



Thesis/Dissertation Manual with LaTeX Master's Degree



**Faculty of Engineering,
King Mongkut's University of Technology
Thonburi
December 2024**

CONTENTS

Chapter 1	1
Basic of LaTeX	1
1.1 Characteristics of LaTeX.....	1
1.2 Software for Writing LaTeX.....	2
Chapter 2	9
Structure of LaTeX for Thesis.....	9
2.1 Basic Structure	9
2.2 Special Characters	12
2.1.1 Commonly Used Special Characters	13
2.1.2 Special Characters in Mathematical Symbols	13
2.1.3 Common Mathematical Symbols	14
2.3 Complete code structure example	16
2.4 Endnotes and Footnotes.....	20
2.5 Creating Cross-References in a Document	21
2.6 Reference and Citation	23
2.7 Setting Pages to Landscape	27
Chapter 3	28
Inserting equations, images, and tables.....	28
3.1 Inserting equations	28
3.2 Inserting pictures	29
3.3 Inserting Tables	32
Chapter 4	36
LaTeX coding for thesis	36
Chapter 5	44
Thesis Guidelines.....	44
5.1 Thesis Printing (already configured in the <code>KMUTTEngiMaster class</code>).....	44
5.2 Page Numbering (already configured in the <code>KMUTTEngiMaster class</code>)	45
5.3 Line Spacing (already configured in the <code>KMUTTEngiMaster class</code>)	45
5.4 Additional Thesis Writing Guidelines	45
Appendix.....	47

Chapter 1

Basic of LaTeX

LaTeX (Lay-tech" or "Lah-tech") is an internationally recognized document management system, ideal for scientific, mathematical, and engineering projects, as well as tasks requiring high credibility, such as theses, books, and research reports. It employs a markup language to define the document structure and includes an automatic page layout system that ensures documents are visually appealing and well-organized. Popular tools like Texmaker and Overleaf enhance convenience in working with LaTeX. A key strength of LaTeX lies in its ability to efficiently handle complex equations and manage references, saving time on formatting while allowing users to focus on content creation. This makes LaTeX particularly suitable for those who require high-quality, standardized documents.

1.1 Characteristics of LaTeX

LaTeX is a document management system designed to meet the demands of complex and high-standard formatting, making it well-suited for scientific, engineering, and academic documents. With outstanding capabilities in handling text, tables, images, and equations, as well as a convenient and efficient citation and bibliography management system, it offers exceptional functionality. Additionally, it supports the management of large-scale documents, is flexible across multiple platforms, and is available at no cost. These features have made it a popular choice among academics and researchers. Despite its steep learning curve, the high-quality and reliable standard of its output continue to make it an essential tool for producing precise and professional documents.

Advantages of LaTeX:

High-quality formatting and layout: LaTeX enables the creation of documents with professional-quality formatting and layout. It offers precise control over page design, text formatting, tables, images, equations, and symbols, ensuring a well-organized and standardized presentation.

Efficient Reference and Bibliography Management: LaTeX provides a robust system for managing references and bibliographies. Users can easily generate page numbers, footnotes, endnotes, and comprehensive lists of references or bibliographies, streamlining the citation process.

Capability to handle large and complex documents: LaTeX excels at managing large and complex documents, such as theses, books, and reports with extensive data. Its capabilities make it an ideal tool for producing substantial academic or professional documents.

Free and Open-Source: LaTeX is free and open-source software available to everyone. It is compatible with various platforms, including Windows, macOS, and Linux, offering flexibility and accessibility to users across different systems.

Elegant Mathematical Typesetting: LaTeX is well-suited for writing and displaying complex mathematical formulas and equations. Its powerful typesetting capabilities make it an excellent choice for academic papers, theses, or scientific documents requiring intricate mathematical expressions.

Disadvantages of LaTeX:

Complexity in Learning and Usage: LaTeX is a complex language. While creating simple documents may not be difficult, advanced formatting and command usage require significant study and understanding of the language's intricacies.

Lack of What You See Is What You Get (WYSIWYG) Interface: LaTeX is not a What You See Is What You Get (WYSIWYG) system, users cannot see the final output as they type. This can make editing and refining documents more difficult compared to tools that offer immediate visual feedback.

Limited flexibility in formatting compared to WYSIWYG systems: Although LaTeX offers diverse and standardized formatting options, achieving highly customized layouts may require additional commands or adjustments. LaTeX follows a "What You Get Is What You Mean" (WYGIWYM) approach, requiring users to precisely specify their desired output, which can be time-consuming to learn.

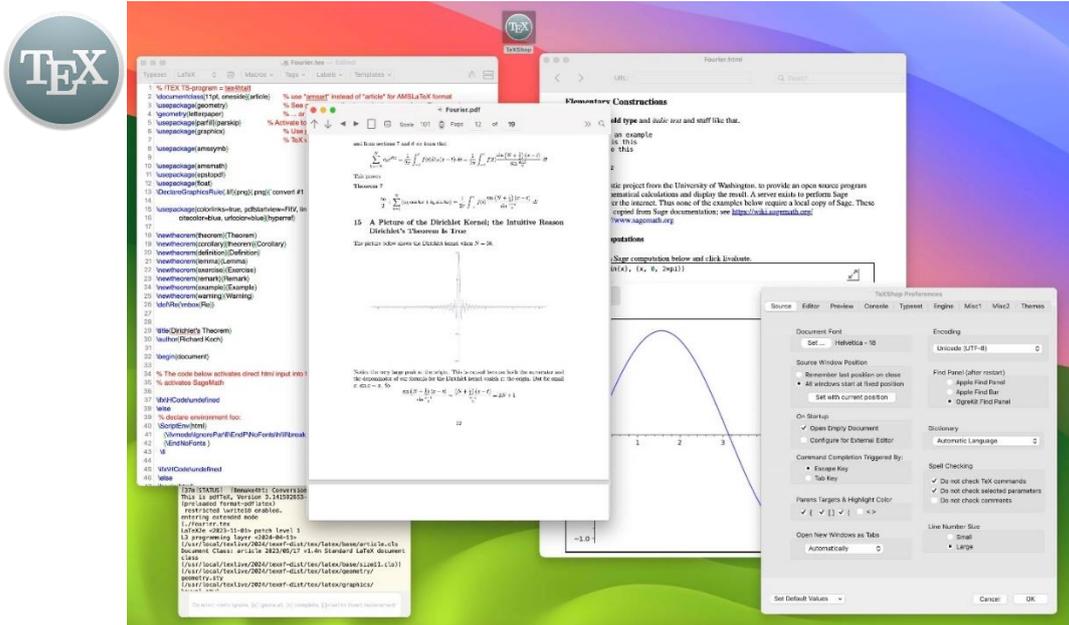
Limited necessity for simple documents: For straightforward documents, LaTeX may be unnecessary, as WYSIWYG word processors often provide quicker and more intuitive solutions

Despite these drawbacks, LaTeX is widely available across operating systems, including Windows, Linux, and macOS. Its plain-text file format ensures compatibility and editability on any platform, producing consistent results regardless of the system or LaTeX distribution used. LaTeX-generated outputs are typically in PDF format, ensuring uniform appearance across all devices and operating systems. In addition to PDF, LaTeX supports other formats such as DVI, PostScript, and HTML, enabling both print and online distribution. This versatility allows documents to be shared and viewed on screens, e-readers, and smartphones seamlessly. For editing, users can choose from various text editors optimized for LaTeX, ensuring flexibility and efficiency in document creation.

1.2 Software for Writing LaTeX

LaTeX is a markup language used for formatting academic documents and producing high-quality printed materials. Several programs are available for writing and managing LaTeX documents, each with its unique features, advantages, and disadvantages as follows:

(1) TeXShop (MacOS) This program is specifically designed for use on MacOS. It is user-friendly and has an intuitive interface.



Source: <https://pages.uoregon.edu/koch/texshop/pictures.html>

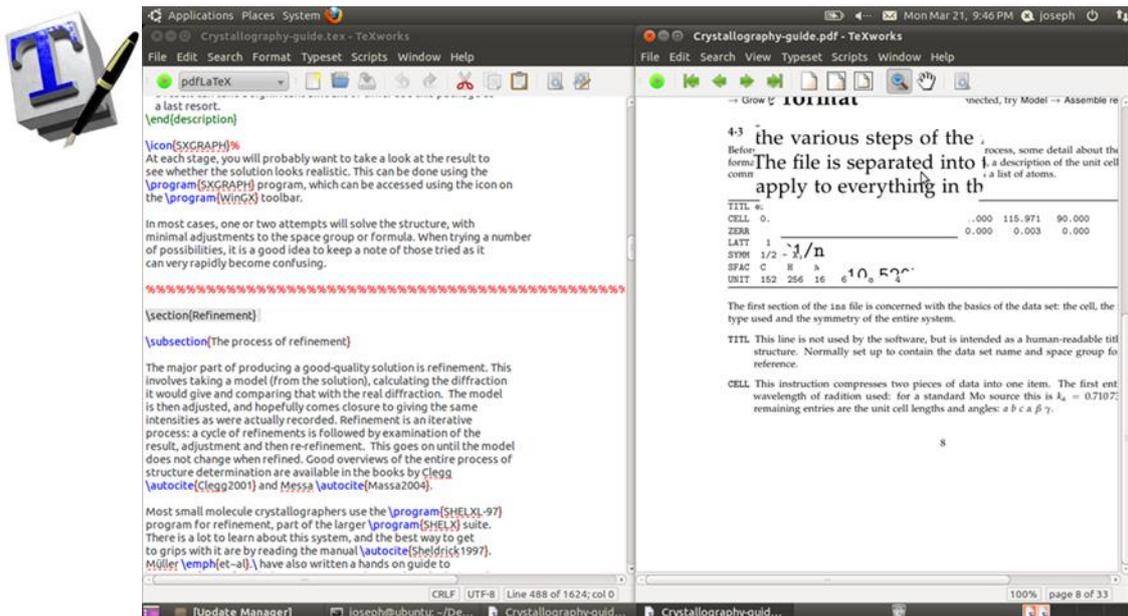
Advantages:

- User-friendly interface
- Fast and stable performance
- Excellent preview display

Disadvantages:

- Only available on MacOS
- Lacks some advanced features found in other programs

(2) TeXworks (Windows, MacOS, Linux) TeXworks is an open-source program designed to work across multiple operating systems with a simple interface.



Source: <https://www.tug.org/texworks/img/texworks-linux.png>

Advantages:

- User-friendly interface
- Cross-platform compatibility
- Excellent preview display

Disadvantages:

- Few advanced features
- Occasional minor bugs

(3) Overleaf (Web-based) Overleaf is an online LaTeX editor that can be accessed via a web browser without needing any software installation



Source: <https://de.overleaf.com/about/features-overview>

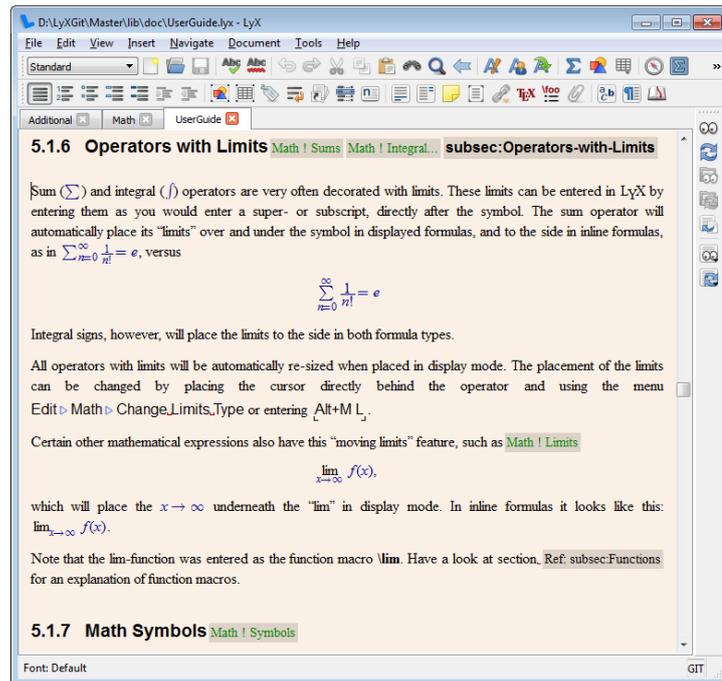
Advantages:

- No software installation required
- Real-time collaboration capability
- A variety of templates and document samples are available

Disadvantages:

- Requires an internet connection
- The free version has some limitations
- Limited file upload size in the free version

(4) LyX (Windows, MacOS, Linux) LyX is a WYSIWYG (What You See Is What You Get) editor focused on ease of use, producing LaTeX results.



Source: <https://www.lyx.org/Screenshots>

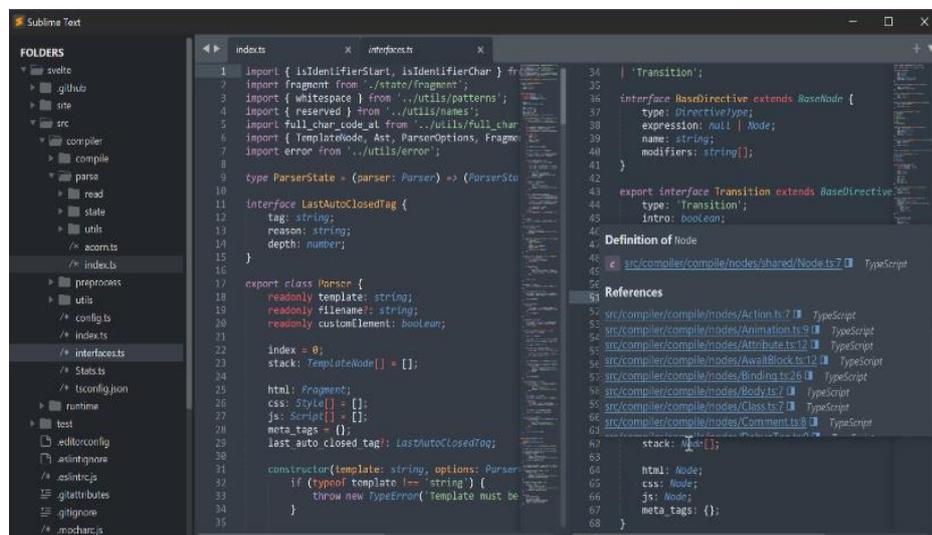
Advantages:

- Easy to use without needing LaTeX code knowledge
- WYSIWYG interface
- Cross-platform support

Disadvantages:

- Limited customization options
- May not suit users needing fine control over document details

(5) Sublime Text with LaTeXTools (Windows, MacOS, Linux) This text editor supports LaTeX writing through the LaTeXTools extension.



Source: <https://www.sublimetext.com/>

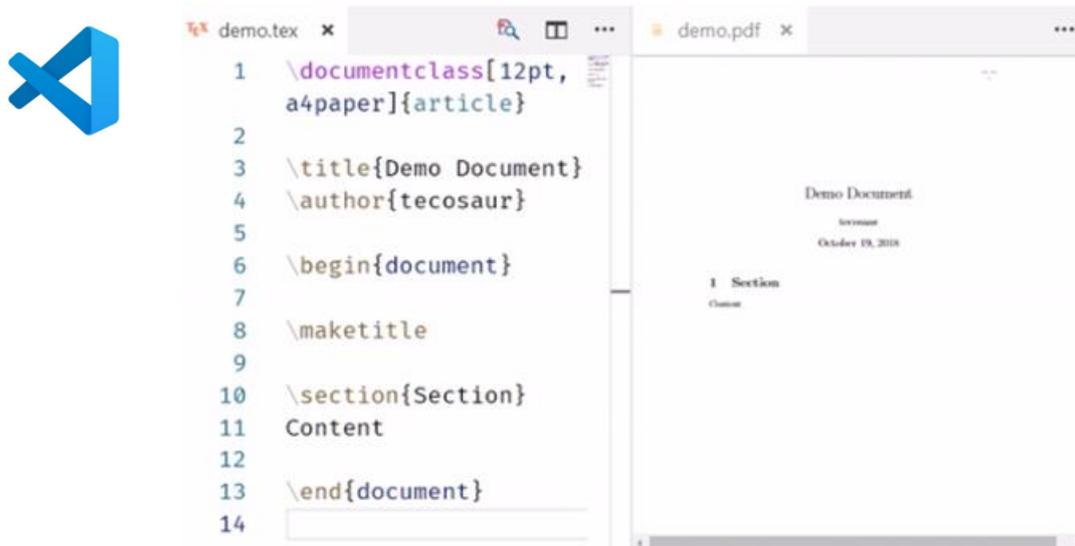
Advantages:

- Modern and user-friendly interface
- Cross-platform compatibility
- Highly customizable features

Disadvantages:

- The free version has feature limitations
- Requires installation and configuration of additional plugins

(6) Visual Studio Code with LaTeX Workshop (Windows, MacOS, Linux) This text editor includes the LaTeX Workshop extension for writing LaTeX.



Source: <https://marketplace.visualstudio.com/items?itemName=James-Yu.latex-workshop>

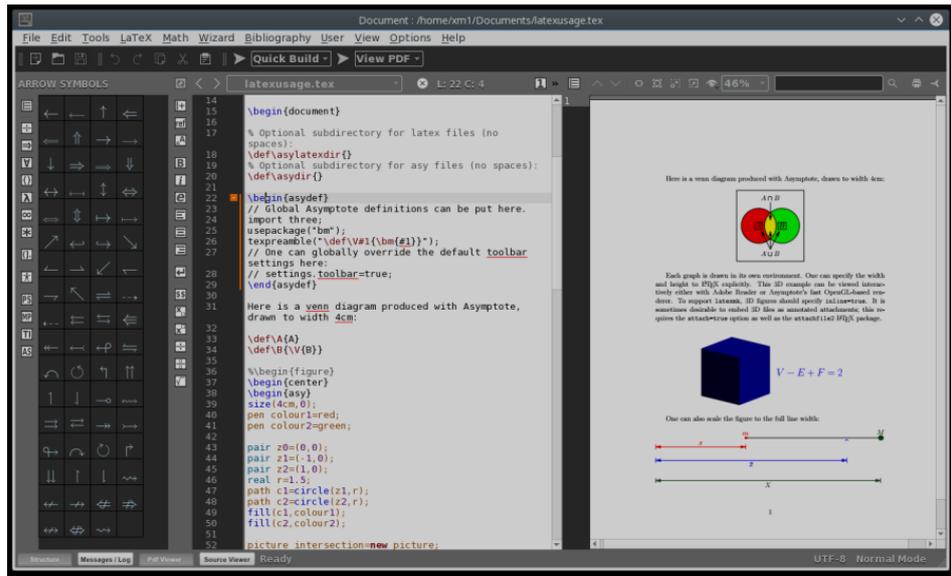
Advantages:

- Cross-platform compatibility
- Excellent preview display
- Comprehensive features and highly customizable

Disadvantages:

- Configuration can be complex for beginners
- Requires installation and configuration of additional plugins

*(7) Texmaker is an open-source LaTeX editor available on multiple platforms (Windows, MacOS, Linux). It features an integrated development environment (IDE), allowing users to write, test, and edit LaTeX documents within a single program.



