



**UNIVERSITAS INDONESIA**

**KEMAMPUAN LLM DALAM LOGICAL INFERENCE PADA  
DATASET BERBAHASA INDONESIA**

**SKRIPSI**

**MIKHAEL DEO BARLI  
1906350572**

**FAKULTAS ILMU KOMPUTER  
PROGRAM STUDI ILMU KOMPUTER  
DEPOK**

## **BULAN TAHUN**



**UNIVERSITAS INDONESIA**

**KEMAMPUAN LLM DALAM LOGICAL INFERENCE PADA  
DATASET BERBAHASA INDONESIA**

**SKRIPSI**

Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Gelar Jurusan Anda

**MIKHAEL DEO BARLI  
1906350572**

**FAKULTAS ILMU KOMPUTER  
PROGRAM STUDI ILMU KOMPUTER  
DEPOK**

## **BULAN TAHUN**

## **HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**

**Nama : Mikhael Deo Barli**

**NPM : 1906350572**

**Tanda Tangan :**

**Tanggal : Tanggal Bulan Tahun**



## **HALAMAN PENGESAHAN**

Skripsi ini diajukan oleh :

Nama : Mikhael Deo Barli

NPM : 1906350572

Program Studi : Ilmu Komputer

Judul Skripsi : Kemampuan LLM dalam Logical Inference pada  
Dataset Berbahasa Indonesia

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana pada Program Studi Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Indonesia.**

## **DEWAN PENGUJI**

Pembimbing 1 : Pembimbing Pertama Anda ( )

Penguji 1 : Penguji Pertama Anda ( )

Penguji 2 : Penguji Kedua Anda ( )

Ditetapkan di : Depok

Tanggal : Tanggal Bulan Tahun



## KATA PENGANTAR

Template ini disediakan untuk orang-orang yang berencana menggunakan L<sup>A</sup>T<sub>E</sub>X untuk membuat dokumen tugas akhir.

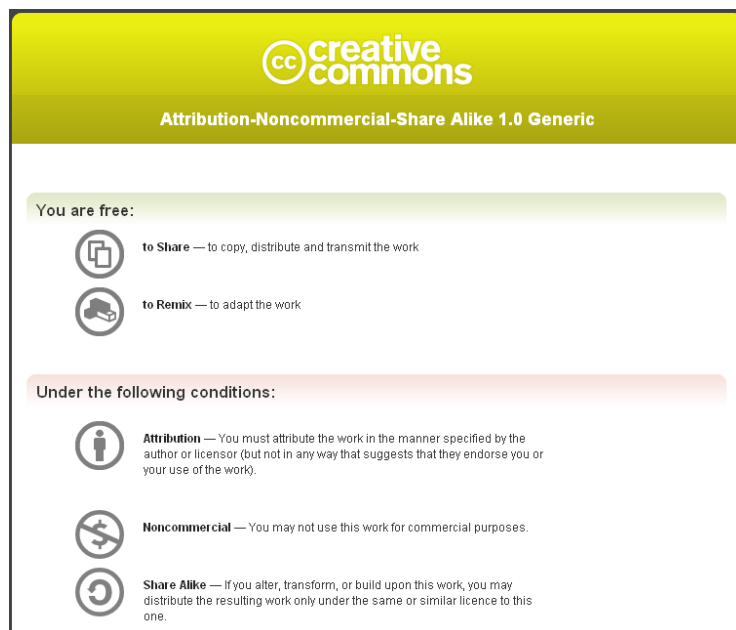
**@todo**

Silakan ganti pesan ini dengan pendahuluan kata pengantar Anda.

Ucapan Terima Kasih:

1. Pembimbing.
2. Dosen.
3. Instansi.
4. Orang tua.
5. Sahabat.
6. Teman.

Penulis menyadari bahwa laporan Skripsi ini masih jauh dari sempurna. Oleh karena itu, apabila terdapat kesalahan atau kekurangan dalam laporan ini, Penulis memohon agar kritik dan saran bisa disampaikan langsung melalui *e-mail* [emailanda@mail.id](mailto:emailanda@mail.id).



*Creative Common License 1.0 Generic*

Terkait template ini, gambar lisensi di atas diambil dari [http://creativecommons.org/licenses/by-nc-sa/1.0/deed.en\\_CA](http://creativecommons.org/licenses/by-nc-sa/1.0/deed.en_CA). Jika ingin mengetahui lebih lengkap mengenai *Creative Common License 1.0 Generic*, silahkan buka <http://creativecommons.org/licenses/by-nc-sa/1.0/legalcode>. Seluruh dokumen yang dibuat dengan menggunakan template ini sepenuhnya menjadi hak milik pembuat dokumen dan bebas didistribusikan sesuai dengan keperluan masing-masing. Lisensi hanya berlaku jika ada orang yang membuat template baru dengan menggunakan template ini sebagai dasarnya.

Penyusun template ingin berterima kasih kepada Andreas Febrian, Lia Sadita, Fahrurrozi Rahman, Andre Tampubolon, dan Erik Dominikus atas kontribusinya dalam template yang menjadi pendahulu template ini. Penyusun template juga ingin mengucapkan terima kasih kepada Azhar Kurnia atas kontribusinya dalam template yang menjadi pendahulu template ini.

Semoga template ini dapat membantu orang-orang yang ingin mencoba menggunakan  $\text{\LaTeX}$ . Semoga template ini juga tidak berhenti disini dengan ada kontribusi dari para penggunanya. Jika Anda memiliki perubahan yang dirasa penting untuk disertakan dalam template, silakan lakukan *fork* repositori Git template ini di <https://gitlab.com/ichlaffterlalu/latex-skripsi-ui-2017>, lalu lakukan *merge request* perubahan Anda terhadap *branch* master. Kami berharap agar *template* ini dapat terus diperbarui mengikuti perubahan ketentuan dari pihak Rektorat Universitas Indonesia, dan hal itu tidak mungkin terjadi tanpa kontribusi dari teman-teman sekalian.

Depok, Tanggal Bulan Tahun

Mikhael Deo Barli

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Mikhael Deo Barli  
NPM : 1906350572  
Program Studi : Ilmu Komputer  
Jenis Karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

Kemampuan LLM dalam Logical Inference pada Dataset Berbahasa Indonesia

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada tanggal : Tanggal Bulan Tahun  
Yang menyatakan

(Mikhael Deo Barli)



## **ABSTRAK**

Nama : Mikhael Deo Barli  
Program Studi : Ilmu Komputer  
Judul : Kemampuan LLM dalam Logical Inference pada Dataset  
Berbahasa Indonesia  
Pembimbing : Pembimbing Pertama Anda

Isi abstrak.

Kata kunci:

*Keyword* satu, kata kunci dua

## **ABSTRACT**

Name : Mikhael Deo Barli  
Study Program : Computer Science  
Title : The Capability of LLM in Logical Inference on Indonesian Language Dataset  
Counselor : Pembimbing Pertama Anda

Abstract content.

Key words:

Keyword one, keyword two

## DAFTAR ISI

<b>HALAMAN JUDUL</b>	<b>i</b>
<b>HALAMAN ORISINALITAS</b>	<b>iii</b>
<b>LEMBAR PENGESAHAN</b>	<b>iii</b>
<b>KATA PENGANTAR</b>	<b>iv</b>
<b>LEMBAR PERSETUJUAN KARYA ILMIAH</b>	<b>vi</b>
<b>ABSTRAK</b>	<b>vii</b>
<b>DAFTAR ISI</b>	<b>ix</b>
<b>DAFTAR GAMBAR</b>	<b>x</b>
<b>DAFTAR TABEL</b>	<b>xi</b>
<b>DAFTAR KODE PROGRAM</b>	<b>xi</b>
<b>DAFTAR LAMPIRAN</b>	<b>xii</b>
<b>1. Pendahuluan</b>	<b>1</b>
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan Penelitian	2
1.4 Batasan Penelitian	2
1.5 Manfaat Penelitian	3
1.6 Posisi Penelitian	3
1.7 Metodologi Singkat	3
1.8 Sistematika Penulisan	4
1.9 Panduan Singkat: Sitasi dan Cross-Reference	5
<b>2. KERANGKA BERPIKIR</b>	<b>6</b>
2.1 Apa itu $\text{\LaTeX}$ ?	6
2.1.1 $\text{\LaTeX}$ Secara Singkat	6
2.1.2 $\text{\LaTeX}$ Kompiler dan IDE	7
2.2 <i>Formatting</i> Teks Dasar	8
2.3 Memasukan Gambar	9
2.4 Membuat Tabel	11
2.4.1 Tabel Panjang (Lintas Baris)	13
2.4.2 Menggabungkan ( <i>Merge</i> ) Baris atau Kolom	15
2.5 Membuat Persamaan Matematis	19
2.5.1 Satu Persamaan	19
2.5.2 Lebih dari Satu Persamaan	20
2.6 Menambahkan Kode Program	23
2.7 Keterkaitan Teori Dengan Penelitian	26
<b>3. EKSPERIMEN DAN HASIL PENELITIAN</b>	<b>28</b>
3.1 Melakukan <i>Cross-Reference</i> ke Suatu Bagian dalam Laporan	28
3.2 Menggunakan BibTeX	29
3.2.1 Menambahkan Referensi	30
3.2.2 Melakukan Sitasi pada Konten Tugas Akhir	31
3.2.3 Mengubah Format Referensi/Sitasi	31
3.3 Membuat Daftar Istilah (Glosarium)	33

3.3.1	Menambahkan Istilah atau Akronim Baru . . . . .	33
3.3.2	Menggunakan Istilah atau Akronim dalam Dokumen . . . . .	34
3.4	Memasukan Berkas PDF . . . . .	34
3.5	Memberikan Catatan . . . . .	38
3.6	<i>Layoutting</i> Tingkat Lanjut . . . . .	38
3.6.1	Menambahkan Tabel/Gambar Panjang secara Lanskap . . . . .	38
3.6.2	<i>Alignment</i> dan <i>Word Wrapping</i> pada Tabel . . . . .	42
3.7	Daftar Isi atau Daftar Konten Lainnya . . . . .	44
3.7.1	Menambahkan Konten ke Daftar Isi/Lampiran Secara Manual . . . . .	44
3.7.2	Menambahkan Daftar Konten <i>Custom</i> . . . . .	44
3.8	Membuat Variabel atau Perintah Baru . . . . .	47
3.9	Pengaturan <i>Header</i> dan <i>Footer</i> . . . . .	48
3.9.1	Konfigurasi Satu Halaman per Lembar . . . . .	49
3.9.2	Konfigurasi untuk Submisi ke UI-ana . . . . .	49
<b>4.</b>	<b>ANALISIS DAN PEMBAHASAN . . . . .</b>	<b>50</b>
4.1	thesis.tex . . . . .	50
4.2	Direktori config . . . . .	50
4.2.1	settings.tex . . . . .	50
4.2.2	istilah.tex . . . . .	50
4.2.3	references.bib . . . . .	51
4.3	Direktori _internals . . . . .	51
4.3.1	hype.indonesia.tex . . . . .	51
4.3.2	uithesis.sty . . . . .	51
4.4	Direktori src/00-frontMatter . . . . .	52
4.5	Direktori src/01-body . . . . .	53
<b>5.</b>	<b>KASUS-KASUS KHUSUS . . . . .</b>	<b>54</b>
5.1	Tugas Akhir Individu S1, Proposal Tesis, dan Tesis S2 . . . . .	54
5.2	Tugas Akhir Kelompok S1 . . . . .	56
5.3	Laporan Ilmiah dan Disertasi S3 . . . . .	57
5.4	Laporan Kerja Praktik . . . . .	58
5.5	Laporan Kegiatan Merdeka Belajar Kampus Merdeka . . . . .	59
<b>6.</b>	<b>PENUTUP . . . . .</b>	<b>61</b>
6.1	Kesimpulan . . . . .	61
6.2	Saran . . . . .	61
	<b>DAFTAR REFERENSI . . . . .</b>	<b>62</b>
	<b>DAFTAR ISTILAH . . . . .</b>	<b>63</b>

