi, hne, hu, it, ja, kk, lt, mai, ms, nb, nds, nl, nn, pl, pt, pt\_BR, ro, ru, sk, sv, tr, ug, uk, zh\_CN, zh\_TW

- Unicode: Ya
- RTL/bidi: Ya
- % !TEX directives: Tidak2
- Syntax Highlighting: Ya, bisa dikustomisasi
- Code Completion: Ya, bisa dikustomisasi
- Code Folding: Ya
- Spell Checking: Ya
- SyncTeX: Ya (namun flag -synctex=1 harus ditambahkan secara manual pada build engine)
- Built-in Output Viewer: Terbatas3 (pratayang PNG dari sebagian kode - misalnya environment yang dipilih - dikonversikan dari DVI/PS/PDF)
- Project Management: Ya

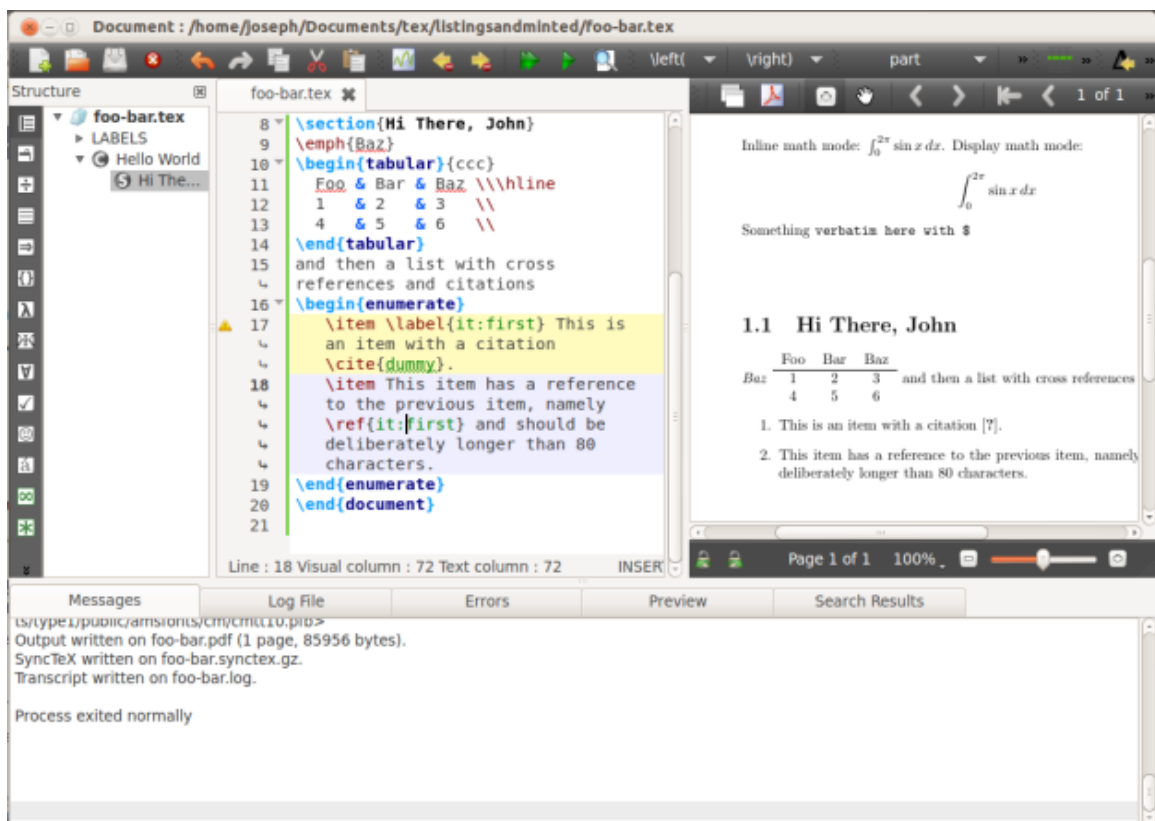


## 6. TeXstudio - texstudio

- OS: Windows XP/Vista/7, OS X, Linux, FreeBSD

## 10 INSTALLATION

- Lisensi: GPL v2
- Bahasa: cs, de, en, es, fr, hu, ja, pt\_BR, zh\_CN
- Unicode: Ya
- RTL/bidi: ?
- % !TeX directives: Ya
- Syntax Highlighting: Ya, bisa dikustomisasi
- Code Completion: Ya, bisa dikustomisasi dan auto-customized
- Code Folding: Ya
- Spell Checking: Ya
- SyncTeX: Ya
- Built-in Output Viewer: Ya, mendukung PDF
- Project Management: Ya
- Saya merekomendasikan TeXstudio sebagai fork yang menarik dari Texmaker yang saya rasa lebih nyaman dan bisa dikustomisasi.
- Berikut ini screenshot TeXStudio.



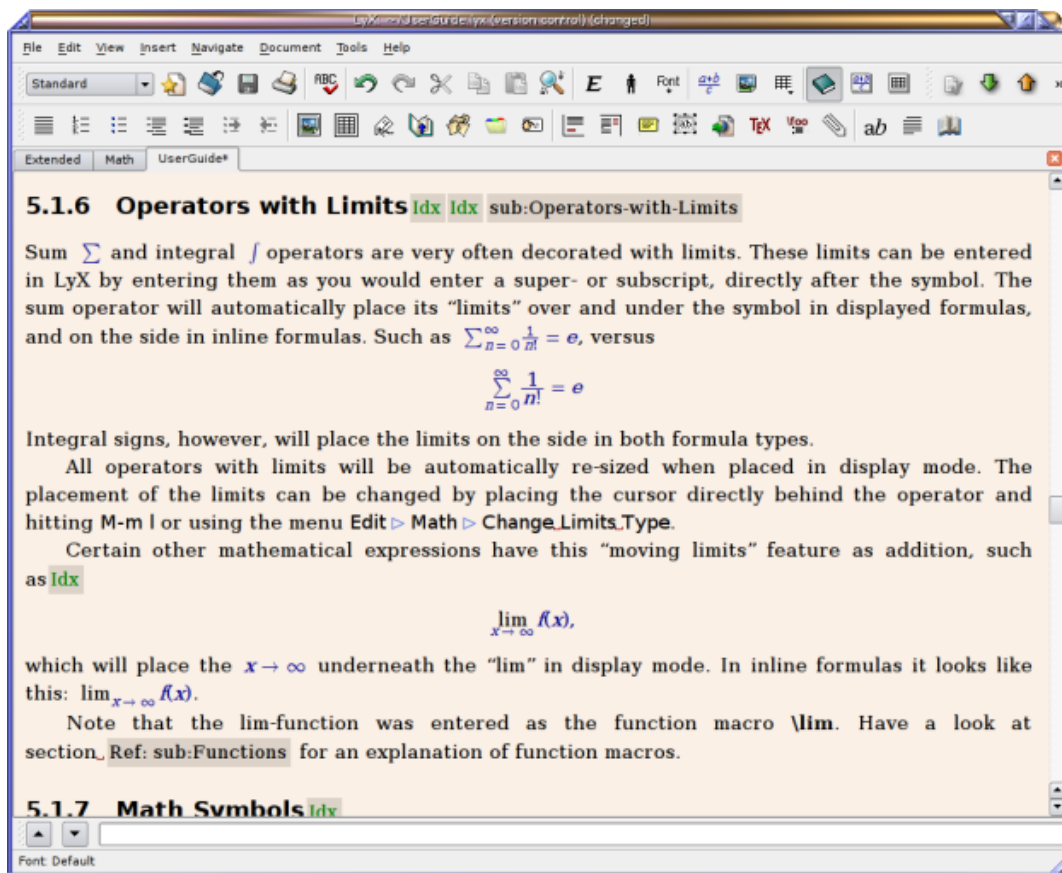
Fitur-fitur lainnya:

- cross-platform
- dukungan penulisan (incremental search, folding, navigasi, auto-completion, custom macros)
- syntax highlighting
- inline interactive spell-checking
- mendukung program-program LaTeX utama, termasuk tikz, pstricks, dan lain-lain
- multi-views: math, structure
- dukungan SVN
- bisa berjalan via USB flash disk
- mendukung syntex
- termasuk penampil PDF, tetapi masih bisa dikonfigurasi untuk memakai viewer eksternal (juga dengan syntex)
- developer dan komunitas yang sangat aktif dan responsive

## 7. LyX

- OS: Windows, Mac, and Linux
  - Lisensi: Open Source
  - Sangat intuitif dan ramah pengguna, dan bisa impor/ekspor ke LaTeX.
  - Terlalu banyak fitur untuk disebutkan, paling bagus: Jika Anda ingin menulis rumus matematika "2-dimensional", LyX cocok untuk itu.





## 8. Sublime Text dengan LaTeX Plugin OS: Windows, Mac, Linux

Ini adalah editor yang sederhana tetapi powerful. Sublime Text mirip Notepad++, tetapi tersedia untuk banyak platform dan sangat mudah diatur untuk LaTeX dengan plugin LaTeXTools atau LaTeXing -keduanya tersedia dari Package Control. Sublime juga mirip TextMate, tetapi dikembangkan lebih aktif dan memiliki komunitas yang besar yang menyediakan plugin-nya. Sublime juga lebih cantik daripada keduanya.

Perhatikan bahwa ini adalah software berbayar, dan meminta lisensi selama periode evaluasi (seharga USD 70). Dimungkinkan untuk menjalankan Sublime Text tanpa membeli lisensi, tetapi Anda akan terus diingatkan bahwa Anda menggunakan salinan yang belum diregistrasikan.

