

CERDAS MENGUASAI LATEX

CERDAS MENGUASAI LATEX

Dalam 24 Jam

Rolly M. Awangga
Informatics Research Center



Kreatif Industri Nusantara

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.’
Imam Syafi’i*

CONTRIBUTORS

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

1 Editor dan Compiler	1
2 Pengaturan Paragraf	17
3 Menambahkan Gambar dan Tabel	23
4 Notasi Matematika Pada Latex	27
5 Sistematika Dokumen	35
6 Referens Pada Latex	39

DAFTAR ISI

Daftar Gambar	xiii
Daftar Tabel	xv
Foreword	xix
Kata Pengantar	xxi
Acknowledgments	xxiii
Acronyms	xxv
Glossary	xxvii
List of Symbols	xxix
Introduction	xxxi
<i>Rolly Maulana Awangga, S.T., M.T.</i>	

1	Editor dan Compiler	1
1.1	Mengenai .tex	1
1.1.1	Keuntungan Latex	2
1.1.2	Perbedaan LateX dari program pengolahan data lainnya	2
1.1.3	Kekurangan pada LateX	2
		ix

1.2	Compiler	2
1.3	Hello LATEX	3
1.4	Mengedit Menggunakan Editor LaTeX	3
1.4.1	Miktex	3
1.4.2	Texworks	8
2	Pengaturan Paragraf	17
2.1	Pembagian bab	17
2.2	Format Cetak	18
2.3	Tanda petik	18
2.4	Kode Program	18
2.5	Menambahkan Spesial Karakter	19
2.6	Menambahkan Chapter	19
2.7	Perintah Include dan Input	20
2.7.1	Include	20
2.7.2	Input	21
3	Menambahkan Gambar dan Tabel	23
3.1	Gambar	23
3.1.1	Posisi Gambar	24
3.2	Membuat Tabel	24
4	Notasi Matematika Pada Latex	27
4.1	Membuat Rumus dengan LaTeX	27
4.2	Penulisan Notasi Matematika	27
4.3	Jenis Font Dalam Notasi Matematika	28
4.4	Rumus Dasar	28
4.4.1	Rumus Pecahan	28
4.4.2	Rumus Akar	29
4.5	Perumusan Menggunakan Superscripts dan Subscripts	29
4.6	Perumusan Array dan Matriks	30
4.7	Perumusan Vektor	30
4.8	Kombinasi Penggunaan Rumus	31
4.9	Penulisan Tata Letak Rumus	32
4.10	Penulisan Simbol	32
5	Sistematika Dokumen	35
5.1	Sistematika Dokumen	35

5.2	Alignment	36
5.3	Document class	36
5.4	Costum Command	37
5.5	Renewcommand	37
5.6	Penomoran	38
5.7	Daftar Pustaka	38
6	Referens Pada Latex	39
6.1	Membuat Penomoran Referensi	39
	Daftar Pustaka	45
	Index	47

DAFTAR GAMBAR

1.1	Ini adalah installer Miktex	4
1.2	Ini adalah Halaman persetujuan install	4
1.3	Ini adalah Halaman istallasi scope	5
1.4	Ini adalah Halaman penentuan ukuran kertas	5
1.5	Ini adalah Halaman penentuan direktori	6
1.6	Ini adalah Halaman memulai install	6
1.7	Ini adalah Proses instalasi	7
1.8	Ini adalah Halaman Execute	7
1.9	Instalasi sudah selesai	8
1.10	Buka Editor Latex	9
1.11	Open File	10
1.12	Pilih file main.tex	11
1.13	main.tex dan main.pdf	11

1.14	Pilih chapter yang ingin di edit	12
1.15	Pilih Save all	13
1.16	Proses Melakukan Compile	13
1.17	Console Output	14
1.18	Errors	15
2.1	Pembagian Bab.	18
2.2	Menambahkan Chapter Baru	19
2.3	Menambahkan Perintah Input Chapter	20
6.1	Ini adalah Contoh Penomoran Referensi	40
6.2	Ini adalah Halaman Google Scholar	40
6.3	Ini adalah Tanda proses awal mengambil reference	40
6.4	Ini adalah Pilihan mengutip	41
6.5	Ini adalah Script BibTex	41
6.6	Ini adalah Direktori pekerjaan	42
6.7	Ini adalah Reference.bib	42
6.8	Ini adalah Proses pemilihan sumber	43
6.9	Ini adalah Compile pdflatex	43
6.10	Ini adalah Compile BibTex	44

DAFTAR TABEL

3.1	Latex Table	25
3.2	Latex Table	26
5.1	Small Table	35

Listings

1.1	Perintah kompilasi latex keluaran pdf	3
1.2	Contoh listing kode hello Latex	3
2.1	Contoh kalimat dalam tanda petik pada Latex	18
2.2	Menambahkan kode program	18
2.3	Contoh kode untuk menambahkan karakter spesial	19
2.4	Penggunaan perintah input untuk menambahkan chapter	20
2.5	Penggunaan perintah includegraphics	20
2.6	Penggunaan perintah include pada main.tex	21
2.7	Penggunaan perintah input untuk menambahkan chapter	21
3.1	Contoh kode untuk menambahkan gambar	23
3.2	Contoh Perintah Membuat Tabel	24
3.3	Memberikan Perintah Tabel	25
4.1	Notasi Matematika Dalam Paragraf	27
4.2	Jenis Font Matematis	28
4.3	Penggunaan Rumus Dasar	28
4.4	Penggunaan Rumus Pecahan	29
4.5	Penggunaan Rumus Akar	29
4.6	Penggunaan Superscripts dan Subscripts	29

4.7	Perintah Pada Superscripts dan Subscripts	29
4.8	Penulisan Array atau Matriks	30
4.9	Penulisan Vektor	30
4.10	Contoh Kombinasi Rumus Sigma	31
4.11	Contoh Kombinasi Rumus	31
4.12	Contoh Penulisan Tata Letak Dalam Rumus	32
4.13	Contoh Penulisan Simbol	32
5.1	Contoh Perintah Membuat Rata Kiri	36
5.2	Contoh Perintah Membuat Rata Kanan	36
5.3	Format Ukuran Font	36
5.4	Struktur Awal Document Pada Latex	37
5.5	Perintah Section pada New Command	37
5.6	Perintah Renewcommand	37
5.7	Memberikan Perintah Numbering	38
5.8	Menambahkan kode perintah bullets	38
5.9	Memberikan Perintah Daftar Pustaka	38
6.1	Penggunaan perintah cite untuk reference	41

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat
Februari, 2019*

ACKNOWLEDGMENTS

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.

ACRONYMS

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association
EPS	Encapsulated PostScript
HTBP	Here Tab Bottom Paragraph
IDE	Integrated Development Environment
GPL	General Public License

GLOSSARY

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasis *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald
compile	Analisis pada kode program untuk mengubah komputer bentuk langsung eksekusi dari program
script	Bahasa yang digunakan untuk menerjemahkan setiap perintah dalam situs pada saat diakses.
listing	Teks yang berisi daftar item berupa perintah langkah-langkah membuat program
bullets	Satu tanda yang dipakai untuk memberikan gambar atau lambang pada latex

SYMBOLS

- A Amplitude
- $\&$ Propositional logic symbol
- a Filter Coefficient

- \mathcal{B} Number of Beats

INTRODUCTION

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

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

BAB 1

EDITOR DAN COMPILER

1.1 Mengenal .tex

Pertama pahami dulu bagaimana badan isi file .tex yang akan kita kerjakan. Download atau lihat salah satu file latex yang akan kita kerjakan. Untuk mengisi latex kita harus mengisinya di dalam komponen yang merupakan tag dengan pembuka begin dan diakhiri dengan end. Kemudian kenali bagian buku terdiri dari part, chapter dan section. Part itu bisa kita andaikan bab, chapter sub bab, dan section adalah bagian.

Kita bisa memisahkan isi dari latex dengan perintah input kemudian di dalam kurung kurawal letak file .tex yang akan kita masukkan kedalam file utama latex tersebut.

LATEX merupakan program pengolahan kata atau sistem persiapan pembuatan dokumen untuk pengetikan sistem TeX, yang dinamakan berdasarkan gaya penulisannya sebagai LaTeX. Nama LaTeX itu sendiri hanya mengacu pada bahasa penulisan yang digunakan pada sebuah dokumen, bukan pada editor yang digunakan untuk menulis dokumen tersebut. Untuk membuat dokumen dalam format LaTeX, sebuah file berformat .tex harus dibuat menggunakan semacam text editor. Walaupun, banyak text editor yang dapat digunakan untuk membuat dokumen LaTeX, beberapa text editor sengaja dibuat khusus untuk menggunakan bahasa LaTeX.

1.1.1 Keuntungan Latex

1. Tersedianya beberapa program untuk melihat hasil pemrosesan latex yang dapat menampilkannya persis seperti hasil cetakan dengan printer
2. Penulisan rumus matematis dapat dilakukan dengan cara sangat mudah dan profesional
3. Banyak jurnal internasional yang menerima artikel artikel dalam format tex
4. Pemakai hanya perlu belajar sedikit perintah yang mudah dipahami yang menyatakan struktur logis sebuah dokumen
5. Latex mendorong pengarang untuk menulis naskah yang tersusun dengan baik

1.1.2 Perbedaan LateX dari program pengolahan data lainnya

1. Latex menggunakan perintah yang diawali dengan backslash sedangkan untuk program olah data lain tidak ada.
2. Peraturan diLateX umumnya bersifat otomatis
3. Peraturan simbol matematika cukup dengan menekan epsilon yang sudah tersedia

1.1.3 Kekurangan pada LateX

1. Tidak memiliki user-friendly seperti MS Word
2. Tidak cocok untuk dokumen yang singkat karena hanya biasanya penuh dengan simbol matematika
3. latex menggunakan perintah yang sudah disesuaikan dengan penggunaannya.