## DAFTAR GAMBAR

Gambar 1.1.	Penjelasan singkat terkait gambar. . . . .	3
Gambar 2.1.	Makara Universitas Indonesia . . . . .	10
Gambar 2.2.	<i>Creative Common License 1.0 Generic</i> . . . . .	11
Gambar 2.3.	<i>User interface</i> dari website <a href="https://www.tablesgenerator.com/">https://www.tablesgenerator.com/</a> . . . .	11
Gambar 2.4.	Keterkaitan konsep hasil studi literatur terhadap penelitian . . . . .	27

## DAFTAR TABEL

Tabel 2.1.	Contoh Tabel . . . . .	13
Tabel 2.2.	Contoh Tabel Panjang . . . . .	15
Tabel 2.3.	Tabel dengan sel yang melebar ke lebih dari satu kolom . . . . .	16
Tabel 2.4.	Tabel dengan sel yang melebar ke lebih dari satu baris . . . . .	17
Tabel 2.5.	Tabel dengan sel yang melebar ke lebih dari satu kolom dan baris . . .	18
Tabel 3.1.	Contoh Tabel: Data Kasus COVID-19 di Asia, 14 September 2020 . .	39
Tabel 3.2.	Contoh Tabel: Perbandingan metode pemodelan <i>access control</i> . . . .	42

## DAFTAR KODE PROGRAM

Kode 2.1. Contoh penggunaan gambar . . . . .	9
Kode 2.2. Contoh penggunaan gambar bersumber . . . . .	10
Kode 2.3. Contoh penggunaan tabel . . . . .	12
Kode 2.4. Contoh penggunaan tabel panjang . . . . .	13
Kode 2.5. Contoh penggunaan tabel dengan sel yang melebar ke lebih dari satu kolom . . . . .	16
Kode 2.6. Contoh penggunaan tabel dengan sel yang melebar ke lebih dari satu baris	16
Kode 2.7. Contoh penggunaan tabel dengan sel yang melebar ke lebih dari satu kolom dan baris . . . . .	17
Kode 2.8. Kode pembuatan Persamaan 2.1 . . . . .	19
Kode 2.9. Kode pembuatan Persamaan 2.2 . . . . .	20
Kode 2.10. Kode pembuatan Persamaan 2.3 . . . . .	21
Kode 2.11. Kode pembuatan Persamaan 2.4, 2.5, dan 2.6 . . . . .	22
Kode 2.12. Kode sampel Java yang cukup panjang . . . . .	23
Kode 2.13. Meng . . . . .	23
Kode 2.14. Kode Python 2 . . . . .	25
Kode 2.15. Kode Python 3 . . . . .	25
Kode 3.1. Daftar referensi di <code>references.bib</code> . . . . .	30
Kode 3.2. Contoh definisi istilah baru . . . . .	33
Kode 3.3. Contoh definisi singkatan/akronim baru . . . . .	33
Kode 3.4. Contoh penggunaan istilah atau akronim dalam dokumen . . . . .	34
Kode 3.5. Kode untuk Tabel 3.2 . . . . .	43
Kode 3.6. Kode Definisi untuk Daftar Aturan Transformasi di <code>_internals/uithesis.sty</code> . . . . .	45

## DAFTAR LAMPIRAN

Lampiran 1. CHANGELOG . . . . .	64
Lampiran 2. Judul Lampiran 2 . . . . .	67

# **BAB 1**

## **PENDAHULUAN**

Bab ini memaparkan latar belakang, permasalahan, tujuan, batasan, manfaat, ringkasan metodologi, serta sistematika penulisan penelitian ini. Penelitian berfokus pada evaluasi kemampuan penalaran logis oleh Model Bahasa Besar (Large Language Models, LLM) ketika bekerja pada dataset berbahasa Indonesia.

### **1.1 Latar Belakang**

Perkembangan *Large Language Model* (LLM) telah mendorong kemajuan signifikan pada berbagai tugas pemrosesan bahasa alami seperti penerjemahan, ringkasan, dan tanya-jawab. Namun, kemampuan LLM untuk melakukan penalaran logis, yaitu menarik inferensi yang benar dari himpunan premis dan aturan formal, masih menghadapi kendala baik pada akurasi maupun keandalan, terutama di kasus yang memerlukan normalisasi, dekomposisi, dan pencarian bukti yang kompleks, serta resolusi logika.

Sebagian besar dataset penalaran dibuat dalam bahasa Inggris, sehingga studi terhadap kemampuan penalaran LLM pada bahasa lain, termasuk Bahasa Indonesia, relatif terbatas. Perbedaan struktur linguistik, idiom, dan masalah tokenisasi serta kualitas terjemahan dapat memengaruhi performa model setelah adaptasi lintas bahasa. Oleh karena itu, diperlukan adaptasi dan evaluasi sistematis pada dataset berbahasa Indonesia serta investigasi pipeline yang menggabungkan modul terjemahan/normalisasi, dekomposisi aturan, mekanisme pencarian bukti, dan resolusi logika.

### **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, rumusan masalah penelitian ini adalah:

1. Sejauh mana LLM mampu melakukan penalaran logis pada dataset berbahasa Indonesia?
2. Bagaimana perbandingan performa antara model open-source berparameter rendah dalam inferensi dataset?
3. Seberapa efektif pipeline *translate*  $\rightarrow$  *decompose*  $\rightarrow$  *search*  $\rightarrow$  *resolve* dalam meningkatkan akurasi inferensi pada data Bahasa Indonesia dibanding dengan penalaran

secara langsung secara naive?

4. Apa saja sumber utama kegagalan (kesalahan translasi, tokenisasi, ambiguitas budaya/linguistik, format data) ketika memindahkan benchmark penalaran dari Bahasa Inggris ke Bahasa Indonesia?

### 1.3 Tujuan Penelitian

**Tujuan umum:** Mengevaluasi dan memperbaiki kemampuan penalaran logis LLM pada dataset berbahasa Indonesia menggunakan pipeline.

**Tujuan khusus:**

1. Mengukur performa beberapa model pada tugas penalaran menggunakan metrik akurasi dan analisis kesalahan.
2. Mengidentifikasi dan mengkategorikan sumber kesalahan serta memberikan rekomendasi praktis untuk pengolahan data dan desain eksperimen berbahasa Indonesia.
3. Menyediakan dataset terjemahan, skrip eksperimen, dan laporan replikasi yang dapat digunakan peneliti lain.

### 1.4 Batasan Penelitian

Agar fokus dan ruang lingkup terukur, penelitian ini dibatasi sebagai berikut:

- **Dataset:** Fokus pada dataset *proof-style* yang telah diterjemahkan ke Bahasa Indonesia, yaitu ProntoQA saja
- **Model:** Eksperimen menggunakan model open-source yang dapat dijalankan lokal maupun server, khususnya dengan kuantisasi.
- **Evaluasi:** Metrik utama adalah akurasi jawaban akhir terhadap ground truth.
- **Sumber daya:** Eksperimen disesuaikan dengan kapasitas komputasi, sampling dev/test split dimulai pada 10% hingga 100% tergantung ketersediaan.

Referensi dari pipeline dan metoode, termasuk skrip seperti `translate_decompose.py`, `negate.py`, dan `search_resolve.py`, serta utilitas evaluasi tersedia pada repositori eksperimen yang menjadi inspirasi implementasi ini, yaitu pada repositori Aristotle LaTeX.

## 1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan memberi manfaat:

- Bagi akademik: memperkaya literatur penalaran LLM pada bahasa Indonesia.
- Bagi praktisi: panduan pemilihan model dan desain pipeline untuk tugas inferensi logis berbahasa lokal.
- Bagi komunitas open-source: dataset dan skrip replikasi yang dapat mempercepat penelitian lanjutan.

## 1.6 Posisi Penelitian

### @todo

Sebutkan posisi penelitian Anda. Ada baiknya jika Anda menggunakan gambar atau diagram. Template ini telah menyediakan contoh cara memasukkan gambar.



**Gambar 1.1:** Penjelasan singkat terkait gambar.

### @todo

Jelaskan Gambar 1.1 di sini. Setiap gambar yang dimasukkan ke tugas akhir **WAJIB** untuk dijelaskan oleh minimal satu paragraf.

## 1.7 Metodologi Singkat

Pendekatan penelitian dirancang sebagai pipeline yang memadukan teknik pemrosesan teks dan mekanisme reasoning simbolik:

1. **Persiapan Data:** Dataset yang ada di translasikan ke Bahasa Indonesia menggunakan model open-source
2. **Dekomposisi Logis:** Mengurai aturan kompleks menjadi bentuk yang lebih sederhana ke dalam First Order Logic (FOL) dan menormalisasi ke Prenex Normal Form (PNF) atau Conjunctive Normal Form (CNF).
3. **Inisialisasi Dua Jalur Pencarian:** Menegasikan konjektur atau pertanyaan untuk membentuk klausa komplemen yang akan dicari.
4. **Search & Resolve:** Melakukan pencarian klausa komplemen dan resolusi logika untuk menyimpulkan kebenaran atau identifikasi kontradiksi.
5. **Evaluasi:** Mengevaluasi hasil akhir dengan menggabungkan jawaban dari dua jalur tersebut dan membandingkannya dengan ground truth dan di agregasi menggunakan metrik akurasi.

## 1.8 Sistematika Penulisan

Sistematika penulisan laporan adalah sebagai berikut:

- Bab 1 PENDAHULUAN  
Bab ini mencakup latar belakang, cakupan penelitian, dan pendefinisian masalah.
- Bab 2 KERANGKA BERPIKIR  
Bab ini mencakup pemaparan terminologi dan teori yang terkait dengan penelitian berdasarkan hasil tinjauan pustaka yang telah digunakan, sekaligus memperlihatkan kaitan teori dengan penelitian.
- Bab 3 EKSPERIMEN DAN HASIL PENELITIAN  
Apa itu Bab 3?
- Bab 4 ANALISIS DAN PEMBAHASAN  
Apa itu Bab 4?
- Bab 5 KASUS-KASUS KHUSUS  
Apa itu Bab 5?
- Bab 6 PENUTUP  
Bab ini mencakup kesimpulan akhir penelitian dan saran untuk pengembangan berikutnya.

## 1.9 Panduan Singkat: Sitasi dan Cross-Reference

Untuk sitasi gunakan BibTeX seperti pada template; contoh pemanggilan:

- Sitasi di dalam kalimat: Menurut `\cite{author:year}` ...
- Sitasi di akhir kalimat: ... sesuai kajian sebelumnya `\citep{author:year}`.

Contoh cross-reference:

- Referensi ke Bab: `\label{bab:1}` kemudian `\ref{bab:1}` (contoh: Bab 1).
- Referensi ke Sub-bab: `\label{sec:metodologi}` lalu `Section~\ref{sec:metodologi}`.

### @todo

Lengkapi bagian dataset dan konfigurasi backend pada Bab 3.



## BAB 2

### KERANGKA BERPIKIR

#### @todo

Bab ini, biasanya namanya adalah "Studi Literatur" atau "Tinjauan Pustaka". Akan tetapi, beberapa fakultas atau dosen pembimbing meminta Bab 2 untuk dinamakan lain, seperti "Kerangka Berpikir".