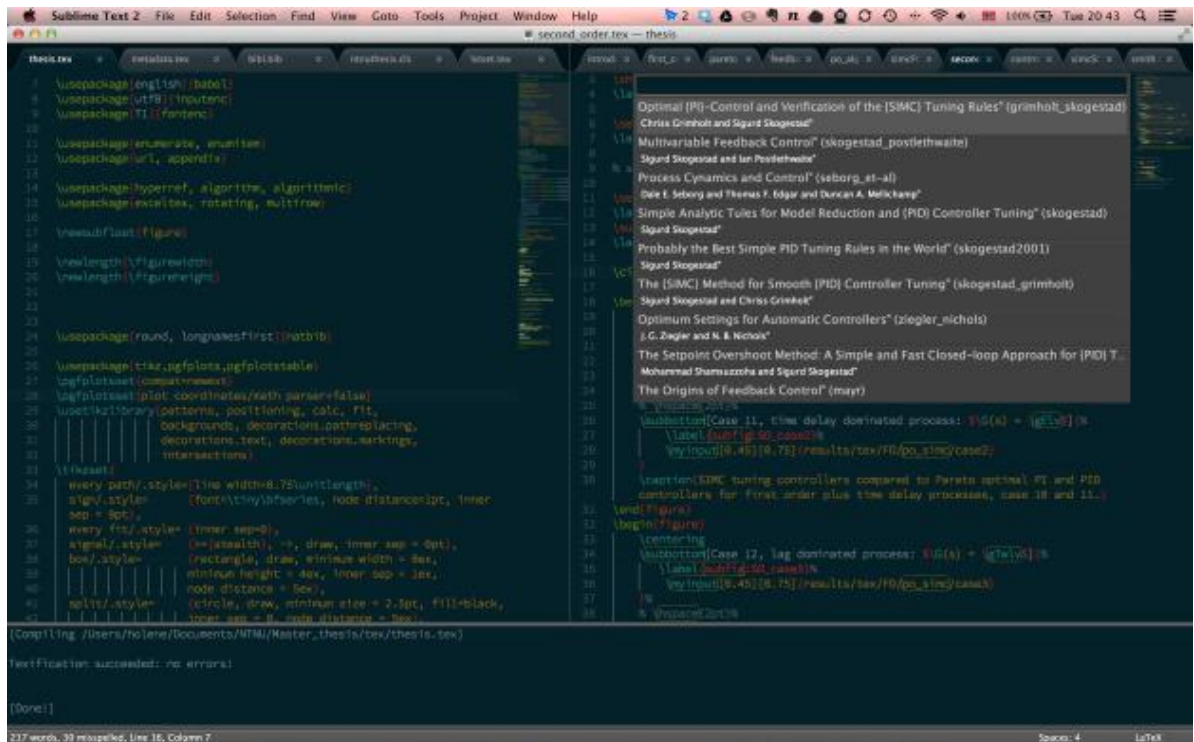
Sublime Text memiliki peralatan yang canggih untuk mengetik, yang Anda tidak mau tinggalkan ketika bekerja dengannya:

- multiple cursors
- go-to ke mana saja
- snippets
- incremental find
- manajemen proyek
- build-systems yang banyak

dan banyak lagi (lihat Perfect Workflow in Sublime Text 2). Skrinsot di bawah juga menampakkan fiturnya untuk menemukan sitas-sitasi (citations) dari Bib-Tex.

Sublime Text ini editor yang hampir sempurna, dengan potensi yang hampir tidak terbatas. Daftar fiturnya panjang sekali. Instal Package Manager, dan paket-paket tambahan dari repositori bisa dipasang dalam beberapa detik saja.

- OS: Windows, Unix
- Lisensi: Free to try, free to buy
- % !TEX directives: Ya
- Syntax highlighting: Ya
- Code completion: Ya
- Code folding: Ya
- Spell check: Ya, baik built-in maupun dengan plugin
- SyncTeX: Ya
- Built-in output viewer: Tidak
- Project management: Ya



## 9. TeXlipse

- OS: Windows, Mac, Linux and others (Java based)
- Lisensi: Open Source

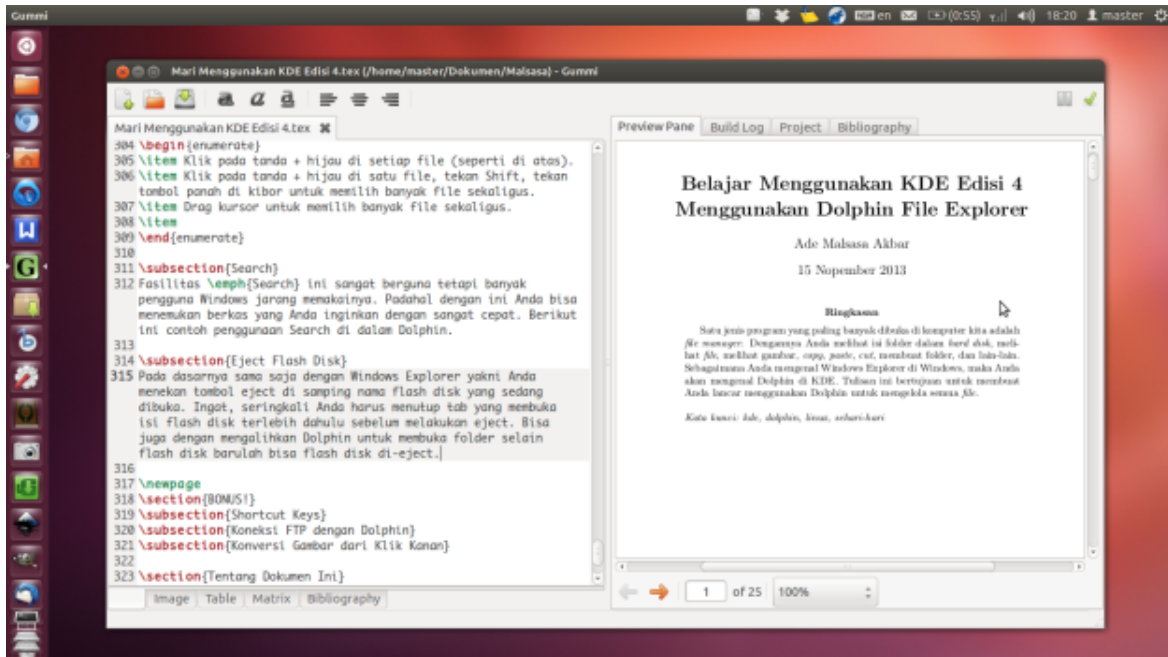
Saya telah berbahagia menggunakan TeXlipse di Eclipse sejak lama, ia memiliki code completion terintegrasi (termasuk entri-entri BibTeX), templat-templat yang mudah dikustomisasi, panel outline, dan secara langsung ia terintegrasi dengan Eclipse itu sendiri yang secara otomatis memiliki shortcuts, version control, dan lain-lain.

Ada plugin penampil PDF untuk Eclipse bernama Pdf4Eclipse dengan dukungan SyncTeX, yang mendukung pencarian maju/mundur di dalam dokumen LaTeX. Karena TeXlipse me-rebuild kode-kode LaTeX secara otomatis (di background) setelah sekali disimpan, maka kode dan pratayang dari dokumen selalu disinkronkan.

## 10. Gummi

- OS: Linux (tersedia versi unstable untuk Windows)
- Lisensi: Open Source

Emacs bagus, tetapi yang seringkali saya pakai adalah Gummi. Ia memiliki panel pratyayang yang sangat berguna untuk mengetahui kesalahan sintaks dan kesalahan format sesegera mungkin. Plus, ketika Anda menyimpan dokumen LaTeX ia akan menyimpan PDF secara otomatis. Fitur lainnya termasuk peralatan bantuan penulisan matriks, memasukkan gambar, dan sitasi (citation).

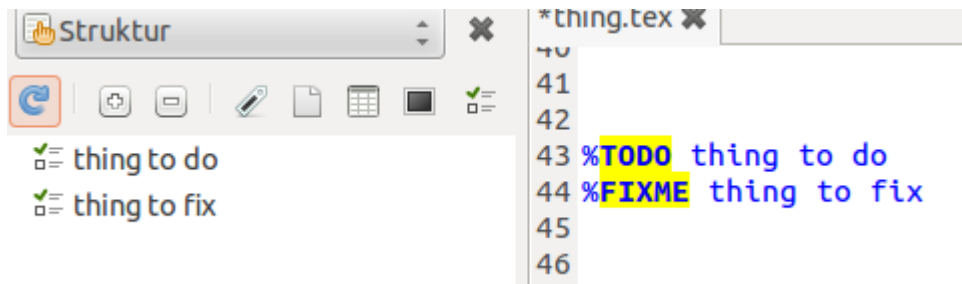


## 11. LaTeXila

- OS: Linux
- Lisensi: Open source
- Unicode: Ya

LaTeXila adalah lingkungan LaTeX terintegrasi untuk GNOME. Ia memiliki antarmuka yang bagus dan jelas. Ia tersedia di Ubuntu Software Center. Anda dapat melihat preview dari apa yang Anda tulis kapanpun Anda mau.

LaTeXila memiliki komentar-komentar ”ajaib” untuk membuat todonotes, yang akan tayang di panel struktur di sebelah kiri. Komentar itu adalah %TODO dan %FIXME, yang harus diikuti oleh teks (jika tidak ada teks, maka tidak ada yang tayang di panel).



## 12. Geany with GeanyLaTeX

- OS: Windows, Mac, Linux dan lain-lain
- Lisensi: Open Source

Editor bagus lainnya adalah Geany. Software ini memiliki plugin untuk LaTeX. Plugin ini di-maintain oleh salah satu developer utama Geany sendiri. Plugin ini memiliki wizard untuk dokumen LaTeX baru, autocompletion, insert environment dengan mudah, dan tentu terdokumentasi dengan baik.