Source: <https://www.xm1math.net/txmaker/>

Advantages:

- **User-friendly Interface:** Texmaker has a simple and intuitive interface, suitable for both beginners and experienced users.
- **Writing Assistance Tools:** Features like auto-completion, table creation tools, and equation editors help streamline the writing process
- **PDF Preview:** Includes a built-in PDF preview, enabling users to immediately check the results of their documents.
- **Error Checking:** Provides tools for error checking in LaTeX code, making it easier to edit documents.
- **Multilingual Support:** Texmaker supports multiple languages, making it accessible to users from various countries.

Disadvantages:

- **Complex Initial Setup:** LaTeX The initial setup might seem complex for new users, especially those unfamiliar with LaTeX.
- **Lack of Real-time Collaboration:** Does not offer real-time collaboration features like online platforms such as Overleaf.
- **Performance:** May face performance issues when handling large or complex documents.
- **Customization:** The interface and features might not be as flexible as other programs like Visual Studio Code.

The programs mentioned above are designed for document creation and require a tool to compile the written code into a finalized document, known as a LaTeX compiler. Various LaTeX compilers are available for different platforms, such as:

- MikTeX for Windows (<https://miktex.org/>)

- MacTeX for macOS (<https://www.tug.org/mactex/>)
- TeXLive for Linux (<https://www.tug.org/texlive/>)

For the purposes of this training session, alongside the accompanying manual, the Texmaker program will be used. Detailed installation instructions for Texmaker can be found in the appendix.

* This training uses Texmaker in conjunction with this manual, with detailed installation instructions provided in the appendix.

Chapter 2

Structure of LaTeX for Thesis

2.1 Basic Structure

LaTeX is a widely used document preparation system, particularly in the fields of science and mathematics, due to its efficiency in handling mathematical formulas and document referencing. The basic structure of the main LaTeX code can be divided into several sections as follows:

```
% (1) Preamble
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage{amsmath}
\usepackage{graphicx}
\title{My First LaTeX Document}
\author{John Doe}
\date{\today}

% (2) Document Beginning
\begin{document}

% (3) Content
\maketitle
\begin{abstract}
This is the abstract of the document.
\end{abstract}
\section{Introduction}
This is the introduction section.
\section{Main Content}
This is the main content section.
\subsection{Subsection}
This is a subsection of the main content.
\begin{equation}
E = mc^2
\end{equation}
\section{Conclusion}
This is the conclusion section.
```

```

\bibliographystyle {plain}
\bibliography{references}
% (4) Document Ending
\end{document}

```

Commands in LaTeX begin with a backslash (\). If a command requires an argument, it is enclosed in curly braces { }. Optional parameters, which are not mandatory, are enclosed in square brackets []. Comments in LaTeX can be added by using the % symbol. Any text following % on the same line is ignored and will not appear in the output document. Because of this functionality, % is considered a special character. To display it in the document, you must use the escape sequence \%.

The code can be divided into the following:

(1) Preamble: The preamble is the section before the main content where various document settings are defined, such as the type of document, the packages to be used, and document configuration commands. An example is shown below:

```

\documentclass{article}           % Define document type
\usepackage[utf8]{inputenc}      % Define character encoding
\usepackage{amsmath}            % Package for symbols and mathematical formulas
\usepackage{graphicx}           % Package for inserting images
\title{My First LaTeX Document}
\author{John Doe}
\date{\today}

```

The documentclass can be adjusted based on the type of document the author intends to create. The text in the curly braces can be modified to match the desired document format such as below:

- article: Used for short articles or reports.
- report: Used for longer reports, such as theses or technical reports.
- book: Used for writing books.
- letter: Used for writing letters.
- beamer: Used for creating presentation slides.
- proc: Used for conference proceedings or academic meetings.
- slides: Used for creating simple slides (beamer is now more commonly used).
- memoir: Used for large, flexible documents such as books, reports, and other extensive documents

Additionally, there are other documentclass options specifically designed for various purposes or for specific publishing or institutional formats, such as `amsart`, `amsbook` (for the American Mathematical Society), and others that can be downloaded as needed.

LaTeX provides a variety of documentclass options designed for specific purposes or tailored to the formatting requirements of publishers or institutions. Examples include `amsart` and `amsbook`, which are created for the American Mathematical Society, among others. Additional document classes can be downloaded as needed to suit specific requirements.

LaTeX also includes a wide range of macros for different applications. A collection of macros grouped for a particular purpose is referred to as a *package*. Users can include these packages by using the command: `\usepackage [option]{ package_name }` Within the preamble the section before the main document content users can define or specify various parameters. For example, commands like: `\title{My First LaTeX Document}` `\author{John Doe}` `\date{\today}` allow the program to organize the provided information according to the formatting rules defined by the article class.

(2) **Document Beginning:** This marks the start of the main content in the document:

```
\begin{document}
```

(3) **Content:** This section includes the main content of the document, such as the introduction, chapters, conclusion, and references.

```
\begin{My First LaTeX Document}      % Title of the document
\author{John Doe}                    % Author
\date{\today}                        % Date of the document
\maketitle                           % Create the document header (title, author, date)

\begin{abstract}
This is the abstract of the document.
\end{abstract}

\section{Introduction}
This is the introduction section.

\section{Main Content}
This is the main content section.
```

```
\subsection{Subsection}
```

This is a subsection of the main content.

```
\begin{equation}
```

$$E = mc^2$$

```
\end{equation}
```

```
\section{Conclusion}
```

This is the conclusion section.

```
\bibliographystyle{plain}
```

```
\bibliography{references}
```

(4) Document Ending: This marks the end of the document

```
\end{document}
```

Sections (2) and (4) are related as they represent an environment in LaTeX, specifically the document environment. This environment is initiated with the macro `\begin{environment}` and concluded with `\end{environment}`.

Definition of LaTeX Commands

- `\documentclass{class}`: Specifies the type of document, such as article, report, book.
- `\usepackage{package}`: Includes additional packages to extend LaTeX's capabilities.
- `\section{section}`: Creates a main section.
- `\subsection{subsection}`: Creates a subsection.
- `\begin{document}` and `\end{document}`: Define the scope of different sections, such as the beginning and ending of the document, tables, figures, equations, etc.
- `\bibliographystyle{key}` and `\bibliography{key}`: Manage document references.

2.2 Special Characters

Typing special characters in LaTeX is essential when you need to input symbols that cannot be directly typed, such as mathematical characters, special letters, or other symbols not included in the standard character set. Unlike Microsoft Word or other common programs, LaTeX requires knowledge of specific code to produce these symbols. Examples are provided below:

2.2.1 Commonly Used Special Characters

LaTeX has a set of special characters that cannot be typed directly, such as \$, %, &, _, {}, #, and \. To print these characters, you need to precede them with a backslash (\).

Symbol:	Name:	Code LaTeX:
\$	Dollar sign	\\$
%	Percent	\%
&	Ampersand	\&
_	Underscore	_
{	Left curly brace	\{
}	Right curly brace	\}
#	Hash (Pound) sign	\#
\	Backslash	\textbackslash

2.2.2 Special Characters in Mathematical Symbols

LaTeX supports a wide range of mathematical symbols that can be used within Math Mode, such as colons, summation symbols, and others. Some mathematical symbols in LaTeX require additional packages, which the author has included in the class, namely amsmath and amssymb. Users can invoke special characters in mathematical symbols using the following commands:

Symbol:	Name:	Code LaTeX:
α	Greek letter alpha	\alpha
β	Greek letter beta	\beta
Σ	Summation symbol	\sum
\int	Integral symbol	\int
π	Pi symbol	\pi

Additionally, using the $\$...\$$ syntax will place the content in Math Mode, formatting the text within as a mathematical equation. For example:

```
\documentclass{article}
\begin{document}
This is inline math mode:  $a^2 + b^2 = c^2$ .
\end{document}
```

In this example, $a^2 + b^2 = c^2$ formatted as an inline mathematical equation. If you need to print the \$ symbol in regular text (not in Math Mode), you must precede it with a backslash (\) to inform LaTeX that you intend to print the symbol as text, not as an indicator of Math Mode:

```
\documentclass{article}
\begin{document}
The price is \$100.
\end{document}
```

In this example, $\$100$ display as \$100.

2.2.3 Common Mathematical Symbols

LaTeX is an essential tool for researchers, mathematicians, and professionals in scientific fields, enabling them to create documents that elegantly and effectively present mathematical equations and symbols. A crucial aspect of using LaTeX is understanding the various mathematical symbols and frequently used commands to write complex equations and text effortlessly. This section provides examples of commonly used mathematical symbols in LaTeX, along with explanations and corresponding code snippets, to help users understand and apply them with ease.

(1) Special Characters and Symbols

Symbol:	Name:	Code LaTeX:
x^2	Square	<code>x^2</code>
*	Asterisk	<code>*</code>
+	Addition	<code>+</code>
-	Subtraction	<code>-</code>
×	Multiplication	<code>\times</code>
÷	Division	<code>\div</code>

(2) Formatting

Symbol:	Name:	Code LaTeX:
$\frac{a}{b}$	Fraction	<code>\frac{a}{b}</code>
\sqrt{x}	Square root	<code>\sqrt{x}</code>
$\sqrt[n]{x}$	n-th root	<code>\sqrt[n]{x}</code>

 \int

Integral

\int

(3) Brackets

In mathematical equations, the size of parentheses adjusts automatically to fit the content inside them. This resizing helps clearly indicate the grouping of operations and their order of execution. For instance, in expressions such as $\ln(x)$ or $\ln\left(\frac{a}{b}\right)$ the size of the parentheses may vary depending on their content. In LaTeX, you can achieve this by using the commands `\left` before the opening parenthesis and `\right` before the closing one. These commands ensure that the parentheses are dynamically sized to match the enclosed content. The following demonstrates how to apply this technique:

Symbol:	Name:	Code LaTeX:
(key)	Parentheses	<code>\left(key\right)</code>
$[key]$	Square brackets	<code>\left[key\right]</code>
$\{key\}$	Curly braces	<code>\left\{key\right\}</code>

(4) Special Symbols

Symbol:	Name:	Code LaTeX:
\sum	Sigma	<code>\sum</code>
π	Pi	<code>\pi</code>
Δ	Delta	<code>\Delta</code>
α	alpha	<code>\alpha</code>
β	beta	<code>\beta</code>
\lim	Limit	<code>\lim</code>
$\sum_{i=1}^n i$	Bounded sigma	<code>\sum_{i=1}^n i</code>
$\int_0^1 x^2 dx$	Bounded integral	<code>\int_{0}^{1} x^2 \, dx</code>
\in	Set membership	<code>\in</code>
\forall	For all	<code>\forall</code>
\exists	There exists	<code>\exists</code>

Note: For a bounded integral, you can use the `\limits` macro in LaTeX to adjust the positioning of the upper and lower bounds. For example, the command `\int\limits_{0}^{1} x^2 \, dx` will render as: $\int_0^1 x^2 dx$

(5) Normal Function

Writing mathematical equations, there is a distinction between italic and upright text usage. Italicized text is typically used for general mathematical variables, whereas upright text is reserved for specific names or functions. LaTeX's math mode, standard text is automatically converted to italics. Therefore, it is the responsibility of the author to ensure the correct formatting. For commonly used functions, LaTeX provides predefined macros. For example:

Symbol:	Name:	Code LaTeX:
$\sin(x)$	Sine	<code>\sin(x)</code>
$\cos(x)$	Cosine	<code>\cos(x)</code>
$\tan(x)$	Tangent	<code>\tan(x)</code>
$\log(x)$	Logarithm	<code>\log(x)</code>
$\ln(x)$	Natural logarithm (base e)	<code>\ln(x)</code>

Note: If a desired function does not have a predefined macro, you can use the command `\mathrm{function_name}` to display the function name in upright text. This approach is also applicable to other mathematical notations, such as the d in derivatives.

2.3 Complete code structure example

LaTeX requires a well-defined structure to efficiently manage its content. The following example demonstrates the essential components of this basic structure, such as specifying the document type, importing packages, defining attributes, and initiating the content. This example is designed to allow users to copy, run, and experiment with the code, making it easier to understand and adapt for document development according to their needs. (You can copy this section and run it to explore the fundamental structure.)

```
% Start document
\documentclass[a4paper, 12pt]{article}

% Import packages
\usepackage[utf8]{inputenc}
\usepackage[thai]{babel}
\usepackage{amsmath, amssymb, graphicx}
\usepackage{hyperref}
\usepackage{listings}
\usepackage{xcolor}
```

```

% Set document properties
\lstset{
  basicstyle=\ttfamily,
  keywordstyle=\color{blue},
  commentstyle=\color{gray},
  stringstyle=\color{red},
  breaklines=true,
  numbers=left,
  numberstyle=\tiny\color{gray},
  frame=single,
  framerule=0.5pt,
  backgroundcolor=\color{white}
}

% Start content
\title{LaTeX Manual}
\author{Author}
\date{\today}

\begin{document}
\maketitle
\documentclass

\section{ Introduction }
LaTeX is a typesetting system used for creating high-quality documents, such as theses,
academic papers, and textbooks. It is highly flexible and powerful for document formatting.

\section{ Getting Started }
To start using LaTeX, you need to install a TeX distribution such as TeX Live (for all operating
systems) or MiKTeX (for Windows).

\subsection{ Basic Structure of a LaTeX Document }
The basic structure of a LaTeX document includes:

% itemize
\begin{itemize}
  \item \texttt{\documentclass\{...\}}: Specifies the type of document
  \item \texttt{\usepackage\{...\}}: Used to import additional packages
  \item \texttt{\begin\{document\} ... \end\{document\}}: Marks the beginning and end of the
document

```

```
\end{itemize}
```

Example of a basic structure:

```
\begin{lstlisting}[language=TeX]
```

```
\documentclass{article}
```

```
\usepackage[utf8]{inputenc}
```

```
\title{ My First Document }
```

```
\author{ Author }
```

```
\date{\today}
```

```
\begin{document}
```

```
\maketitle
```

```
\section{ Introduction }
```

This is a basic example of a LaTeX document.

```
\end{document}
```

```
\end{lstlisting}
```

```
\section{ Text Formatting }
```

You can format text in LaTeX in various ways, such as bold, italics, and underline.

Example:

```
\begin{lstlisting}[language=TeX]
```

```
\textbf{ Bold Text }
```

```
\textit{ Italic Text }
```

```
\underline{ Underlined Text }
```

```
\end{lstlisting}
```

```
\section{ Including Mathematical Formulas }
```

LaTeX has special capabilities for handling mathematical formulas. You can include formulas inline or in separate lines.

Example:

```
\begin{lstlisting}[language=TeX]
```

Inline formula: $E = mc^2$

Separate line formula:

```
\begin{equation}
```

$E = mc^2$

```
\end{equation}
```

```
\end{lstlisting}
```

```
\section{ Inserting Images }
```

You can insert images in a LaTeX document using the `\graphicsx` package.

Example:

```
\begin{lstlisting}[language=TeX]
```

```
\usepackage{graphicx}
```

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

```
\centering
```

```
\includegraphics[width=0.5\textwidth]{example-image}
```

```
\caption{ Image Caption }
```

```
\label{fig:example}
```

```
\end{figure}
```

```
\end{lstlisting}
```

```
\section{ Creating Tables }
```

Creating tables in LaTeX can be done using `\texttt{tabular}`

Example:

```
\begin{lstlisting}[language=TeX]
```

```
\begin{tabular}{|c|c|c|}
```

```
\hline
```

```
Header 1 & Header 2 & Header 3 \\
```

```
\hline
```

```
Data 1 & Data 2 & Data 3 \\
```

```
Data 4 & Data 5 & Data 6 \\
```

```
\hline
```

```
\end{tabular}
```

```
\end{lstlisting}
```

```
\section{ Creating Lists }
```

LaTeX provides environments for creating both ordered and unordered lists.

Example:

```
\begin{lstlisting}[language=TeX]
```

```
\begin{itemize}
```

```
  \item Item 1
```

```
  \item Item 2
```

```
\end{itemize}
```

```
\begin{enumerate}
```

```
  \item Item 1
```

```
  \item Item 2
```

```
\end{enumerate}
```

```
\end{lstlisting}
```

```
\section{ Referencing }
```

Creating references in LaTeX is straightforward.

Example:

```
\begin{lstlisting}[language=TeX]
```

```
\section{สิ่งที่เกี่ยวข้อง}
```

```
\label{sec:related}
```

Section [\ref{sec:related}](#) for more information.

```
\end{lstlisting}
```

```
\section{ Conclusion }
```

This is a basic example of using LaTeX for document management. I hope this guide will be useful in getting started with LaTeX.

```
\end{document}
```

2.4 Endnotes and Footnotes

Adding endnotes and footnotes in LaTeX can be done in various ways, utilizing different packages that support reference management. This section introduces methods for using footnotes, endnotes, and related packages.

Footnote: A footnote is additional information, or a brief explanation placed at the bottom of the same page as the main content. The author uses a reference number or symbol in the main text, which corresponds to the number displayed at the bottom of the page. This allows the reader to access supplementary information immediately without having to navigate to another page. To add a footnote, the `\footnote{your text here}` command is used. For example:

<pre>\documentclass{article} \begin{document} <i>This is the text in the article.</i> \footnote{<i>This is the text of Footnote</i>}. \end{document}</pre>	<p>This is the text in the article.¹</p> <hr/> <p>¹ This is the text of Footnote</p>
--	--

The footnote text will appear at the bottom of the page. Additionally, footmisc package allows for extensive customization of footnotes, such as formatting and spacing adjustments. Below is an example of how to use the footmisc package:

<pre>\documentclass{article} \usepackage[perpage]{footmisc} \begin{document} <i>This is the text in the article.</i> \footnote {<i>This is the text in the article in Footnote 1</i>}. <i>This is the text in the article.</i> \footnote{ <i>This is the text in the article in Footnote 2</i>}. \end{document}</pre>	<pre>% Reset footnote on each page</pre>
---	--

Endnote: An endnote is used to add information or explanations that cannot be included in the main content, like a footnote, but it is placed at the end of the document instead of at the bottom of the page. In the document, a reference number or symbol is used in the main text, allowing the reader to refer to additional details at the end of the document. To add an endnote, the endnotes package must be used, which has already been installed in the code.

<pre>\let\footnote=\endnote \begin{document} This is the text in the article. \footnote{ <i>This is the text of Endnote</i>} \theendnotes \end{document}</pre>	<pre>% Key endnote</pre>
--	--------------------------

2.5 Creating Cross-References in a Document

Cross-referencing within a document allows for referencing components such as chapters, sections, figures, tables, or equations. This feature makes navigation and referencing much more convenient. Furthermore, if the position of these components changes such as due to reordering chapters or adding/removing figures LaTeX will automatically update the references to ensure accuracy.