Untuk memulai penelitian, dibutuhkan kerangka berpikir yang sesuai untuk permasalahan yang ingin dipecahkan. Untuk membentuk kerangka berpikir yang sesuai, perlu dikaitkan dengan hasil studi literatur yang telah dilakukan. Oleh karena itu, pada bab ini, akan dijelaskan hasil studi literatur yang telah dilakukan yang telah dikaitkan dengan kerangka kerja untuk penelitian ini.

## 2.1 Apa itu L<sup>A</sup>T<sub>E</sub>X?

### 2.1.1 L<sup>A</sup>T<sub>E</sub>X Secara Singkat

Berdasarkan Clark (2010):

L<sup>A</sup>T<sub>E</sub>X is a family of programs designed to produce publication-quality typeset documents. It is particularly strong when working with mathematical symbols.

The history of L<sup>A</sup>T<sub>E</sub>X begins with a program called TEX. In 1978, a computer scientist by the name of Donald Knuth grew frustrated with the mistakes that his publishers made in typesetting his work. He decided to create a typesetting program that everyone could easily use to typeset documents, particularly those that include formulae, and made it freely available. The result is TEX.

Knuth's product is an immensely powerful program, but one that does focus very much on small details. A mathematician and computer scientist by the name of Leslie Lamport wrote a variant of TEX called L<sup>A</sup>T<sub>E</sub>X that focuses on document structure rather than such details.

Dokumen L<sup>A</sup>T<sub>E</sub>X sangat mudah, seperti halnya membuat dokumen teks biasa. Ada beberapa perintah yang diawali dengan tanda '\'. Seperti perintah `\\` yang digunakan untuk memberi baris baru. Perintah tersebut juga sama dengan perintah `\newline`. Pada bagian ini akan

sedikit dijelaskan cara manipulasi teks dan perintah-perintah  $\text{\LaTeX}$  yang mungkin akan sering digunakan. Jika ingin belajar hal-hal dasar mengenai  $\text{\LaTeX}$ , silakan kunjungi:

- <http://frodo.elon.edu/tutorial/tutorial/>, atau
- <http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/>

### 2.1.2 $\text{\LaTeX}$ Kompiler dan IDE

Untuk menggunakan  $\text{\LaTeX}$  (pada konteks hanya sebagai pengguna), tidak perlu banyak tahu mengenai hal-hal didalamnya. Dengan menggunakan *Integrated Development Environment* (IDE), penggunaan  $\text{\LaTeX}$  akan serupa dengan pembuatan dokumen secara visual, layaknya OpenOffice Writer atau Microsoft Word. Orang-orang yang menggunakan  $\text{\LaTeX}$  relatif lebih teliti dan terstruktur mengenai cara penulisan yang dia gunakan, karena  $\text{\LaTeX}$  memaksa untuk seperti itu.

Untuk mencoba  $\text{\LaTeX}$ , diperlukan kompiler dan IDE. Bagi pengguna Microsoft Windows dan Mac OS, instalasi kompiler  $\text{\LaTeX}$  dapat menggunakan MikTeX (<https://miktex.org/download>). Bagi pengguna Linux, instalasi kompiler  $\text{\LaTeX}$  dapat menggunakan Texlive (<http://www.tug.org/texlive/>). Distro-distro *mainstream* di Linux seperti Ubuntu biasanya telah menyediakan *package texlive* melalui *package manager*. Apabila ingin melakukan instalasi Texlive melalui *package manager*, lakukan instalasi *package texlive-full* atau setidaknya *texlive-science* agar prasyarat *template* ini tersedia secara lengkap.

Beberapa text editor atau IDE yang dapat digunakan adalah sebagai berikut:

- TeXstudio (direkomendasikan),
- TeXWorks (biasanya bawaan dari MikTeX),
- Texmaker, atau
- Microsoft Visual Studio Code, dengan *plugin*  $\text{\LaTeX}$  Workshop. Untuk menggunakan *plugin* tersebut, diperlukan instalasi MikTeX dan Perl. Alternatif lain untuk persyaratan tersebut adalah menggunakan *plugin* Remote - WSL jika memiliki distro Windows Subsystem for Linux (WSL) 2 yang sudah terpasang *texlive*.

## 2.2 *Formatting Teks Dasar*

Hal pertama yang mungkin ditanyakan adalah bagaimana membuat huruf tercetak tebal, miring, atau memiliki garis bawah. Pada Texmaker, Anda bisa melakukan hal ini seperti halnya saat mengubah dokumen dengan OO Writer. Namun jika tetap masih tertarik dengan cara lain, ini dia:

- **Bold**

Gunakan perintah `\textbf{}` atau `\bo{}`.

Contoh: **Contoh hasil tulisan** atau **Contoh hasil tulisan**.

- *Italic*

Gunakan perintah `\textit{}` atau `\f{}`.

Contoh: *Contoh hasil tulisan* atau *Contoh hasil tulisan*.

- Underline

Gunakan perintah `\underline{}`.

Contoh: Contoh hasil tulisan.

- Overline

Gunakan perintah `$\overline{}`.

Contoh: Contoh hasil tulisan.

- *superscript*

Gunakan perintah `$\{}`.

Contoh: *Contoh hasil tulisan*.

- *subscript*

Gunakan perintah `$\_{}`.

Contoh: *Contoh hasil tulisan*.

Ada beberapa hal lain yang bisa digunakan.

- Kombinasi **Bold** dan *Italic*:

Gunakan perintah `\bi{}`.

Contoh: ***Contoh hasil tulisan***.

- Menebalkan teks formula matematis:

Gunakan perintah `\m{}`.

Contoh:  $\alpha \beta$

- Menebalkan teks formula matematis, sekaligus meletakkannya di tengah:

Gunakan perintah `\mc{}`.

Contoh:

$\alpha \beta$

- Menggunakan *monospaced font* untuk kode: Gunakan perintah `\texttt{}` atau `\code{}`.

Contoh: Contoh hasil tulisan atau Contoh hasil tulisan.

Perintah `\f`, `\bo`, `\bi`, `\m`, `\mc`, dan `\code` hanya dapat digunakan jika package `_internals/uithesis` digunakan.

## 2.3 Memasukan Gambar

Setiap gambar dapat diberikan caption dan diberikan label. Label dapat digunakan untuk menunjuk gambar tertentu. Jika posisi gambar berubah, maka nomor gambar juga akan diubah secara otomatis. Begitu juga dengan seluruh referensi yang menunjuk pada gambar tersebut. Contoh sederhana adalah Gambar 2.1, yang bisa dibuat dengan menggunakan Kode 2.1. Harap diingat pada aturan Tugas Akhir UI, caption harus selalu diletakkan di bawah gambar.

```
1 \begin{figure}
2   \centering
3   \includegraphics[width=0.25\textwidth]
4     {assets/pics/makara_kuning.png}
5   \caption{Makara Universitas Indonesia}
6   \label{fig:testGambar}
7 \end{figure}
```

**Kode 2.1:** Contoh penggunaan gambar

Berikut adalah penjelasan dari Kode 2.1:

- Baris ke-2: `\centering` digunakan untuk membuat gambar berada di tengah.
- Baris ke-3 dan 4: `\includegraphics` digunakan untuk memasukkan gambar. `width=0.25\textwidth` digunakan untuk mengatur lebar gambar sebesar 25% dari lebar teks (dari ujung margin kiri ke ujung margin kanan).
- Baris ke-5: `\caption` digunakan untuk memberikan *caption* pada gambar. *Caption* tersebut diletakkan setelah `includegraphics` agar *caption* berada di bawah gambar.
- Baris ke-6: `\label` digunakan untuk memberikan label pada gambar. Label ini bisa di-

gunakan di suatu paragraf untuk merujuk pada gambar tersebut, dengan cara menuliskan `\ref{label}` pada paragraf.



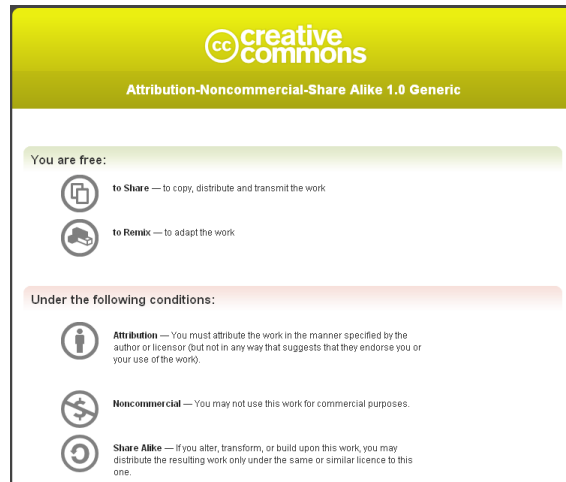
**Gambar 2.1:** Makara Universitas Indonesia

Anda juga bisa memasukkan sitasi atau URL sumber gambar, jika gambar tersebut bukan Anda sendiri yang membuatnya. Contoh sederhana adalah Gambar 2.2, yang bisa dibuat dengan menggunakan Kode 2.2.

```
1 \begin{figure}
2   \centering
3   \includegraphics[width=0.50\textwidth]
4     {assets/pics/creative_commons.png}
5   \captionsource{\license.}{\url{https://creativecommons.org/licenses/by-nc-sa/1.0/}}
6   \label{fig:testGambarBersumber}
7 \end{figure}
```

**Kode 2.2:** Contoh penggunaan gambar bersumber

Pada baris ke-5, `\captionsource` digunakan untuk memberikan caption dan sumber gambar. Dalam kasus ini, sumber gambar merupakan sebuah URL, sehingga ditandai dengan perintah `\url{}`. Jika sumber gambar merupakan sebuah buku, jurnal, atau dokumen, maka bisa dilakukan sitasi menggunakan perintah `\cite{}`. Contoh: `\captionsource{Sesuatu}{\cite{latex:intro}}`.

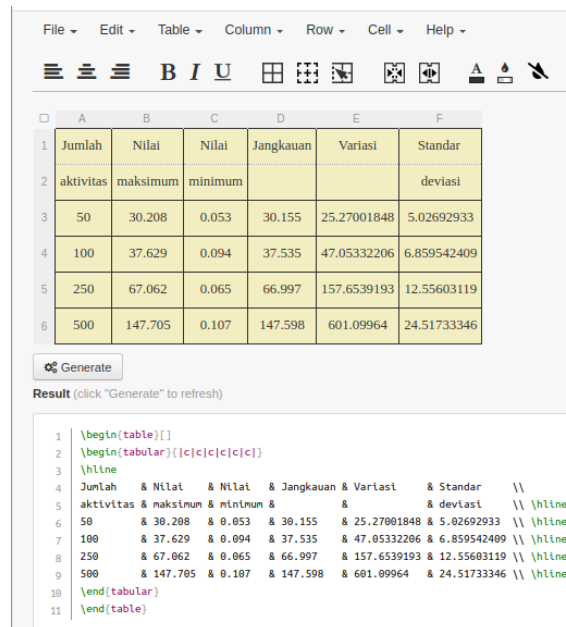


**Gambar 2.2:** *Creative Common License 1.0 Generic.*

**Sumber:** <https://creativecommons.org/licenses/by-nc-sa/1.0/>

## 2.4 Membuat Tabel

Tabel pada  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  dapat dibuat secara visual dengan bantuan *website* seperti <https://www.tablesgenerator.com/>. Dengan menggunakan *website* ini, maka pembuatan tabel akan menjadi lebih mudah. *User interface* dari *website* dapat dilihat pada Gambar 2.3.