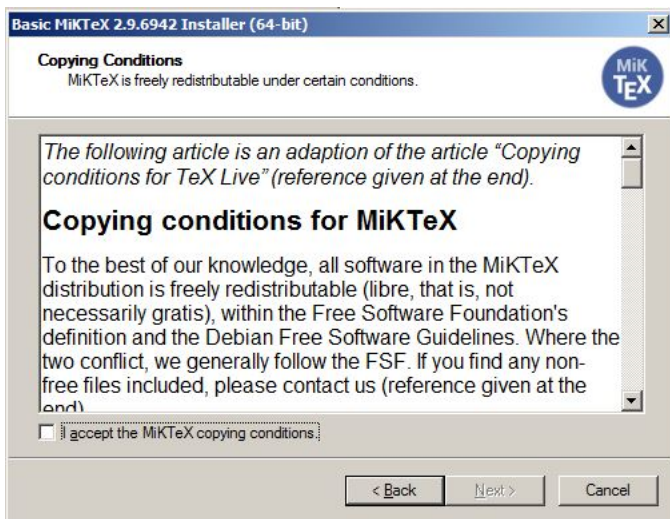
1.2 Compiler

Kemudian untuk dapat menuliskan kode LaTeX kita harus menggunakan editor LaTeX. Oleh karena itu pastikan kita sudah meng-install aplikasi editor LaTeX seperti texworks, texmaker, winedt dll. Untuk dapat melihat perintah yang sudah kita lakukan, kita harus melakukan compile. *Compile* merupakan proses eksekusi yang dilakukan untuk melihat hasil dari perintah yang telah kita buat dalam LaTeX yang dapat kita preview dalam bentuk file dengan ekstensi PDF. Editor LaTeX yang saya praktekan disini adalah texmaker. Kemudian untuk dapat menuliskan kode LaTeX kita harus menggunakan editor LaTeX. Oleh karena itu pastikan kita sudah meng-install aplikasi editor LaTeX seperti texworks, texmaker, winedt dll. Untuk dapat melihat perintah yang sudah kita lakukan, kita harus melakukan compile. *Compile* merupakan proses eksekusi yang dilakukan untuk melihat hasil dari perintah yang telah kita buat

Name ^	Date modified	Type
jurnal	11/03/2019 13:43	File folder
Keleketex	11/03/2019 13:49	File folder
yogi	20/12/2018 13:01	File folder
basic-miktex-2.9.6942-x64	01/02/2019 9:38	Application
Git-2.21.0-32-bit	06/03/2019 14:03	Application
pkm pid line follower java	14/01/2019 10:17	Text Document
pkm usulan 2018	08/01/2019 8:35	Microsoft Excel W
Proposal PKM-KC Christopel	08/01/2019 8:25	Microsoft Word D
seleksipenerimaanbeasiswahp	20/12/2018 8:55	WinRAR archive
Sublime Text Build 3176 Setup	20/12/2018 12:25	Application
tutorial	11/03/2019 13:00	Text Document

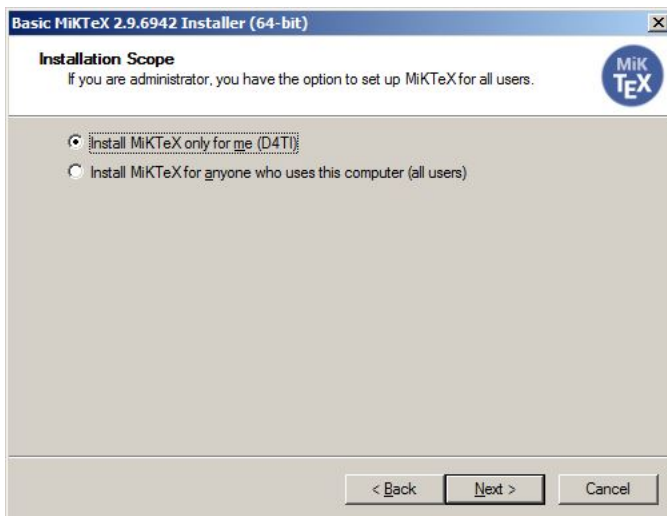
Gambar 1.1 Ini adalah installer Miktex

2. *Double-click* pada installer Miktex seperti pada gambar 1.1
3. Maka akan muncul halaman awal installer seperti pada gambar 1.2

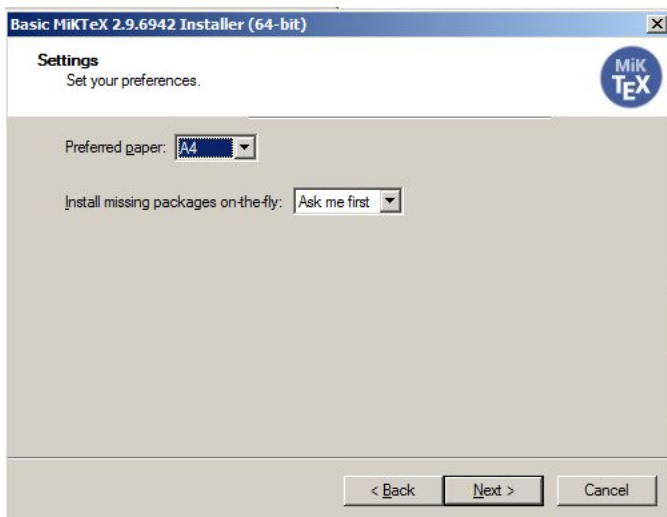


Gambar 1.2 Ini adalah Halaman persetujuan install

4. Klik *Next* maka akan muncul Halaman instalasi scope seperti pada gambar 1.3
5. Klik *Next* maka akan muncul Halaman penentuan ukuran kertas seperti pada gambar 1.4

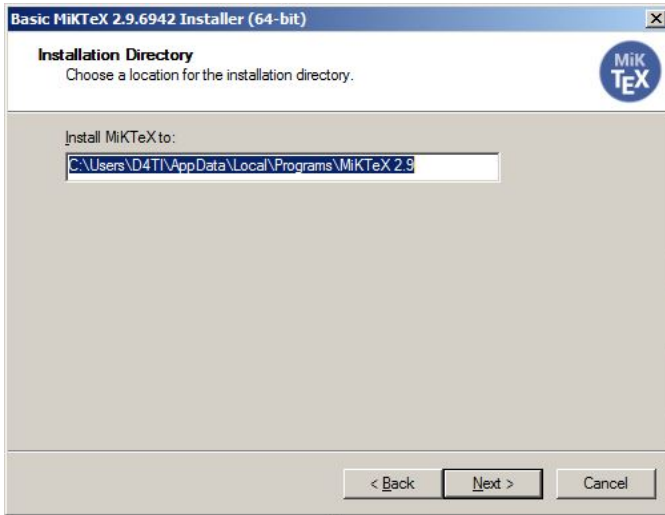


Gambar 1.3 Ini adalah Halaman istallasi scope



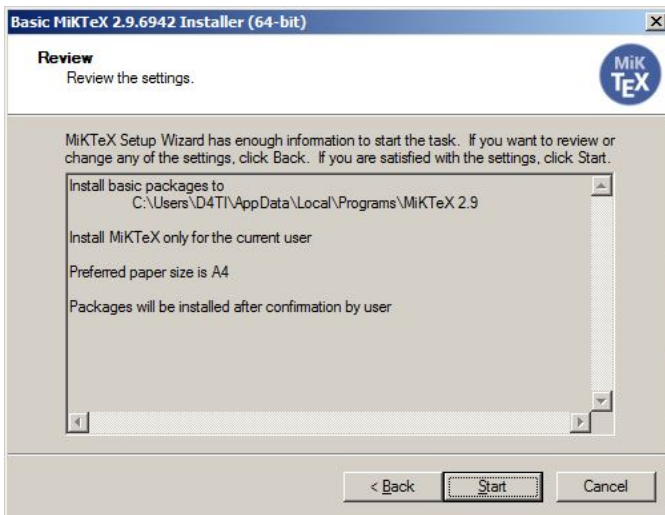
Gambar 1.4 Ini adalah Halaman penentuan ukuran kertas

6. Klik *Next* untuk memilih direktori penyimpanan program seperti pada gambar 1.5



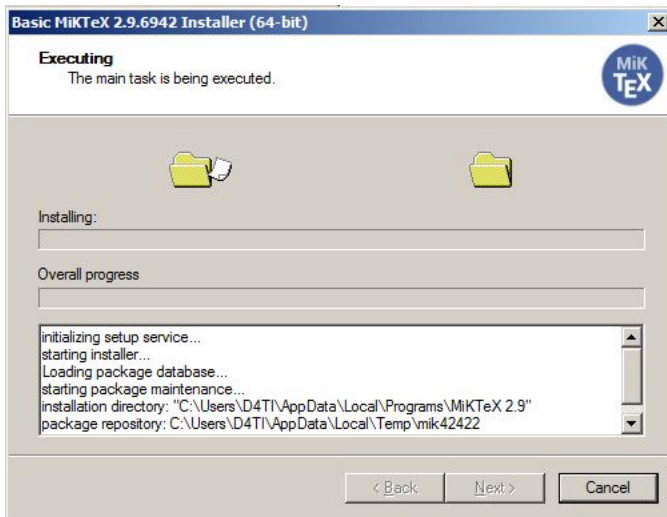
Gambar 1.5 Ini adalah Halaman penentuan direktori