There are three primary commands for creating cross-references in LaTeX:

- `\label{ key}`: Used to define a component and assign a unique identifier (key) for referencing.
- `\ref{ key}`: Retrieves the number associated with the referenced component identified by the key.
- `\pageref{ key}`: Retrieves the page number of the component identified by the key.

The key serves as an identifier in the document, allowing LaTeX to associate it with the corresponding component. When the key is referenced, LaTeX replaces it with the relevant information, such as the number (`\ref{ key }`) or page number (`\pageref{ key }`). Keys can be named in any language, such as Thai or English, but it is common practice to structure them as *ident:name_id*, where:

ident indicates the type of element being referenced (e.g., chapter, fig, table).

name_id provides a descriptive name for the element.

In general practice, the following identifiers (*ident*) are commonly used to represent various elements within a document:

<i>ident</i>	Used for	<i>ident</i>	Used for
sec	chapter, section, ...	fig	figure
tab	table	eq	equation

Note: Some authors may choose to differentiate between chapters and sections by using separate identifiers, such as chap. The choice of identifiers can be customized based on clarity and conciseness.

Below demonstrates how to define and reference elements within a document using these identifiers.

```

\documentclass{report}
\begin{document}

\chapter{Introduction} % Begin Chapter 1 Introduction

\label{sec: Introduction } % Define label sec:Introduction

\section{Research Background} % Begin section 1.1
\label{sec:ResearchBG} % Define label sec: ResearchBG

According to the equation
\begin{equation}
a^2 + b^2 = c^2

```

```

\label{eq:Pythagoras} % Define label eq:Pythagoras
\end{equation}

, a Pythagorean equation  $\dots$ 

\section{Literature Review} % Begin section 1.2

% Cross referencing eq:Pythagoras and sec:ResearchIntro

According to the equation~(\ref{eq:Pythagoras}) in section~\ref{sec:ResearchIntro} in page
\pageref{eq:Pythagoras} ,  $\dots$ 

\end{document}

```

Note: LaTeX does not limit cross-referencing to only the elements mentioned earlier. Users can create custom elements and define references for them. For example, you can create unique environments with their own counters and design them to support cross-referencing. Detailed instructions for this can be found in basic LaTeX manuals or on resources like LaTeX.org.

When using `\ref{key}`, LaTeX retrieves only the numerical value associated with the referenced element. To enhance convenience and accuracy, the `KMUTTEngiMaster` class defines additional macros for more efficient referencing. Details on these macros are available in Section 4.1.5. By leveraging these features, you can adapt LaTeX's cross-referencing system to meet the specific needs of your document.

2.6 Reference and Citation

There are several methods for citing references in LaTeX, depending on the desired citation style. These include numeric citations, author-year citations, and citations based on the order of appearance in the document. Each citation style requires different packages, such as `natbib` and `biblatex`, as well as the use of a bibliography file (`.bib`) or directly writing the citations within the LaTeX document.

2.6.1 Biblatex

`biblatex` is a versatile package for citations, offering extensive customization options.

```

\documentclass{article}

\usepackage[style=authoryear]{biblatex}

\addbibresource{references.bib}

%Authors must specify the type of their document in the references.bib file:

\begin{document}

References \parencite{label1} and \textcite{label2}

```

```
\printbibliography
\end{document}
```

In the references.bib file, the document type must be specified, with the type indicated on the first line of the entry, such as @article for articles and @book for books. The bibliographic details should be provided within the brackets, such as the author's name, title, and journal name. Here is an example:

```
@article{label1,
  author = {Author1, A.},
  title = {Title of the article},
  journal = {Journal Name},
  year = {2020},
  volume = {1},
  pages = {1-10}
}
@book{label2,
  author = {Author2, B.},
  title = {Title of the Book},
  publisher = {Publisher Name},
  year = {2019}
}
```

2.6.2 Numeric

Numeric citations display numbers in brackets, such as [1], [2], [3], indicating the order of citations as they appear in the document. Numbers should be ordered according to their appearance in the thesis content, without separating by language or publication type. The citation numbers should be aligned to the left margin, with the corresponding bibliographic details printed in the bibliography section.

```
\documentclass{article}
\usepackage{cite}

%% Authors need to specify the document type in the references.bib file:

\begin{document}
Citation \cite {label1} and \cite{label2}

\bibliographystyle{plain}
\bibliography{references}
\end{document}
```

Numeric citations are concise and easy to read, preventing the text from appearing cluttered. They are also space-efficient, making them ideal for scientific and technical documents that involve numerous citations. However, the drawback is that numeric citations do not provide immediate information about the source within the text, requiring the reader to consult the bibliography for full details. This method may not be suitable for emphasizing the origin of ideas.

2.5.3 Author-Year

The author-year citation style, such as (Author1, 2020) or (Author2, 2019), is commonly used in research that requires clear identification of the source within the text. All referenced information should be grouped and listed alphabetically. For theses written in Thai, the Thai references should be listed first, followed by English references. Conversely, for theses written in English, English references should be listed first, followed by Thai references.

```
\documentclass{article}
\usepackage{natbib}
% Authors need to specify the document type in the references.bib file:
\begin{document}
Citation \citep{label1} and \citet{label2}
\bibliographystyle{plainnat}
\bibliography{references}
\end{document}
```

The author-year citation style is advantageous for providing clear attribution, allowing readers to immediately identify the author and publication year of the work. This makes it particularly suitable for research in the social sciences and humanities, where the author and context are of paramount importance. However, this style can clutter the text, especially when citations are frequent, and may complicate citation management due to the potential for redundancy when citing the same author multiple times.

2.6.4 Manually

Citations in a document can be written directly without the need for a '.bib' file. Bibliographic information can be included at the end of the document.

```
\documentclass{article}
\begin{document}
Citation [1] and [2]
\begin{thebibliography}{9}
```

```

\begin{thebibliography}{99}
%-----
%Books Name
%(1)
\bibitem{Smith} Smiths, J.E., 1990,
\textbf{Engineer}, 2\textsuperscript{nd} ed.,
Amalin, Bangkok, pp. 120--150.
%-----
%Article in a journal
%(2)
\bibitem{Yeh} Yeh, T.S., Boy, T.Y. and Sacks, M.D., 1988,
\textquotedblleft Effect of Particle Size Distribution on the
\textbf{Journal of the American Ceramic Society},
Vol. 71, No. 12, pp. 484--487.
%-----
%Article in a proceedings
%(3)
\bibitem{Merin} Merin, U. and Daufin, G., 1989,
\textquotedblleft Separation Process Using Inorganic Membrane
\textbf{International Conference on Inorganic Membranes},
6 July 1989, Paris, pp. 272--278.
%-----
\end{thebibliography}

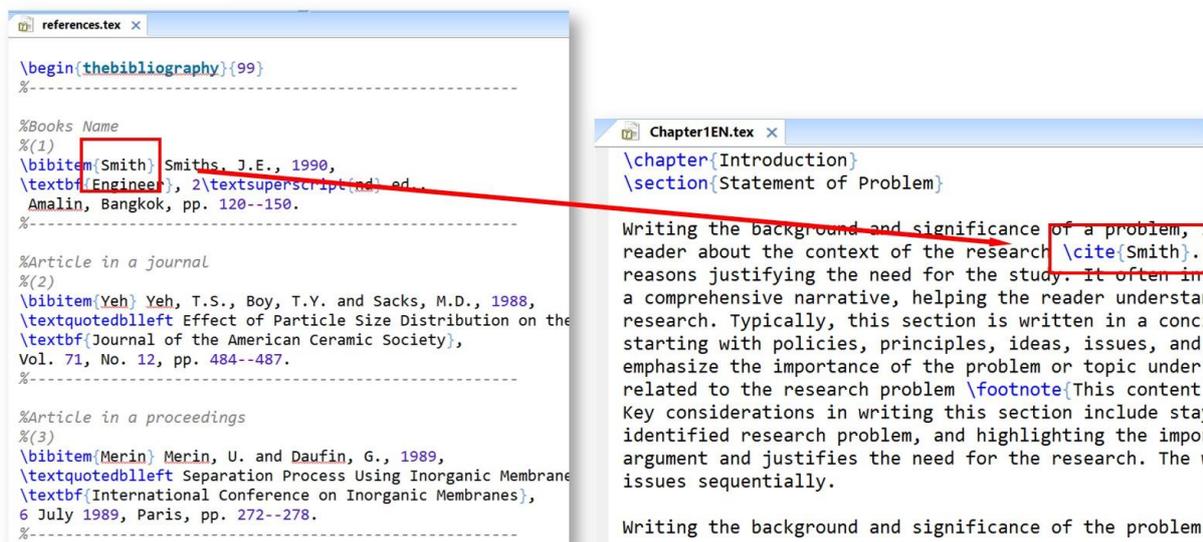
\end{document}

```

Thus, the choice of citation style depends on the nature of the document and the author's requirements. Numeric citations are well-suited for scientific reports, author-year citations are ideal for social sciences and humanities research, and biblatex is preferred by those who need flexibility and detailed bibliography management.

Additionally, for reference within the content, it can be done by including information through the `\bibitem` command in the `reference.tex` file. Each reference must have a unique ID, and you can cite the desired reference in the document using the `\cite{ id }` command. If multiple sources need to be cited at once, it can be done using commands like `\cite{ id1 , id2 }`, as shown in the example.

references.tex



To insert references in the document, use the ID specified in the `\bibitem` command by calling it with the `\cite{id}` command in the document. If multiple references are cited consecutively, they can be separated by a comma, such as `\cite{id1, id2}`.

2.7 Setting Pages to Landscape

In some cases, certain pages of a thesis may need to be set to landscape orientation for better visualization of information. In LaTeX, this can be achieved using the `pdfscape` or `lscape` packages, which allow for rotating specific pages to landscape orientation. Both packages function similarly, but `pdfscape` offers additional functionality by rotating the page in the PDF viewer

2.6.1 Using `pdfscape` or `lscape`

- **Package Installation:** The following packages must be installed in the document's preamble. The difference is that `pdfscape` rotates both the page and its content in the PDF viewer, whereas `lscape` only rotates the content, not the page itself. For convenience, it is recommended to install both packages.

```
\usepackage{pdfscape}
\usepackage{lscape}
```

- **Using the landscape environment:** To set a page to landscape orientation, use the `landscape` environment, which will rotate everything within the environment to landscape orientation. Once the `landscape` environment ends, the document will automatically revert to its original portrait orientation.

```
\begin{landscape}
% Content to be displayed in landscape orientation, such as text, tables, or images.
\end{landscape}
```

Additionally, users can modify other options of the `geometry` package, such as `margin`, `top`, `bottom`, `left`, and `right`, to adjust the page margins as needed. To return to portrait orientation, include the `\newpage` command at the end of the `landscape` content to ensure that the following page reverts to portrait orientation.

Chapter 3

Inserting equations, images, and tables

3.1 Inserting equations

Inserting equations in LaTeX is very popular in academic documents. LaTeX has commands and environments that help manage and display mathematical equations in an orderly manner. It can be done by inserting equations on the same line (inline) or displaying equations in a new line (display), which can be explained as follows:

3.1.1 Inline Math

To insert an equation on the same line, use the `$... $` or `\(... \)` symbols to insert the equation into the text.

```
This is an inline equation: $E=mc^2$ within the text.  
or  
This is an inline equation: \( E=mc^2 \) within the text.
```

The result will appear on the same line.

3.1.2 Display Math

To display equations on a new line separated from the contents, use the `\[... \]` symbol, `Environment equation`, or `align` symbol.

```
\[  
E = mc^2  
\]
```

or:

```
\begin{equation}  
E = mc^2  
\end{equation}
```

The equation will be displayed on a new line, separate from the text.

3.1.3 Numbering equations

`Environment equation` or `align` will automatically number the equations, resulting in an equation with the number on the right side of the equation. For example:

```
\begin{equation}
a^2 + b^2 = c^2
\end{equation}
```

3.1.4 Environment align for multi-line equations

To display multi-line equations or equations with multiple parts, you can use the align environment, which helps to arrange the equations beautifully. The align environment helps to arrange equations with multiple lines easily and beautifully. & is used to align each line.

For example:

```
\begin{align}
a &= b + c \\
d &= e - f
\end{align}
```

3.1.5 Indent of Multiline Equations

The environment multiline can be used to write equations with multiple lines or sequences of calculations. This is suitable for very long equations that cannot be written on a single line.

For example:

```
\begin{multiline}
a + b + c + d + e + f + g + h + i + j + k + l \\
= m + n + o + p + q + r + s + t + u + v + w + x
\end{multiline}
```

3.2 Inserting pictures

LaTeX supports inserting pictures in various formats, such as .png, .jpg, and .pdf. Inserting pictures in LaTeX can be done with the graphicx package, which provides the ability to manipulate images and control their display in the document.

3.2.1 Installing the graphicx package

To insert pictures in LaTeX, you need to install the graphicx package, which is a basic package that allows you to insert images using the \includegraphics command. The installation method is as follows:

```
\documentclass{article}
\usepackage{graphicx} %Package for inserting images
\begin{document}
```

```
% Details
\end{document}
```

Note: If the package is not installed, the system will prompt to install it from the internet.

3.2.2 `\includegraphics` Command

Use the `\includegraphics` command and specify the file type of the image to be used. For example, this command will insert the image `filename.png` in the same folder as the LaTeX file. By default, the image is displayed in full size but can be resized as desired.

```
\includegraphics{filename.png}
```

3.2.3 Scaling and Positioning

Image scaling and positioning can be done by adding parameters to the `\includegraphics` directive. For example:

```
\includegraphics[width=0.5\textwidth]{filename.png}
```

In example as below: the image will be scaled to half the width of the text width `\textwidth`. Other parameters can also be used, such as `height`, `scale`, and `angle`, for example this command will reduce the image to 75% of its actual size and rotate it 90 degrees.

```
\includegraphics[scale=0.75, angle=90]{filename.png}
```

3.2.4 Environment `figure`

To organize images systematically and display captions and image numbers, you can use Environment `figure`, for example:

```
\begin{figure}[h]
\centering %Centering the image on the page
\includegraphics[width=0.5\textwidth]{filename.png}
\caption{Caption}
\label{fig:example}
\end{figure}
```

In addition, the `\centering` command is used to center an image on the page, `\caption` is used to add a caption to an image, and `\label` is used to refer to an image in a document. For example:

In figure~\ref{fig:example}, we see the following...

3.2.5 Inserting Multiple Images into a Single Image

The `subfigure` or `subcaption` environment allows you to insert multiple images into a single image. This example shows two images on the same line, using a `subfigure` for each image, and a caption for each image.

```
\usepackage{subcaption}

\begin{figure}[h]
\centering
\begin{subfigure}{0.4\textwidth}
\centering
\includegraphics[width=\linewidth]{image1.png}
\caption{ figure 1}
\end{subfigure}
\hspace{1cm}
\begin{subfigure}{0.4\textwidth}
\centering
\includegraphics[width=\linewidth]{image2.png}
\caption{ figure 2}
\end{subfigure}
\caption{ figure 1 and figure 2}
\end{figure}
```

3.2.6 Floating Figures

When using the `figure` environment, images may not appear where they appear in the code because LaTeX handles the positioning of images appropriately in the document. This behavior can be controlled using the `h`, `t`, `b`, and `p` options. The `[htbp]` parameter, enclosed in square brackets, tells LaTeX where in the document you want the image to appear:

- h (here): Place the image where the code appears in the document.
- t (top): Place the image at the top of the page
- b (bottom): Place the image at the bottom of the page.
- p (page of floats): Place images on pages with only images and tables.

```
\begin{figure}[htbp]
\centering
\includegraphics[width=0.5\textwidth]{filename.png}
\caption{คำอธิบายรูปภาพ}
\end{figure}
```

3.3 Inserting Tables

Inserting tables in LaTeX is about creating a structured table that can be formatted efficiently. LaTeX has specific commands and environments for creating and modifying tables, such as `tabular` and `table`. The details of their usage and functions are explained below:

3.3.1 Environment `tabular`

The `tabular` environment is a basic command used to create a table in LaTeX, excluding control over the position of the table in the document. The basic structure of a `tabular` is as follows:

```
\begin{tabular}{|c|c|c|}
\hline
Column 1 & Column 2 & Column 3 \\ \hline
Data 1 & Data 2 & Data 3 \\ \hline
Data 4 & Data 5 & Data 6 \\ \hline
\end{tabular}
```

In the code above:

- `|c|c|c|` specifies the alignment of each column
(c for center, l for left align, r for right align)
the `|` symbol indicates a vertical line between columns
- `\hline` is used to create a horizontal line between rows
- `&` is used to separate data between columns
- `\\` is used to indicate the end of a row and the beginning of a new row

The result is a table with 3 columns and 2 rows of data, with borders around the columns and rows.

3.3.2 Environment table

The Environment table is used to control the position of a table in the document (making it float) and to add captions and numbers to the table. Example usage:

```
\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|} \hline
Column 1 & Column 2 & Column 3 \\ \hline
Data 1 & Data 2 & Data 3 \\ \hline
Data 4 & Data 5 & Data 6 \\ \hline
\end{tabular}
\caption{ Table Description }
\label{tab:example}
\end{table}
```

In the code above:

- `\begin{table}[h]` is used to start the table environment with the `[h]` option which specifies that the table will appear in the position closest to the code (like inserting an image)
- `\centering` is used to center the table
- `\caption{ Table Description }` is used to add a caption below the table
- `\label{tab:example}` is used to specify a label to refer to the table

Note: It is common practice to place the caption of a table above the table itself. Typically, the `\caption{}` and `\label{}` commands are placed before the tabular environment. However, in the `KMUTTEngiMaster` class, this setting is pre-configured, allowing these commands to be placed after the table instead.

3.3.3 Column Sizing

Table columns can be resized using column alignment options such as `p{width}`. For example, we want the second column to be 4 cm wide, and the first and third columns to be 2 cm wide.

```
\begin{tabular}{|p{2cm}|p{4cm}|p{2cm}|}
\hline
Column 1 & Column 2 & Column 3 \\ \hline
Data 1 & Data 2 & Data 3 \\ \hline
\end{tabular}
```

```
\end{tabular}
```

3.3.4 Merging Cells

To merge cells in a table, you can use the `\multicolumn` and `\multirow` commands from the `multirow` package. For example:

```
\usepackage{multirow}
\begin{tabular}{|c|c|c|}
\hline
\multirow{2}{*}{Row 1} & Column 2 & Column 3 \\ \cline{2-3}
& Column 2a & Column 3a \\ \hline
\multicolumn{3}{|c|}{Combined Column} \\ \hline
Row 2 & Column 2b & Column 3b \\ \hline
\end{tabular}
```

In the above code:

- `\multirow{2}{*}{Row 1}` is used to combine two cells in the same row
- `\multicolumn{3}{|c|}{Combined Column}` is used to combine multiple columns in a single cell.