**Gambar 2.3:** *User interface* dari *website* <https://www.tablesgenerator.com/>

Di sisi lain, tabel juga dapat diberi label dan caption seperti pada gambar. Caption pada tabel terletak pada bagian atas tabel. Contoh kode yang menyusun suatu tabel sederhana

dapat dilihat pada Kode 2.3.

```

1 \begin{table}
2   \centering
3   \caption{Contoh Tabel}
4   \label{tab:basic}
5   \begin{tabular}{| l | c r |} %
6     \hline % garis lurus horizontal
7     & kol 1 & kol 2 \\ % baris 1
8     \hline %
9     baris 1 & 1 & 2 \\ % baris 2
10    baris 2 & 3 & 4 \\ % baris 3
11    baris 3 & 5 & 6 \\ % baris 4
12    baris 4 & 7 & 8 \\ % baris 5
13    baris 5 & 9 & 10 \\ % baris 6
14    \hline
15    \bo{jumlah} & \bo{25} & \bo{30} \\ % baris 7
16    \hline
17  \end{tabular}
18 \end{table}

```

**Kode 2.3:** Contoh penggunaan tabel

Berikut adalah penjelasan dari Kode 2.3:

- Baris ke-2: `\centering` digunakan untuk membuat tabel berada di tengah.
- Baris ke-3: `\caption` digunakan untuk memberikan *caption* pada tabel. *Caption* tersebut diletakkan setelah `\begin{tabular}` agar *caption* berada di atas tabel.
- Pada baris ke-3, terdapat argumen `| l | c r |` yang artinya adalah sebagai berikut:
  - `|` digunakan untuk membuat garis vertikal pada tabel.
  - `l` digunakan untuk membuat suatu kolom menjadi rata kiri.
  - `c` digunakan untuk membuat suatu kolom menjadi rata tengah.
  - `r` digunakan untuk membuat suatu kolom menjadi rata kanan.
- Baris ke-4: `\label` digunakan untuk memberikan label pada tabel. Label ini bisa digunakan di suatu paragraf untuk merujuk pada tabel tersebut, dengan cara menuliskan `\ref{label}` pada paragraf.
- Baris ke-5: `\begin{tabular}` digunakan untuk memulai pembuatan tabel.
- `\hline` digunakan untuk membuat garis horizontal pada tabel.
- `&` digunakan untuk memisahkan antar kolom.
- `\\` digunakan untuk memisahkan antar baris.
- Baris ke-17 dan 18: Kode untuk mengakhiri pembuatan tabel.

Hasil dari Kode 2.3 akan menjadi Tabel 2.1.

**Tabel 2.1:** Contoh Tabel

	kol 1	kol 2
baris 1	1	2
baris 2	3	4
baris 3	5	6
baris 4	7	8
baris 5	9	10
<b>jumlah</b>	<b>25</b>	<b>30</b>

### 2.4.1 Tabel Panjang (Lintas Baris)

Adapun untuk membuat tabel panjang yang bisa melebihi dari satu halaman, gunakan perintah `\begin{longtable}` sebagai pengganti `\begin{table}`. Di dalam `longtable` tidak perlu lagi ada `\begin{tabular}`. Kemudian, tambahkan tanda `\\` setelah baris `\label{...}`, agar tidak menimbulkan error saat menampilkan *caption* di bagian atas tabel. Kemudian, untuk membatasi header yang ingin diulang pada halaman-halaman berikutnya, gunakan perintah `\endhead`. Contoh kode pembuatan tabel panjang dapat dilihat pada Kode 2.4.

```

1 \begin{longtable}{| l | c r |}
2   \caption{Contoh Tabel Panjang}
3   \label{tab:long} \\
4   \hline
5   & kol 1 & kol 2 \\
6   \hline
7   \endfirsthead % batas akhir header yang akan muncul di halaman pertama
8   \caption[]{}{Contoh Tabel Panjang (sambungan)} \\
9   \hline
10  & kol 1 & kol 2 \\
11  \hline
12  \endhead % batas akhir header yang akan muncul di halaman berikutnya
13  \hline
14  \endfoot % batas akhir footer yang akan muncul di halaman berikutnya
15  \hline
16  \endlastfoot % batas akhir footer yang akan muncul di halaman terakhir
17  baris 1 & 1 & 2 \\
18  baris 2 & 3 & 4 \\
19  baris 3 & 5 & 6 \\
20  baris 4 & 7 & 8

```

```

21   baris 5 & 9 & 10 \\
22   baris 6 & 11 & 12 \\
23   baris 7 & 13 & 14 \\
24   baris 8 & 15 & 16 \\
25   baris 9 & 17 & 18 \\
26   baris 10 & 19 & 20 \\
27   baris 11 & 21 & 22 \\
28   baris 12 & 23 & 24 \\
29   baris 13 & 25 & 26 \\
30   baris 14 & 27 & 28 \\
31   baris 15 & 29 & 30 \\
32   baris 16 & 31 & 32 \\
33 \end{longtable}

```

**Kode 2.4:** Contoh penggunaan tabel panjang

Terdapat lima bagian pada sebuah tabel panjang:

1. Awalan (*header*) di halaman pertama, umumnya disebut sebagai *firsthead*.

Pada Kode 2.4, bagian ini didefinisikan di baris 2 sampai 7. Bagian ini diakhiri definisinya dengan perintah `\endfirsthead`. Pada bagian ini, *caption* yang digunakan merupakan *caption* asli. *caption* dan label hanya perlu didefinisikan di bagian awal tabel, di halaman pertama saja. Anda juga tetap bisa menggunakan *captionsource* apabila dibutuhkan.

2. Awalan (*header*) di halaman berikutnya, umumnya disebut sebagai *head*.

Pada Kode 2.4, bagian ini didefinisikan di baris 8 sampai 12. Bagian ini diakhiri definisinya dengan perintah `\endhead`. Terkait penggunaan *caption* sambungan:

- Jika Anda menggunakan `\caption{}`, maka untuk menuliskan *caption* sambungan, gunakan perintah `\caption[]{}{}`. Pada kasus Kode 2.4, *caption* sambungan didefinisikan di baris 8 menggunakan perintah `\caption[]{}{}`.
- Jika Anda menggunakan `\captionsource{}`, maka untuk menuliskan *caption* sambungan, gunakan perintah `\captionsourcecont{}{}`.

3. Akhiran (*footer*) yang muncul di halaman pertama hingga sebelum terakhir, umumnya disebut sebagai *foot*.

Bagian ini diakhiri definisinya dengan perintah `\endfoot`. Pada Kode 2.4, bagian ini didefinisikan di baris 13 sampai 14. Dalam kasus ini, akhiran tabel hanya berupa garis horizontal.

4. Akhiran (*footer*) di halaman terakhir, umumnya disebut sebagai *lastfoot*.

Bagian ini diakhiri definisinya dengan perintah `\endlastfoot`. Pada Kode 2.4, bagian

ini didefinisikan di baris 15 sampai 16. Dalam kasus ini, akhiran tabel hanya berupa garis horizontal.

5. Isi dari tabel. Pada Kode 2.4, bagian ini didefinisikan di baris 17 sampai 32. Isi dari tabel akan diletakkan di antara awalan tabel (*header*) dan akhiran tabel (*footer*).

Hasil dari Kode 2.4 akan menjadi Tabel 2.2.

**Tabel 2.2:** Contoh Tabel Panjang

	kol 1	kol 2
baris 1	1	2
baris 2	3	4
baris 3	5	6
baris 4	7	8
baris 5	9	10
baris 6	11	12
baris 7	13	14
baris 8	15	16
baris 9	17	18
baris 10	19	20
baris 11	21	22
baris 12	23	24
baris 13	25	26
baris 14	27	28
baris 15	29	30
baris 16	31	32

### 2.4.2 Menggabungkan (*Merge*) Baris atau Kolom

Ada jenis tabel lain yang dapat dibuat dengan L<sup>A</sup>T<sub>E</sub>X berikut beberapa diantaranya. Contoh-contoh ini bersumber dari <https://en.wikibooks.org/wiki/LaTeX/Tables>.

#### Contoh 1: Menggabungkan Kolom

Contoh penggunaan tabel dengan sel yang melebar ke lebih dari satu kolom dapat dilihat pada Kode 2.5. Pada contoh ini, sel pada baris 1, kolom 3 dan 4 digabungkan menjadi

satu dengan menggunakan perintah `\multicolumn{3}{c}{Week 1}`.

```

1 \begin{table}
2   \centering
3   \captionsource{Tabel dengan sel yang melebar ke lebih dari satu
4     kolom}{\url{https://en.wikibooks.org/wiki/LaTeX/Tables}, dengan modifikasi.}
5   \label{tab:rowSpanning}
6   \begin{tabular}{|l|l|*{6}{c|}}
7     % Baris 1
8     \hline % buat garis horizontal dari kolom pertama ke kolom terakhir
9     No & Name & \multicolumn{3}{c}{Week 1} & \multicolumn{3}{c}{Week 2} \\
10    % Baris 2
11    \cline{3-8} % buat garis horizontal dari kolom 3 sampai 8
12    & A & B & C & A & B & C \\
13    % Baris 3
14    \hline
15    1 & Lala & 1 & 2 & 3 & 4 & 5 & 6 \\
16    % Baris 4
17    2 & Lili & 1 & 2 & 3 & 4 & 5 & 6 \\
18    % Baris 5
19    3 & Lulu & 1 & 2 & 3 & 4 & 5 & 6 \\
20    \hline
21  \end{tabular}
22 \end{table}

```

**Kode 2.5:** Contoh penggunaan tabel dengan sel yang melebar ke lebih dari satu kolom

Hasil dari Kode 2.5 akan menjadi Tabel 2.3.

**Tabel 2.3:** Tabel dengan sel yang melebar ke lebih dari satu kolom

**Sumber:** <https://en.wikibooks.org/wiki/LaTeX/Tables>, dengan modifikasi.

No	Name	Week 1			Week 2		
		A	B	C	A	B	C
1	Lala	1	2	3	4	5	6
2	Lili	1	2	3	4	5	6
3	Lulu	1	2	3	4	5	6

## Contoh 2: Menggabungkan Baris

Contoh penggunaan tabel dengan sel yang melebar ke lebih dari satu baris dapat dilihat pada Kode 2.6. Pada contoh ini, sel pada kolom 1, baris 2 dan 3 digabungkan menjadi satu dengan menggunakan perintah `\multirow{2}{*}{Kedua}`.

```

1 \begin{table}
2   \centering

```

```

3 \captionsource{Tabel dengan sel yang melebar ke lebih dari satu
  baris}{\url{https://en.wikibooks.org/wiki/LaTeX/Tables}, dengan modifikasi.}
4 \label{tab:columnSpanning}
5 \begin{tabular}{|l|c|l|} % border vertikal ditandai dengan |
6     % Baris 1
7     \hline % buat garis horizontal dari kolom pertama ke kolom terakhir
8     Percobaan & Iterasi & Waktu \\
9     % Baris 2
10    \hline
11    Pertama & 1 & 0.1 sec \\ \hline
12    % Baris 3 (kolom 1 melebar 2 baris)
13    \multirow{2}{*}{Kedua} & 1 & 0.1 sec \\
14    % Baris 4 (kolom 1 dikosongkan karena sudah ditimpa)
15    & 3 & 0.15 sec \\
16    % Baris 5 (kolom 1 melebar 3 baris)
17    \hline
18    \multirow{3}{*}{Ketiga} & 1 & 0.09 sec \\
19    % Baris 6 (kolom 1 dikosongkan karena sudah ditimpa)
20    & 2 & 0.16 sec \\
21    % Baris 7 (kolom 1 dikosongkan karena sudah ditimpa)
22    & 3 & 0.21 sec \\
23    \hline
24 \end{tabular}
25 \end{table}

```

**Kode 2.6:** Contoh penggunaan tabel dengan sel yang melebar ke lebih dari satu baris

Hasil dari Kode 2.6 akan menjadi Tabel 2.4.