7. Klik *Next* maka akan muncul halaman untuk memulai proses install seperti pada gambar 1.6



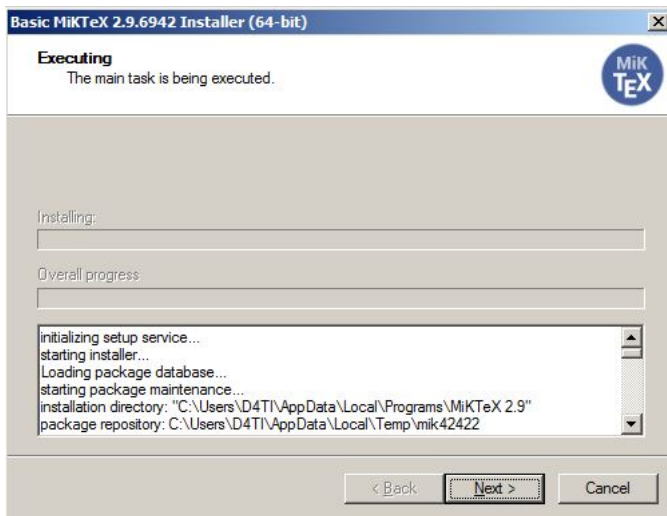
Gambar 1.6 Ini adalah Halaman memulai install

8. Klik *Start* maka proses installasi dimulai seperti pada gambar 1.7



Gambar 1.7 Ini adalah Proses installasi

9. Jika sudah selesai proses installasinya maka akan muncul halaman execute seperti pada gambar 1.8



Gambar 1.8 Ini adalah Halaman Execute

10. Klik *Next* maka akan muncul halaman seperti pada gambar 1.9 maka instalasi sudah selesai dilakukan dan program siap digunakan.

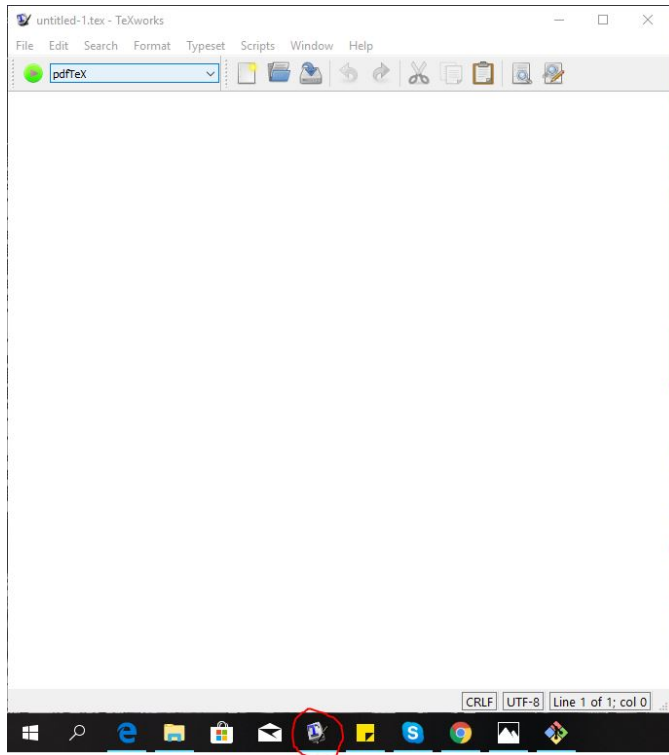


Gambar 1.9 Instalasi sudah selesai

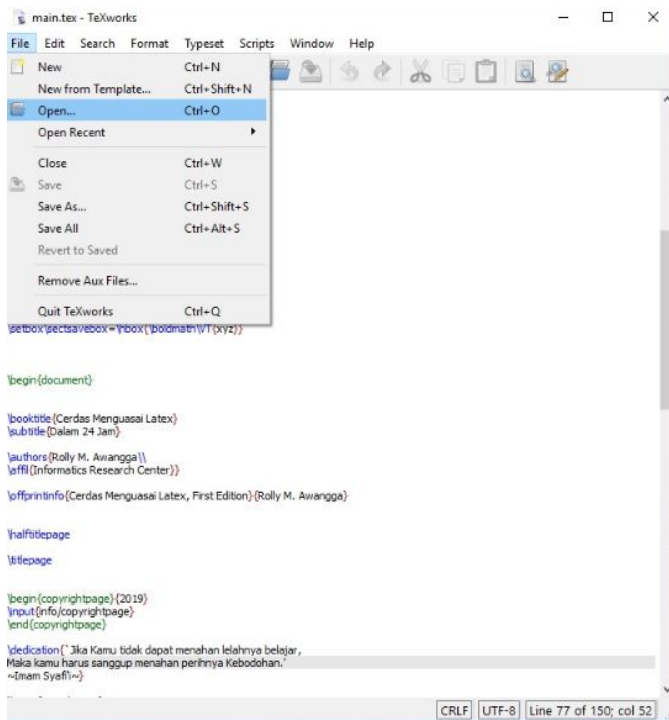
1.4.2 Texworks

Bagaimana cara mengedit sebuah file pada LaTeX? Berikut penjelasannya :

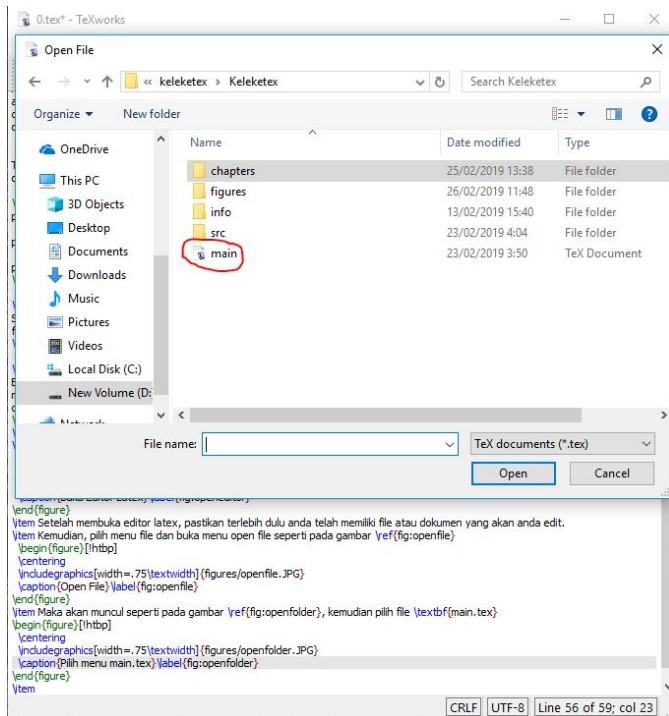
1. Pertama pastikan anda telah meng-install Editor latex yang akan digunakan (Editor latex yang digunakan disini adalah TeXworks)
2. Buka Editor latex yang sudah kalian install 1.10
3. Setelah membuka editor latex, pastikan terlebih dulu anda telah memiliki file atau dokumen yang akan anda edit.
4. Kemudian, pilih menu file dan buka menu open file seperti pada gambar 1.11
5. Setelah memilih menu file kemudian buka file main.tex seperti pada gambar 1.12, kemudian pilih file **main.tex**
6. Main.tex merupakan file utama dalam sebuah dokumen atau laporan ilmiah yang berisi file-file tambahan dari direktori lain. Jika anda membuka file main.tex, maka kita dapat melihat hasil atau isi dari laporan dalam bentuk preview beres-tesensi PDF pada main.pdf seperti pada gambar 1.13
7. Setelah itu pastikan terlebih dahulu file dari chapter mana yang ingin kita rubah atau tambahkan