## CHAPTER 2

---

### YOUR FIRST DOCUMENT

---

#### Pengertian Latex

TEX merupakan perangkat lunak pengolah dokumen yang terutama ditujukan menghasilkan dokumen yang berisi simbol-simbol matematik. TEX diciptakan oleh

Donald E. Knuth (Mei1977) sebagai bahasa pembentuk dokumen (document formatting language). LaTeX adalah sistem typesetting yang dapat digunakan untuk membuat artikel, buku, surat, dan publikasi lain berkualitas tinggi. LaTeX berbasiskan pada TeX, bahasa typesetting aras bawah yang didesain oleh Donald E.

Knuth. LaTeX tidak bekerja seperti pengolah kata WYSIWYG(what you see is what you get), jenis persiapan dokumen yang sudah banyak dipakai oleh banyak orang. Dengan LaTeX, Anda tidak harus peduli dengan pemformatan dokumen, hanya tentang penulisan dokumen.

Perangkat lunak TEX memiliki kemampuan yang baik untuk mengolah dokumen-dokumen yang berkualitas tinggi. Kelemahannya, perintah perintahnya sulit digunakan untuk menuliskan dokumen terstruktur yang terdiri dari unsure unsure bab, sub-bab, paragraph, table dan gambar bernomor, dsb.

Versi LATEX yang sudah baku ini memiliki beberapa kekuatan, diantaranya:

- Standard yang sangat baik untuk menyiapkan tulisan teks, formula teknis, dan tabel-tabel
- Kemudahan penggunaan oleh penulis naskah.
- Portabilitas dokumen pada berbagai platform
- Adaptabilitas terhadap banyak bahasa (multilingual support)
- Ketersediaan secara meluas dan bebas

Sebuah dokumen LATEX memiliki struktur yang dicirikan dengan blok yang diapit oleh pasangan perintah `\begin` dan `\end`. Untuk menyatakan jenis dokumen yang akan diolah, setiap dokumen harus dimulai dengan perintah:

```
\documentclass{...}
```

Membuat dokumen dengan latex sangat sederhana. Anda bisa memulai membuat dokumen latex dengan mengetikkan kode latex lalu ditambah dengan konten yang sederhana yaitu teks. Latex menggunakan kode-kode perintah yang terkontrol yang nantinya akan menentukan seperti apa hasil akhir dari dokumen yang anda buat. Setelah anda mengetikkan kode-kode perintah latex, maka compiler dari editor latex dapat mengkompilasinya menjadi file .pdf.

### Kelas Dokumen

Jenis dokumen yang akan diolah ditentukan oleh perintah pertama dalam bentuk:

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

Dalam perintah diatas, "class" dapat diganti oleh `article`, `report`, `book`, atau `slides` untuk menuliskan artikel, laporan, buku, atau transparansi untuk seminar. Sedangkan pada bagian "option" dapat dituliskan satu atau beberapa pilihan berikut: `10pt`, `11pt`, `12pt` untuk menyatakan ukuran font utama yang digunakan didalam dokumen `a4paper`, `letterpaper` menyatakan ukuran kertas yang digunakan `titlepage`, `notitlepage` untuk menyatakan apakah halaman judul akan dibuat terpisah dari badan dokumen atau tidak `twocolumn` untuk menampilkan dokumen dalam bentuk dua kolom `twoside`, `oneside` untuk menyatakan apakah dokumen akan dicetak pada satu sisi atau dua sisi dari kertas. contoh dasar menggunakan kode perintah dalam latex yaitu:

```
\documentclass{article}
```

```
\begin{document}
```

Hello World!

`\end{document}`

Kode diatas jika di compile maka akan muncul tulisan "Hello World!" dalam bentuk file .pdf.

### Perintah-Perintah LATEX

#### 1. Spasi dalam Latex

Ada perintah khusus untuk membuat spasi dengan panjang tertentu baik secara horizontal maupun vertikal, yaitu :

- Jika ingin membuat jarak dengan panjang tertentu antara 2 baris, dapat menggunakan tanda ' `\\` ' di akhir baris. Dan juga dapat menentukan sendiri panjang baris kosong dengan menggunakan perintah seperti contoh berikut ini :

```

        baris 1 \\
        \vspace{2cm}
        baris 2 \\

```

Dengan perintah ini, Latex akan mengosongkan baris-baris sepanjang 2 cm. Tanpa menggunakan perintah ini untuk membuat spasi dalam teks dokumen, Latex akan tetap menganggapnya 1 spasi.

- Jika ingin membuat spasi sejauh beberapa centimeter antara 2 kata dibutuhkan perintah sebagai berikut :

```

kata 1 \hspace{2cm} kata 2

```

Dengan perintah ini, Latex akan membuat spasi sejauh 2 centimeter. Jadi, secara umum aturan yang dapat dipakai adalah akhiri paragraf dengan tanda ' `\\` ' dan berikan 1 baris kosong antara tiap-tiap paragraf dan 1 spasi kosong antara masing-masing kata.

#### 2. Alignment dalam Latex

Alignment/perataan baris pada Latex adalah rata kiri, rata kanan, atau rata tengah. Semua dokumen dalam Latex secara default diatur memiliki perataan justified (rata kanan kiri).

- Jika ingin mengatur dokumen rata kiri digunakan perintah sebagai berikut :

```

        \begin{raggedright}
        isi dokumen yang diatur dengan rata kiri
        \end{raggedright}

```



- Jika ingin mengatur dokumen rata kanan digunakan perintah sebagai berikut :

```

\begin{raggedleft}
isi dokumen yang diatur dengan rata kanan
\end{raggedleft}

```

- Jika ingin mengatur dokumen rata tengah digunakan perintah sebagai berikut :

```

\begin{center}
isi dokumen yang diatur dengan rata tengah
\end{center}

```

### 3. Bahasa dalam Latex

Latex dapat menggunakan tulisan mengikuti aturan ejaan yang dimiliki bahasa tertentu. Kemampuan ini diatur oleh babel package. Mengubah peraturan bahasa dengan menggunakan babel akan secara otomatis mengubah nama-nama dari unit struktur dokumen (misalnya Abstract, Chapter, Index) menjadi terjemahannya.

Perintah yang mengatur latex untuk menggunakan babel bahasa Indonesia seperti berikut :

```

\dokumenclass {a4paper, 12pt}{report}

\usepage[bahasa]{babel}

\begin{document}

.....

\end{document}

```

### 4. Keterangan dalam Latex

Jika ingin menambahkan keterangan pada file yang tidak ingin tercetak, caranya dengan menambahkan tanda % diawal setiap baaris keterangan. Contoh :

```

\dokumenclass {a4paper, 12pt}{report}

\usepage[bahasa]{babel}

\begin{document}

```

ini baris keterangan, baris ini tidak akan tercetak dalam file keluaran

```

.....

```

`\end{document}`

## 5. Font dalam Latex

Ada 3 jenis fonts dalam Latex :

- Roman. Cara menggunakannya seperti dibawah ini :

`{\rmfamily teks yang ingin diformat }`

- Sans serif. Cara menggunakannya seperti dibawah ini :

`{\sffamily teks yang ingin diformat }`

- Typewriter. Cara menggunakannya seperti dibawah ini :

`{\ttfamily teks yang ingin diformat }`

Ada 4 bentuk font dalam Latex :

- Italic. Cara mengaturnya sebagai berikut :

`{\itshape teks yang ingin diformat }`

- Slanted. Cara mengaturnya sebagai berikut :

`{\slshape teks yang ingin diformat }`

- Vertical. Cara mengaturnya sebagai berikut :

`{\upshape teks yang ingin diformat }`

- SMALL CAPS. Cara mengaturnya sebagai berikut :

`{\scshape teks yang ingin diformat }`

### Ukuran Font

Ada beberapa macam ukuran font dalam Latex. Untuk menggunakan ukuran-ukuran itu caranya sebagai berikut :

- Tiny

`{\tiny teks yang ingin diformat }`

- Scriptsize

`{\scriptsize teks yang ingin diformat }`

- Footnotesize

`{\footnotesize teks yang ingin diformat }`

- Small

`{\small teks yang ingin diformat }`

- Normal

`{\normalsize teks yang ingin diformat }`

- Large

`{\large teks yang ingin diformat }`

- Larger

`{\LARGE teks yang ingin diformat }`

- Largest

`{\LARGE teks yang ingin diformat }`

- Huge

`{\huge teks yang ingin diformat }`

- Huger

`{\Huge teks yang ingin diformat }`

## 6. Struktur Dasar Sebuah Dokumen Latex

### Document Class

Document class dalam Latex berguna untuk menentukan layout halaman, jenis heading, dan berbagai perintah dan environment yang digunakan untuk mengatur style dokumen. Cara mendeklarasikannya sebagai berikut :

`\dokumentclass {class}`

Ada beberapa jenis document class yang bisa dipakai dalam sebuah dokumen Latex, yaitu :

- report : dapat digunakan untuk membuat laporan (report) baik dalam bidang bisnis, teknik, hukum, akademis, atau ilmu pengetahuan.
- article : dapat digunakan untuk membuat paper, artikel sebuah jurnal atau majalah, review, paper untuk konferensi, atau catatan riset.

- book : dapat digunakan untuk membuat buku dan thesis.
- letter : dapat digunakan untuk membuat surat.

Biasanya kelas 'article' adalah yang paling sering digunakan untuk sembarang jenis dokumen.

### Document Class Option

Merupakan pilihan yang tersedia pada kelas dokumen yang bisa ditentukan sendiri isinya. Opsi pada suatu kelas dokumen dituliskan sebagai berikut :

```
\documentclass [option1, option2]{class}
```

Default opsi yang digunakan oleh Latex sebagai berikut :

- Ukuran kertas yang digunakan adalah A4.
- Ukuran font yang digunakan adalah 10pt untuk semua kelas dokumen.
- Layout halaman yang digunakan adalah two-sided printing khusus untuk kelas book dan report; dan one-sided printing khusus untuk kelas article dan letter.
- Halaman judul yang terpisah dibagian awal dokumen khusus untuk kelas book dan report.