**Tabel 2.4:** Tabel dengan sel yang melebar ke lebih dari satu baris

**Sumber:** <https://en.wikibooks.org/wiki/LaTeX/Tables>, dengan modifikasi.

Percobaan	Iterasi	Waktu
Pertama	1	0.1 sec
Kedua	1	0.1 sec
	3	0.15 sec
Ketiga	1	0.09 sec
	2	0.16 sec
	3	0.21 sec

### Contoh 3: Menggabungkan Baris dan Kolom

Contoh penggunaan tabel dengan sel yang melebar ke lebih dari satu kolom dan baris dapat dilihat pada Kode 2.7.

```

1 \begin{table}
2   \centering
3   \captionsource{Tabel dengan sel yang melebar ke lebih dari satu kolom dan
4     baris}{\url{https://en.wikibooks.org/wiki/LaTeX/Tables}, dengan modifikasi.}
5   \label{tab:mixSpanning}
6   \begin{tabular}{|cc|c|c|c|c|} % border vertikal ditandai dengan |, semua kolom rata
7     tengah.
8     % 2-row 2-col span (rata tengah), dan 4-col span (rata tengah)
9     \hline % buat garis horizontal dari kolom pertama ke kolom terakhir
10    \multicolumn{2}{|c|}{\multirow{2}{*}{Element}} & \multicolumn{4}{c|}{Title} \\
11    % kelanjutan border untuk 2-row 2-col span, dan 4 kolom biasa
12    \cline{3-6} % buat garis horizontal dari kolom 3 sampai 6
13    \multicolumn{2}{|c|}{& A & B & C & D} \\
14    % 2-row span (rata kiri), dan 5 kolom biasa
15    \hline
16    \multicolumn{1}{|l|}{\multirow{2}{*}{Type}} & X & 1 & 2 & 3 & 4 \\
17    % kelanjutan border untuk 2-row span, dan 5 kolom biasa
18    \cline{2-6}
19    \multicolumn{1}{|l|}{& Y & 0.5 & 1.0 & 1.5 & 2.0} \\
20    % kelanjutan bordering untuk 2-row span, dan 5 kolom biasa
21    \hline
22    \multicolumn{1}{|l|}{\multirow{2}{*}{Resource}} & I & 10 & 20 & 30 & 40 \\
23    % kelanjutan border untuk 2-row span, dan 5 kolom biasa
24    \cline{2-6}
25    \multicolumn{1}{|l|}{& J & 5 & 10 & 15 & 20} \\
26    \hline
27 \end{tabular}
28 \end{table}

```

**Kode 2.7:** Contoh penggunaan tabel dengan sel yang melebar ke lebih dari satu kolom dan baris

Hasil dari Kode 2.7 akan menjadi Tabel 2.5.

**Tabel 2.5:** Tabel dengan sel yang melebar ke lebih dari satu kolom dan baris  
**Sumber:** <https://en.wikibooks.org/wiki/LaTeX/Tables>, dengan modifikasi.

Element		Title			
		A	B	C	D
Type	X	1	2	3	4
	Y	0.5	1.0	1.5	2.0
Resource	I	10	20	30	40
	J	5	10	15	20

## 2.5 Membuat Persamaan Matematis

Di  $\text{\LaTeX}$ , kita dapat membuat persamaan matematis baik yang terdiri dari satu persamaan maupun lebih dari satu persamaan. Anda bisa mencoba mengikuti dan memahami contoh kode yang ada di *template* ini untuk kebutuhan tugas akhir Anda. Menggunakan  $\text{\LaTeX}$  juga perlu latihan dan lihai memahami dokumentasi.

### 2.5.1 Satu Persamaan

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1} \quad (2.1)$$

Persamaan 2.1 di atas adalah persamaan garis. Persamaan tersebut dapat disusun dengan menggunakan perintah `\align`, seperti yang ditunjukkan pada Kode 2.8. Penggunaan perintah `\noindent` sebelum memanggil *environment* `align` diperlukan agar persamaan yang dicetak tidak tergeser mengikuti indentasi yang umumnya muncul di awal paragraf. Terdapat beberapa perintah yang digunakan untuk menyusun Persamaan 2.1, yaitu:

- Perintah `\cfrac{\{\}\{\}}` untuk menulis pecahan dalam bentuk vertikal. Argumen pertama adalah pembilang (di atas), sedangkan argumen kedua adalah penyebut (di bawah).
- Perintah `\_{\}` untuk menulis *subscript*.

Di luar *environment* `align`, kita juga bisa menambahkan persamaan tersebut ke daftar persamaan dengan menggunakan perintah `\addequtotoc{label}{caption}`. Perintah `\label` pada persamaan matematis hanya akan memberikan label, namun tidak menambahkan ke daftar persamaan. Perintah `\addequtotoc` harus ditambahkan di luar *environment* `align`, karena perintah tersebut harus dijalankan di mode *typesetting* teks biasa agar tidak terjadi *error*.

```
1 \noindent \begin{align}\label{equ:garis}
2   \cfrac{y - y_{1}}{y_{2} - y_{1}} =
3   \cfrac{x - x_{1}}{x_{2} - x_{1}}
4 \end{align}
5
6 \addequtotoc{equ:garis}{Persamaan garis}
```

**Kode 2.8:** Kode pembuatan Persamaan 2.1

Persamaan bola berikut, yang ditunjukkan oleh 2.2, merupakan contoh lain menyusun sebuah persamaan di  $\text{\LaTeX}$ .

$$\underbrace{\overline{ab}}_{\text{pada bola } |\overline{ab}| = r} = \sqrt[2]{x_b - x_a^2 y_b - y_a^2 ||z_b - z_a^2} \quad (2.2)$$

Terdapat beberapa perintah yang digunakan untuk menyusun Persamaan 2.2, yaitu:

- Perintah `\sqrt{}` untuk menulis akar.
- Perintah `\underbrace{}` untuk menulis *brace* di bawah suatu ekspresi. Biasanya digunakan untuk memberikan anotasi terhadap suatu ekspresi.
- Perintah `\overline{}` untuk menulis garis di atas suatu ekspresi.
- Perintah `\text{}` untuk menulis teks biasa di dalam persamaan.
- Di dalam teks biasa, kita bisa menuliskan kembali sebuah persamaan matematis dengan menambahkan tanda `$` di awal dan di akhir persamaan tersebut. Di  $\text{\LaTeX}$ , terdapat dua mode *typesetting*, yaitu mode teks dan mode matematika. Fungsi `$` adalah untuk mengubah mode teks menjadi mode matematika.
- Perintah `\vert` untuk menulis tanda garis vertikal.

```

1 \noindent \begin{align}\label{equ:bola}
2   \underbrace{\{\overline{ab}\}}_{\text{pada bola } \{\overline{ab}\} = r\$}
3   = \sqrt[2]{(x_{\{b\}} - x_{\{a\}})^2 + (y_{\{b\}} - y_{\{a\}})^2 +
4     \vert\vert z_{\{b\}} - z_{\{a\}} \vert\vert^2}
5 \end{align}
6
7 \addequtoc{equ:bola}{Persamaan bola}

```

**Kode 2.9:** Kode pembuatan Persamaan 2.2

Suatu persamaan yang dibuat menggunakan *environment* `align` akan secara otomatis memiliki indeks (nomor) dari persamaan. Perintah `align` ini juga dapat digunakan untuk menulis lebih dari satu persamaan.

## 2.5.2 Lebih dari Satu Persamaan

Persamaan 2.3 adalah contoh persamaan matriks yang dibuat menggunakan  $\text{\LaTeX}$ .

$$|\overline{a} * \overline{b}| = |\overline{a}| |\overline{b}| \sin \theta \quad (2.3)$$

$$\begin{aligned} \overline{a} * \overline{b} &= \begin{vmatrix} \hat{i} & x_1 & x_2 \\ \hat{j} & y_1 & y_2 \\ \hat{k} & z_1 & z_2 \end{vmatrix} \\ &= \hat{i} \begin{vmatrix} y_1 & y_2 \\ z_1 & z_2 \end{vmatrix} - \hat{j} \begin{vmatrix} x_1 & x_2 \\ z_1 & z_2 \end{vmatrix} + \hat{k} \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} \end{aligned}$$

Pada Persamaan 2.3 dapat dilihat beberapa baris persamaan menjadi satu bagian dari persamaan tersebut. Kode untuk menyusun kumpulan persamaan di Persamaan 2.3 dapat dilihat pada Kode 2.10. Eksekusi perintah `\addeqtotoc{label}{caption}` juga cukup dilakukan sekali saja.

```

1 \noindent \begin{align}\label{equ:matricks}
2   |\overline{a} * \overline{b}| &= |\overline{a}| |\overline{b}| \sin\theta
3   \\[0.2cm]
4   \overline{a} * \overline{b} &=
5   \begin{array}{| c c c |}
6     \hat{i} & x_{1} & x_{2} \\
7     \hat{j} & y_{1} & y_{2} \\
8     \hat{k} & z_{1} & z_{2}
9   \end{array} \nonumber \\[0.2cm]
10  &= \hat{i} \begin{vmatrix} y_{1} & y_{2} \\ z_{1} & z_{2} \end{vmatrix}
11  - \hat{j} \begin{vmatrix} x_{1} & x_{2} \\ z_{1} & z_{2} \end{vmatrix}
12  + \hat{k} \begin{vmatrix} x_{1} & x_{2} \\ y_{1} & y_{2} \end{vmatrix}
13  \\[0.2cm]
14  \nonumber
15 \end{align}
16 \addeqtotoc{equ:matricks}{Persamaan matricks}

```

**Kode 2.10:** Kode pembuatan Persamaan 2.3

Terdapat beberapa perintah yang digunakan untuk menyusun Persamaan 2.3, yaitu:

- Perintah `\begin{array}` dan `\end{array}` untuk membuat matriks. Cara kerja *environment* `array` sama dengan *environment* `tabular` yang digunakan untuk membuat tabel.
- Perintah `\sin` untuk menulis fungsi sinus.
- Perintah `\theta` untuk menulis simbol theta ( $\theta$ ).
- Perintah `\hat{}` untuk menulis tanda aksen *hat* (^) di atas suatu huruf.