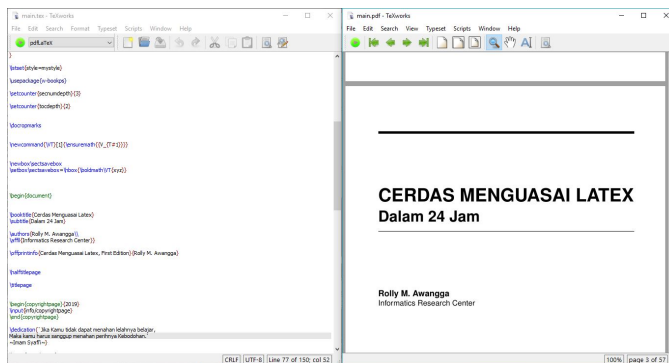
Gambar 1.10 Buka Editor Latex



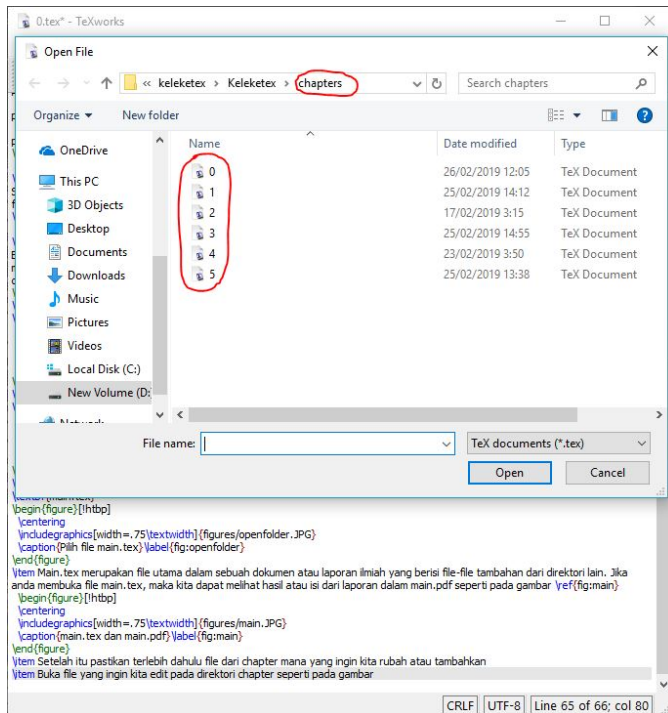
Gambar 1.11 Open File



Gambar 1.12 Pilih file main.tex

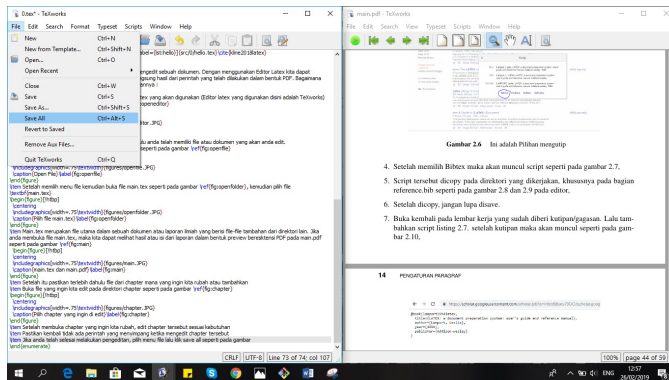


Gambar 1.13 main.tex dan main.pdf



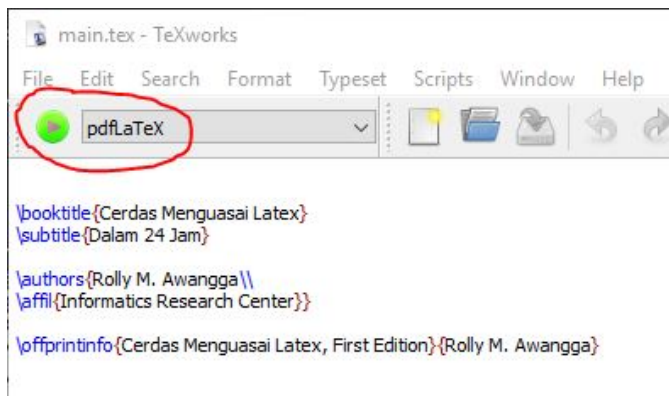
Gambar 1.14 Pilih chapter yang ingin di edit

8. Buka file yang ingin kita edit pada direktori chapter seperti pada gambar 1.14
9. Setelah membuka chapter yang ingin kita rubah, edit chapter tersebut sesuai kebutuhan
10. Pastikan kembali tidak ada perintah yang menyimpang ketika mengedit chapter tersebut
11. Jika anda telah selesai melakukan pengeditan, pilih menu file lalu klik save all seperti pada gambar 1.15



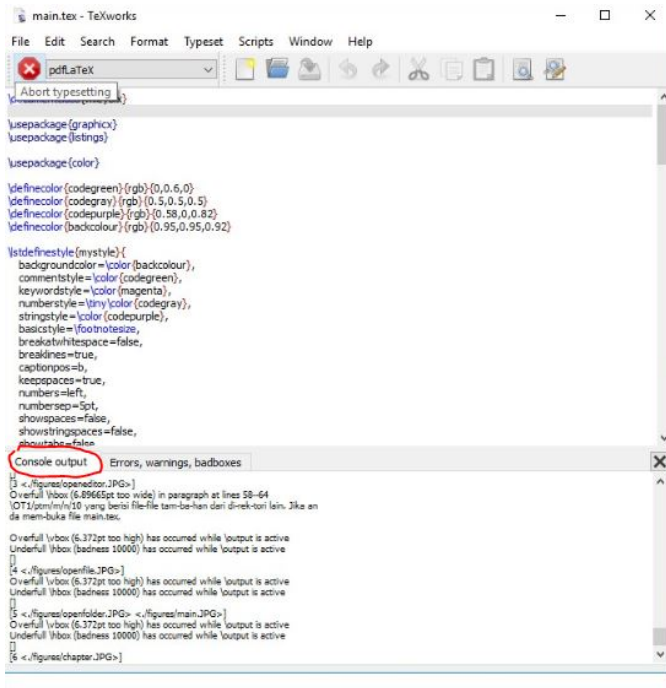
Gambar 1.15 Pilih Save all

12. Setelah itu agar perintah yang telah kita buat dapat dijalankan, buka kembali main.tex lalu pilih button *typeset* dengan format *pdfLaTeX* untuk melakukan compile seperti pada gambar 1.16



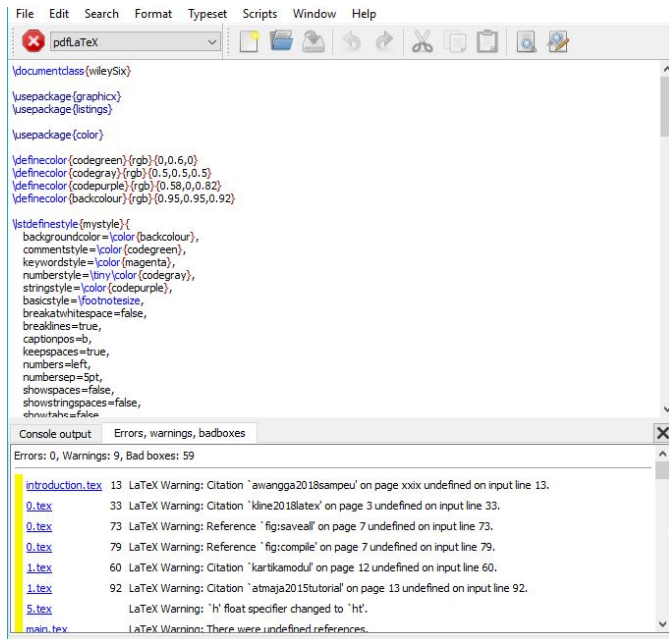
Gambar 1.16 Proses Melakukan Compile

13. Setelah melakukan compile pastikan kembali tidak ada perintah yang error pada *console outputnya* seperti pada gambar 1.17



Gambar 1.17 Console Output

14. Jika terjadi error kita dapat melihat error tersebut dan memperbaikinya pada bagian *Error;Warnings,Badboxes* seperti pada gambar 1.18



Gambar 1.18 Errors

BAB 2

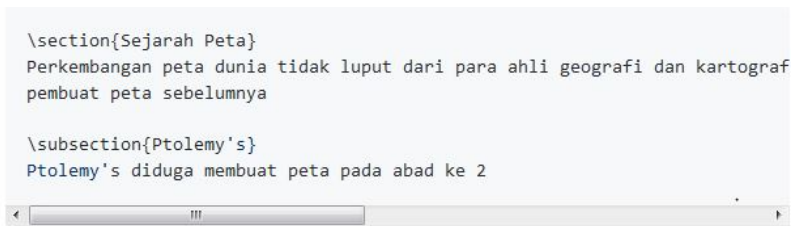
PENGATURAN PARAGRAF

2.1 Pembagian bab

Secara default pembagian bab pada latex menggunakan perintah *section*, *subsection*, *subsubsection* dan *subsubsubsection*. Untuk mengatur kedalaman suatu dokumen pada bab bab tertentu, kita dapat menggunakan perintah berikut ini pada bagian Preamble : `setcounter{secnumdepth}` `setcounter{tocdepth}`

Opsi yang digunakan pada syntax `secnumdepth` pada perintah `verbcounter=` seperti perintah diatas, berarti Anda telah merubah kedalaman bab yang Anda perbaharui sampai dengan level 5 yaitu *section* – *subsection* – *subsubsection* – *paragraph* – *subparagraph*.

Sedangkan pada perintah dari opsi `tocdepth` berfungsi untuk membuat table of contents atau menampilkan kedalaman bab sampai dengan level 5, namun jika tidak di setel maka pada bagian level 3 kebawah tidak akan dapat ditampilkan pada bagian `toc 2.1`.



Gambar 2.1 Pembagian Bab.

2.2 Format Cetak

Pada format LaTeX teks mempunyai bentuk plaintext, yang artinya teks tersebut belum diformat. Pada proses formatting teks dapat dilakukan dengan bahasa tersendiri yaitu bahasa markup. Hal paling mendasar antara lain cetak tebal, miring dan garis bawah. Cetak tebal menggunakan perintah *textbf*, cetak miring menggunakan perintah *textit* dan garis bawah menggunakan perintah *underline*.