Opsi diatas dapat dimodifikasikan sebagai berikut :

- Ukuran kertas. Dapat ditentukan sendiri ukuran kertasnya. Cara penulisannya :

```
\documentclass [ a3paper ] {class}
          atau
\documentclass [ letterpaper ] {class}
```

- Ukuran font. Dapat memilih ukuran 10pt, 11pt, atau 12pt. Cara penulisannya :

```
\documentclass [ a4paper, 11pt ] {class}
```

Setelah menentukan ukuran font yang dipakai, semua font yang ada dalam dokumen akan diatur sesuai dengan ukuran yang ditentukan.

Layout halaman. Dapat ditentukan dengan pilihan berikut :

- oneside : jika ingin layout one-sided printing saat menggunakan kelas book dan report.
- twoside : jika ingin layout two-sided printing saat menggunakan kelas article.

- `titlepage` : jika ingin kelas `article` untuk memiliki halaman judul yang terpisah dibagian awal dokumen.
- `draft` : berguna untuk mengatur Latex supaya menandai masalah-masalah yang timbul seperti masalah pemenggalan kata ( pemenggalan kata tidak tepat ) atau masalah perataan tulisan ( ada baris tertentu melebihi batas kanan dokumen ).

### **Paket-Paket dalam Latex**

Merupakan fungsi-fungsi yang dipakai untuk menambah kemampuan Latex melakukan pengaturan dokumen. Cara menggunakan paket yang sudah tersedia/terintegrasi di dalam Latex sebagai berikut :

```
\documentclass {class}

\usepackage [ option ] {nama paket}

\begin{document}

.....

\end{document}
```

Beberapa paket yang tersedia dalam Latex sebagai berikut :

- `graphicx` : dapat menghasilkan gambar grafis dan juga membuat Latex mampu menampilkan gambar yang kita sertakan dalam dokumen.
- `hyperref` : dapat menghasilkan dokumen yang memiliki dynamic link ke alamat tertentu.
- `babel` : dapat mengenali format bahasa yang digunakan.
- `color` : dapat menghasilkan teks dokumen yang memiliki warna sesuai warna yang ditentukan.
- `makeidx` : dapat menghasilkan indeks dari dokumen yang dibuat.

### **Document Environment**

Merupakan bagian dalam sebuah dokumen Latex dimana isi sebenarnya dari dokumen itu sendiri ditempatkan.

```
\documentclass {class}

\begin{document}
```

```

.....
.....

\end{document}

```

Struktur `\begin . . . \end` inilah yang disebut dengan environment. Environment membatasi bagian teks yang akan diatur dengan aturan tertentu.

### **Penulisan Judul**

Judul dalam sebuah dokumen Latex diletakan pada awal document environment. Cara penulisannya sebagai berikut :

```

\documentclass [ a4paper, 12pt ] {report}

\begin{document}

\title{Judul Dokumen}

\author{Nama Penulis}

\date{Tanggal Pembuatan}

\maketitle

.....
.....

\end{document}

```

### **Abstrak**

Pada dokumen kelas article dan report umumnya memiliki abstrak/ringkasan. Latex memiliki cara khusus untuk menuliskan abstrak. Penulisannya sebagai berikut :

```

\documentclass [ a4paper, 12pt ] {report}

\begin{document}

\title{Judul Dokumen}

\author{Nama Penulis}

\date{Tanggal Pembuatan}

```

```

\maketitle

\begin{abstract}

isi abstrak

\end{abstract}

.....

\end{document}

```

Jika ingin mengubah judul abstrak digunakan perintah sebelum `\begin{abstract}` :  
`\renewcommand{\abstractname}{Ringkasan Laporan}`

Contoh diatas dapat mengganti judul abstrak menjadi ” Ringkasan Laporan ”.

### Daftar Berurut

Ada 3 cara penulisan daftar berurut, yaitu :

Daftar dengan penomoran dengan menggunakan simbol (Bulleted List), contoh :

Mobil

Montor

Sepeda

Bus

Cara penulisannya dalam Latex sebagai berikut :

```

\begin{itemize}

\item . . . .

\item . . . .

\item . . . .

\item . . . .

.....

.....

\end{itemize}

```

### Daftar Isi

Untuk menampilkan daftar isi dapat menggunakan perintah :

```
\tableofcontents
```

Perintah ini diletakan pada bagian dimana daftar isi tersebut ditempatkan. Biasanya daftar isi ditempatkan setelah abstrak/kata pengantar.

Untuk menampilkan daftar gambar dapat menggunakan perintah :

```
\listoffigures
```

Untuk menampilkan daftar tabel dapat menggunakan perintah :

```
\listoftables
```

Latex menghasilkan file berekstensi \*.toc untuk daftar isi, daftar gambar, dan daftar tabel. Jika daftar isi, daftar gambar dan daftar tabel tidak menampilkan keseluruhan struktur dokumen dengan benar, dapat diatur sendiri isinya dengan perintah berikut ini :

```
\addcontentsline{toc}{struktur}{teks yang ingin ditampilkan pada daftar isi}
```

Struktur dapat diisi dengan chapter, section, subsection, dll, tergantung dengan bagian dokumen yang ingin dimasukkan dalam daftar isi. Dengan perintah diatas, Latex akan menghasilkan baris baru dalam daftar isi dan akan secara otomatis menentukan nomor halaman bagian tersebut.

### Gambar

Agar Latex mendapatkan gambar dalam dokumen, maka perlu mendeklarasikan penggunaan paket graphicx pada bagian preamble. Cara mendeklarasikannya adalah :

```
\usepackage{graphicx}
```

Untuk menempatkan sebuah gambar dalam dokumen Latex, dapat dengan cara berikut :

```
\begin{figure}[htbp]
```

```
\caption{Nama Gambar}
```

```
\begin{center}
```

```
\includegraphics[width=2cm,height=3cm\columnwidth]{nama file gambar}
```



```
\end{center}
```

```
\end{figure}
```

Ada beberapa hal yang perlu diketahui dalam format perintah diatas :

- Panjang dan lebar dari gambar yang akan ditampilkan dapat ditentukan sesuai keinginan. Isi dari width dapat diisi dengan lebar gambar dan isi dari height dapat diisi dengan tinggi gambar itu; keduanya harus dilengkapi dengan dimensi dari ukuran panjang yang digunakan.
- File gambar yang ingin dimasukkan dalam dokumen, harus diletakkan pada direktori yang sama dengan direktori file dokumen (\*.tex).
- Pengaturan posisi gambar dapat diatur sesuai dengan 2 hal :
  - Perataan terhadap tepi dokumen : dengan mengubah `\begin{center}` dan juga `\end{center}` dapat menentukan posisi gambar terhadap tepi dokumen.
  - Huruf-huruf pada `\begin{figure}[htbp]` berfungsi sebagai pengatur posisi gambar pada suatu halaman.
    - \* h : tabel diletakan persis ditempat perintah tersebut dituliskan pada dokumen.
    - \* t : tabel diletakan dibagian atas halaman.
    - \* b : tabel diletakan dibagian bawah halaman.
    - \* p : tabel diletakan pada sebuah halaman khusus yang hanya memuat tabel itu saja.

Saat menggunakan h, Latex akan otomatis menempatkan gambar dihalaman baru jika tidak ada cukup ruang untuk gambar tersebut ditempat perintah gambar dituliskan.

Format gambar standar Latex adalah \*.eps. Tetapi gambar dengan format \*.jpg juga bisa digunakan.

### Daftar Pustaka

Untuk menampilkan daftar pustaka pada akhir sebuah dokumen Latex menggunakan format perintah sebagai berikut :

```
\begin {thebibliography}{99}
```

```
\bibitem {label untuk referensi}{keterangan pustaka yang digunakan}
```

```
.....
.....
```

`\end{thebibliography}`

Beberapa hal yang perlu diketahui dalam perintah diatas :

- Angka 99 memberitahukan Latex bahwa penomoran maksimal Daftar pustaka adalah 99.
- Label untuk referensi diisikan keyword yang akan digunakan saat membuat rujukan ke pustaka yang bersangkutan.
- Keterangan pustaka diisi informasi mengenai : penulis, judul pustaka, edisi, penerbit, kota penerbit, dan tahun penerbit.

### **Notasi Matematika dalam Latex**

#### **Penulisan Notasi Matematika dalam Paragraf**

Untuk menyisipkan notasi matematika dalam suatu kalimat/paragraf menggunakan perintah berikut ini :

`\begin{math} . . . . . \end{math}` atau

`$ . . . . . $`

Titik-titik tersebut diisi dengan notasi matematika yang disisipkan.

#### **Paragraf Khusus Matematika**

Untuk menuliskan notasi matematika yang panjang, dapat memilih untuk menuliskannya dalam paragraf baru. Perintahnya :

`\begin{displaymath}`

`. . . . .`

`\end{displaymath}`

Titik-titik tersebut diisi dengan notasi matematika yang disisipkan.

#### **Font dalam Matematika**

Ada beberapa perintah yang digunakan untuk mengubah jenis font yang dipakai dalam notasi matematika, seperti berikut ini :

Perintah `$\mathrm{x y z}$` akan menghasilkan :

$xyz$

Perintah `\mathsf{x y z}` akan menghasilkan :

$$\mathsf{xyz}$$

Perintah `\mathtt{x y z}` akan menghasilkan :

$$\mathtt{xyz}$$

Perintah `\mathit{x y z}` akan menghasilkan :

$$\mathit{xyz}$$

Perintah `\mathbf{x y z}` akan menghasilkan :

$$\mathbf{xyz}$$

Perintah `\mathcal{x y z}` akan menghasilkan :

$$\mathcal{xyz}$$

Untuk menuliskan font matematika dalam bentuk superscripts dan subscripts digunakan aturan berikut ini :

Superscripts, cara penulisannya dengan perintah `\sp{ . . . }` atau dengan tanda  $^$ .

Subscripts, cara penulisannya dengan perintah `\sb{ . . . }` atau dengan tanda  $_$ .