Sedangkan dibawah ini dapat dilihat bahwa dengan cara yang sama, Persamaan 2.4, 2.5, dan 2.6 memiliki nomor persamaannya masing-masing.

$$\int_a^b f(x) dx = \int_b^c f(x) dx = \int_a^c f(x) dx \quad (2.4)$$

$$\lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = 0 \quad \text{jika pangkat } f(x) < \text{pangkat } g(x) \quad (2.5)$$

$$a^{m^{a^n \log b}} = b^{\frac{m}{n}} \quad (2.6)$$

Kode yang menyusun Persamaan 2.4, 2.5, dan 2.6 dapat dilihat pada Kode 2.11. Perbedaan penyusunan tiga argumen tersebut dibandingkan dengan Persamaan 2.3 adalah penggunaan *label* pada setiap persamaan. Pada Persamaan 2.3, *label* diletakkan pada bagian awal persamaan pertama, sehingga hanya satu *label* yang digunakan. Pada Persamaan 2.4, 2.5, dan 2.6, *label* diletakkan pada awal setiap persamaan, sehingga setiap persamaan memiliki *label* masing-masing. Selain itu, jika ingin memasukkan setiap persamaan secara terpisah ke Daftar Persamaan, eksekusi perintah `\addcontentsline{label}{caption}` juga harus dilakukan terpisah untuk setiap persamaan.

```

1 \noindent \begin{align}
2   \label{equ:integral}
3   \int_a^b f(x) \, dx + \int_b^c f(x) \, dx = \int_a^c f(x) \, dx \\
4   \label{equ:limit}
5   \lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = 0 \hspace{1cm}
6   \text{jika pangkat } f(x) < \text{pangkat } g(x) \\
7   \label{equ:eksponen}
8   a^{m^{a^n \log b}} = b^{\frac{m}{n}}
9 \end{align}
10
11 \addcontentsline{equ:integral}{Persamaan integral}
12 \addcontentsline{equ:limit}{Persamaan limit}

```

```
13 \addequtotoc{equ:eksponen}{Persamaan eksponen dan logaritma}
```

**Kode 2.11:** Kode pembuatan Persamaan 2.4, 2.5, dan 2.6

Terdapat beberapa perintah yang digunakan untuk menyusun Persamaan 2.4, 2.5, dan 2.6, yaitu:

- Perintah `\int` untuk menulis simbol integral. Untuk batas bawah dan batas atas integral, bisa dituliskan dengan menggunakan perintah *subscript* dan *superscript*.
- Perintah `\lim` untuk menulis simbol limit.
- Perintah `\to` untuk menulis panah ke kanan.
- Perintah `\infty` untuk menulis simbol tak hingga.
- Perintah `\log` untuk menulis fungsi logaritma. Pangkat basis logaritma dituliskan dengan menggunakan perintah *superscript* sebelum perintah `\log`.

## 2.6 Menambahkan Kode Program

Pada  $\text{\LaTeX}$ , kode program seringkali disebut *listing*. *Syntax highlighting* kini sudah bisa dilakukan secara otomatis oleh *library* yang ada di  $\text{\LaTeX}$ . Sudah tidak perlu lagi membuat skrip manual untuk menambahkan *syntax highlighting* sendiri. Kode 2.12 adalah contoh kode program (*listing*) Java yang dicetak oleh  $\text{\LaTeX}$ .

```
1 package com.sample.service.dummy.route;
2 import java.util.List;
3 import com.sample.interfaces.RoutingList;
4 import com.sample.route.RootRoutingListImpl;
5
6 public class VeryLongClassNameSampleRouteImpl extends RootRoutingListImpl implements
    RoutingList {
7     @Override
8     public List<String> getRouteList() {
9         return routeList;
10    }
11 }
```

**Kode 2.12:** Kode sampel Java yang cukup panjang

Sintaks untuk memasukkan kode program ke dalam dokumen  $\text{\LaTeX}$  adalah sebagai berikut:

```
1 \lstinputlisting[language=Java, caption=Kode sampel Java yang cukup panjang,
    label=code:java]{assets/codes/2-sample.java}
```

**Kode 2.13:** Meng

Terdapat tiga argumen yang digunakan pada perintah `\lstinputlisting`:

- `language` digunakan untuk menentukan bahasa pemrograman yang digunakan. Untuk menggunakan suatu dialek bahasa pemrograman yang berbeda dari *default*, misalkan versi Python3 dari Python, gunakan perintah `language={ [3]Python }`.
- `caption` digunakan untuk memberikan *caption* pada kode program. Argumen ini sifatnya opsional, jika ada, maka *caption* akan ditampilkan di bawah kode program. Jika argumen ini tidak ada, maka *caption* tidak akan ditampilkan dan kode tidak bisa masuk ke daftar kode program.
- `label` digunakan untuk memberikan label pada kode program untuk rujukan di dalam dokumen (*cross-reference*). Argumen ini tidak boleh didefinisikan jika argumen `caption` tidak didefinisikan.

Terdapat empat kelompok bahasa pemrograman (dan dialek) yang didukung oleh implementasi listings pada *template* ini, yaitu:

- **Bahasa pemrograman yang didukung secara *default* oleh listings** (menurut Heinz et al. (2024)):

ABAP, ACSL, Ada, Algol, Ant, Awk, bash, Basic, C++, C, Caml, Clean, Cobol, Comal, command.com (Windows Batch), csh, Delphi, Eiffel, Elan, erlang, Euphoria, Fortran, GCL, Go (golang), Gnuplot, Haskell, HTML, IDL, inform, Java, JVMIS, ksh, Lisp, Logo, Lua, make, Mathematica, Matlab, Mercury, MetaPost, Miranda, Mizar, ML, Modelica, Modula-2, MuPAD, NASTRAN, Oberon-2, OCL, Octave, Oz, Pascal, Perl, PHP, PL/I, Plasm, POV, Prolog, Promela, PSTricks, Python, R, Reduce, Rexx, RSL, Ruby, S, SAS, Scilab, sh, SHELXL, Simula, SQL, tcl, TeX, VBScript, Verilog, VHDL, VRML, XML, dan XSLT.

- **Dialek yang didukung secara *default* oleh listings** (menurut Heinz et al. (2024), diambil beberapa contoh):

- Dialek Assembly: `[Motorola68k]{Assembler}`, `[x86masm]{Assembler}`,
- Dialek Awk: `[gnu]{Awk}` (GNU Awk), `[POSIX]{Awk}`,
- Dialek C: `[ANSI]{C}` (default), `[Handel]{C}`, `[Objective]{C}` (Objective-C), `[Sharp]{C}` (C#),
- Dialek Caml: `[Objective]{Caml}` (OCaml), `[light]{Caml}` (default),
- Dialek C++: `[11]{C++}`, `[ANSI]{C++}`, `[GNU]{C++}`, `[Visual]{C++}` (Visual

- C++), [ISO]{C++} (default),
- Dialek Java: []{Java} (default), [AspectJ]{Java},
- Dialek Pascal: [Borland6]{Pascal}, [XSC]{Pascal}, [Standard]{Pascal} (default),
- Dialek Python: [2]{Python} (default, Python 2), [3]{Python} (Python 3),
- Dialek TeX: [LaTeX]{TeX} (LaTeX), [AlLaTeX]{TeX}, [plain]{TeX} (plain TeX, default),
- Dialek tcl: []{Tcl} (default Tcl), [tk]{Tcl} (Tcl/Tk).
- **Bahasa pemrograman yang ditambahkan pada *template* ini:**  
 ABS, Acceleo, Batch, Clojure, CSS, D, Dart, Docker, F# (FSharp), GDScript (Godot), GLSL, Groovy, HSL, JavaScript, Julia, Kotlin, Markdown, PowerShell, Rust, Scala, Scheme, Solidity, Swift, TOML, TypeScript, dan YAML.
- **Dialek yang ditambahkan pada *template* ini:**
  - Dialek HTML: [v5]{HTML} (HTML5),
  - Dialek Java: [v9]{Java} (Java 9 Modules), [ContextJ]{Java}, [DeltaJ]{Java}, dan [FOP]{Java}.

Berikut adalah contoh penggunaan dialek bahasa pemrograman dalam menuliskan kode program. Dalam kasus ini, Kode 2.14 merupakan kode Python versi 2. Sedangkan, Kode 2.15 merupakan kode Python versi 3. Perbedaan antara kedua versi tersebut cukup signifikan, salah satunya adalah pada perintah `print`. Pada Python 2, perintah `print` tidak memerlukan tanda kurung karena merupakan sebuah *keyword*, sedangkan pada Python 3, perintah `print` memerlukan tanda kurung karena merupakan sebuah *function*.

```
1 # Simple addition function
2 def add(a, b):
3     return a + b
4
5 result = add(3, 5)
6 print "The result is:", result
```

**Kode 2.14:** Kode Python 2

```
1 # Simple addition function
2 def add(a, b):
3     return a + b
4
5 result = add(3, 5)
6 print("The result is:", result)
```

**Kode 2.15:** Kode Python 3

Kode yang mencetak Kode 2.14 dan Kode 2.15 adalah sebagai berikut:

```
1 \lstinputlisting[language=Python, caption=Kode Python 2,
   label=code:python2]{assets/codes/2-sample-python2.py}
2 \lstinputlisting[language={[3]Python}, caption=Kode Python 3,
   label=code:python3]{assets/codes/2-sample-python3.py}
```

Secara *default*, dialek yang digunakan untuk Python pada *library listings* adalah Python 2. Sehingga untuk mencetak kode Python 3, perlu digunakan dialek Python 3 (`{[3]Python}`).

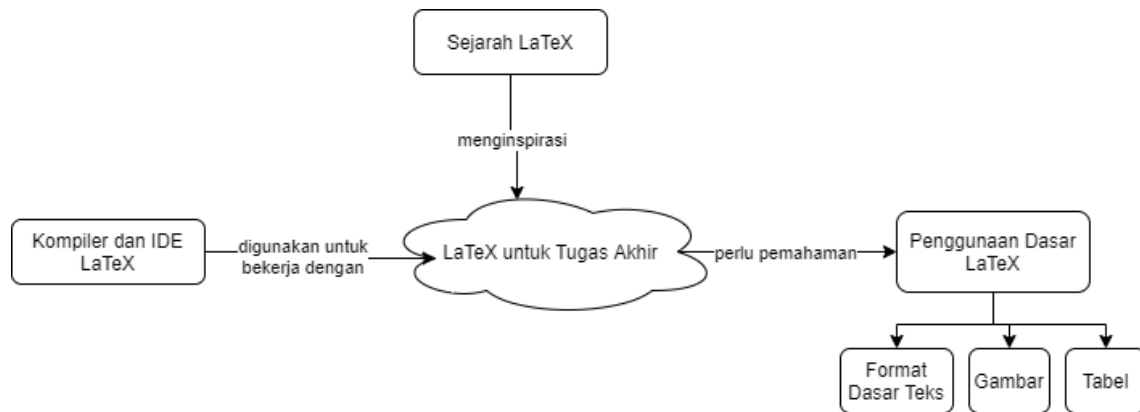
Kode program yang dicetak oleh L<sup>A</sup>T<sub>E</sub>X bersifat *auto-wrapped*. Jika suatu baris kode program melebihi batas lebar halaman, maka kode program tersebut akan dipindahkan ke baris berikutnya. *Auto-wrapped* ini berguna agar Anda tidak perlu memberikan *line break* manual pada kode Anda, Anda bisa menyampaikan kode program Anda apa adanya.

**Catatan:** Jangan lupa untuk menjelaskan kode melalui paragraf, terutama pada bagian-bagian yang perlu penjelasan lebih. Setiap kode perlu dijelaskan, kalau bisa rujuk ke setiap baris, karena belum tentu pembaca mau membaca kode Anda. Tetapi, pembaca tetap perlu mengetahui ide di balik kode yang Anda buat, dan mengapa kode tersebut dibuat.

## 2.7 Keterkaitan Teori Dengan Penelitian

### @todo

Ada baiknya setelah menjelaskan teori-teori, Anda menjelaskan apa kaitan teori tersebut dengan penelitian Anda. Hal ini tentunya membantu pembaca dalam memahami bahwa teori yang Anda paparkan memang penting untuk memahami penelitian Anda nantinya.