2.3 Tanda petik

Tanda petik di Latex menggunakan petik miring dan petik satu. Petik miring biasanya berada pada sebelah angka satu di keyboard dan diakhiri petik satu. Ingat fungsi tanda petik hanya untuk melakukan quote atau pengutipan langsung. Untuk istilah bahasa Inggris gunakan miring atau italic.

```
1 'kalimat dalam tanda petik'
```

Listing 2.1 Contoh kalimat dalam tanda petik pada Latex

2.4 Kode Program

Agar kita dapat memasukan kode program, kita dapat menggunakan perintah *lstlisting*. Perintah ini berfungsi untuk memasukkan atau menambahkan kode program apapun ke dalam file yang terpisah. Untuk memasukan perintah *lstlisting* kita perlu menulis parameter *caption* dan *label* untuk memberikan penjelasan keterangan kode program dan sebagai sumber referensi dari label kode program.

```

1 \begin{lstlisting}[caption=Contoh kalimat dalam tanda petik di Latex,
   label={lst:tandapetik}]
2 'kalimat dalam tanda petik'
3 \end{lstlisting}

```

Listing 2.2 Menambahkan kode program

2.5 Menambahkan Spesial Karakter

Untuk menambahkan karakter spesial pada LaTeX kita dapat menggunakan tanda *backslash* didepan karakter yang ingin kita tandai. Terdapat beberapa karakter yang tidak bisa langsung digunakan seperti tanda *ampersand*. Selain itu format pemberian kutipan pada LaTeX berbeda dengan pemberian kutipan pada editor lainnya, cara memasukkan karakter spesial menggunakan listing 2.3

```

1 \&
2 \%
3 \$
4 \#
5 \{ \}
6 \_
7 \"dalam petik\"
8 'dalam petik'
9 jika spesial karakter menjadi banyak atau satu baris gunakan verb
10 contoh :
11 \verb|%$'%&$&'%'%'&'%|

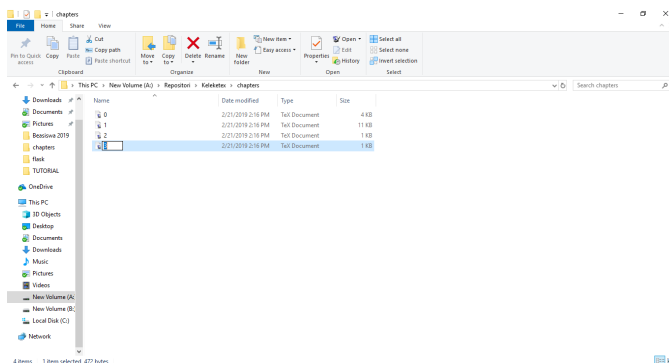
```

Listing 2.3 Contoh kode untuk menambahkan karakter spesial

2.6 Menambahkan Chapter

Berikut ini merupakan langkah-langkah untuk menambahkan *chapter* baru.

1. Pertama kita buat *chapter* baru pada repositori kita di folder *chapters*, seperti pada gambar 2.2.



Gambar 2.2 Menambahkan Chapter Baru

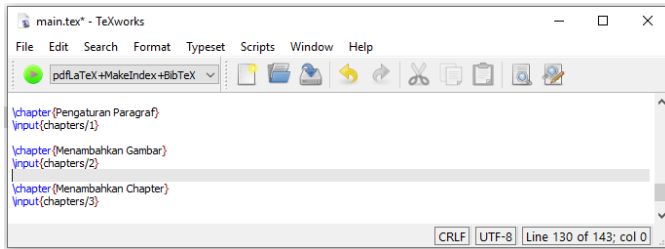
2. Kemudian kita tambahkan kode seperti pada *listing 2.7* yang berfungsi untuk memanggil *chapter* yang baru kita tambahkan pada file *main.tex* seperti pada gambar 2.3.

```

1 \chapter{Judul Bab}
2 \input{lokasi_chapter}

```

Listing 2.4 Penggunaan perintah input untuk menambahkan chapter



Gambar 2.3 Menambahkan Perintah Input Chapter

3. Terakhir, compile file `main.tex` untuk melihat chapter baru yang telah kita tambahkan pada file `main.pdf`.

2.7 Perintah Include dan Input

Ketika kita mengerjakan dokumen-dokumen besar menggunakan latex, tentunya kita akan memecah file menjadi beberapa bagian atau chapters agar isi dari dokumen tersebut lebih terorganisir atau terstruktur. Untuk itu latex memiliki 2 perintah untuk membantu kita semua agar dapat melakukan hal ini. Perintah tersebut tidak lain adalah perintah **include** dan **input**.

2.7.1 Include

Perintah **include** bisa kita gunakan pada batang tubuh dokumen dan pada bagian `main.tex`, untuk menyisipkan isi file yang lain. Perintah include bisa digunakan untuk menyisipkan file chapter pada `main.tex`, gambar, link dan sebagainya. Pada gambar misalnya, biasanya akan ditambahkan perintah *includegraphics* atau pada `main.tex` kita hanya perlu menambahkan perintah **include** pada chapter atau bagian yang ingin kita sisipkan. Hal yang perlu kita ketahui adalah latex akan memulai halaman baru sebelum memproses isi dari *namafile*. Contoh penggunaan perintah *includegraphics* dapat kita lihat pada gambar 2.3 dengan kode pemrograman yang dilakukan seperti pada listing 2.6

```

1 \begin{figure} [!htbp]
2   \centering
3   \includegraphics[width=.75\textwidth]{figures/inputchapter.png}
4   \caption{Menambahkan Perintah Input Chapter}\label{fig:inputchapter}
5 \end{figure}

```

Listing 2.5 Penggunaan perintah includegraphics

Dan berikut adalah contoh penggunaan perintah **include** pada main.tex dapat kita lihat dengan listing ??

```

1 \maketitle
2 \include{section/dedication}
3 \include{section/acknowledgements}
4 \include{section/abstract}
5
6 \begin{romanpages}
7 \tableofcontents
8 \listoffigures
9 \end{romanpages}
10
11 \include{section/syaratumum}
12 \include{section/chapter1}
13 \include{section/chapter2}
14 \include{section/chapter3}
15 \include{section/chapter4}
16 \include{section/chapter5}
17 \include{section/chapter6}
18
19 \appendix
20 \include{section/appendix1}
21 \include{section/appendix2}

```

Listing 2.6 Penggunaan perintah include pada main.tex

Dari contoh listing ?? dapat kita ketahui bahwa kode diawali dengan perintah *include* kemudian di ikuti dengan lokasi section dan chapter.

2.7.2 Input

Berbeda dengan perintah include, perintah ini tidak menyebabkan pergantian halaman pada saat menyisipkan isi file *nama-file*. Perintah **input** biasanya digunakan pada main.tex untuk menambahkan bagian-bagian file atau chapter-chapter pada sebuah dokumen dengan format yang sama seperti pada listing 2.7

```

1 \chapter{Judul Bab}
2 \input{lokasi_chapter}

```

Listing 2.7 Penggunaan perintah input untuk menambahkan chapter

Untuk penerapan atau pengimplementasian perintah input sendiri sudah cukup dijelaskan pada section menambahkan chapter sebelumnya.

BAB 3

MENAMBAHKAN GAMBAR DAN TABEL

3.1 Gambar

Untuk dapat menambahkan gambar pada latex biasanya hal yang harus dilakukan adalah mendeklarasikan penggunaan paket `graphicx` pada bagian preamble agar latex dapat menempatkan gambar sesuai dengan yang kita inginkan di dalam sebuah dokumen. Cara menambahkan gambar dapat dilihat seperti pada listing 3.1.

```
1 \begin{figure}[!htbp]
2   \includegraphics[width=0.75\textwidth]{figures/namagambar.JPG}
3   \caption{penjelasan keterangan gambar.}
4   \label{labelgambar}
5 \end{figure}
```

Listing 3.1 Contoh kode untuk menambahkan gambar

Beberapa hal yang harus kita perhatikan dari format perintah penambahan gambar :

1. File gambar yang ingin kita masukkan kedalam dokumen harus diletakkan pada direktori yang sama dengan direktori file dokumen (`.tex`) yang telah kita buat.

2. Panjang dan Lebar suatu gambar dapat diubah sesuai dengan yang kita inginkan. Perintah *width* berfungsi untuk mengatur lebar gambar, sedangkan perintah *height* berfungsi untuk mengatur tinggi gambar tersebut.
3. Dengan mengatur **width** dan **height** kita dapat memasukkan gambar meskipun gambar tersebut memiliki ukuran dimensi yang besar.
4. Format gambar standar latex adalah **.eps** (*Encapsulated PostScript*) namun kita juga dapat menggunakan format **.jpg**.

3.1.1 Posisi Gambar

Pengaturan posisi gambar dapat kita tentukan melalui 2 hal :

- Perataan tepian dokumen : Dari contoh yang sudah kita lihat pada listing kita bisa menambahkan perintah tertentu untuk mengubah posisi dari suatu gambar, misalkan kita menambahkan perintah *center* maka posisi gambar akan berubah ketengah sesuai dengan perintah yang diberikan.
- Huruf-huruf seperti [**htbp**] yang terdapat pada perintah diatas juga memiliki fungsi untuk mengatur gambar pada suatu halaman.
 - h** : berfungsi untuk meletakkan gambar persis ditempat perintah tersebut dituliskan didalam dokumen.
 - t** : berfungsi untuk meletakkan gambar dibagian atas halaman.
 - b** : berfungsi untuk meletakkan gambar dibagian bawah halaman.
 - p** : berfungsi untuk meletakkan gambar pada sebuah halaman khusus yang memuat hanya gambar itu saja.
- Saat kita menggunakan **h**, latex akan secara otomatis menempatkan gambar di-halaman baru jika tidak ada cukup ruang untuk memuat gambar tersebut dengan perintah yang telah dituliskan pada gambar.

3.2 Membuat Tabel

Latex memiliki banyak keunggulan dalam membuat dokumen selain membuat format penulisan dokumen menjadi akurat dan tertata dengan rapi, latex juga mempermudah pengguna dalam penulisan dokumen yakni tidak perlu memperhatikan penulisan karena latex secara otomatis dapat memperbaharuihnya.