Contoh :

```
\begin{displaymath}
y=x\sb{1}\sp{2}+x\sb{2}\sp{2}
\end{displaymath}
```

### Penulisan Pecahan

Untuk menghasilkan notasi pecahan dengan format perintah sebagai berikut:

```
\frac{numerator}{denominator}
```

Contoh :

```
\begin{displaymath}
\frac{a+2b}{a-1}
\end{displaymath}
```

### Penulisan Array dan Matriks

Sebuah array/matriks dituliskan dalam environment tabular sama seperti cara pembuatan tabel. Perintahnya sebagai berikut :

```
\begin{displaymath}
\left (
\begin{array}{rrr}
0 & 45 & 23 \\
34 & -93 & 68 \end{array}
\right )
\end{displaymath}
```

Beberapa hal yang perlu diketahui dari format perintah itu :

- Sama seperti penulisan tabel, huruf r dibagian belakang `\begin{array}{rrr}` fungsinya menentukan posisi dari masing-masing komponen matriks tersebut. Dalam hal ini komponen masing-masing matriks dibuat menjadi rata kanan.
- Tanda kurung yang digunakan adalah tanda kurung kurawal. Bagian kurung buka dan tutup didefinisikan masing-masing.

### Penulisan Vektor

Penulisan vektor dalam Latex menggunakan perintah berikut ini :

```
\begin{displaymath}
\vac{variabel}
\end{displaymath}
```

Contoh :

```
\begin{displaymath}
\vac{a}
\end{displaymath}
```



## CHAPTER 3

---

# STRUCTURING YOUR DOCUMENT (SECTION AND PARAGRAPH)

---

Membuat dokumen yang sangat mendasar dalam pelajaran sebelumnya, namun saat menulis makalah, perlu menyusun konten ke dalam unit logika. Untuk mencapai hal ini, LaTeX menawarkan kepada perintah untuk menghasilkan judul bagian dan mencatatnya secara otomatis. Perintah untuk membuat judul bagian

sangat mudah:

```
\section { }  
\subsection { }  
\subsubsection { }  
\paragraph { }  
\subparagraph { }
```

Sedangkan dokumen kelas report dan book selain memiliki perintah-perintah di atas memiliki juga perintah :

```
\part {... }  
  
\chapter {... }
```

```
\frontmatter
```

```
\mainmatter
```

```
\backmatter
```

Argumen yang diberikan pada perintah-perintah ini adalah nama bab, subbab, dll.

Dalam naskah buku yang dituliskan dengan kelas dokumen book, frontmatter digunakan untuk menandai halaman judul, daftar isi, kata pengantar, daftar gambar, dsb.), mainmatter untuk menandai bagian tulisan utama, dan backmatter untuk menandai daftar pustaka, indeks, daftar istilah, dsb. Perintah `\chapter`, `\section`, `\subsection`, dan `\subsubsection` secara otomatis memberikan nomor pada nama bagian, bab, dsb. Jika nomor ini tidak diinginkan, perintah yang ekuivalen adalah `\chapter*`, `\section*`, `\subsection*`, dan `\subsubsection*`.

Perintah bagian diberi nomor dan akan muncul dalam daftar isi dokumen. Paragraf tidak diberi nomor dan tidak akan ditampilkan dalam daftar isi. Berikut contoh output menggunakan bagian:

```
1 Section
Hello World!
1.1 Subsection
Structuring a document is easy
```

Untuk mendapatkan hasil ini, kami hanya perlu menambahkan beberapa baris ke program kami dari pelajaran 1:

```
\documentclass {article }

\title {Title of my document }
\date {2013-09-01 }
\author {John Doe }

\begin {document }

\maketitle
\pagenumbering {gobble }
\newpage
\pagenumbering {arabic }

\section {Section }

Hello World!

\subsection {Subsection }
```

Structuring a document is easy!

```
\end {document }
```

Gambar berikut menunjukkan struktur hirarkis dari semua elemen:

```
1. Section
  Hello World
  1.1 Subsection
    Structuring a document is easy!
    1.1 Subsection
      More Text
      Paragraph      Some more text
      Subparagraph   Even more text
2. Another Section
```

Saya telah menggunakan kode berikut untuk mendapatkan output ini:

```
\documentclass {article }

\begin {document }

\section {Section }

  Hello World!

\subsection {Subsection }

  Structuring a document is easy!

\subsubsection {Subsubsection }

    More text.

\paragraph {Paragraph }

    Some more text.

\subparagraph {Subparagraph }

    Even more text.

\section {Another section }

\end {document }
```

Contoh struktur dokumen berkelas article dan book :



```

\documentclass {article }

\usepackage {... }

\begin {document }

    \maketitle

    \section {... }

    \section {... }

    \subsection {... }

    \subsubsection {... }

    \section

\end {document }

```

Sangat mudah untuk menyusun dokumen menjadi beberapa bagian menggunakan LaTeX. di LaTeX sangat mudah untuk memiliki format yang konsisten di seluruh kertas. Berikut perintah-perintah latex, seperti:

\* part

*part* berfungsi untuk membuat pembagian bab, biasanya dibuat dalam halaman yang terpisah. Adapun penggunaannya adalah sebagai berikut:

```
\part {[Judul] }
```

\* chapter

*chapter* merupakan bab utama yang memuat judul. Penggunaannya demikian:

```
\chapter {[Judul] }
```

\* section

*section* merupakan pasal dari suatu bab. Contoh penggunaannya adalah sebagai berikut:

```
\section {[Judul] }
```

\* subsection

*subsection* berfungsi untuk membuat sub pasal atau pasal baru di bawah judul pasal.

```
\subsection {[Judul] }
```

*subsubsection* berfungsi untuk membuat sub pasal di bawahnya lagi dari sub pasal yang ada.

```
\subsubsection* {[Judul] }
```

\* paragraph

*paragraph* berguna untuk membuat alinea kalimat, cara penggunaannya adalah sebagai berikut:

```
\paragraph {kalimat }
```

\* subparagraph

*subpragraph* berfungsi untuk membuat alinea baru di dalam alinea yang sudah ada.

Cara penggunaannya adalah demikian:

```
\subparagraph {kalimat }
```

Contoh struktur dokumen berikut ini:

```
\part {Memulai LATEX } % ini adalah contoh penggunaan part
\chapter {Menggunakan LATEX } % ini adalah contoh penggunaan chapter
\section {Penggunaan Class dalam penulisan dokumen }
% ini adalah contoh penggunaan section
\subsection {Penyertaan Package }
% ini adalah contoh penggunaan subsection
\paragraph {Penyertaan package berguna
untuk menambahkan fungsi
kedalam dokumen/naskah yang kita buat.
Bentuk penulisannya
adalah sebagai berikut: }
% ini adalah contoh penggunaan
paragraph
```

### 3.1 Komentar

Fungsi dari komentar adalah untuk menampilkan catatan dari naskah yang kita buat, namun tidak ditampilkan pada saat file dicetak. Contoh penggunaannya adalah sebagai berikut:

```
\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }
\begin {document }
```

Halo, ini adalah contoh penulisan menggunakan LaTeX.  
ukuran font dari naskah ini adalah 12. Pencetakan akan menggunakan kertas A4, yang akan dicetak dalam satu sisi. Naskah ini berbentuk bu dan akan ditampilkan kedalam bahasa Indonesia.  
komentar ini tidak akan ditampilkan pada saat dilakukan pencetakan naskah.

```
\end {document }
```

### 3.2 Membuat judul dokumen

Untuk judul dokumen, perintahnya adalah sebagai berikut:

```
\title { }
\maketitle
```

Adapun contohnya adalah sebagai berikut:

```

\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }
\title {Membuat Dokumen dengan \LaTeX { } }
\begin {document }
\maketitle

```

Halo, ini adalah contoh penulisan menggunakan LaTeX, dengan ukuran font Pencetakan akan menggunakan kertas A4, yang akan dicetak dalam satu sis Naskah ini berbentuk buku dan akan ditampilkan kedalam bahasa Indonesia judul akan ditampilkan secara otomatis pada awal dokumen ketika dokumen dikonversi ke format DVI,HTML, ataupun PDF. \end {document }

### 3.3 Pembuatan Paragraph

```

\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }
\begin {document }
\paragraph {Kata Pembuka }

```

Halo, ini adalah contoh penulisan menggunakan LaTeX, dengan ukuran font Pencetakan akan menggunakan kertas A4, yang akan dicetak dalam satu sis Naskah ini berbentuk buku dan akan ditampilkan kedalam bahasa Indonesia Paragraf akan ditampilkan secara otomatis pada awal dokumen ketika dokumen dikonversi ke format DVI,HTML, ataupun PDF. \end {document }

### 3.4 Memisahkan Baris

Untuk memisahkan baris, Anda bisa menggunakan perintah sebagai berikut:

```

\\
atau
\nline

```

Adapun contoh penggunaannya adalah demikian:

Contoh 1:

```

\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }
\begin {document }

```

Halo, ini adalah contoh penulisan menggunakan LaTeX, dengan ukuran font Pencetakan akan menggunakan kertas A4, yang akan dicetak dalam satu sis Naskah ini berbentuk buku dan akan ditampilkan kedalam bahasa Indonesia

Tulisan ini akan ditampilkan dengan penambahan satu baris.

```

\end {document }

```

Contoh 2:

```

\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }
\begin {document }

```

Halo, ini adalah contoh penulisan menggunakan LaTeX, dengan ukuran font Pencetakan akan menggunakan kertas A4, yang akan dicetak dalam satu sis Naskah ini berbentuk buku dan akan ditampilkan kedalam bahasa Indonesia

```

\linebreak
Tulisan ini akan ditampilkan dengan penambahan satu baris.
\end {document }

```

### 3.5 Berpindah Halaman

Untuk berpindah halaman, Anda bisa menggunakan perintah sebagai berikut:

```

\newpage

```

Contohnya adalah sebagai berikut

```

\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }
\title {Membuat Dokumen dengan \LaTeX { } }
\author {R. Kresno Aji (masaji@ai.co.id) }
\date {17 Agustus 2004 }
\begin {document }
\maketitle

```

Halo, ini adalah contoh penulisan menggunakan LaTeX, dengan ukuran font Pencetakan akan menggunakan kertas A4, yang akan dicetak dalam satu sis Naskah ini berbentuk buku dan akan ditampilkan kedalam bahasa Indonesia judul akan ditampilkan secara otomatis pada awal dokumen ketika dokumen dikonversi ke format DVI,HTML, ataupun PDF.

```

\newpage
\chapter {Halaman Baru }
\end {document }

```

### 3.6 Environment

LATEX menyediakan environmen yang berupa:

- \* Itemize

berfungsi untuk membuat daftar yang tidak memiliki urutan.