**Gambar 2.4:** Keterkaitan konsep hasil studi literatur terhadap penelitian

### @todo

Jelaskan Gambar 2.4 di sini. Setiap gambar pada tugas akhir butuh penjelasan. Gambar hadir untuk mempermudah membaca memahami konteks, tetapi tidak bisa berdiri sendiri tanpa penjelasan.

## BAB 3

### EKSPERIMEN DAN HASIL PENELITIAN

Bab ini menjelaskan tentang hal-hal *advanced* dalam  $\text{\LaTeX}$ . Hal ini mencakup bagaimana cara menulis persamaan matematis di  $\text{\LaTeX}$ , menambahkan daftar isi, catatan, PDF, menambahkan kode, bahkan menambahkan perintah baru.

#### **@todo**

Sejatinya bab ini digunakan untuk membahas inti dari penelitian Anda. Sesuaikan saja dengan kebutuhan Anda: misalkan bab tiga Anda adalah penjelasan terkait desain sistem.

### 3.1 Melakukan *Cross-Reference* ke Suatu Bagian dalam Laporan

Dengan menggunakan  $\text{\LaTeX}$ , Anda tidak perlu lagi melakukan referensi ke suatu bagian atau objek dalam laporan secara manual. Anda cukup melakukan referensi ke bagian/gambar/kode/persamaan yang Anda inginkan dengan menggunakan perintah `\ref`. Anda tidak perlu lagi mengubah referensi secara manual setiap kali ada perubahan letak pada bagian tersebut, karena  $\text{\LaTeX}$  akan melakukannya secara otomatis. Selain itu, pada berkas *Portable Document Format* (PDF) yang dihasilkan oleh  $\text{\LaTeX}$ , referensi tersebut akan memiliki *link* yang langsung mengarahkan pembaca ke posisi objek atau bagian yang direferensikan. Untuk melakukan *cross-reference*, pertama kali tandai bagian yang ingin Anda referensikan dengan menggunakan suatu label, melalui perintah `\label{...:....}`. Label tidak boleh mengandung spasi. Berikut ini adalah konvensi penamaan label dan cara melakukan referensi yang digunakan dalam *template* ini:

- `\label{bab:[nomorBab]}` untuk sebuah bab.

Contoh: `\label{bab:3}`

Cara referensi: `\bab~\ref{bab:3}`

Hasil referensi: Bab 3.

- `\label{sec:[....]}` untuk sebuah subbab.

Contoh: `\label{sec:crossReference}`

Cara referensi: `\sect~\ref{sec:crossReference}`

Hasil referensi: Subbab 3.1.

- `\label{appendix:[...]}` untuk sebuah bab/subbab lampiran.  
 Contoh: `\label{appendix:changelog}`  
 Cara referensi: `\apdx~\ref{appendix:changelog}`  
 Hasil referensi: Lampiran 1.
- `\label{equ:[...]}` untuk sebuah persamaan matematis.  
 Contoh: `\label{equ:matriks}`  
 Cara referensi: `\equ~\ref{equ:matriks}`  
 Hasil referensi: Persamaan 2.3.
- `\label{fig:[...]}` untuk sebuah gambar.  
 Contoh: `\label{fig:testGambar}`  
 Cara referensi: `\pic~\ref{fig:testGambar}`  
 Hasil referensi: Gambar 2.1.
- `\label{tab:[...]}` untuk sebuah tabel.  
 Contoh: `\label{tab:Tabel1}`  
 Cara referensi: `\tab~\ref{tab:tab1}`  
 Hasil referensi: Tabel 2.2.
- Untuk sebuah kode sumber, label diletakkan sebagai argumen dari `\lstinputlisting` seperti: `\lstinputlisting[... , label=code:...]`.  
 Contoh: `\lstinputlisting[language=Java, caption=Kode sampel Java, label=code:java]`  
 Cara referensi: `\lst~\ref{code:java}`  
 Hasil referensi: Kode 2.12.

### 3.2 Menggunakan BibTeX

BibTeX adalah *library* dalam L<sup>A</sup>T<sub>E</sub>X yang dapat membantu Anda untuk menuliskan sitasi. Dengan menggunakan BibTeX, Anda tidak perlu memikirkan format penulisan referensi atau sitasi. *Formatting* akan dilakukan secara otomatis sesuai dengan format sitasi yang digunakan. Secara *default*, *template* ini menggunakan format sitasi APA. Namun, format tersebut dapat diubah sesuai dengan peraturan yang dimiliki oleh fakultas, dosen pembimbing, atau dosen penguji Anda.

### 3.2.1 Menambahkan Referensi

Anda bisa menambahkan bahan bacaan yang ingin Anda jadikan referensi ke dalam berkas `references.bib`. Contoh isi kode *references.bib* saat ini dapat dilihat di Kode 3.1.

```
1 @book{book:sample,
2   author   = {Ahmad Surahmad and Bedu Pian Sebedu},
3   year     = {2020},
4   month    = {Januari},
5   pages    = {},
6   title    = {Buku Test},
7   isbn      = {999-9-999-99999-9},
8   journal  = {Judul Jurnal},
9   doi      = {99.9999/9-999-99999-9},
10  publisher = {Universitas Antah Berantah},
11  address   = {Depak}
12 }
```

**Kode 3.1:** Daftar referensi di `references.bib`

Format suatu objek referensi pada BibTeX adalah sebagai berikut:

```
@[tipe-referensi]{[kode-untuk-sitasi]
  title      = {Judul Buku},
  ....
}
```

Kode untuk sitasi dapat berisi karakter non-spasi yang bisa digunakan untuk melakukan sitasi di dalam konten laporan. Terdapat empat belas tipe referensi yang bisa digunakan pada BibTeX:

- `article`: Digunakan untuk merujuk ke sebuah artikel dalam suatu majalah, buku, atau koleksi artikel lainnya.
- `book`: Digunakan untuk merujuk ke sebuah buku.
- `booklet`: Digunakan untuk merujuk ke sebuah buku saku.
- `inbook`: Digunakan untuk merujuk ke sebuah bab atau subbab dalam suatu buku.
- `incollection`: Digunakan untuk merujuk ke sebuah bab atau subbab dalam suatu koleksi atau seri buku.
- `mastersthesis`: Digunakan untuk merujuk ke sebuah tesis karya mahasiswa magister (S2).
- `manual`: Digunakan untuk merujuk ke suatu buku manual.
- `phdthesis`: Digunakan untuk merujuk ke sebuah tesis karya mahasiswa doktoral (S3).

- `proceedings`: Digunakan untuk merujuk ke sebuah *paper* ilmiah yang dipublikasikan dalam suatu *conference* atau prosiding.
- `techreport`: Digunakan untuk merujuk ke suatu laporan teknis (misal: draf konvensi teknologi terbaru).
- `unpublished`: Digunakan untuk merujuk ke suatu hal yang tidak dipublikasikan.
- `misc`: Digunakan untuk merujuk ke hal-hal lain yang tidak masuk ke kategori-kategori yang telah disebutkan.

### 3.2.2 Melakukan Sitasi pada Konten Tugas Akhir

Berikut ini adalah contoh kalimat yang menggunakan sitasi:

”Kalimat menurut Surahmad and Sebedu (2020) terdiri dari subjek, predikat, dan objek (Surahmad and Sebedu, 2020).”

Berikut adalah kode yang digunakan untuk melakukan sitasi pada kalimat tersebut:

```
1 "Kalimat menurut \cite{book:sample} terdiri dari subjek, predikat, dan objek
   \citep{book:sample}."
```

Ada format sitasi yang memiliki cara penulisan yang berbeda berdasarkan posisi sitasi, ada juga yang tidak. Format sitasi APA membedakan penulisan sitasi pada isi kalimat dengan akhir kalimat, sedangkan format sitasi IEEE tidak. *Template* ini menggunakan format sitasi APA secara *default*, sehingga diperlukan pembeda berdasarkan posisi sitasi. Untuk melakukan sitasi pada isi kalimat, di mana sitasi tersebut umumnya sebagai subjek, objek, atau keterangan pada kalimat, gunakan perintah `\citep`. Sedangkan untuk melakukan sitasi pada akhir kalimat, di mana sitasi tersebut umumnya sebagai rujukan suatu gagasan, gunakan perintah `\cite`.

Perlu diperhatikan bahwa `\citep` hanya bisa digunakan untuk format sitasi yang butuh membedakan posisi sitasi. Penggunaan `\citep` pada format sitasi seperti IEEE akan menimbulkan error. Jika Anda menggunakan format seperti itu, cukup gunakan `\cite` dimanapun posisi sitasi Anda.

### 3.2.3 Mengubah Format Referensi/Sitasi

Sejak versi *template* 2.0.2, format referensi *default* telah diganti menjadi APA dari sebelumnya IEEE karena banyaknya permintaan dosen penguji untuk menggunakan format

APA. Pada dasarnya, peraturan Rektor UI terkait Tugas Akhir menyerahkan format referensi sesuai dengan aturan fakultas. Namun, mayoritas dari fakultas atau dosen pembimbing di Universitas Indonesia menggunakan APA sebagai format sitasinya. Oleh karena itu, jika fakultas atau dosen pembimbing/penguji Anda meminta format sitasi yang berbeda selain APA, Anda bisa menggantinya dengan mengikuti tahapan berikut:

1. Pada berkas `uithesis.sty`, terdapat bagian **Package**. Cari konfigurasi "Format sitasi".
2. Hilangkan tanda komentar (*uncomment*) pada bagian konfigurasi format yang akan digunakan, misal: APA. Pastikan hanya satu jenis konfigurasi format yang di-*uncomment*.
3. Cari "Konfigurasi khusus sitasi APA" di bagian **Ubah Istilah Penulisan**.
  - Jika Anda akan menggunakan format APA, hilangkan tanda komentar (*uncomment*) pada bagian konfigurasi tersebut.
  - Jika Anda akan menggunakan format selain APA, jadikan bagian konfigurasi tersebut sebagai komentar (*comment*).
4. Tidak semua format sitasi mengenal perbedaan pada sitasi di awal/tengah kalimat atau di akhir kalimat. Contoh format yang mengenal perbedaan tersebut adalah APA dan MLA. IEEE dan ACM tidak mengenal format tersebut.
  - Jika format sitasi yang akan digunakan mengenal perbedaan tersebut, ganti sitasi pada akhir kalimat atau tempat lain yang membutuhkan model sitasi dengan *parentheses* (kurung) dengan menggunakan perintah `\citep`.
  - Jika format sitasi yang akan digunakan tidak mengenal perbedaan tersebut, pastikan semua sitasi menggunakan perintah `\cite`.
5. Jika muncul pesan error seperti `[nama-format].bst not found`, itu tandanya format tersebut tidak tersedia secara bawaan dari BibTeX. Unduh berkas terkait dahulu dari CTAN, lalu letakkan di direktori `_internals`. Contoh format sitasi yang membutuhkan berkas eksternal adalah MLA (konfigurasi MLA sudah tersedia di `uithesis.sty`, namun berkas `mla.bst` belum tersedia).
6. Jika konfigurasi format sitasi belum tersedia di `uithesis.sty`, ikuti langkah-langkah berikut:
  - (a) Tambahkan konfigurasi baru di `uithesis.sty`, pada bagian **Package** > "Format sitasi". Contoh bisa mengikuti dengan format-format lain yang sudah tersedia, namun silakan sesuaikan dengan kebutuhan format sitasi yang akan digunakan.
  - (b) Jika format sitasi yang akan digunakan mengenal perbedaan pada sitasi di awal/ten-

gah kalimat atau di akhir kalimat, gunakan *package* *natbib* sehingga mendukung *command* sitasi `\citep`.