3.3.5 Using booktabs for high-quality tables

The `booktabs` package allows creating high-quality, more professional-looking tables by manipulating their borders. For example:

```
\usepackage{booktabs}
\begin{tabular}{ccc}
\toprule
Column 1 & Column 2 & Column 3 \\
\midrule
Data 1 & Data 2 & Data 3 \\
Data 4 & Data 5 & Data 6 \\
\bottomrule
\end{tabular}
```

In the above code:

- `\toprule` creates a thin, beautiful top border
- `\midrule` creates a center border that emphasizes the separation between the topic and data
- `\bottomrule` creates a bottom border

3.3.6 Using “tabularx” for fixed-width table management

The tabularx package automatically manages tables that need to fit within the width of the page. For example:

```
\usepackage{tabularx}
% X is used to automatically resize columns so that the table fits the width of the page
(\textwidth)
\begin{tabularx}{\textwidth}{|X|X|X|}
\hline
Column 1 & Column 2 & Column 3 \\ \hline
Data 1 & Data 2 & Data 3 \\ \hline
Data 4 & Data 5 & Data 6 \\ \hline
\end{tabularx}
```

Chapter 4

LaTeX coding for thesis

Main folders, classes, and code

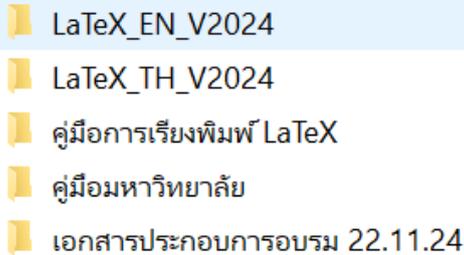
Each part of the code is separated according to the chapter of the complete thesis. Users can use it by editing the content in the sub-files contained in various folders, which are arranged according to the chapter for ease of use. In addition, there are examples written within each chapter, in which users can edit the sample content to be according to own thesis. Important files and folders that users must know are as follows:

(1) Main folders

The files required to use LaTeX code for thesis typesetting are packed in the main folder named **KMUTTEngiMaster-master**. Users need to download the files and **save all in a safety and stable drive**, such as drive D. The folder looks like the image below.

A screenshot of a file explorer showing a folder named "KMUTTEngiMaster-master". The folder icon is a yellow square with a white document symbol.

When you click on the main folder, you will find many folders and files, all of which are important for the completion of the thesis, as follows:

A screenshot of the contents of the "KMUTTEngiMaster-master" folder. The folder is highlighted in light blue. It contains five items: a folder named "LaTeX_EN_V2024", a folder named "LaTeX_TH_V2024", a folder named "คู่มือการเรียงพิมพ์ LaTeX", a folder named "คู่มือมหาวิทยาลัย", and a folder named "เอกสารประกอบการอบรม 22.11.24". Each item has a yellow folder icon.

Use only **the LaTeX_EN_V2024** folder if your thesis is written in **English**. The primary files and folders that users need to understand and work continuously for formatting the thesis are as shown in the image below. These include the following folders: Itx, Chapter1 – Chapter5, PostChapter, PreChapter, and Template (this folder contains examples; if there is no data in Chapter1 – Chapter5, the content from Template will be used automatically).

Additionally, there are two essential files:

1. KMUTTEngiMaster.cls (this file should not be edited).
2. MainCode (this file is used to compile the thesis document).

The importance and usage details of each folder/file are explained in the following.

 _ltx	
 Chapter1	Chapter1 contains the file Chapter1.tex and other related files such as image files in Chapter 1.
 Chapter2	Chapter2 contains the file Chapter2.tex and other related files such as image files in Chapter 2.
 Chapter3	Chapter3 contains the file Chapter3.tex and other related files such as image files in Chapter 3.
 Chapter4	Chapter4 contains the file Chapter4.tex and other related files such as image files in Chapter 4.
 Chapter5	Chapter5 contains the file Chapter5.tex and other related files such as image files in Chapter 5.
 PostChapter	PostChapter contains all the files that are at the end or after Chapter 5.
 PreChapter	PreChapter contains all the files in the preface to Chapter 1.
 Templates	Templates is used for example and content preparation only.
 KMUTTEngiMaster.cls	
 litbib.bib	
 MainCode.tex	

ltx: This folder is designated for storing or saving images intended for use in the thesis. The university logo has already been saved in this folder for the front cover and flyleaf. Additionally, users can save any image files here or organize them into categories by chapter for easier access. By categorizing images by chapter, users can maintain better organization and retrieve images conveniently based on their preferences.

Chapter1 – Chapter5: Within this folder, you can find separate files for each chapter, from Chapter 1 to Chapter 5. Users can edit these files to match the content of thesis. Be sure to save all changes after editing each file to avoid issues caused by the program's auto-saving feature. Once you have completed all chapters (Chapter1 – Chapter5), proceed to fill in your personal details in the PreChapter and PostChapter folders or you can do this step either before or after creating the chapters, depending on your preference. These steps are essential for compiling the final document as detailed in section **Creating a complete thesis**. It is important not to rename any files in the code during the thesis compilation process unless you can modify all references in the code accurately. Renaming files without updating all references may result in errors during code execution.

PostChapter: This LaTeX code file is used for formatting the closing sections of the thesis. Users can input their details here to generate pages such as biography, appendices, and references. The required files for these sections are provided in this folder.

 Appendices.tex

 CurriculumVitae.tex

 references.tex

The **Appendices.tex** file is used to create appendices.

The **CurriculumVitae.tex** file is used to create Curriculum Vitae

The **reference.tex** file is used to create References

The **references.tex** file is used to create the reference list. Since the university's referencing style does not fully align with the standard formats integrated into the system, it is necessary to

manually input the references in the **references.tex** file. Authors can freely edit and cite references within the content. However, a fundamental understanding of the university's specific referencing style is required to ensure accuracy.

PreChapter: This is a LaTeX code file for formatting the preliminary sections of the thesis. Authors can input their details directly into this file. This section includes the abstract, list of abbreviations, list of symbols, and acknowledgments. These elements can be edited as follows:

 **abbreviations.tex**

 **abstract.tex**

 **abstractEN.tex**

 **acknowledgements.tex**

 **symbols.tex**

The **abbreviations.tex** file is used for formatting the list of abbreviations.

The **abstract.tex** file is used for formatting the abstract.

The **abstractEN.tex** file is used for formatting the English abstract.

The **acknowledgements.tex** file is used for formatting the acknowledgments.

The **symbol.tex** file is used for formatting the list of symbols.

Template: This folder serves as an example template for thesis formatting. Inside, you will find a folder named **Examples**, which contains sample thesis chapters provided. Users can refer to these sample files as a guide or choose not to use them at all. Examples of how to write each section can also be found in the files discussed in sections 1-4. However, if the user does not create content for **Chapter 1 – Chapter 5**, this folder will be automatically.

(2) Class

"Class" refers to the type of document created, such as an article, report, book, or slide. The typesetting of the thesis of the Faculty of Engineering, King Mongkut's University of Technology Thonburi uses a created class named `KMUTTEngiMaster`, which is already written in a file named `KMUTTEngiMaster.cls`. This class will be called in the typesetting code, which is in the preamble section, which is already set. Therefore, the user does not need to do anything with the class section, such as editing the file name or editing some details in the file, because the class section is bound to many parts of the code in other files. If this section is edited, the user will need to edit in many, which will create complications and may cause errors during the code execution.

(3) Create pdf in Main Code

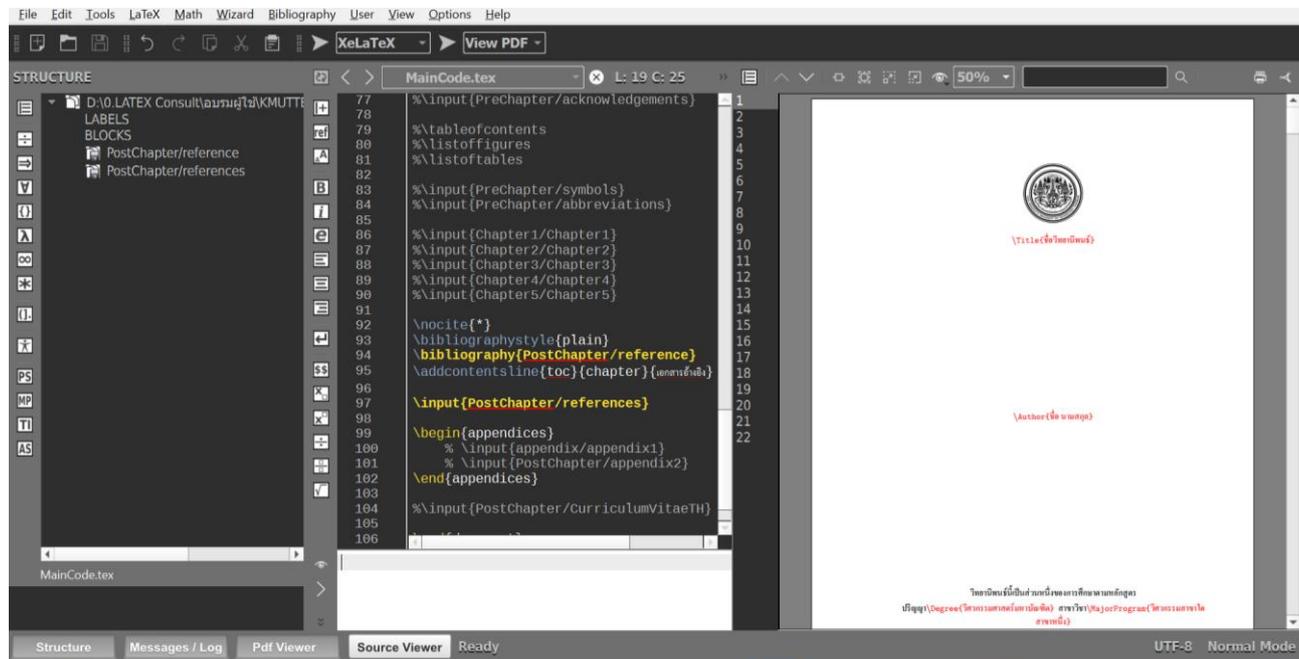
In the overall formatting process, once users have prepared all files, including Chapter 1 to Chapter 5 and other supporting files, the author has created a file named `MainCode.tex` for running

the code to compile the complete thesis document. Users must input their personal details in this file to generate the finalized thesis.

MainCode.tex

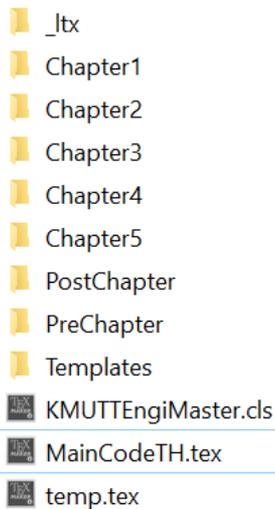
When users open MainCode.tex and run it, the compiled thesis will appear on the right-hand side of the program. As shown in the example, any red-highlighted sections will automatically disappear once users have filled in all the required information in MainCode.tex.

If red characters remain after running, it indicates incomplete information. You need to should check for missing variables by observing the red text. For example, if `\Title{Thesis Title}` is still highlighted in red, users should verify whether this variable has been activated and the thesis title has been filled in the MainCode.tex file. If not, remove the % symbol in front of the command (as the % symbol is a comment, and commented lines will not be executed).

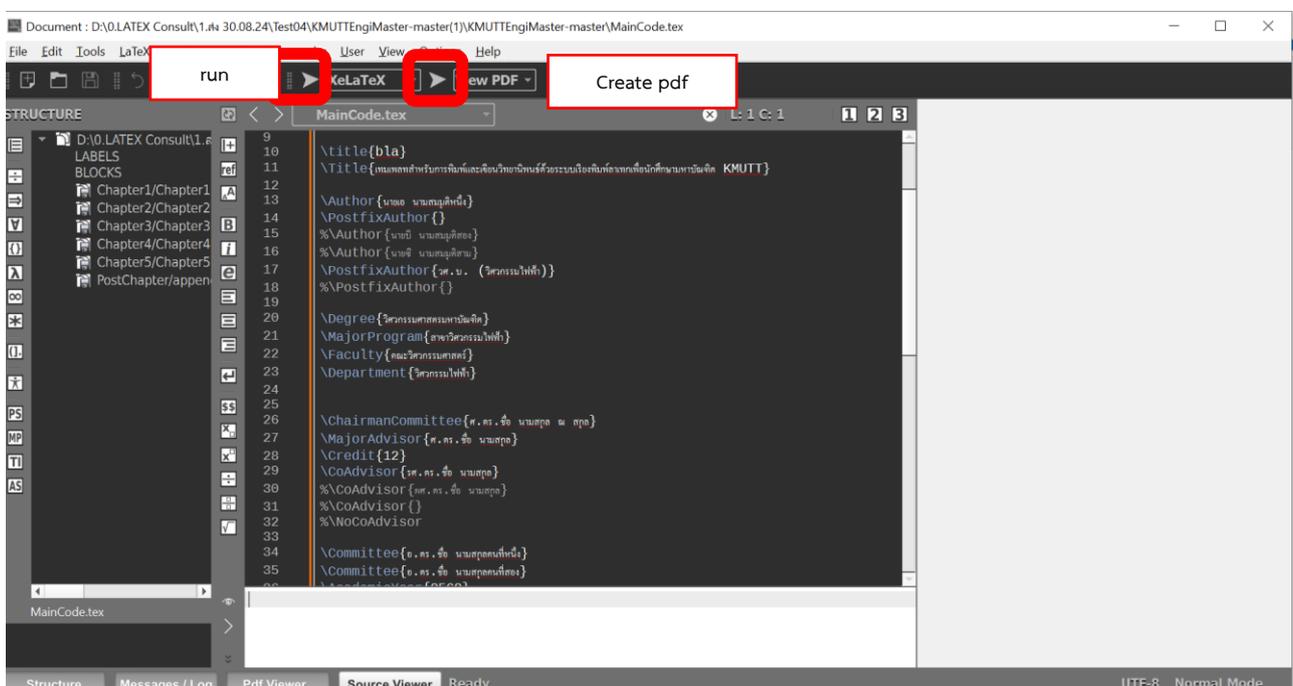


(4) Creating a complete thesis

Once the user has edited all the files completely, you should check the accuracy of all files and folders once again before proceeding to compile the final document as follows:



After organizing all the files and folders, the user should return to the MainCode.tex file to input your personal details as shown below:



Next, click **Run** and wait until the program finishes processing. Then, click **Create PDF** to generate the thesis file named **MainCode.pdf**. Be sure to check the file every time through the window on the right-hand side. If any information is missing, try clicking **Run** again and wait for it to finish before creating the PDF. You may need to repeat this process several times.

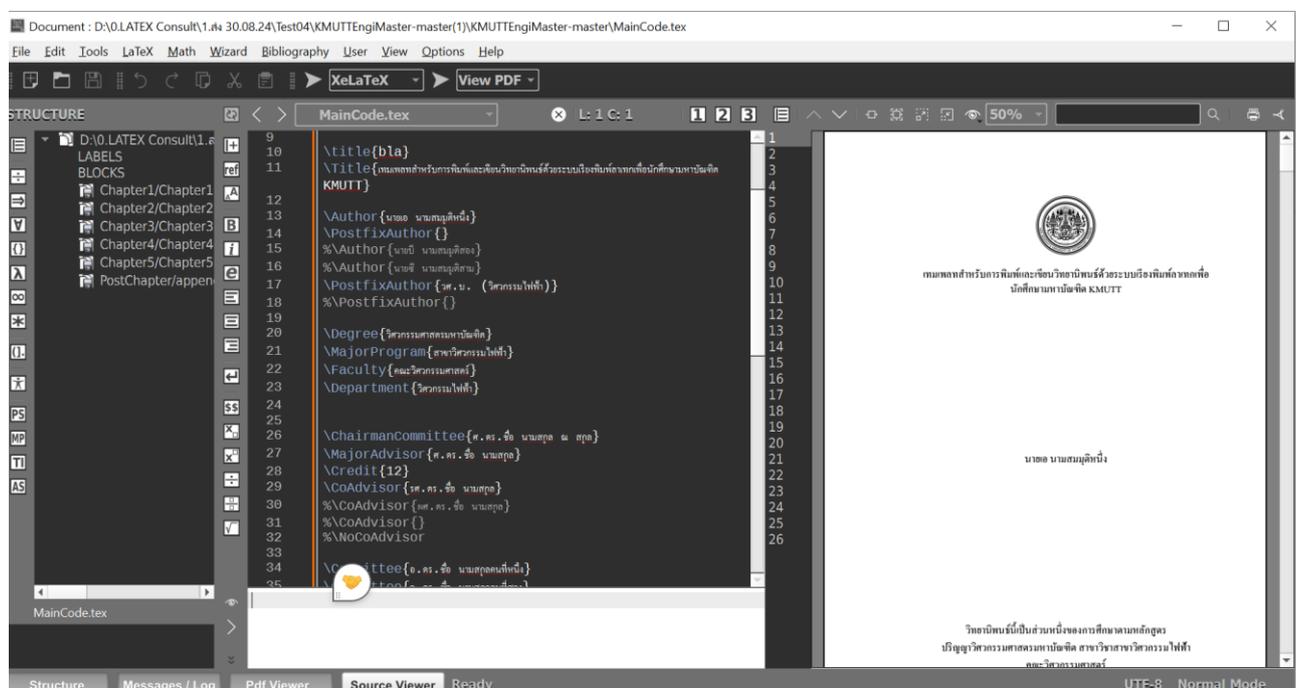
Once completed, you can rename the PDF file as desired. Double-check everything before using it, and the final thesis will appear as shown on the right. The PDF file will automatically be saved in the designated folder.

Each time you compile a LaTeX document, the system generates several auxiliary files with extensions such as .aux, .toc, .lot, .lof, and .blg. These files play a crucial role in assembling specific elements of your document, such as bibliographic references, table of contents, lists of tables, and figures.

In the initial state of the working directory (before any compilation), these auxiliary files do not exist. Consequently, critical document elements like cross-references and indices may not appear correctly after the first compilation.

To ensure that all document elements are correctly updated, it is recommended to compile your LaTeX document multiple times—at least twice. During the first run, LaTeX gathers the necessary data to populate references and lists. The second run processes this data and integrates it into the final document structure. Further run process may require, for example to arrange some information.

By following this multi-pass compilation process, you can ensure that your document is accurately formatted with all references and supplementary information correctly displayed.



Note: When you click **Run** and **Create PDF** again, make sure that the previously created PDF file is not open. If the PDF is open, the program will not be able to save the newly generated file. Therefore, after reviewing the PDF, if you need to make changes, you must close the PDF file before running the program again to ensure that the latest edited version can be saved.

(5) Cross-Referencing in Documents

Cross-referencing in LaTeX is explained in Section 2.5, where the use of `\ref{key}` retrieves only the numerical sequence of the referenced element. In common practice, references often include the type of element being cited, such as “Section x.x.x” or “Figure x.x.”