Salah satu keunggulan latex yang lainnya yaitu dapat membuat tabel, untuk dapat membuat tabel kita harus menggunakan perintah *table*. Selain itu kita juga perlu menambahkan referensi pada tabel yang terdapat dalam kalimat berdasarkan labelnya. Contoh perintah pembuatan tabel dapat kita lihat seperti pada listing ??.

```
1 \begin{table}[h]
2 \caption{LateX Table}
```

```

3 \centering
4 \begin{tabular}{|c|c|}
5 \hline
6 \textbf{Bagian I}&\textbf{Bagian II}\\
7 \hline
8 Cover&judul\\
9 \hline
10 Kata pengantar&abstrak\\
11 \hline
12 daftar isi&si\\
13 \hline
14 kesimpulan&penutup\\
15 \hline
16 \end{tabular}
17 \label{table:permisalan}
18 \end{table}

```

Listing 3.2 Contoh Perintah Membuat Tabel

Hasil output :

Tabel 3.1 Latex Table

Bagian I	Bagian II
Cover	judul
Kata pengantar	abstrak
daftar isi	si
kesimpulan	penutup

Kemudian untuk menyesuaikan tabel sesuai dengan keinginan, kita juga dapat membuat tabel tanpa menggunakan garis tepi. Perintah untuk membuat tabel tanpa garis tepi dapat kita lihat pada listing ??

```

1 \begin{table}[h]
2 \caption{Latex Table}
3 \centering
4 \begin{tabular}{c|c}
5 \hline
6 \textbf{Bagian I}&\textbf{Bagian II}\\
7 \hline
8 Cover&judul\\
9 \hline
10 Kata pengantar&abstrak\\
11 \hline
12 daftar isi&si\\
13 \hline
14 kesimpulan&penutup\\
15 \hline
16 \end{tabular}
17 \label{table:permisalan}
18 \end{table}

```

Listing 3.3 Memberikan Perintah Tabel

Tabel 3.2 Latex Table

Bagian I	Bagian II
Cover	judul
Kata pengantar	abstrak
daftar isi	si
kesimpulan	penutup

Hasil output :

Bisa kita lihat bawah garis tepi pada sisi kiri dan kanan pada tabel tidak ada karena perintah yang dimasukan adalah perintah membuat tabel tanpa garis tepi. Perintah yang dimasukan yaitu `—c—c—` dimana perintah tersebut berfungsi untuk mengatur garis tepi pada tabel dan sesuai dengan kolom tabel yang diinginkan.

BAB 4

NOTASI MATEMATIKA PADA LATEX

4.1 Membuat Rumus dengan LaTeX

Sebagai aplikasi editor pengolah dokumen, LATEX memiliki kemampuan yang mampu menghasilkan dokumen berisi notasi-notasi matematis. Agar dapat menghasilkan dokumen yang berisikan notasi-notasi matematis maka kita harus berada dalam *Mathematics Environment*. Terdapat beberapa perintah yang bisa digunakan dalam membuat rumus pada latex. Kita dapat menggunakan perintah *equation*, *displaymath* ataupun menggunakan $. Kita juga dapat menyelipkan rumus didalam suatu kalimat di sebuah paragraf dengan menggunakan perintah $$.$$

4.2 Penulisan Notasi Matematika

Pada latex kita dapat menuliskan suatu notasi matematika yang cukup panjang dalam suatu paragraf baru. Penulisan Notasi Matematika dalam suatu paragraf dapat dilihat pada listing 4.1.

```
1 \begin{displaymath}  
2 .....  
3 .....  
4 .....  
5 .....  
6 .....  
7 .....  
8 .....  
9 .....  
10 .....
```

```
3 \end{displaymath}
```

Listing 4.1 Notasi Matematika Dalam Paragraf

4.3 Jenis Font Dalam Notasi Matematika

Ada beberapa perintah pada yang dapat digunakan untuk mengubah jenis font notasi matematis dalam latex. Beberapa perintah tersebut dapat kita lihat pada listing 4.2.

```
1 1. $\mathrm{x y z}$
2 2. $\mathsf{x y z}$
3 3. $\mathtt{x y z}$
4 4. $\mathit{x y z}$
5 5. $\mathbf{x y z}$
```

Listing 4.2 Jenis Font Matematis

Hasil output :

xyz

xyz

xyz

xyz

xyz

4.4 Rumus Dasar

Rumus dasar ini terdiri dari 3 notasi yaitu penjumlahan, pengurangan, dan perkalian. Contoh kode untuk rumus dasar bisa dilihat pada listing 4.3.

```
1 $$ a+b$$ %Notasi untuk penjumlahan
2 $$ a-b$$ %Notasi untuk pengurangan
3 $$ a \times b$$ %Notasi untuk perkalian
```

Listing 4.3 Penggunaan Rumus Dasar

Hasil output:

$$a + b$$

$$a - b$$

$$a \times b$$

4.4.1 Rumus Pecahan

Rumus pecahan yang dimaksud adalah notasi per pada pembagian. Contoh kode untuk rumus pecahan bisa dilihat pada listing 4.4.

```

1 $$ a/b $$ %Notasi per dengan output satu baris
2 $$ \frac {a}{b} $$ %Notasi per dengan output dua baris

```

Listing 4.4 Penggunaan Rumus Pecahan

Hasil output:

$$a/b$$

$$\frac{a}{b}$$

4.4.2 Rumus Akar

Rumus akar dapat dilihat pada listing 4.5. 4.5.

```

1 $$ \sqrt[a]{b} $$ %Notasi akar dengan output b akar a
2 $$ \sqrt{\sqrt[a]{a}} $$ %Notasi akar dengan output a akar di dalam akar

```

Listing 4.5 Penggunaan Rumus Akar

Hasil output:

$$\sqrt[a]{b}$$

$$\sqrt{\sqrt[a]{a}}$$

4.5 Perumusan Menggunakan Superscripts dan Subscripts

Penulisan *Superscripts* dan *Subscripts* biasanya digunakan untuk membuat sebuah rumus dengan menghasilkan pangkat diatas dan pangkat dibawah pada suatu rumus. Cara penulisan penggunaan ini adalah dengan menggunakan perintah **sp** dan perintah **sb**. Untuk contoh penerapan perintah *Superscripts* dan *Subscripts* dapat kita lihat pada listing 4.6.

```

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

```

Listing 4.6 Penggunaan Superscripts dan Subscripts

Hasil output :

$$y = x_1^2 + x_2^2$$

Atau kita juga dapat menggunakan perintah lain seperti pada listing 4.7.

```

1 \begin{displaymath}
2 f(x) = e^{\text{x}_1}
3 \end{displaymath}
4 %Penggunaan tanda " ^ " digunakan sebagai Superscripts
5 %Penggunaan tanda " _ " digunakan sebagai Subscripts

```

Listing 4.7 Perintah Pada Superscripts dan Subscripts

Hasil output :

$$f(x) = e^{x_1}$$

4.6 Perumusan Array dan Matriks

Dalam LaTeX, kita dapat menuliskan rumus sebuah array pada environment **tabular**. Perintah untuk membuat array dan matriks dapat kita lihat pada listing 4.8.

```
1 \begin{displaymath}
2 \left(
3 \begin{array}{rrr}
4 0 & 55 & 23 \\
5 34 & -83 & 68 \end{array}
6 \right)
7 \end{displaymath}
```

Listing 4.8 Penulisan Array atau Matriks

Hasil output :

$$\begin{pmatrix} 0 & 55 & 23 \\ 34 & -83 & 68 \end{pmatrix}$$

Ada beberapa hal yang perlu kita ketahui dalam penulisan rumus array atau matriks ini :

- Penulisan array memiliki kesamaan seperti saat membuat format tabel
- Perintah **”rrr”** berfungsi untuk menentukan posisi dari masing-masing komponen matriks tersebut
- Tanda kurung kurawal **”()”** berfungsi untuk mendefinisikan bagian kurung buka dan kurung tutup pada sebuah matriks

4.7 Perumusan Vektor

Dalam LaTeX, perumusan dengan format *vektor* kita dapat menuliskannya dengan perintah seperti pada listing 4.9.

```
1 \begin{displaymath}
2 \vec{variabel}
3 \end{displaymath}
```

Listing 4.9 Penulisan Vektor

Contoh kita akan mengubah variabel x kedalam satuan vektor. Maka hasil outputnya adalah :

$$\vec{x} = a + b$$

4.8 Kombinasi Penggunaan Rumus

Pada section ini kita akan mempelajari bagaimana mengkombinasikan sebuah rumus dari penulisan dasar rumus Subscript, Superscript, Akar Pangkat, Pecahan dan sejenisnya. Contoh pertama dapat kita lihat pada listing 4.10.

```

1  $$\sum^{\infty}_{n=1} \frac{1}{n}$$
2
3
4  % \sum = Perintah yang menghasilkan output lambang Sigma
5  % " ^ " = Menempatkan perintah diatas rumus (superscripts)
6  % {\infty} = Perintah yang menghasilkan output lambang Infinity
7  % " _ " = Menempatkan perintah dibawah rumus (subscript)
8  % \frac = Memberikan nilai pecahan pada rumus

```

Listing 4.10 Contoh Kombinasi Rumus Sigma

Hasil output :

$$\sum_{n=1}^{\infty} \frac{1}{n}$$

Setelah melihat salah satu penggunaan kombinasi rumus diatas kita bisa melakukan kombinasi rumus lainnya. Seperti yang akan diperlihatkan pada listing 4.11

```

1  $\sqrt{\frac{a^2}{3b^3+1}}$
2  $\lim_{n \rightarrow \infty} \frac{1}{n}=0$
3  $\int_a^b x^2 \, dx$
4  $\lim_{n \rightarrow \infty} \frac{1}{n}=0$
5  $\int_a^b x^2 \, dx$
6  $\sum_{n=1}^{\infty} \frac{1}{n}$