- \* Enumerate

berfungsi untuk membuat daftar yang berurutan.

- \* Flushleft

untuk membuat kalimat rata kiri.

- \* Center

berfungsi untuk membuat kalimat dengan format center.

- \* Flushright

berfungsi untuk membuat kalimat rata kanan.

- \* Footnote

berfungsi untuk membuat catatan kaki.

- \* Verbatim

berfungsi untuk membuat kalimat / karakter yang ditulis

\* Table

berfungsi untuk membuat tabel.

### 1. Pembuatan daftar berurutan

Untuk membuat daftar yang berurutan, Anda bisa menggunakan perintah berikut ini:

```
\begin {enumerate }
  \item
\end {enumerate }
```

Contohnya adalah demikian:

```
\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }
\begin {document }
```

Halo, ini adalah contoh penulisan menggunakan LaTeX, dengan ukuran font Pencetakan akan menggunakan kertas A4, yang akan dicetak dalam satu sis Naskah ini berbentuk buku dan akan ditampilkan kedalam bahasa Indonesia Daftar secara berurutan akan ditampilkan secara otomatis pada awal dokumen ketika dokumen dikonversi ke format DVI,HTML, ataupun PDF.

Pada bab ini, kita akan membahas:

```
\begin {enumerate }
  \item item satu
  \item item dua
\end {enumerate }
\end {document }
```

### 2. Penggunaan rata kiri, rata kanan dan center

Untuk membuat dokumen LATEX menjadi rata kiri perintahnya adalah demikian:

```
\begin {flushleft }
[kalimat]
\end {flushleft }
```

Contohnya adalah sebagai berikut:

```
\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }
\begin {document }
\begin {flushleft }
```

Halo, ini adalah contoh penulisan menggunakan LaTeX, dengan ukuran font Pencetakan akan menggunakan kertas A4, yang akan dicetak dalam satu sis Naskah ini berbentuk buku dan akan ditampilkan kedalam bahasa Indonesia dan berada di sebelah kiri.

```
\end {flushleft }
```

```
\end {document }
```

Untuk membuat dokumen LATEX menjadi rata kanan, perintahnya adalah demikian:

```
\begin {flushright }  
[kalimat]  
\end {flushright }
```

Contohnya adalah sebagai berikut:

```
\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }  
\begin {document }  
\begin {flushright }
```

Halo, ini adalah contoh penulisan menggunakan LaTeX, dengan ukuran font Pencetakan akan menggunakan kertas A4, yang akan dicetak dalam satu sis Naskah ini berbentuk buku dan akan ditampilkan kedalam bahasa Indonesia dan terletak rata kanan.

```
\end {flushright }  
\end {document }
```

Untuk membuat dokumen LATEX menjadi center perintahnya adalah demikian:

```
\begin {center }  
[kalimat]  
\end {center }
```

Contohnya adalah sebagai berikut:

```
\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }  
\begin {document }  
\begin {center }
```

Halo, ini adalah contoh penulisan menggunakan LaTeX, dengan ukuran font Pencetakan akan menggunakan kertas A4, yang akan dicetak dalam satu sis Naskah ini berbentuk buku dan akan ditampilkan kedalam bahasa Indonesia dan terletak center.

```
\end {center }  
\end {document }
```

### 3. Pembuatan footnote

Untuk pembuatan footnote pada dokumen LATEX, Anda bisa memberikan perintah sebagai berikut:

```
\footnote { ... }
```

Contohnya adalah demikian:

```
\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }
\begin {document }
```

Halo, ini adalah contoh penulisan menggunakan LaTeX, dengan ukuran font Pencetakan akan menggunakan kertas A4, yang akan dicetak dalam satu sis Naskah ini berbentuk buku dan akan ditampilkan kedalam bahasa Indonesia \footnote {Ini adalah contoh penggunaan footnote }

```
\end {document }
```

#### 4. Penulisan apa adanya dengan verbatim

Seperti halnya pada penulisan dalam format HTML, dengan menggunakan tag ; pre ĩ. LATEX juga menyediakan fasilitas ini. Adapun formatnya adalah sebagai berikut:

```
begin {verbatim }
[kalimat]
end {verbatim }
```

Contohnya adalah sebagai berikut:

```
\begin {verbatim }
\documentclass[12pt,a4paper,oneside,bahasa,dvips] {book }
\begin {document }
```

Pada bab ini, kita akan membahas:

```
\begin {itemize }
\item item satu
\item item dua
\end {itemize }
\end {document }
end {verbatim }
```

Maka jika dilakukan pencetakan, hasilnya akan tampak sebagai berikut:

Pada bab ini, kita akan membahas:

```
\begin {itemize }
\item item satu
\item item dua
\end {itemize }
```

#### 5. Pembuatan Tabel

Untuk membuat tabel pada dokumen LATEX, perintahnya adalah sebagai berikut:

```
\begin {tabular }
\end {tabular }
```

Untuk jelasnya, Anda bisa meniru langkah di bawah ini:

```
\hline
```

```
\begin {tabular } { |c |c |c | }
No. & \bf Uraian & Jumlah \ \
\hline
```

- & Pembelian alat-alat kantor & Rp. 250.000 \ \ \cline {2-2 }

```
\hline
\end {tabular }
```

## 6. Mengubah bentuk dan ukuran font

Ada beberapa mode perubahan font pada LATEX, seperti bisa Anda lihat pada penjelasan berikut ini:

Untuk memperkecil huruf, perintahnya adalah demikian:

```
\small
```

Untuk memperbesar huruf, perintah sebagai berikut:

```
\large
\LARGE
\Huge
```

Contohnya adalah demikian:

```
\documentclass[12pt] {article }
\begin {document }

\end {document }
```

## 7. Membuat daftar pustaka

Akhir dari pembuatan dokumen atau naskah ilmiah adalah dengan membuat daftar pustaka atau referensi. Pada LaTeX, hal ini sudah tersedia. Anda hanya perlu menggunakannya saja. Adapun perintahnya adalah sebagai berikut:

```
\bibliographystyle {plain }
\begin {thebibliography } {Reference }
\bibitem
\end {thebibliography }
```

Untuk jelasnya, Anda bisa melihat contoh di bawah ini:

```
\bibliographystyle {plain }
\begin {thebibliography } {Reference }
\bibitem A Guide to LaTeX.
\end {thebibliography }
```





## CHAPTER 4

---

### PACKAGES EXPLAINED

---



## CHAPTER 5

---

# TYPESETTING MATH IN LATEX

---



## CHAPTER 6

---

### ADDING A PICTURE

---

#### Memasukkan Gambar

Gambar adalah elemen penting dalam sebagian besar dokumen ilmiah.

LaTeX menyediakan beberapa pilihan untuk menangani gambar dan membuat tampilannya sesuai dengan kebutuhan Anda. Pada artikel ini dijelaskan bagaimana memasukkan gambar dalam format yang paling umum, cara mengecilkan, memperbesar dan memutarnya, dan bagaimana mereferensikannya di dalam dokumen Anda.

```
\documentclass {article }  
  
\usepackage {graphicx }  
  
\graphicspath { {images/ } }  
  
\begin {document }
```

The universe is immense and it seems to be homogeneous,

in a large scale, everywhere we look at.

```
\includegraphics {universe }
```

There's a picture of a galaxy above

```
\end {document }
```

Lateks tidak bisa mengatur gambar dengan sendirinya, jadi kita perlu menggunakan paket grafisnya. Untuk menggunakannya, kami menyertakan baris berikut dalam basa-basi: `\usepackage {graphicx }` Perintah `\graphicspath { {images / } }` memberitahu LaTeX bahwa gambar disimpan dalam folder bernama gambar di bawah direktori saat ini. Perintah `\includegraphics {universe }` adalah salah satu yang benar-benar menyertakan gambar dalam dokumen. Disini alam semesta adalah nama file yang berisi gambar tanpa ekstensi, maka alam semesta.PNG menjadi alam semesta. Nama file gambar tidak boleh berisi spasi putih atau beberapa titik.

Catatan: Ekstensi file diperbolehkan disertakan, tapi sebaiknya hilangkan itu. Jika ekstensi file dihilangkan maka akan meminta LaTeX untuk mencari semua format yang didukung. Untuk lebih jelasnya lihat bagian tentang menghasilkan resolusi tinggi dan gambar beresolusi rendah.

#### Jalur folder ke gambar

Saat mengerjakan dokumen yang berisi beberapa gambar, mungkin menyimpan foto tersebut dalam satu atau beberapa folder terpisah sehingga proyek Anda lebih teratur. Pada contoh di pengenalan perintah `\graphicspath { {images / } }` memberitahu LaTeX untuk melihat folder gambar. Jalannya relatif terhadap direktori kerja saat ini. Path ke folder bisa relatif (disarankan) jika berada di lokasi yang sama dengan file utama .tex atau di salah satu sub-folder, atau mutlak jika Anda harus menentukan path yang tepat. Sebagai contoh:

%Path in Windows format:

```
\graphicspath { {c:/user/images/ } }
```

%Path in Unix-like (Linux, OsX) format

```
\graphicspath { {/home/user/images/ } }
```

Perhatikan bahwa perintah ini membutuhkan garis miring (trailing slash) dan jalur di antara kawat gigi ganda. Anda juga dapat mengatur beberapa jalur jika gambar disimpan di lebih dari satu folder. Misalnya, jika ada dua folder bernama images1 dan images2, gunakan perintahnya.

```
\graphicspath { {images1/ } {images2/ } }
```

Jika tidak ada jalur yang disetel, LaTeX akan mencari gambar di folder tempat file

.tex disimpan.