To facilitate this, the KMUTTEngiMaster class defines macros for referencing elements in both Thai and English. The language version is determined by the english option. These macros allow for more precise and user-friendly cross-referencing as follow:

Thai Version					
Macro	Used for	Result	Macro	Used for	Result
<code>\chapref{key}</code>	chapter	บทที่ x	<code>\figref{key}</code>	figure	รูปที่ x.x
<code>\secref{key}</code>	section/ subsection/ paragraph ...	หัวข้อที่ x.x.x	<code>\tabref{key}</code>	table	ตารางที่ x.x
	<code>\eqref{key}</code>		equation	สมการที่ (x.x)	
English Version (first letter lowercase)					
Macro	Used for	Result	Macro	Used for	Result
<code>\chapref{key}</code>	chapter	chapter x	<code>\figref{key}</code>	figure	figure x.x
<code>\secref{key}</code>	section/ subsection/ paragraph ...	section x.x.x	<code>\tabref{key}</code>	table	table x.x
	<code>\eqref{key}</code>		equation	equation (x.x)	
English Version (first letter uppercase)					
Macro	Used for	Result	Macro	Used for	Result
<code>\Chapref{key}</code>	chapter	Chapter x	<code>\Figref{key}</code>	figure	Figure x.x
<code>\Secref{key}</code>	section/ subsection/ paragraph ...	Section x.x.x	<code>\Tabref{key}</code>	table	Table x.x
	<code>\Eqref{key}</code>		equation	Equation (x.x)	

The code example in section 2.5 can then be modified as follows:

```

\documentclass{report}
\begin{document}

\chapter{Introduction} % Begin Chapter 1 Introduction

\label{sec: Introduction } % Define label sec:Introduction

\section{Research Background} % Begin section 1.1
\label{sec:ResearchBG} % Define label sec: ResearchBG

According to the equation
\begin{equation}
a^2 + b^2 = c^2
\label{eq:Pythagoras} % Define label eq:Pythagoras

```

```
\end{equation}
```

```
, a Pythagorean equation  $\dots$ 
```

```
\section{Literature Review}
```

```
% Begin section 1.2
```

```
% Cross referencing eq:Pythagoras and sec:ResearchIntro
```

```
According to the \eqref{eq:Pythagoras} in \secref{sec:ResearchIntro} in page \pageref{eq:Pythagoras},  
 $\dots$ 
```

```
\end{document}
```

Chapter 5

Thesis Guidelines

5.1 Thesis Printing (already configured in the KMUTTEngiMaster class)

5.1.1 The paper used for printing the thesis must be white, 80-gram, unlined, in standard A-4 size, and printed on one side only.

5.1.2 The font used should be black, clear, easy to read, and consistent throughout the entire document.

5.1.3 For Thai-language theses, use AngsanaUPC size 16 for the main content, and size 18-22 in bold for headings.

5.1.4 For English-language theses, use Times New Roman size 12 for the main content, and size 13-15 in bold for headings*

* Typesetting Thai alphabets in the English version

In the English version of the KMUTTEngiMaster class, the default font is Times New Roman, which does not support Thai characters. If Thai text needs to be typeset, users must switch to the pre-configured AngsanaUPC font. The following commands are provided to handle font switching:

● Switching to Thai Font:

```
\selectFontTH
```

This command changes the font to AngsanaUPC, allowing Thai text to be displayed correctly.

● Switching Back to English Font:

```
\selectFontEN
```

This command reverts the font back to Times New Roman for English text.

Example:

```
This is English text in Times New Roman.
```

```
\selectFontTH
```

```
นี่คือตัวอย่างข้อความภาษาไทยในฟอนต์ AngsanaUPC.
```

```
\selectFontEN
```

```
Now back to English text in Times New Roman.
```

Further examples of inserting Thai text, please refer to **Chapter 1.tex** and **references.tex**. Additionally, the file needs to be encoded to support the Unicode, i.e. in UTF-8.

Note: Although the `KMUTTEngiMaster` class can be configured to automatically switch fonts between Thai and Latin characters using the `ucharclasses` package with macros like `\setTransitionTo` and `\setTransitionFrom`, it is recommended to handle font switching manually as above. This approach ensures smoother typesetting, especially in sections containing a mix of Thai and English content.

5.2 Page Numbering (already configured in the `KMUTTEngiMaster` class)

5.2.1 For the preliminary section of a Thai thesis, use Thai letters in alphabetical order (๑, ๒, ๓, etc.). For an English thesis, use lowercase Roman numerals (i, ii, iii, etc.), printed in the top right corner.

5.2.2 For the main content until the end of the thesis, use Arabic numerals (1, 2, 3, etc.), printed in the top right corner with consistent margins. The first page of each chapter and appendix does not need a page number.

5.2.3 The font for page numbers in a Thai thesis should be AngsanaUPC 16 pt, and in an English thesis, Times New Roman 12 pt, consistent throughout the document.

5.3 Line Spacing (already configured in the `KMUTTEngiMaster` class)

5.3.1 Use 1.15 line spacing for Thai-language theses formatted as single line, and 1.5 line spacing for English-language theses formatted as single line.

5.4 Additional Thesis Writing Guidelines

5.4.1 Margins: Left 4 cm, Right 2 cm, Top 3 cm, Bottom 2 cm throughout the document. (already configured in the `KMUTTEngiMaster` class)

5.4.2 For name titles (in English), use Mr./Mrs. or Miss (without a period after "Miss"), followed by one space, then the first name, one space, and the last name.

5.4.3 There should be one space before and after a number when followed by text (in both Thai and English).

5.4.4 Leave one space between Thai and English characters.

5.4.5 Leave one space before and after parentheses, but do not insert a space between the parentheses and the text inside.

5.4.6 Do not use commas (,) in Thai abstracts, except for numbers.

5.4.7 Do not insert spaces before commas (,) colons (:) or semi-colons (;), but leave one space after these punctuation marks.

5.4.8 Avoid using passive voice in Thai abstracts (if translated from an English abstract, rewrite it in active voice).

5.4.9 Verify that Thai terms translated from English follow the Royal Institute of Thailand's guidelines. For example, Property = สมบัติ (not คุณสมบัติ) / Kinetics = จลนพลศาสตร์, etc.

5.4.10 For words using the Thai repetition mark (๑), no space is needed before it, but leave one space after it.

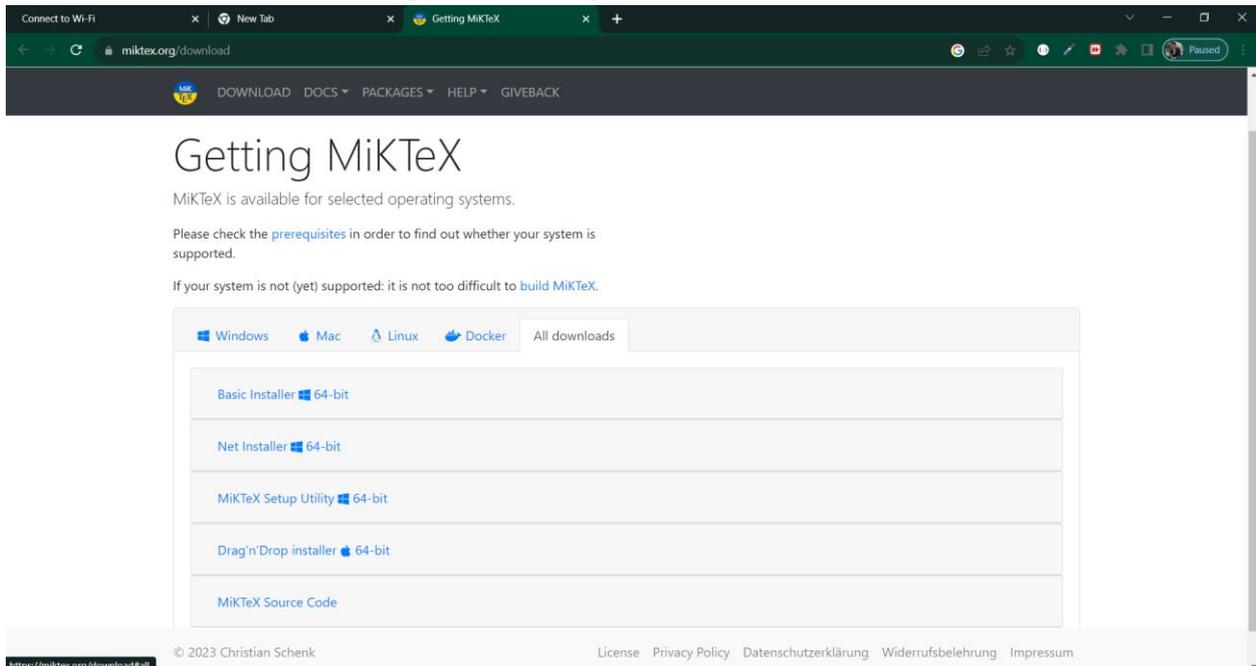
5.4.11 For headings spanning two or more lines, the second and subsequent lines should be left-aligned.

Appendix

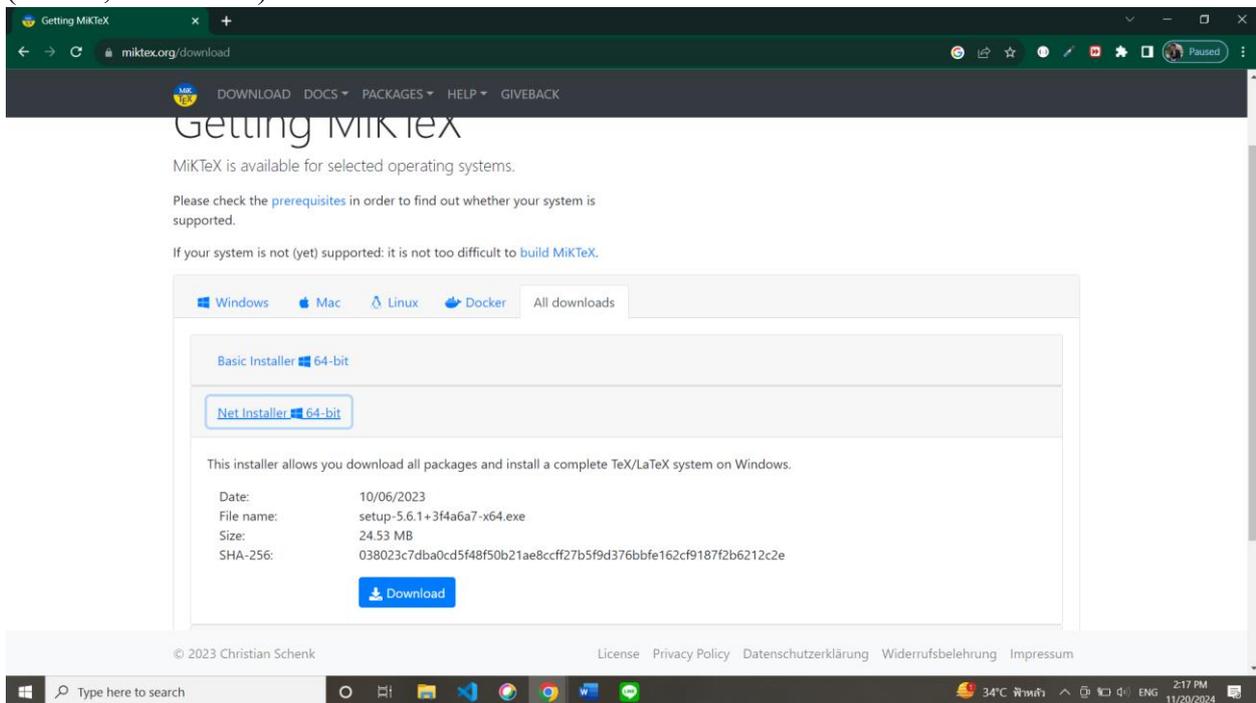
Installing the MikTeX and Texmaker programs (Using Both)

Installing MikTeX

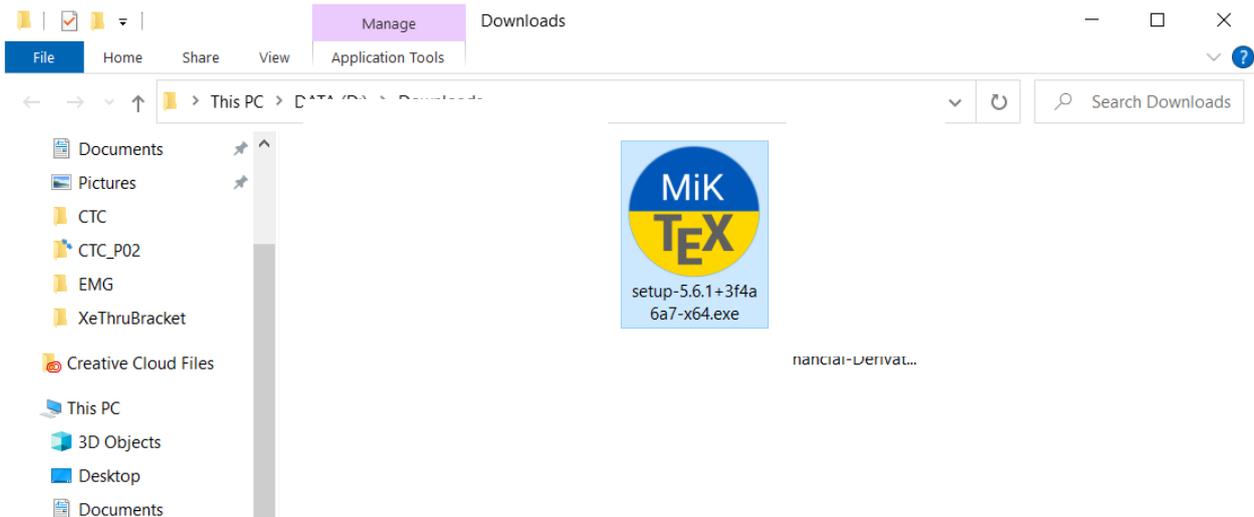
1. Download MikTeX from <https://miktex.org/download>



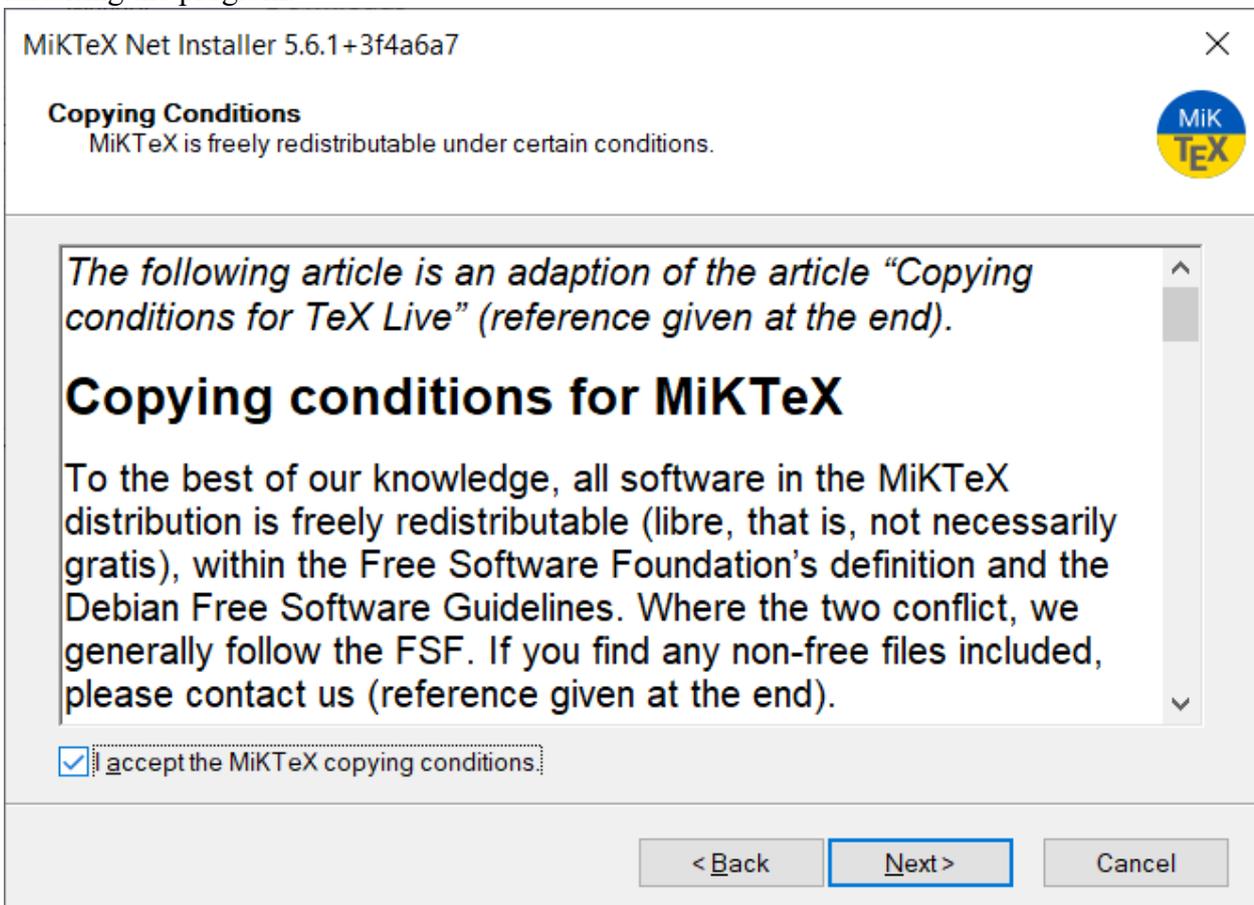
2. Select All download and press download program in Net installer 64-bit version for Windows (If Mac, select Mac)



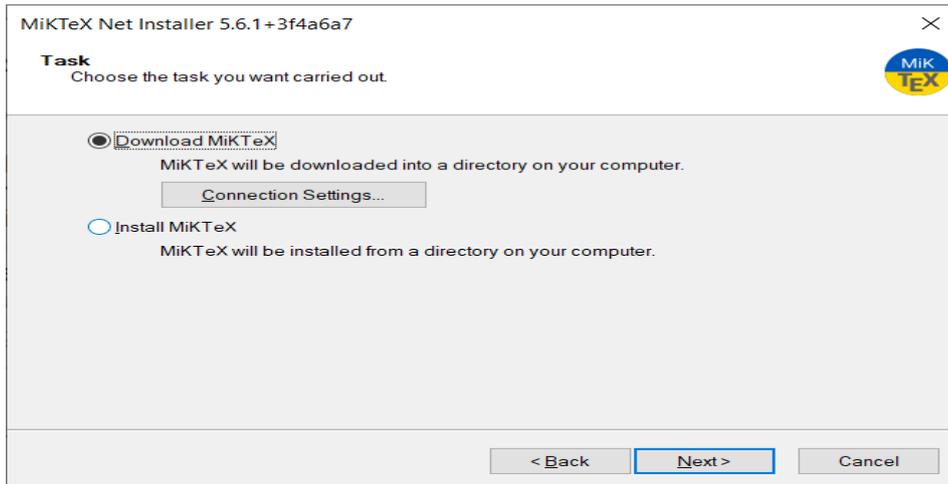
3. When the download is complete, double click the setup file.