```

Listing 4.11 Contoh Kombinasi Rumus

Hasil output :

$$1. \sqrt{\frac{a^2}{3b^3+1}}$$

$$2. \lim_{n \rightarrow \infty} \frac{1}{n} = 0$$

$$3. \int_a^b x^2 \, dx$$

$$4. \lim_{n \rightarrow \infty} \frac{1}{n} = 0$$

$$5. \int_a^b x^2 \, dx$$

$$6. \sum_{n=1}^{\infty} \frac{1}{n}$$

4.9 Penulisan Tata Letak Rumus

Pada section ini kita kan mempelajari bagaimana menempatkan penulisan rumus sesuai dengan tata letak yang kita inginkan. Misalkan kita ingin membuat suatu rumus dan hasil dari rumus tersebut ingin kita letakkan di tengah, kiri atau pun kanan. Contoh penerapan tata letak dalam sebuah rumus dapat kita lihat pada listing 4.12.

```

1 $\begin{array}{ccc} (x+y)(x-y) & = & x^2-xy + yx-y^2 \\ & \& & \& & x^2-y^2 \\ & \& & \& & x^2 + 2xy + y^2 \end{array}$
2 %{\ccc} = center:center:center
3
4 $\begin{array}{lcr} (x+y)(x-y) & = & x^2-xy + yx-y^2 \\ & \& & \& & x^2-y^2 \\ & \& & \& & x^2 + 2xy + y^2 \end{array}$
5 %{\lcr} = left:center:right
6
7 $\begin{array}{rrcl} (x+y)(x-y) & = & x^2-xy + yx-y^2 \\ & \& & \& & x^2-y^2 \\ & \& & \& & x^2 + 2xy + y^2 \end{array}$
8 %{\rcr} = right:center:left

```

Listing 4.12 Contoh Penulisan Tata Letak Dalam Rumus

Hasil output :

$$\begin{array}{lcl}
 (x+y)(x-y) & = & x^2 - xy + yx - y^2 \\
 1. & & = x^2 - y^2 \\
 (x+y)^2 & = & x^2 + 2xy + y^2 \\
 \\
 (x+y)(x-y) & = & x^2 - xy + yx - y^2 \\
 2. & & = x^2 - y^2 \\
 (x+y)^2 & = & x^2 + 2xy + y^2 \\
 \\
 (x+y)(x-y) & = & x^2 - xy + yx - y^2 \\
 3. & & = x^2 - y^2 \\
 (x+y)^2 & = & x^2 + 2xy + y^2
 \end{array}$$

4.10 Penulisan Simbol

Banyak penulisan simbol yang dapat kita implementasikan pada latex. Beberapa contoh perintah penggunaan simbol pada latex dapat kita lihat seperti pada listing 4.13

```

1 $\pi$
2 $\phi$
3 $\rho$
4 $\sigma$
5 $\epsilon$
6 $\delta$
7 $\theta$
8 $\kappa$

```

```

9  $\alpha$
10 $\beta$
11 $\gamma$
12 $\omega$
13 $\zeta$
14 $\eta$
15 $\iota$
16 $\lambda$
17 $\mu$
18 $\nu$
19 $\xi$
20 $\tau$
21 $\upsilon$
22 $\chi$
23 $\psi$

```

Listing 4.13 Contoh Penulisan Simbol

Hasil output :

1. π = Merupakan simbol **pi**
2. ϕ = Merupakan simbol **phi**
3. ρ = Merupakan simbol **rho**
4. σ = Merupakan simbol **sigma**
5. ϵ = Merupakan simbol **epsilon**
6. δ = Merupakan simbol **delta**
7. θ = Merupakan simbol **theta**
8. κ = Merupakan simbol **kappa**
9. α = Merupakan simbol **alpha**
10. β = Merupakan simbol **beta**
11. γ = Merupakan simbol **gamma**
12. ω = Merupakan simbol **omega**
13. ζ = Merupakan simbol **zeta**
14. η = Merupakan simbol **eta**
15. ι = Merupakan simbol **iota**
16. λ = Merupakan simbol **lambda**
17. μ = Merupakan simbol **mu**
18. ν = Merupakan simbol **nu**

- 19. ξ = Merupakan simbol **xi**
- 20. τ = Merupakan simbol **tau**
- 21. v = Merupakan simbol **upsilon**
- 22. χ = Merupakan simbol **chi**
- 23. ψ = Merupakan simbol **psi**

BAB 5

SISTEMATIKA DOKUMEN

5.1 Sistematika Dokumen

Dalam format latex memiliki kemampuan untuk membuat suatu susunan yang struktural atau tertata dengan rapi seperti *bab*, *subbab*, dan sebagainya.

Tabel 5.1 Small Table

Struktur	penulisan
Bagian	part
Bab	chapter
subbab	section
subsubbab	subsection
subsubsubbab	subsubsection
paragraf berjudul	paragraph
anak paragraf berjudul	subparagraph

5.2 Alignment

Alignment adalah perataan baris pada LaTeX, terdapat 3 jenis pada perataan tersebut yaitu rata kiri, rata kanan atau rata tengah. pada dokumen LaTeX memiliki perataan yang secara default sudah diatur dan memiliki perataan justified atau biasa disebut rata kanan. Misalkan saat ini kita ingin melakukan pengaturan dokumen rata kiri, maka perintah yang dimasukan yaitu :

```
1 \begin{raggedleft}
2
3 isi dokumen yang diatur dengan rata kanan
4
5 \end{raggedleft}
```

Listing 5.1 Contoh Perintah Membuat Rata Kiri

atau bisa juga kira masukan perintah yang sama untuk rata kiri.

```
1 \begin{raggedleft}
2
3 isi dokumen yang diatur dengan rata kanan
4
5 \end{raggedleft}
```

Listing 5.2 Contoh Perintah Membuat Rata Kanan

5.3 Document class

Pada dokumen Latex terdapat atau mempunyai beberapa struktur yang dicirikan dengan blok yang diberi apit oleh perintah begin dan end. Latex memberikan pilihan Class dokumen yang bisa di pakai, antara lain aadlah Book, Report, Article dan lain sebagainya. Class document book merupakan Class Document yang paling tepat untuk menulis, karena dapat mendukung table of contents yang dapat berfungsi langsung untuk generate daftar isi secara langsung.

Perintah format ukuran font pada latex dapat kita lihat seperti pada listing 5.3

```
1 \tiny{teks yang ingin diformat}
2 \scriptsize{teks yang ingin diformat}
3 \footnotesize{teks yang ingin diformat}
4 \small{teks yang ingin diformat}
5 \normal{teks yang ingin diformat}
6 \large{teks yang ingin diformat}
7 \larger{teks yang ingin diformat}
8 \largest{teks yang ingin diformat}
9 \huge{teks yang ingin diformat}
10 \huger{teks yang ingin diformat}
```

Listing 5.3 Format Ukuran Font

Dalam memberikan penulisan judul pada format latex biasanya di letakkan pada awal document, untuk cara penulisan nya dapat dilakukan seperti pada listing ?? dibawah ini :

```

1 \documentclass{ A4 Papper / Ukuran yang di inginkan }
2 \begin{document}
3 \begin{Title} %Judul Dokumen
4 \author{Nama Penulis}
5 \date{Tanggal Pembuatan}
6 \maketitle{}
7 \end{document}

```

Listing 5.4 Struktur Awal Document Pada Latex

5.4 Costum Command

Sesuai dengan namanya Costum Command, dimana ke unggulan latex ada fitur yang satu ini, Pembuat dokumen ini dapat membuat macro untuk kebutuhan yang sifatnya spesifik dan berulang-ulang, dimana costum cummad dapat melakukan tanda bintang berjejer sebagai penanda garis. Pada Pengaturan huruf lateks dibuat dengan menggunakan tag atau perintah khusus yang menyediakan beberapa cara untuk memformat dokumen Anda. Kadang-kadang perintah standar tidak cukup untuk memenuhi beberapa kebutuhan spesifik.

5.5 Renewcommand

Perintah command merupakan perintah dasar dalam pembentukan format sebuah dokumen yang ada pada latex. Lalu apakah fungsi dari perintah *renewcommand*? Bagaimana mengubah perintah *section* pada command dalam suatu class? Atau bagaimana mengubah command latex pada bagian intinya daripada hanya dari command *section*-nya saja?

Perintah *section* tidak terdefinisi pada latex kernel namun dia terdefinisi oleh class file yang saling berhubungan. Contoh pengimplementasiannya dapat kita lihat pada dokumentasi listing 5.5 :

```

1 \newcommand\section{\@startsection {section}{1}{\z@}%
2 {-3.5ex \@plus -1ex \@minus -.2ex}%
3 {2.3ex \@plus .2ex}% Format Yang Ditandai
4 {\normalfont\Large\bfseries}}

```

Listing 5.5 Perintah Section pada New Command

Perintah *section* diatas menampilkan heading pada dokumen yang dibuat. Pada keterangan **Format Yang Ditandai** seperti pada listing diatas, dapat di artikan bahwa format heading memiliki space bernilai positive didalamnya, kita bisa mengubah format tersebut menjadi negative dengan perintah *renewcommand* seperti pada listing 5.6

```

1 \renewcommand\section{\@startsection {section}{1}{\z@}%
2 {-3.5ex \@plus -1ex \@minus -.2ex}%
3 {-1em}%
4 {\normalfont\Large\bfseries}}

```

Listing 5.6 Perintah Renewcommand

Preamble diatas memberikan efek yang berbeda pada dokumen. Package *titlesec* memberikan hasil yang lebih deklaratif untuk membuat perubahan yang sama, namun ia tetap harus mengasumsikan definisi asli kurang lebih seperti pada class dalam dokumen yang dibuat. Jadi pada dasarnya, perintah *renewcommand* dapat mengubah format suatu dokumen berdasarkan class file dari format *command* yang sudah ditentukan sebelumnya.