## Mengubah ukuran gambar dan memutar gambar

Jika kita ingin menentukan lebih lanjut bagaimana LaTeX memasukkan gambar kita ke dalam dokumen (panjang, tinggi, dll), kita dapat melewati pengaturan tersebut dengan format berikut:

ShareLaTeX is a great professional tool to edit online,

share and backup your \LaTeX projects. Also offers a rather large help documentation.

```
\includegraphics[scale=1.5] {lion-logo }
```

Perintah `\includegraphics [scale = 1.5] {singa-logo }` akan menyertakan gambar singa-logo dalam dokumen, skala parameter tambahan = 1,5 akan melakukan hal itu, skala gambar 1.5 dari ukuran sebenarnya. Anda juga bisa menskalakan gambar dengan lebar dan tinggi tertentu.

Seperti yang mungkin sudah Anda duga, parameter di dalam kurung [`width = 3cm`, `height = 4cm`] menentukan lebar dan tinggi gambar. Anda dapat menggunakan unit yang berbeda untuk parameter ini. Jika hanya parameter lebar yang dilewati, tinggi akan diperkecil untuk menjaga rasio aspek. Unit panjang juga bisa relatif terhadap beberapa elemen dalam dokumen. Jika Anda ingin, misalnya, buat gambar dengan lebar yang sama seperti teksnya:



ShareLaTeX is a great professional tool to edit online,  
share and backup your \LaTeX projects. Also offers a  
rather large help documentation.

```
\includegraphics[width=\textwidth]{universe }
```

Alih-alih `\textwidth` Anda dapat menggunakan panjang LaTeX default lainnya: `\columnsep`, `\linewidth`, `\textheight`, `\paperheight`, dll. Lihat panduan referensi untuk penjelasan lebih lanjut tentang unit-unit ini. Ada pilihan lain yang sama saat menyertakan gambar di dalam dokumen Anda, untuk memutarinya. Hal ini dapat dengan mudah dicapai di LaTeX:

ShareLaTeX is a great professional tool to edit online,  
share and backup your \LaTeX projects. Also offers a  
rather large help documentation.

```
\includegraphics[scale=1.2, angle=45]{lion-logo }
```

Sudut parameter = 45 memutar gambar 45 derajat berlawanan arah jarum jam.  
Untuk memutar gambar searah jarum jam gunakan angka negatif.

Pada bagian sebelumnya dijelaskan bagaimana memasukkan gambar dalam dokumen Anda, namun kombinasi teks dan gambar mungkin tidak terlihat seperti yang kita harapkan. Untuk mengubah ini kita perlu mengenalkan lingkungan baru.

In the next example the figure will be positioned  
right below this sentence.

```
\begin{figure}[h]
```

```
\includegraphics[width=8cm]{Plot }
```

```
\end{figure }
```

Lingkungan gambar digunakan untuk menampilkan gambar sebagai elemen mengambang di dalam dokumen. Ini berarti Anda menyertakan gambar di

dalam lingkungan gambar dan Anda tidak perlu khawatir dengan penempatannya, LaTeX akan memposisikannya sedemikian rupa sehingga sesuai dengan alur dokumen. Bagaimanapun, terkadang kita perlu lebih banyak kontrol terhadap cara gambar ditampilkan. Parameter tambahan dapat dilewatkan untuk menentukan posisi gambar. Pada contoh, `\begin{figure}[h]`, parameter di dalam kurung mengatur posisi gambar ke sini. Di bawah tabel untuk membuat daftar nilai posisi yang mungkin.

h

Tempatkan float di sini, yaitu kira-kira pada titik yang sama terjadi pada teks sumber (namun tidak tepat di tempat)

T

Posisi di bagian atas halaman.

B

Posisi di bagian bawah halaman.

p

Tuliskan halaman khusus hanya untuk mengapung. ! Menimpa parameter internal yang digunakan LaTeX untuk menentukan posisi float bagus”.

H

Tempatkan pelampung tepat di lokasi kode LaTeX. Membutuhkan paket float. Ini agak setara dengan h !

Pada contoh berikut, Anda dapat melihat gambar di bagian atas dokumen, meskipun telah dinyatakan di bawah teks.

In this picture you can see a bar graph that shows  
the results of a survey which involved some important  
data studied as time passed.

`\begin{figure}[t]`

`\includegraphics[width=8cm]{Plot }`

`\centering`

```
\end {figure }
```

Perintah tambahan `\centering` akan memusatkan gambar. Penyelarasan standar dibiarkan.

Mungkin juga membungkus teks di sekitar gambar. Bila dokumen berisi gambar kecil ini membuatnya terlihat lebih baik.

```
begin {wrapfigure } {r } {0.25 \textwidth } %this figure will be at
```

```
the right
```

```
\centering
```

```
\includegraphics[width=0.25 \textwidth] {mesh }
```

```
\end {wrapfigure }
```

There are several ways to plot a function of two variables, depending on the information you are interested in. For instance, if you want to see the mesh of a function so it easier to see the derivative you can use a plot like the one on the left.

```
\begin {wrapfigure } {l } {0.25 \textwidth }
```

```
\centering
```

```
\includegraphics[width=0.25 \textwidth] {contour }
```

```
\end {wrapfigure }
```

Di sisi lain, jika Anda hanya tertarik nilai tertentu Anda bisa menggunakan plot kontur, Anda Bisa menggunakan kontur plot, Anda bisa menggunakan konturnya Plot, Anda bisa menggunakan plot kontur, bisa Anda gunakan plot kontur, Anda bisa menggunakan plot kontur, Anda bisa menggunakan plot kontur, seperti yang ada di sebelah kiri.

Di sisi lain, jika Anda hanya tertarik nilai tertentu Anda bisa menggunakan plot kontur, Anda

Bisa menggunakan kontur plot, Anda bisa menggunakan konturnya  
 plot, Anda bisa menggunakan plot kontur, Anda bisa menggunakan  
 plot kontur, Anda bisa menggunakan plot kontur,  
 Anda bisa menggunakan plot kontur,

Untuk perintah di contoh kerja, Anda harus mengimpor paket wrapfig. Tambahkan ke pembukaan basa `\usepackage {wrapfig}`. Sekarang Anda dapat menentukan lingkungan wrapfigure dengan menggunakan perintah `\begin {wrapfigure } {1 } {0.25 \textwidth } \end {wrapfigure }`. Perhatikan bahwa lingkungan memiliki dua parameter tambahan yang disertakan dalam kawat gigi. Di bawah kode ini dijelaskan dengan lebih detail:

`{1 }`

Ini mendefinisikan kesejajaran gambar. Atur 1 untuk kiri dan kanan. Selanjutnya, jika Anda menggunakan buku atau format yang serupa, gunakan sebagai gantinya untuk tepi luar dan saya untuk tepi bagian dalam halaman.

`{0.25 \textwidth }`

Ini adalah lebar kotak gambar. Ini bukan lebar gambar itu sendiri, itu harus diatur dalam perintah `includegraphics`. Perhatikan bahwa panjangnya relatif terhadap lebar teks, tapi unit normal juga bisa digunakan (cm, mm, mm, dll). Lihat panduan referensi untuk daftar unit.

`\centering`

Ini sudah dijelaskan, namun dalam contoh ini gambar akan dipusatkan dengan menggunakan wadahnya sebagai referensi, bukan keseluruhan teks.

### Captioning, labeling dan referensiing

Gambar captioning untuk menambahkan deskripsi singkat dan memberi label untuk referensi lebih lanjut adalah dua alat penting saat mengerjakan teks panjang.

Keterangan Mari kita mulai dengan contoh caption

`\begin {figure }[h]`

`\caption {Example of a parametric plot (  $\sin (x)$ ,  $\cos(x)$ ,`

`x  $\sin (x)$  ) }`

`\centering`

```
\includegraphics[width=0.5 \textwidth] {spiral }

\end {figure }
```

Ini sangat mudah, cukup tambahkan `\caption {Some caption }` dan di dalam kawat gigi tulis teks yang akan ditampilkan. Penempatan keterangan bergantung pada tempat Anda menempatkan komando; Jika itu di atas kata-kata yang di bawah itu maka judulnya akan di atasnya, jika di bawah maka judul juga akan ditetapkan di bawah gambar. Teks juga bisa ditempatkan tepat setelah gambar. Paket `sidecap` menggunakan kode yang mirip dengan yang ada pada contoh sebelumnya untuk mencapai hal ini.

```
\documentclass {article }

\usepackage[rightcaption] {sidecap }

\usepackage {graphicx } %package to manage images

\graphicspath { {images/ } }

\begin {SCfigure }[0.5][h]

\caption {Example of a parametric plot.
This caption will be on the right }

\includegraphics[width=0.6 \textwidth] {spiral }

\end {SCfigure }
```

Ada dua perintah baru

```
\ usepackage [rightcaption] {sidecap }
```

Seperti yang mungkin Anda harapkan, baris ini akan mengimpor paket yang dinamai `sidecap`, namun ada parameter tambahan: `rightcaption`. Parameter ini menetapkan penempatan judul di sebelah kanan gambar, Anda juga dapat menggunakan `leftcaption`. Dalam dokumen seperti `outercaption` dan `innercaption` juga tersedia. Nama-nama ini bersifat deskriptif.

```
\ begin {SCfigure } [0.5] [h] \ end {SCfigure }
```

Mendefinisikan sebuah lingkungan yang mirip dengan gambar. Parameter pertama adalah lebar keterangan relatif terhadap ukuran gambar, seperti yang dideklarasikan di dalam dokumen. Parameter kedua `h` bekerja sama persis seperti pada lingkungan gambar. Lihat bagian penempatan untuk informasi lebih lanjut.