4. Select I accept MikTeX copyright conditions if you agree with the policy and want to continue installing the program.

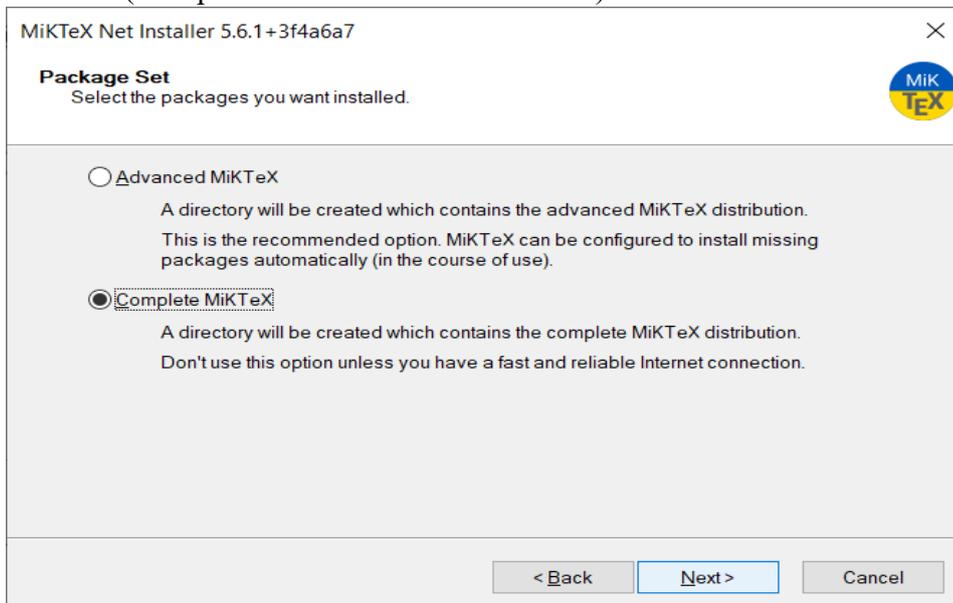


5. Select Download MikTeX

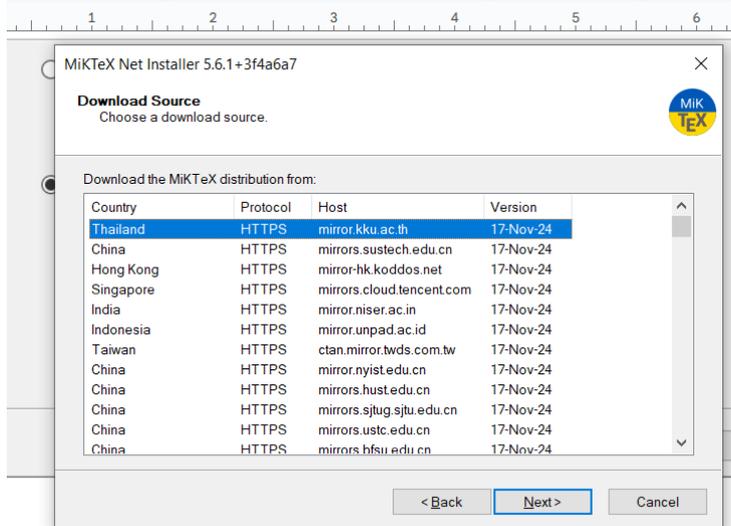


6. Select Complete MikTeX

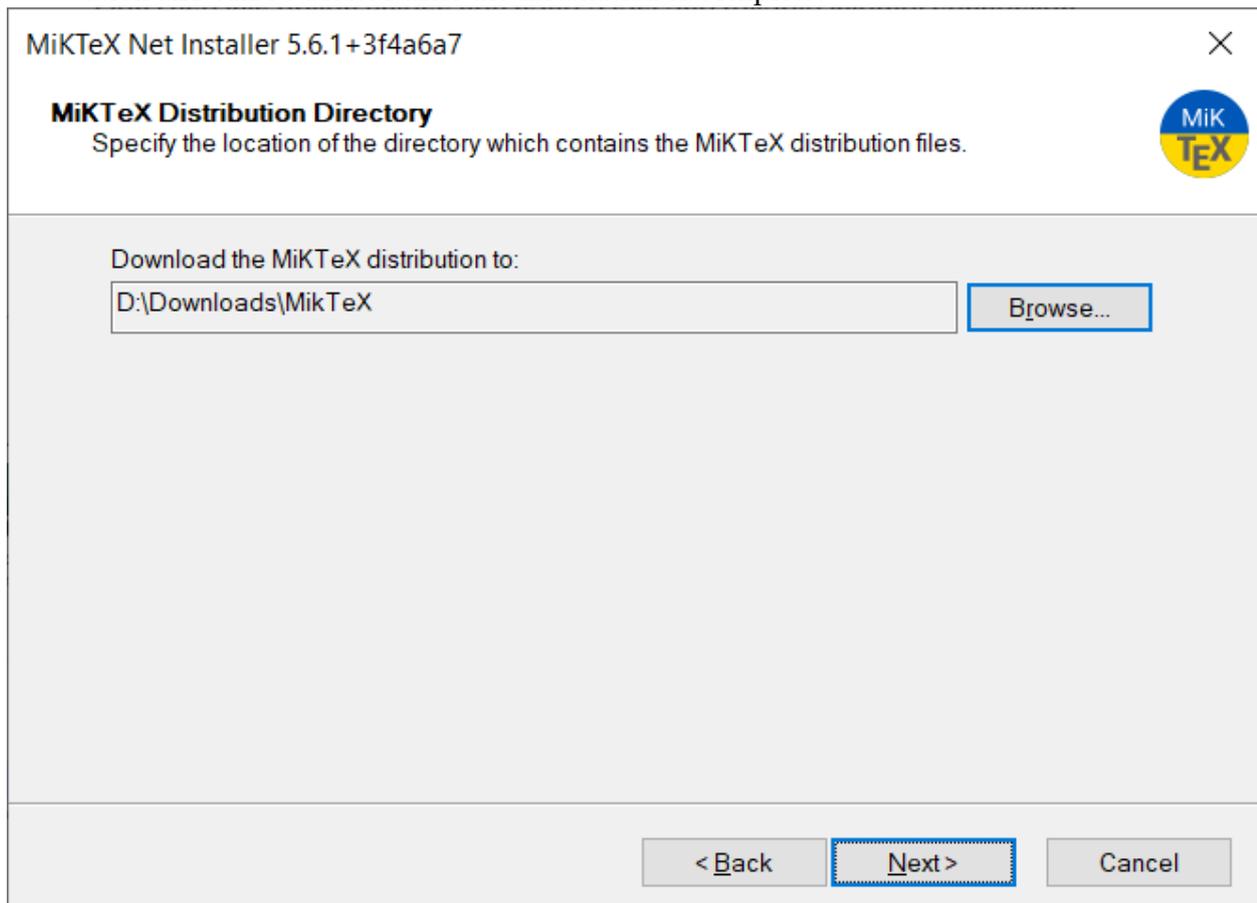
Note In practice, Advanced MikTeX installation takes less space because only the necessary packages are installed. If additional packages are called, the program will download them if the user allows it (Complete MikTeX is recommended).



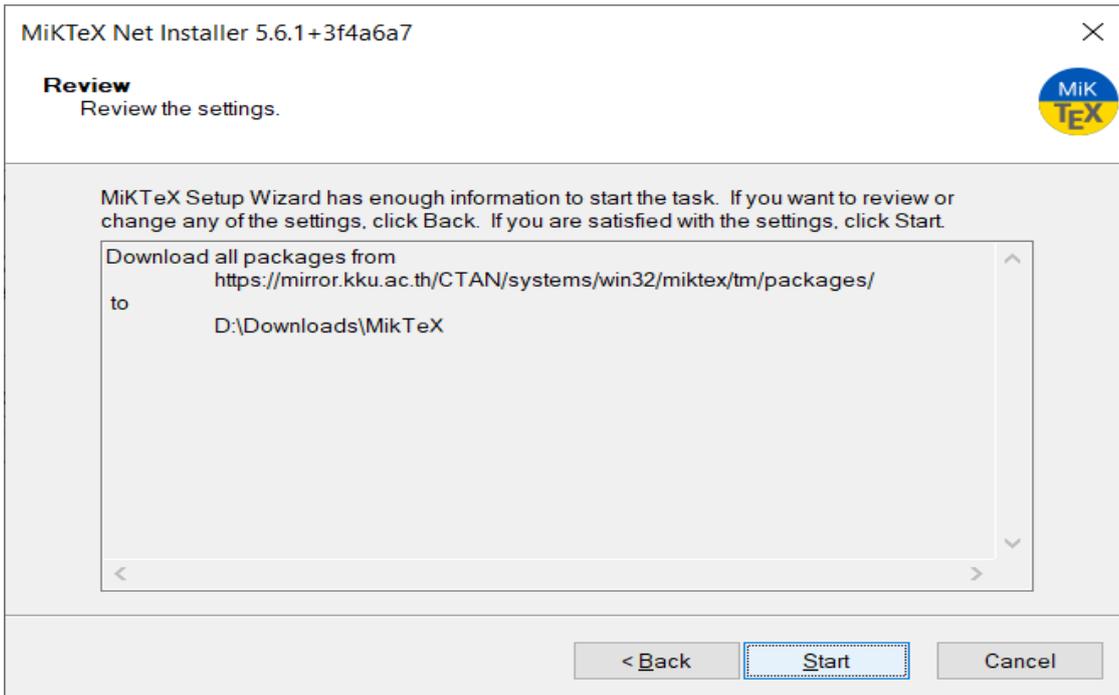
7. Select the appropriate server for downloading. In case the program has an error during downloading, it is recommended to open the setup file again and change the server.



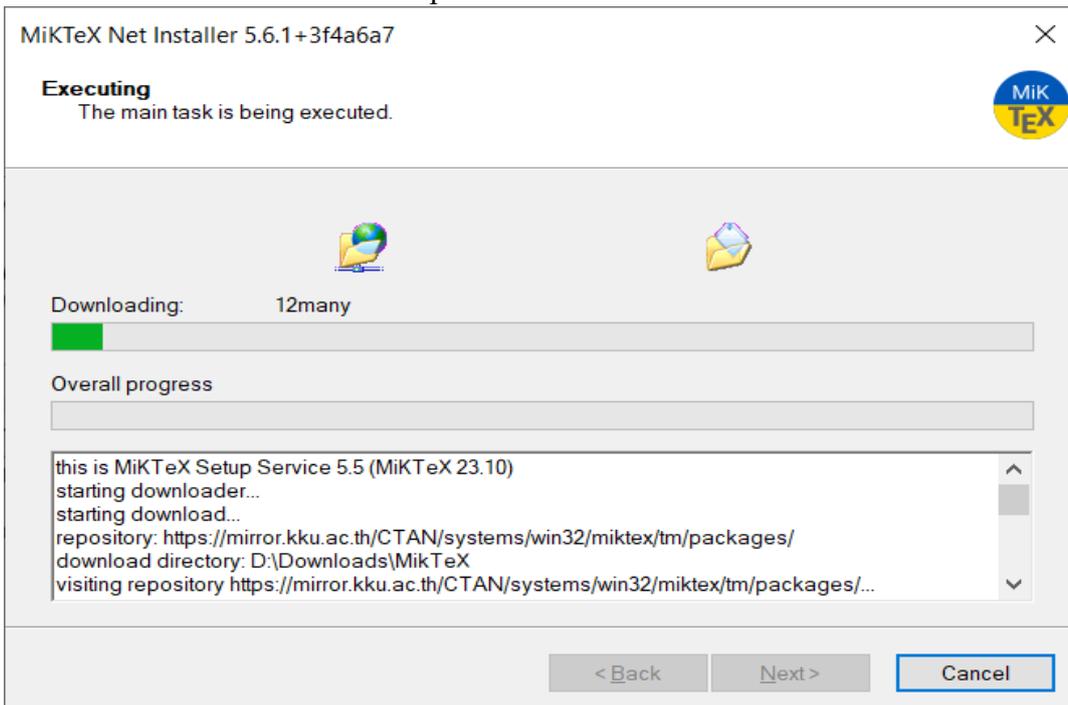
8. Select the file to store the data. This is temporary installation data that will be used only for the installation and can be deleted after the installation is complete.

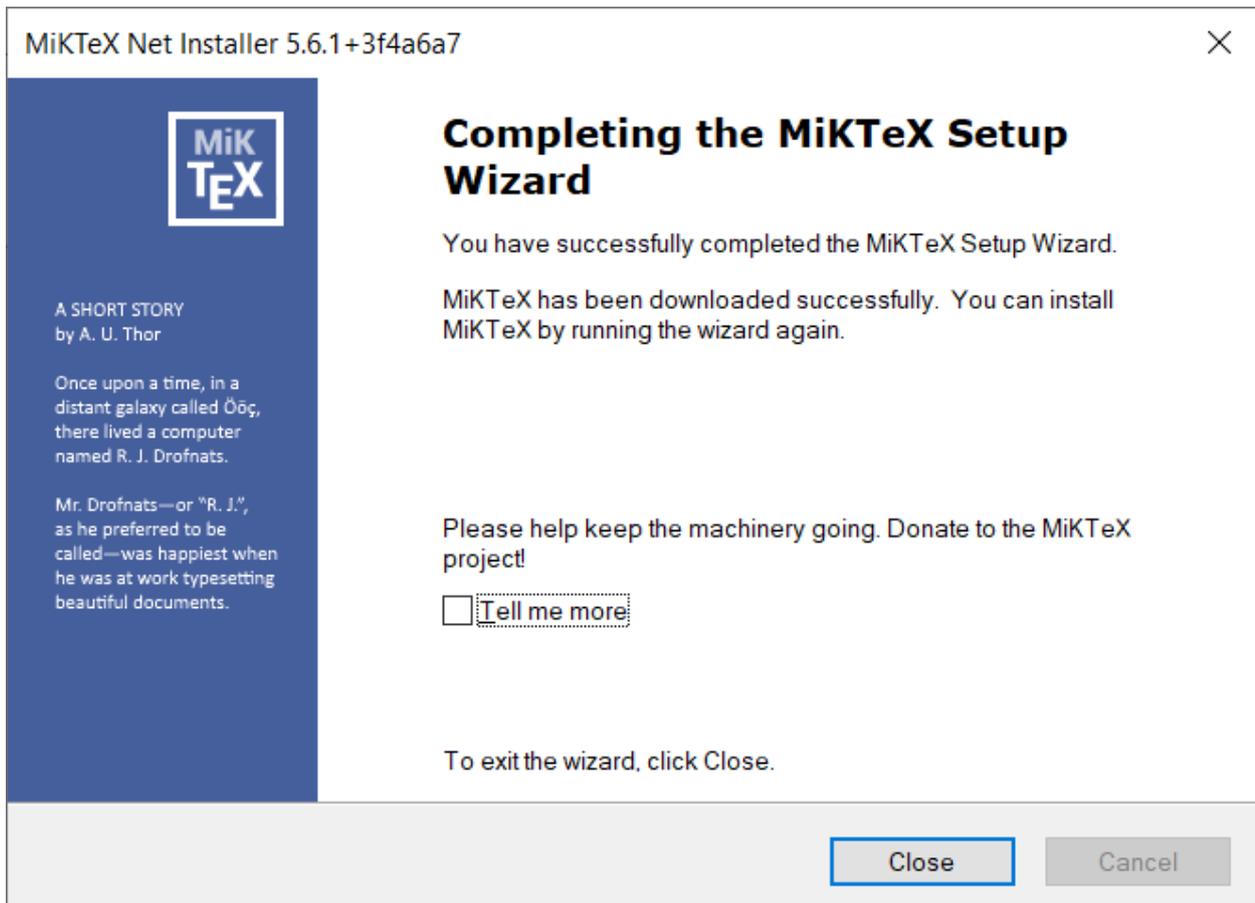
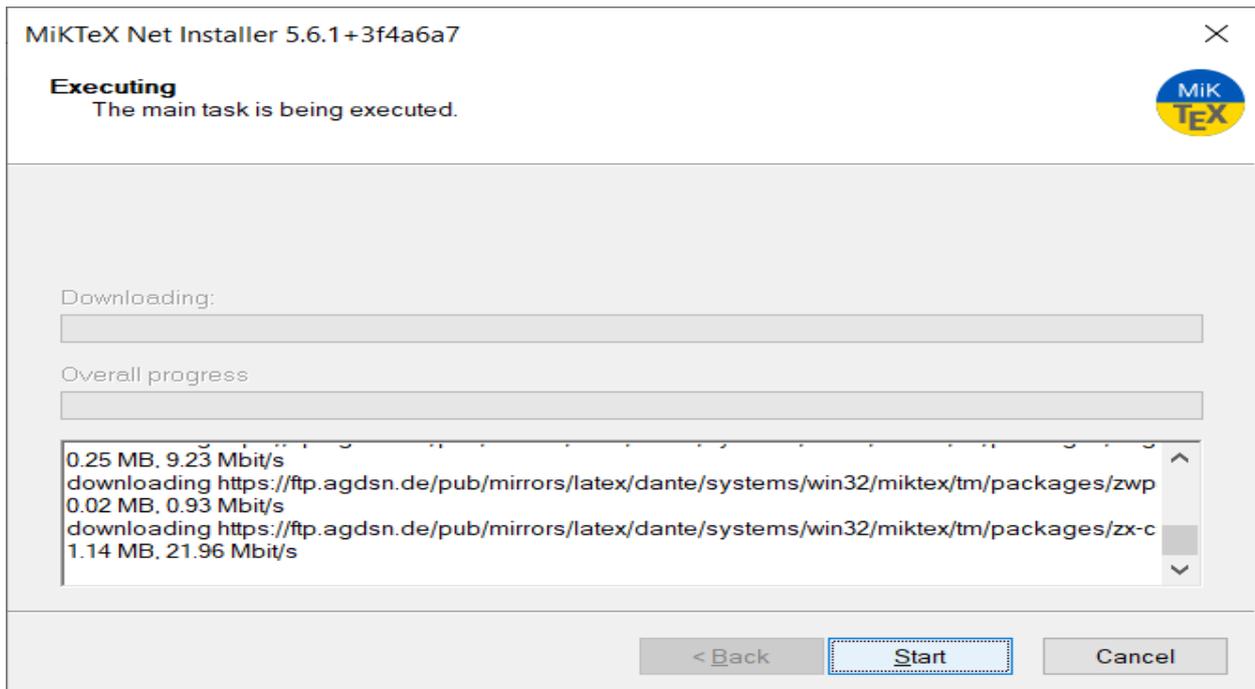


9. Select Start

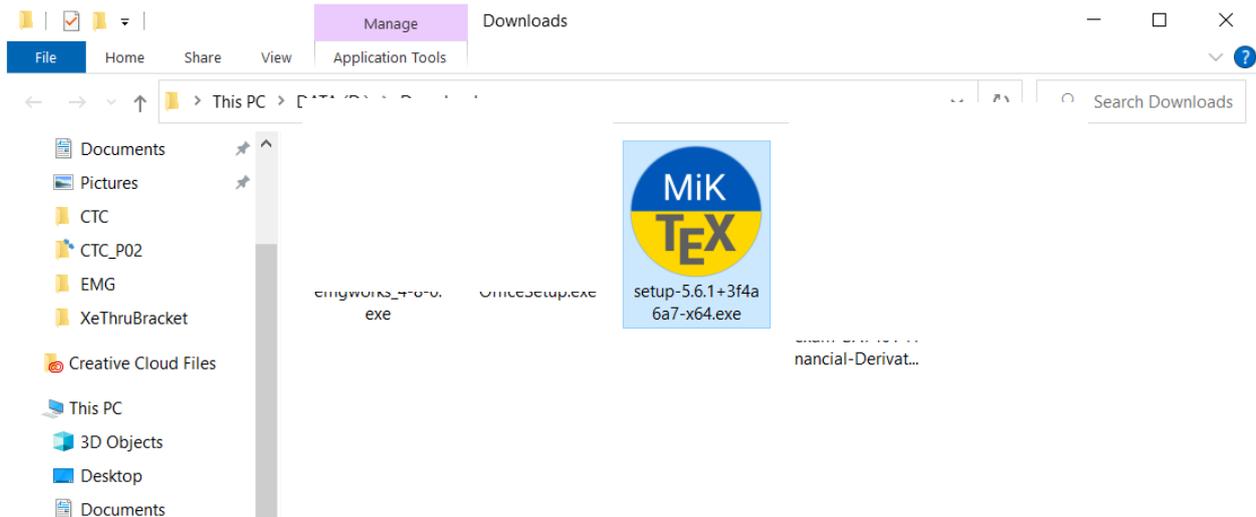


10. Wait until the download is complete, then press Start as shown in the next picture and press Close when the download is complete.