5.6 Penomoran

Perintah penomoran pada latex biasanya menggunakan format *Numbering* atau format *Bullets*. Perintah yang digunakan pada format *Numbering* adalah *enumerate* sedangkan untuk *Bullets* yang menyerupai poin menggunakan *itemize*.

Numbering merupakan perintah yang digunakan untuk membuat daftar berurut dengan penomoran menggunakan angka (numbered list), yang biasanya diberikan pada awal baris baru. Sedangkan *Bullets* atau poin adalah perintah yang digunakan untuk membuat daftar berurut dengan penomoran berupa symbol atau poin (bulleted list). Pada listing 5.8 adalah perintah untuk memasukan listing pada penomoran .

```
1 \begin{enumerate}
2   \item 1
3   \item 2
4 \end{enumerate}
```

Listing 5.7 Memberikan Perintah Numbering

Sedangkan *Bullets* atau poin adalah perintah yang digunakan untuk membuat daftar berurut dengan penomoran berupa symbol atau poin (bulleted list) 5.8.

```
1 \begin{itemize}
2   \item
3   \item
4 \end{itemize}
```

Listing 5.8 Menambahkan kode perintah bullets

5.7 Daftar Pustaka

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

```
1 \begin{thebibliography}{99}
2 \bibitem{label untuk referensi}{keterangan pustaka yang digunakan}
3 .....
4 .....
5 \end{thebibliography}
```

Listing 5.9 Memberikan Perintah Daftar Pustaka

BAB 6

REFERENS PADA LATEX

6.1 Membuat Penomoran Referensi

Untuk menambahkan referensi atau melakukan sanitasi pada latex kita dapat menggunakan berbagai macam cara. Salah satu cara sederhana yang dapat kita gunakan adalah dengan menggunakan environment yang di sebut *thebibliography*. Namun, kebanyakan orang saat ini menggunakan *BibTeX* untuk melakukan sanitasi sebagai acuan referensi. Dengan menggunakan *BibTeX* kita dapat mengatur sitasi sendiri secara terpisah dalam format file .bib [3]. Disaat mengutip maupun menggunakan sanitasi diperkenankan untuk memberi keterangan referensi atau sumber asal suatu kutipan dan gagasan. Untuk mengetahui bagaimana menambahkan referensi pada latex, kita dapat melihat langkah-langkahnya seperti pada gambar 6.1.

Bagaimana cara membuatnya di Latex? berikut cara membuatnya:

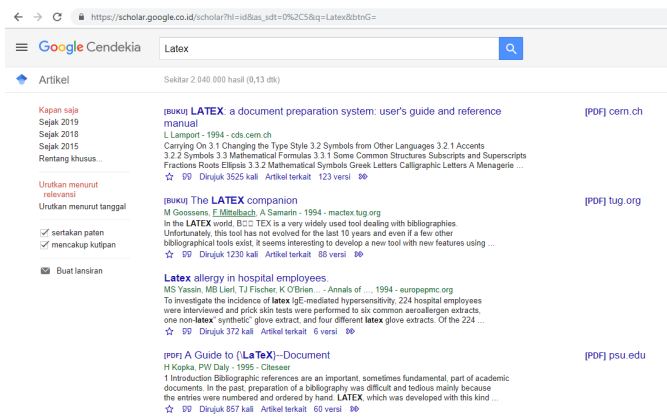
1. Cari materi yang akan dikutip melalui Google Scholar seperti pada gambar 6.2 ,
2. Setelah selesai mengutip jangan lupa untuk mengambil script bibtexnya dengan cara klik pada tanda kutip seperti pada gambar 6.3,

7.1 ASCII

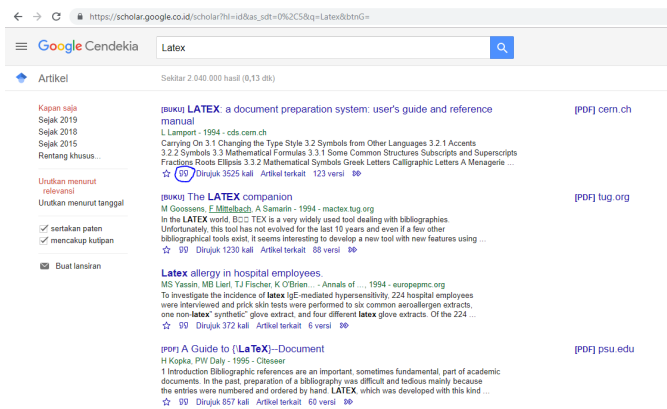
7.1.1 Definisi ASCII

Berdasarkan artikel yang ditulis oleh hieronymus [2] ASCII atau American Standard Code for Information Interchange merupakan sebuah pengkodean berstandar Internasional yang berupa kode huruf dan simbol, seperti Hex dan Unicode dan juga merupakan simbol tambahan dari database. ASCII bersifat universal contohnya 124

Gambar 6.1 Ini adalah Contoh Penomoran Referensi

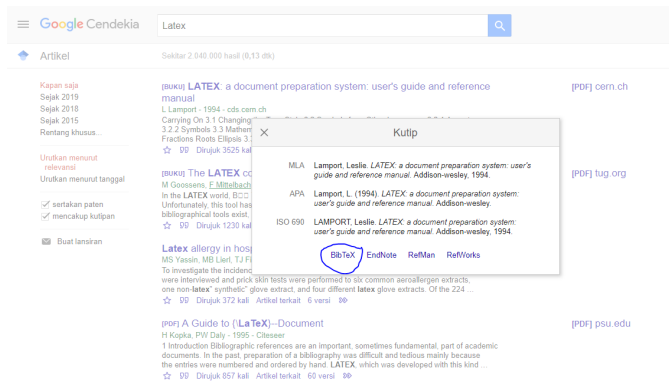


Gambar 6.2 Ini adalah Halaman Google Scholar



Gambar 6.3 Ini adalah Tanda proses awal mengambil reference

3. Maka akan muncul seperti gambar 6.4, lalu pilih Bibtex.



Gambar 6.4 Ini adalah Pilihan mengutip

4. Setelah memilih Bibtex maka akan muncul script seperti pada gambar 6.5,

```

< → ↺ 🔒 https://scholar.googleusercontent.com/scholar.bib?q=info:d9buru79OCU:scholar.google
@book{lampport1994latex,
  title={LATEX: a document preparation system: user's guide and reference manual},
  author={Lampport, Leslie},
  year={1994},
  publisher={Addison-wesley}
}

```

Gambar 6.5 Ini adalah Script BibTeX

- Script tersebut dicopy pada direktori yang dikerjakan, khususnya pada bagian reference.bib seperti pada gambar 6.6 dan 6.7 pada editor,
- Setelah dicopy, jangan lupa disave.
- Buka kembali pada lembar kerja yang sudah diberi kutipan/gagasan. Lalu tambahkan script listing 6.1. setelah kutipan maka akan muncul seperti pada gambar 6.8,

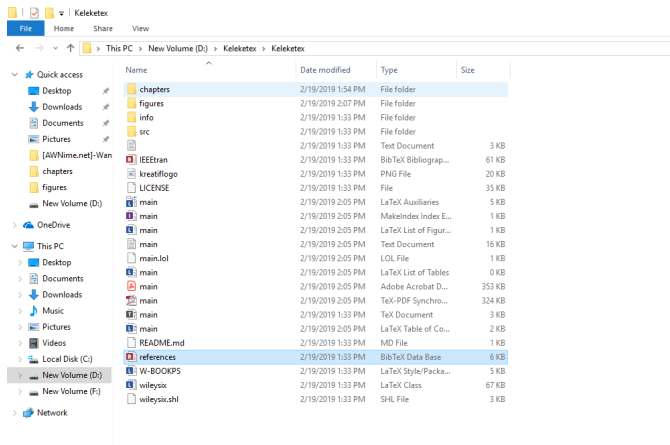
```

1 \cite{armiati2018national}
2 'di dalam kurung kurawal diisi sesuai sumber'

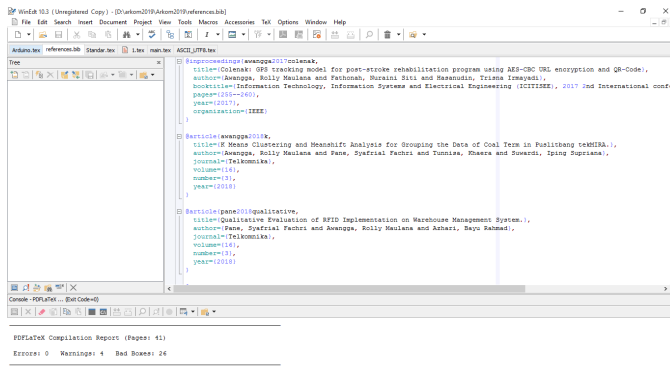
```

Listing 6.1 Penggunaan perintah cite untuk reference

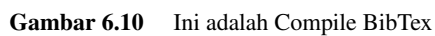
- Pilih insert dan save.
- Untuk proses compilenya dilakukan 2 kali yaitu pada main.tex pilih Tex lalu pilih pdflatex dan Bibtex, dilakukan berulang minimal 3 kali compile. Seperti pada gambar 6.9 untuk pdflatex dan 6.10 untuk BibTeX.



Gambar 6.6 Ini adalah Direktori pekerjaan



Gambar 6.7 Ini adalah Reference.bib



DAFTAR PUSTAKA

1. R. Awangga, "Sampeu: Servicing web map tile service over web map service to increase computation performance," in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.
2. M. Kline, "Latex," 2018.
3. B. T. Atmaja, "Tutorial latex sederhana," 2015.

Index

disruptif, xxxi
modern, xxxi