Anda bisa melakukan pengelolaan format caption yang lebih canggih. Periksa bagian bacaan lebih lanjut untuk referensi. Label dan referensi silang

Angka, sama seperti elemen lainnya dalam dokumen LaTeX (persamaan, tabel, plot, dll) dapat dirujuk dalam teks. Ini sangat mudah, cukup tambahkan label ke gambar atau lingkungan `SCfigure`, kemudian gunakan label itu untuk merujuk gambarnya.

```
\begin {figure }[h]
\centering
\includegraphics[width=0.25 \textwidth] {mesh }
\caption {a nice plot }
\label {fig:mesh1 }
\end {figure }
```

As you can see in the figure \ref {fig:mesh1 }, the  
function grows near 0. Also, in the page \pageref {fig:mesh1 }  
is the same example.

Ada tiga perintah yang menghasilkan rujukan silang dalam contoh ini.

```
\ label {fig: mesh1 }
```

Ini akan menetapkan label untuk gambar ini. Karena label dapat digunakan dalam beberapa jenis elemen di dalam dokumen, sebaiknya gunakan awalan, seperti ara: pada contoh.

```
\ ref {fig: mesh1 }
```

Perintah ini akan memasukkan nomor yang ditugaskan ke gambar. Ini otomatis dihasilkan dan akan diperbarui jika memasukkan beberapa gambar lain sebelum yang direferensikan.

```
\pageref{fig: mesh1 }
```

Ini akan mencetak nomor halaman dimana gambar yang direferensikan akan muncul.

Keterangan adalah wajib untuk referensi gambar.

Karakteristik hebat lainnya dalam dokumen LaTeX adalah kemampuan untuk menghasilkan daftar angka secara otomatis. Ini sangat mudah.

```
\Daftar Gambar
```

Perintah ini hanya bekerja pada gambar teks, karena menggunakan judul di tabel.

Contoh di atas mencantumkan gambar di artikel ini. Catatan Penting: Bila menggunakan referensi silang, proyek LaTeX Anda harus dikompilasi dua kali, jika tidak, rujukan, rujukan halaman dan tabel angka tidak akan berfungsi.

Membangkitkan citra beresolusi tinggi dan resolusi rendah

Sejauh ini saat menentukan nama file gambar di perintah `\includegraphics`, kami telah menghapus ekstensi file. Namun, itu tidak perlu, meski sering berguna. Jika ekstensi file dihilangkan, LaTeX akan mencari format gambar yang didukung di direktori tersebut, dan akan mencari berbagai ekstensi dalam urutan default (yang dapat dimodifikasi). Hal ini berguna untuk beralih antara lingkungan pengembangan dan produksi. Dalam lingkungan pengembangan (saat artikel / laporan / buku masih dalam proses), sebaiknya gunakan gambar dengan resolusi rendah (biasanya dalam format .png) untuk penyusunan preview dengan cepat. Di lingkungan produksi (saat versi final artikel / laporan / buku diproduksi), sebaiknya sertakan gambar dengan resolusi tinggi.

Hal ini dilakukan oleh

Tidak menentukan ekstensi file dalam perintah `\includegraphics`, dan  
Menentukan ekstensi yang diinginkan dalam basa-basi.

Jadi, jika kita memiliki dua versi gambar, `venndiagram.pdf` (resolusi tinggi) dan `venndiagram.png` (resolusi rendah), maka kita bisa memasukkan baris berikut dalam pembukaan untuk menggunakan versi .png saat mengembangkan laporan -

```
\DeclareGraphicsExtensions { .png,.pdf }
```

Perintah di atas akan memastikan bahwa jika dua file ditemukan dengan nama dasar yang sama namun ekstensi yang berbeda (misalnya `venndiagram.pdf` dan `venndiagram.png`), maka versi `.png` akan digunakan terlebih dahulu, dan dalam ketiadaan versi `.pdf` akan digunakan, ini juga merupakan ide bagus jika beberapa versi resolusi rendah tidak tersedia.

Begitu laporan telah dikembangkan, untuk menggunakan versi resolusi tinggi `.pdf`, kita dapat mengubah baris dalam basa-basi yang menentukan urutan pencarian ekstensi ke

```
\DeclareGraphicsExtensions { .pdf,.png }
```

Memperbaiki teknik yang dijelaskan pada paragraf sebelumnya, kita juga dapat menginstruksikan LaTeX untuk menghasilkan versi resolusi rendah `.png` gambar dengan cepat saat menyusun dokumen jika ada PDF yang belum dikonversi ke PNG. Untuk mencapainya, kita bisa memasukkan yang berikut dalam basa-basi setelah `\usepackage {graphicx}`

```
\usepackage {epstopdf}
```

```
\epstopdfDeclareGraphicsRule { .pdf } { png } { .png } { convert
```

```
#1 \OutputFile }
```

```
\DeclareGraphicsExtensions { .png,.pdf }
```

Jika `venndiagram2.pdf` ada tapi tidak `venndiagram2.png`, file `venndiagram2-pdf-convert-to.png` akan dibuat dan dimuat di tempatnya. Perintah yang mengonversi `# 1` bertanggung jawab atas konversi dan parameter tambahan dapat dilewatkan antara konversi dan `# 1`. Misalnya - `convert -density 100 # 1`.

Ada beberapa hal penting yang perlu diingat:

Agar konversi otomatis berhasil, kita perlu memanggil `pdflatex` dengan opsi `-shell-escape`.

Untuk versi produksi terakhir, kita harus memberi komentar pada \



`epstopdfDeclareGraphicsRule`, sehingga hanya file PDF beresolusi tinggi yang dimuat. Kita juga perlu mengubah urutan prioritas.

### Tentang tipe gambar di LaTeX

#### Getah

Saat kompilasi dengan latex, kita hanya bisa menggunakan gambar EPS, yaitu format vektor. `pdflatex` Jika kita kompilasi menggunakan "`pdflatex`" untuk menghasilkan PDF, maka kita bisa menggunakan sejumlah format gambar -

JPG: Pilihan terbaik jika kita ingin menyisipkan foto

PNG: Pilihan terbaik jika kita ingin memasukkan diagram (jika versi vektor tidak dapat dihasilkan) dan tangkapan layar

PDF: Meskipun kita terbiasa melihat dokumen PDF, PDF juga bisa menyimpan gambar

EPS: Citra EPS bisa diikutkan menggunakan paket `epstopdf` (kita hanya perlu menginstal

paketnya, kita tidak perlu menggunakan `\usepackage { }` untuk memasukkannya ke dalam dokumen kami.)

#### Format vektor atau format Bit-map?

Gambar bisa berupa format vektor format bit-map. Umumnya kita tidak perlu khawatir tentang hal itu, tapi jika kita memang mengetahui format gambarnya, kita bisa menggunakan informasi itu untuk memilih format gambar yang sesuai untuk disertakan dalam dokumen latex kita. Jika kita memiliki gambar dalam format vektor, kita harus mencari PDF atau EPS. Jika kita memilikinya dalam format bit-map, kita harus memilih JPG atau PNG, karena menyimpan gambar bit-map dalam PDF atau EPS membutuhkan banyak ruang disk.

## CHAPTER 7

---

## GENERATE A TABLE OF CONTENTS

---



## CHAPTER 8

---

### ADDING BIBLIOGRAPHY

---



## CHAPTER 9

---

### ADDING FOOTNOTES

---



## CHAPTER 10

---

### CREATE TABLES WITH LATEX

---





## CHAPTER 11

---

### USING TABLES THE SMART WAY

---



## CHAPTER 12

---

# PLOTS VISUALIZING YOUR DATA WITH PGFGPLOTS

---



## CHAPTER 13

---

## ELECTRIC CIRCUIT WITH CIRCUITIKZ

---



## CHAPTER 14

---

### SOURCE CODE HIGHLIGHTING IN LATEX USING THE LISTING PACKAGE (LISTING)

---





## REFERENCES

---

1. J. S. Kilby, "Invention of the Integrated Circuit," *IEEE Trans. Electron Devices*, **ED-23**, 648 (1976).
2. R. W. Hamming, *Numerical Methods for Scientists and Engineers*, Chapter N-1, McGraw-Hill, New York, 1962.
3. J. Lee, K. Mayaram, and C. Hu, "A Theoretical Study of Gate/Drain Offset in LDD MOSFETs" *IEEE Electron Device Lett.*, **EDL-7**(3). 152 (1986).
4. A. Berenbaum, B. W. Colbry, D.R. Ditzel, R. D Freeman, and K.J. O'Connor, "A Pipelined 32b Microprocessor with 13 kb of Cache Memory," in Int. Solid State Circuit Conf., Dig. Tech. Pap., p. 34 (1987).



## REFERENCES

---

- [Kil76] J. S. Kilby, "Invention of the Integrated Circuit," *IEEE Trans. Electron Devices*, **ED-23**, 648 (1976).
- [Ham62] R. W. Hamming, *Numerical Methods for Scientists and Engineers*, Chapter N-1, McGraw-Hill, New York, 1962.
- [Hu86] J. Lee, K. Mayaram, and C. Hu, "A Theoretical Study of Gate/Drain Offset in LDD MOSFETs" *IEEE Electron Device Lett.*, **EDL-7**(3). 152 (1986).
- [Ber87] A. Berenbaum, B. W. Colbry, D.R. Ditzel, R. D Freeman, and K.J. O'Connor, "A Pipelined 32b Microprocessor with 13 kb of Cache Memory," in *Int. Solid State Circuit Conf.*, Dig. Tech. Pap., p. 34 (1987).