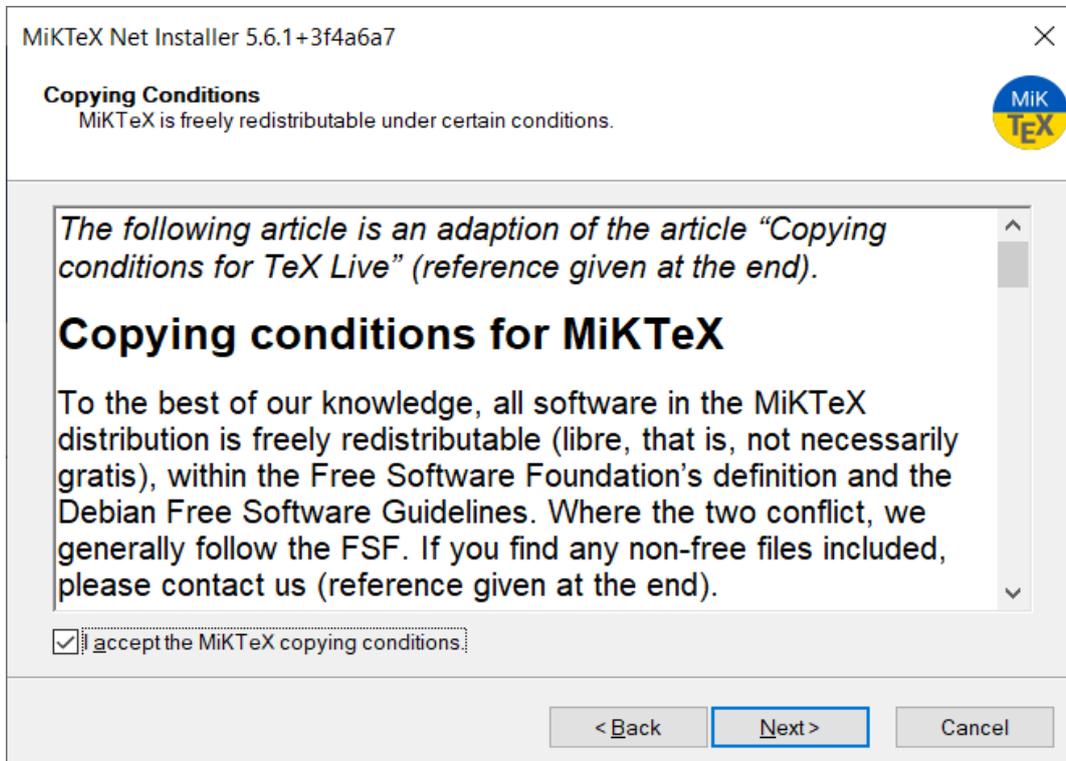




11. Click here again to install the program.



12. Click Next



13. Select Install MiKTeX then proceed with the steps as shown in the next picture

MiKTeX Net Installer 5.6.1+3f4a6a7 ✕

Task
Choose the task you want carried out.

Download MiKTeX
MiKTeX will be downloaded into a directory on your computer.
[Connection Settings...](#)

Install MiKTeX
MiKTeX will be installed from a directory on your computer.

[< Back](#) [Next >](#) [Cancel](#)

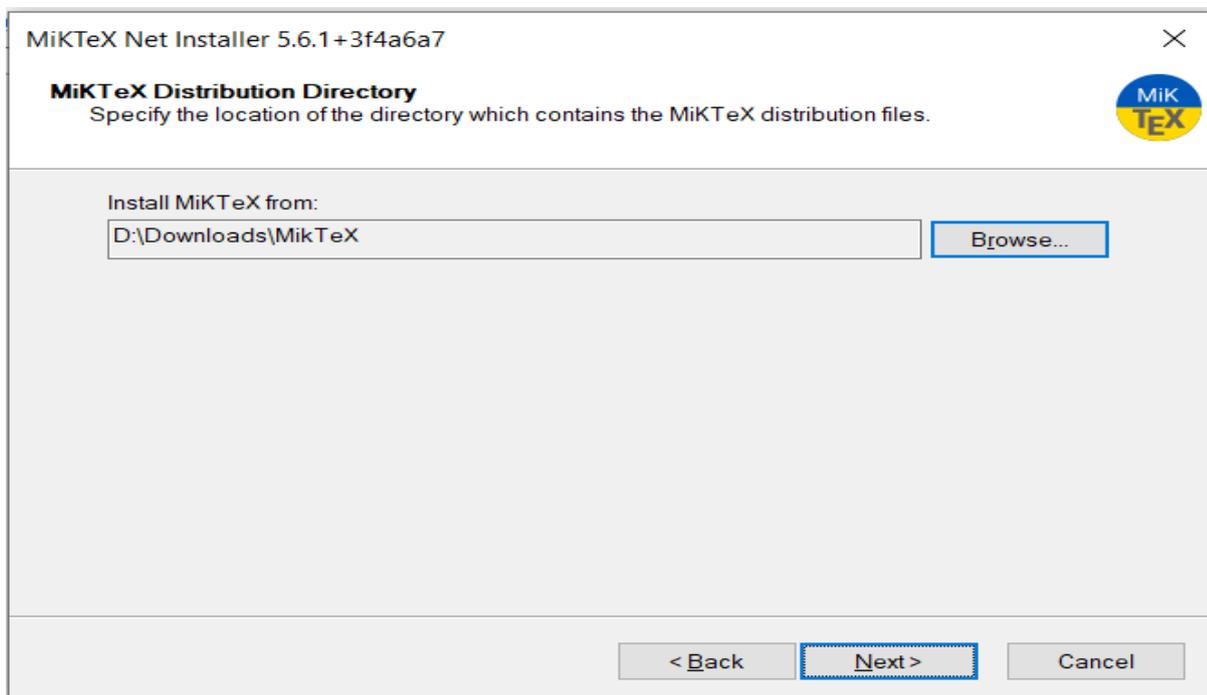
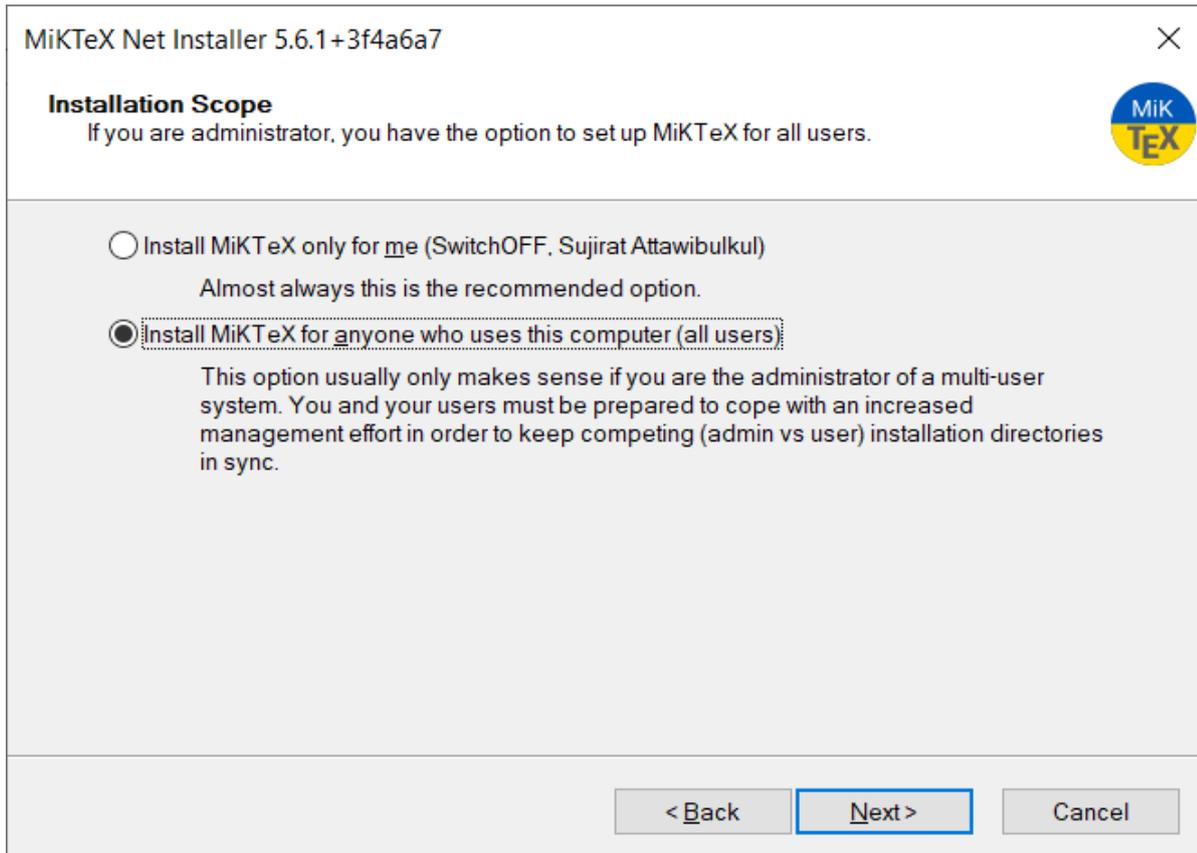
MiKTeX Net Installer 5.6.1+3f4a6a7 ✕

Package Set
Select the packages you want installed.

Advanced MiKTeX
This is the recommended option. MiKTeX can be configured to install missing files automatically (in the course of use).

Complete MiKTeX

[< Back](#) [Next >](#) [Cancel](#)



MiKTeX Net Installer 5.6.1+3f4a6a7 ✕

Settings
Set your preferences.



Preferred paper: ▾

Install missing packages on-the-fly: ▾

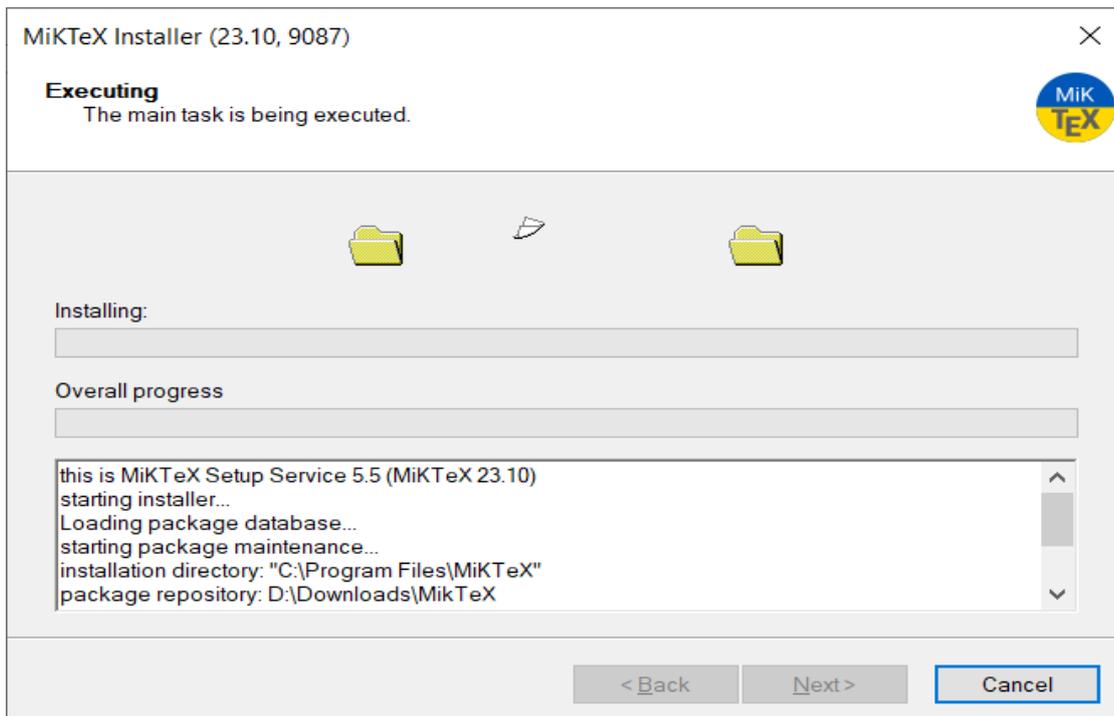
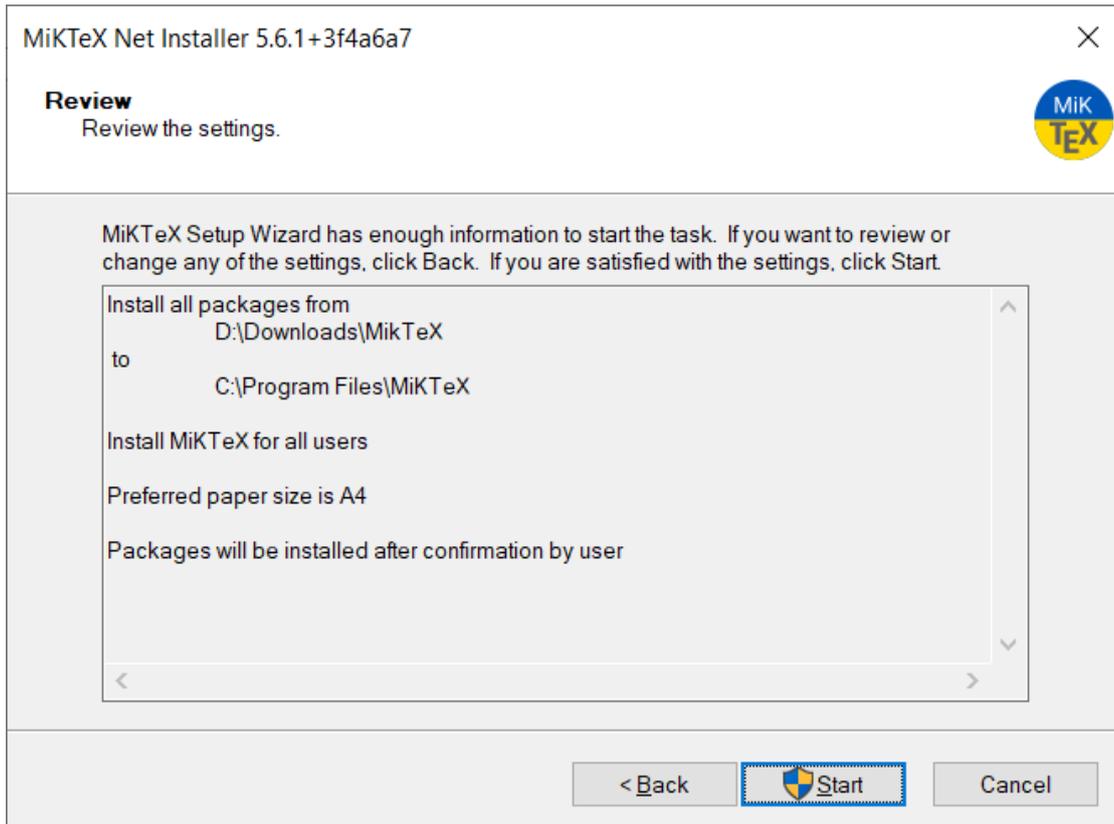
MiKTeX Net Installer 5.6.1+3f4a6a7 ✕

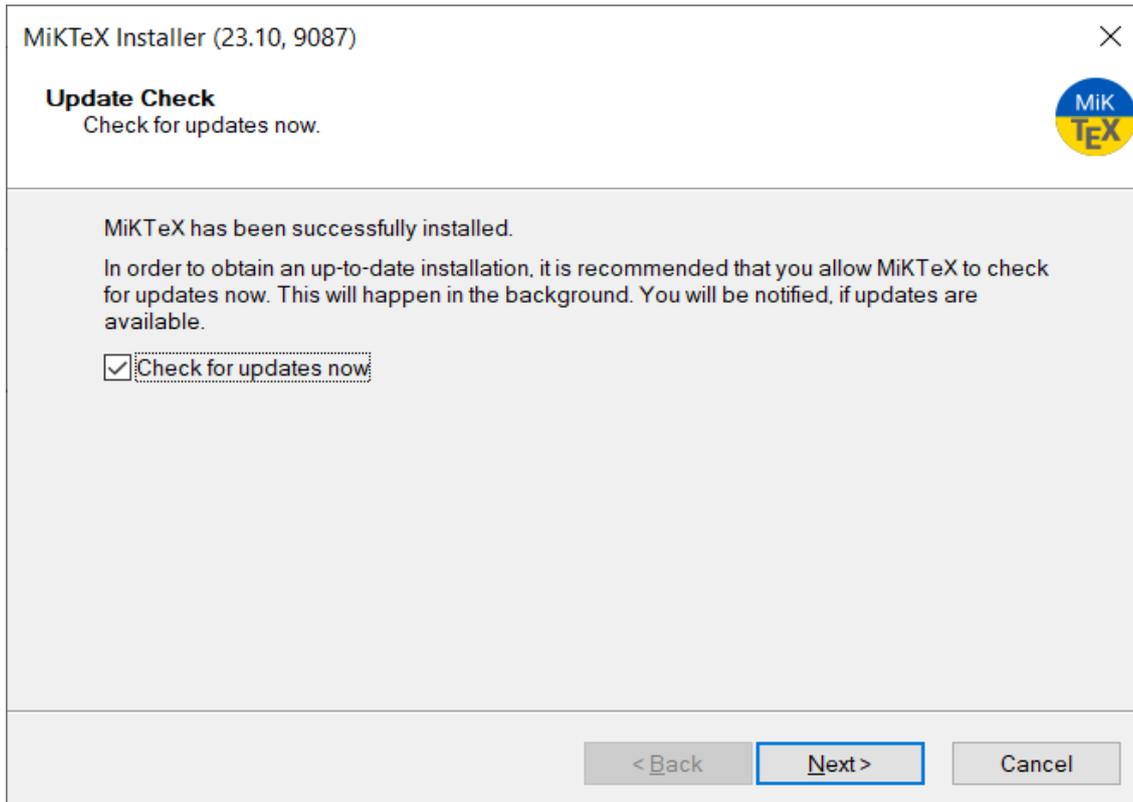
Settings
Set your preferences.



Preferred paper: ▾

Install missing packages on-the-fly: ▾





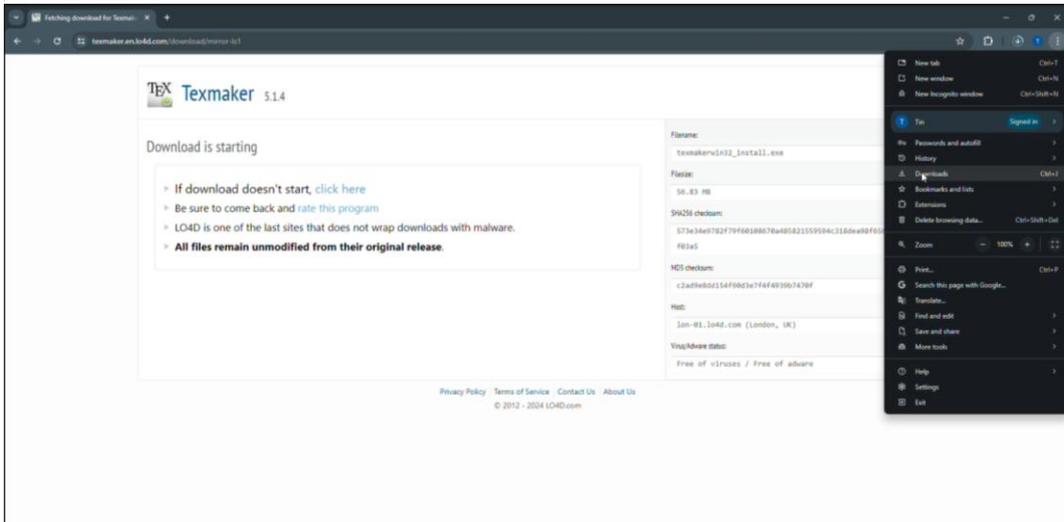
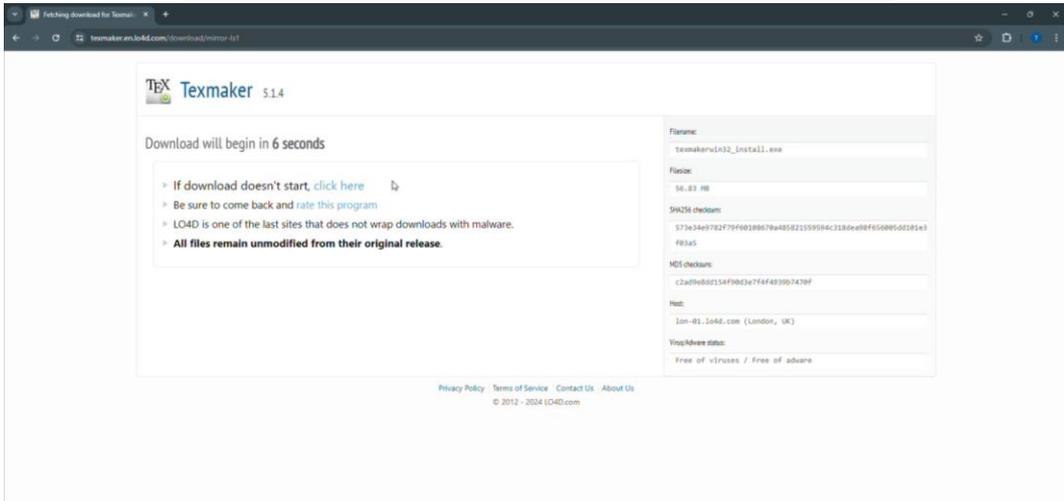
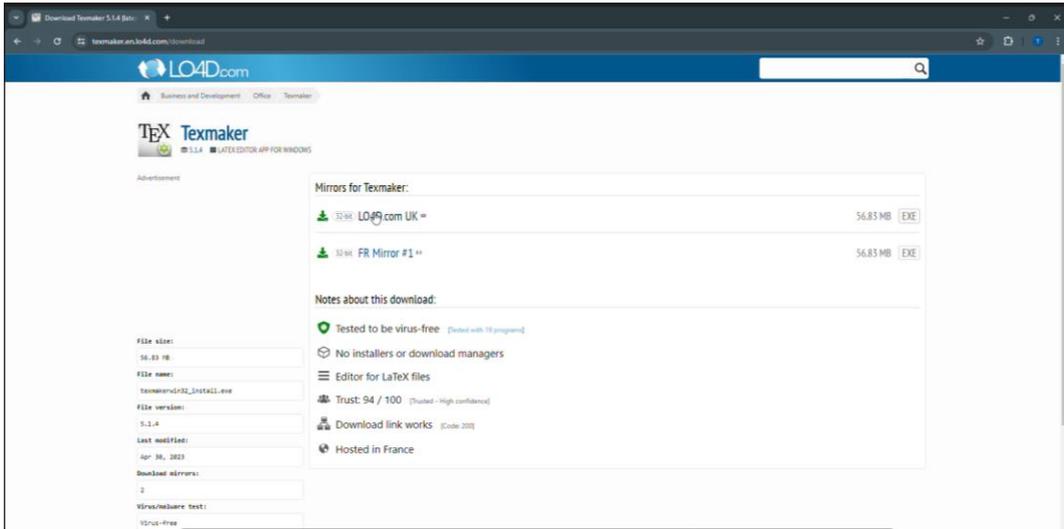
Installation of Texmaker

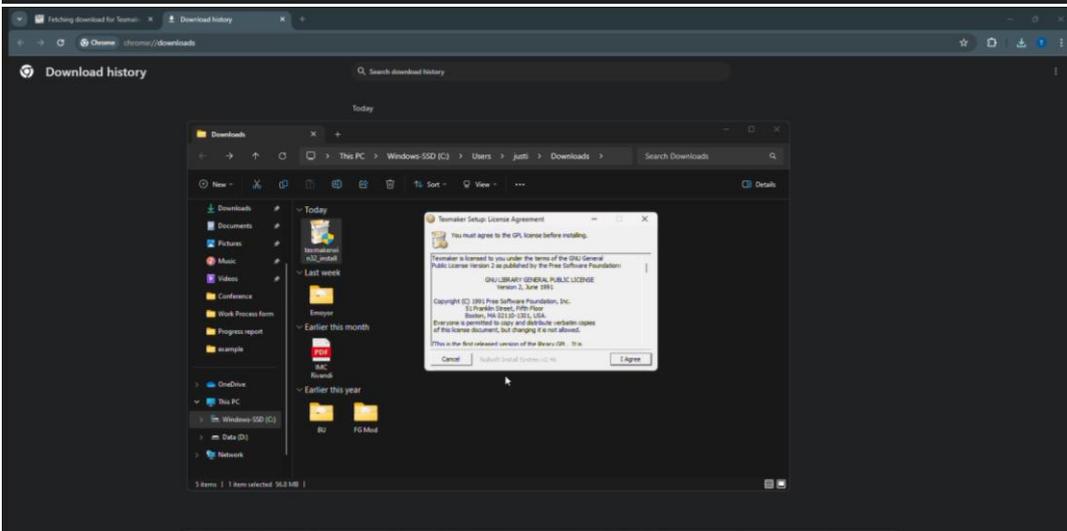
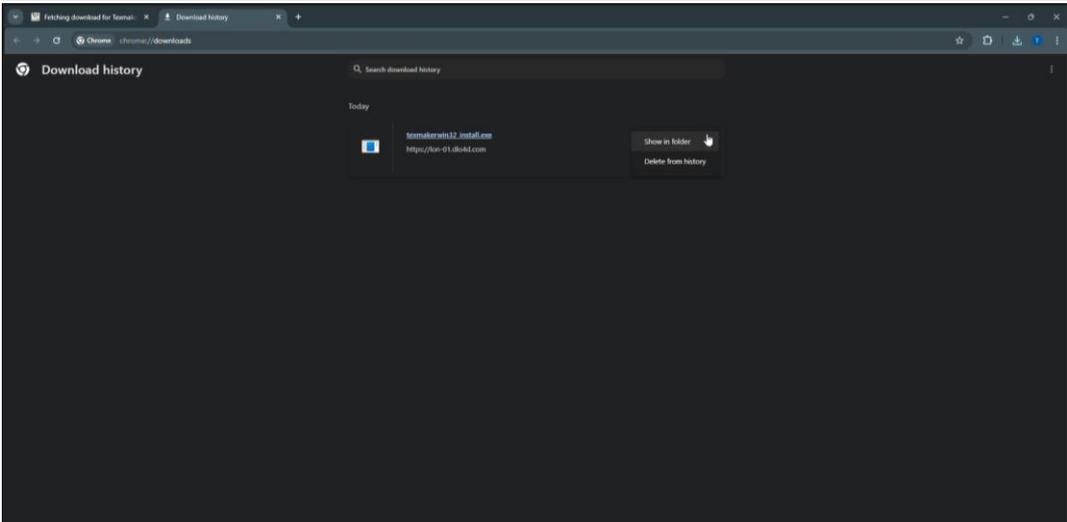
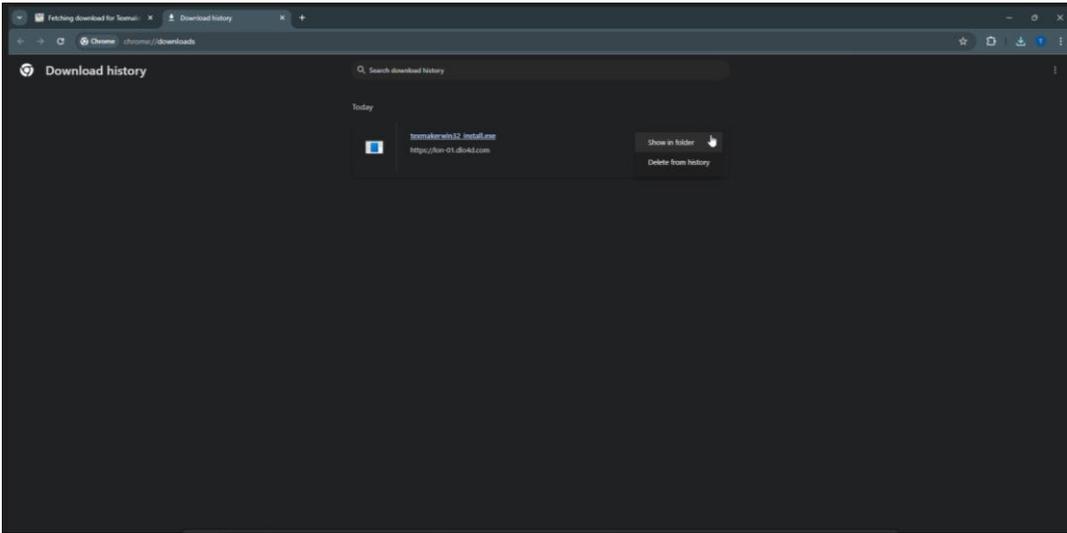
Google search results for 'Texmaker' showing 'People also ask' and 'People also search for' sections.

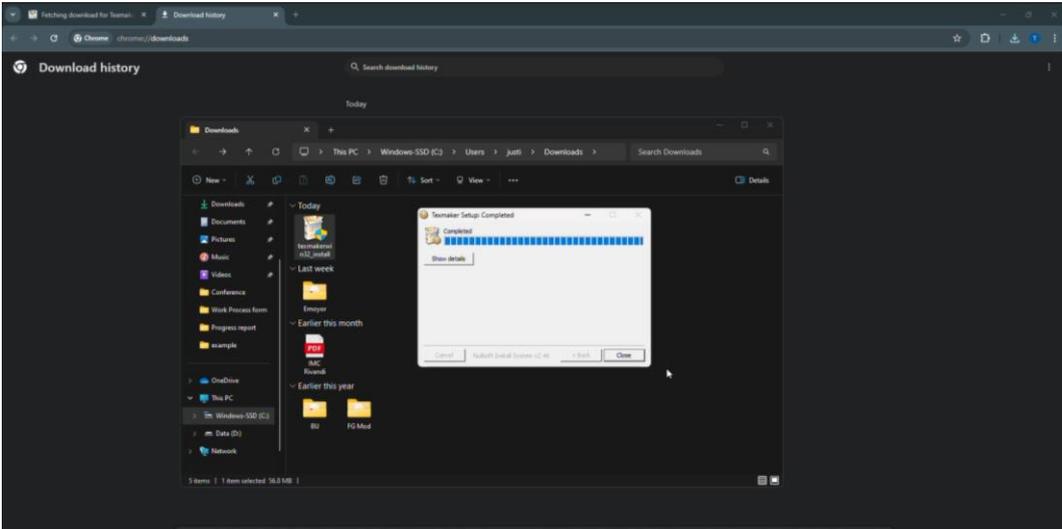
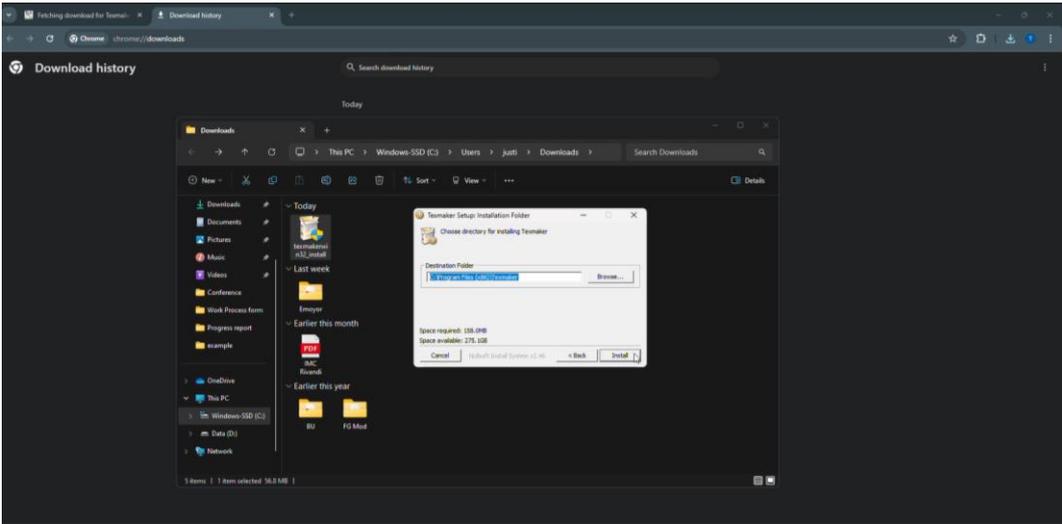
LO4D.com download page for Texmaker. The page includes the following information:

- Software to edit and typeset LaTeX documents, used for academic papers, presentations and more.
- Download for Windows (56.83 MB, Apr 30, 2023)
- User Rating: 4.1/5 (127)
- File Tested

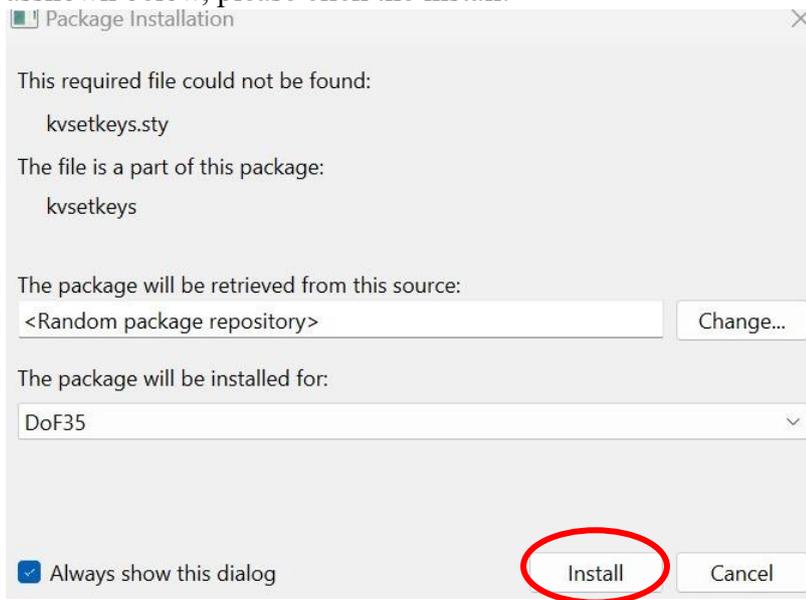
Quote: "A LaTeX text editor with all of the tools and pointers necessary including complex LaTeX commands, labels, footnotes and indexes as well as PDF creation in a clean interface."







When using the program for the first time, there will be a window to install the package as shown below, please click the install.



References

Kopka, H., & Daly, P. W. (2003). Guide to LaTeX (4th ed.). Addison-Wesley Professional.

Lamport, L. (1994). LaTeX: A document preparation system (2nd ed.). Addison-Wesley.

LaTeX Project. (n.d.). User's guide.

Retrieved from <https://www.latex-project.org/help/documentation/>

Overleaf. (n.d.). Documentation: Document classes. Overleaf. Retrieved from https://www.overleaf.com/learn/latex/Document_classes

Overleaf. (n.d.). Inserting equations. Retrieved from <https://www.overleaf.com/learn>

Overleaf. (n.d.). Overleaf: Learn. Retrieved from <https://www.overleaf.com/learn>

Wikibooks. (n.d.). LaTeX. In Wikibooks. Retrieved from <https://en.wikibooks.org/wiki/LaTeX>

Wikibooks. (n.d.). LaTeX Wikibook: Mathematics. Retrieved from <https://en.wikibooks.org/wiki/LaTeX/Mathematics>

คู่มือการเขียนและพิมพ์วิทยานิพนธ์ระดับบัณฑิตศึกษา คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี (ฉบับปรับปรุงเมื่อ 1 ตุลาคม 2560)