### 3.3 Membuat Daftar Istilah (Glosarium)

Daftar istilah atau glosarium adalah daftar kata atau frasa yang digunakan dalam dokumen beserta definisinya. Daftar frasa tersebut bisa berupa istilah, atau berupa singkatan/akronim. Template ini sudah menggunakan *library* *glossaries*. Berikut adalah langkah-langkah untuk mendefinisikan istilah baru atau singkatan/akronim baru dan menggunakannya dalam dokumen Anda.

#### 3.3.1 Menambahkan Istilah atau Akronim Baru

Untuk menambahkan istilah atau akronim baru, buka berkas `config/istilah.tex` dan tambahkan definisi istilah atau akronim baru menggunakan perintah `\newglossaryentry` atau `\newacronym`.

```
1 \newglossaryentry{latex}{
2   name={\LaTeX},
3   description={A document preparation system for high-quality typesetting}
4 }
```

**Kode 3.2:** Contoh definisi istilah baru

Pada Kode 3.2, ditunjukkan bahwa `\newglossaryentry` memiliki 3 argumen, yaitu:

- Argumen *positional* pertama: Merupakan kode panggilan untuk istilah tersebut. Kode tersebut yang nanti akan digunakan untuk memanggil istilah tersebut dalam dokumen.
- Argumen *keyword* `name`: Merupakan istilah yang akan dicetak ke dalam dokumen jika definisi ini dipanggil. Contoh: Jika `\gls{latex}` dipanggil, maka yang akan dicetak adalah  $\text{\LaTeX}$ .
- Argumen *keyword* `description`: Definisi (deskripsi) dari istilah tersebut. Deskripsi tersebut nantinya akan muncul di halaman Daftar Istilah.

```
1 \newacronym{pdf}{PDF}{Portable Document Format}
```

**Kode 3.3:** Contoh definisi singkatan/akronim baru

Pada Kode 3.3, ditunjukkan bahwa `\newacronym` memiliki 3 argumen, yaitu:

- Argumen pertama: Merupakan kode panggilan untuk akronim tersebut.

- Argumen kedua: Merupakan akronim (dalam bentuk singkatan) yang akan dicetak ke dalam dokumen jika definisi ini dipanggil menggunakan `\acrshort`. Contoh: Jika `\acrshort{pdf}` dipanggil, maka yang akan dicetak adalah PDF.
- Argumen ketiga: Kepanjangan dari akronim tersebut. Kepanjangan ini akan dicetak ke dalam dokumen jika definisi ini dipanggil menggunakan `\acrlong` atau `\acrfull`. Contoh:
  - `\acrlong{pdf}` akan mencetak *Portable Document Format*.
  - `\acrfull{pdf}` akan mencetak *Portable Document Format (PDF)*.

### 3.3.2 Menggunakan Istilah atau Akronim dalam Dokumen

Setelah mendefinisikan istilah atau akronim, Anda dapat menggunakannya dalam dokumen dengan perintah `\gls`, `\glspl`, `\acrshort`, `\acrlong`, atau `\acrfull`. Contoh:

```
1 \gls{latex} adalah sistem persiapan dokumen untuk pengetikan berkualitas tinggi.
2 \acrfull{pdf} adalah format berkas yang digunakan untuk representasi dokumen dua
  dimensi.
3 \acrlong{pdf} merupakan format berkas yang dibuat oleh Adobe.
4 \acrshort{pdf} dapat di-\f{edit} menggunakan Adobe Acrobat.
```

**Kode 3.4:** Contoh penggunaan istilah atau akronim dalam dokumen

Kode 3.4 akan menghasilkan kalimat berikut:

$\text{\LaTeX}$  adalah sistem persiapan dokumen untuk pengetikan berkualitas tinggi. *Portable Document Format* (PDF) adalah format berkas yang digunakan untuk representasi dokumen dua dimensi. *Portable Document Format* merupakan format berkas yang dibuat oleh Adobe. PDF dapat di-*edit* menggunakan Adobe Acrobat.

### 3.4 Memasukan Berkas PDF

Untuk memasukan berkas *Portable Document Format* (PDF) dapat menggunakan perintah `\inpdf` yang menerima satu buah argumen. Argumen ini berisi nama berkas yang akan digabungkan dalam laporan. PDF yang dimasukan dengan cara ini akan memiliki header dan footer seperti pada halaman lainnya.

Ini adalah berkas pdf yang dimasukan dalam dokumen laporan.

Cara lain untuk memasukan PDF adalah dengan menggunakan perintah `\putpdf` dengan satu argumen yang berisi nama berkas pdf. Berbeda dengan perintah sebelumnya, PDF yang dimasukan dengan cara ini tidak akan memiliki footer atau header seperti pada halaman lainnya.

Untitled

Ini adalah berkas pdf yang dimasukkan dalam dokumen laporan.

### 3.5 Memberikan Catatan

Ada dua perintah untuk memberikan catatan penulisan dalam dokumen yang Anda kerjakan, yaitu:

- `\todo`

Contoh:

**@todo**

Contoh bentuk todo.

- `\todoCite`

**@todo**

Contoh:

Referensi

### 3.6 Layoutting Tingkat Lanjut

#### 3.6.1 Menambahkan Tabel/Gambar Panjang secara Lanskap

Ketika Anda ingin memasukkan tabel atau gambar yang ukurannya cukup panjang ke samping, Anda diperkenankan untuk menyajikan konten tersebut dengan orientasi *landscape*. Caranya cukup mudah, yaitu dengan menambahkan `\begin{landscape}` di sebelum konten dan `\end{landscape}` di setelah konten. Format ini kompatibel juga dengan `longtable` untuk tabel yang panjang dan lebar. Contoh penggunaannya adalah pada Tabel 3.1.

**Tabel 3.1:** Contoh Tabel: Data Kasus COVID-19 di Asia, 14 September 2020**Sumber:** <https://worldometers.info/coronavirus>

#	Country, Other	Cases		Deaths		Recovered		Active	Critical	.../1M pop			Population
		Total	New	Total	New	Total	New			Tot Cases	Deaths	Tests	
1	India	4850887	5884	79784	30	3780107	3063	990996	8944	3508	58	41395	1382752528
2	Iran	404648	2619	23313	156	348013	1771	33322	3798	4805	277	42594	84209239
3	Bangladesh	339332	1812	4759	26	243155	2512	91418		2056	29	10560	165021623
4	Saudi Arabia	325651		4268		302870		18513	1326	9325	122	163863	34922248
5	Pakistan	302020	539	6383	4	289806	377	5831	551	1362	29	13388	221741906
6	Turkey	291162		7056		258833		25273	1267	3445	83	100796	84522503
7	Iraq	290309		8014		224705		57590	546	7186	198	46610	40399964
8	Philippines	265888	4699	4630	259	207504	249	53754	1048	2420	42	28018	109874163
9	Indonesia	221523	3141	8841	118	158405	3395	54277		808	32	9751	274108479
10	Israel	156823	1219	1126	7	115128	130	40569	529	17050	122	297533	9197590
11	Qatar	121740		205		118682		2853	37	43358	73	246111	2807805
12	Kazakhstan	106855	52	1634		100627	12	4594	221	5677	87	136625	18821980
13	Kuwait	94764		560		84995		9209	94	22124	131	157765	4283219
14	Oman	90222	476	790	10	83928	157	5504	171	17580	154	60252	5131974
15	China	85194	10	4634		80415	16	145	2	59	3	111163	1439323776
16	UAE	79489		399		69451		9639		8017	40	819752	9914483
17	Japan	75218		1439		66899		6880	180	595	11	13576	126395837
18	Bahrain	60307		212		53681		6414	29	35209	124	731472	1712845
19	Singapore	57454	48	27		56764		663		9805	5	389287	5859703
20	Nepal	54159		345		38697		15117		1852	12	28745	29240966

**Tabel 3.1:** Contoh Tabel: Data Kasus COVID-19 di Asia, 14 September 2020 (sambungan)**Sumber:** <https://worldometers.info/coronavirus>

#	Country, Other	Cases		Deaths		Recovered		Active	Critical	.../1M pop			Population
		Total	New	Total	New	Total	New			Tot Cases	Deaths	Tests	
21	Uzbekistan	47620	333	394	4	44002	136	3224	246	1419	12	41050	33566409
22	Armenia	45969	107	919	3	41693	34	3357		15507	310	81279	2964385
23	Kyrgyzstan	44928	47	1063		41023	101	2842	24	6864	162	40900	6545664
24	Afghanistan	38772	56	1425	5	32073	435	5274	93	992	36	2741	39100693
25	Azerbaijan	38327		562		35756		2009		3773	55	98716	10157722
26	Palestine	30574		221		20082		10271		5966	43	66248	5124685
27	Lebanon	24310		241		8334		15735	113	3565	35	94995	6819062
28	S. Korea	22285	109	363	5	18489	263	3433	157	435	7	41948	51278298
29	Malaysia	9946	31	128		9203	7	615	11	307	4	42286	32449426
30	Maldives	9173		32		7326		1815	12	16911	59	240315	542438
31	Tajikistan	9049		72		7816		1161		945	8		9579764
32	Syria	3540		155		842		2543		201	9		17583867
33	Thailand	3475	2	58		3312		105	1	50	0.8	10728	69836028
34	Jordan	3314		24		2206		1084	13	324	2	95814	10223646
35	Sri Lanka	3234		12		3005	9	217		151	0.6	11844	21431662
36	Myanmar	3015	83	24	4	699		2292		55	0.4	3518	54484197
37	Georgia	2392	165	19		1369		1004		600	5	118041	3987576
38	Yemen	2011		583		1212		216		67	19		29955256
39	Cyprus	1526		22		1281		223	2	1262	18	274810	1209149
40	Vietnam	1063		35		918		110		11	0.4	10348	97516308

**Tabel 3.1:** Contoh Tabel: Data Kasus COVID-19 di Asia, 14 September 2020 (sambungan)**Sumber:** <https://worldometers.info/coronavirus>

#	Country, Other	Cases		Deaths		Recovered		Active	Critical	.../1M pop			Population
		Total	New	Total	New	Total	New			Tot Cases	Deaths	Tests	
41	Taiwan	499	1	7		476	1	16		21	0.3	3770	23825661
42	Mongolia	311				300	2	11	1	95		18720	3288830
43	Cambodia	275				274		1		16		6926	16765404
44	Bhutan	245	1			161	2	84		317		151934	773324
45	Brunei	145		3		139		3		331	7	124633	438328
46	Timor-Leste	27				25		2		20		3888	1323423
47	Laos	23				22	1	1		3		6138	7296716

### 3.6.2 Alignment dan Word Wrapping pada Tabel

Mulai versi 2.1.0, Anda bisa melakukan *word wrapping* dalam tabel, dengan *alignment* sesuai yang diinginkan. Karakter *alignment* dapat ditambahkan pada konfigurasi tabel, contohnya adalah: `\begin{tabular}{|P0.5\textwidth|p{0.4\textwidth}|}`.

- p untuk *alignment justified* atas dengan *word wrapping*.
- m untuk *alignment justified* tengah dengan *word wrapping*.
- b untuk *alignment justified* bawah dengan *word wrapping*.
- P untuk *alignment* kiri-atas.
- L untuk *alignment* kiri-tengah.
- B untuk *alignment* kiri-bawah.
- U untuk *alignment* tengah-atas.
- C untuk *alignment* tengah-tengah.
- O untuk *alignment* tengah-bawah.
- E untuk *alignment* kanan-atas.
- R untuk *alignment* kanan-tengah.
- T untuk *alignment* kanan-bawah.

Contoh pemanfaatan *alignment* dan *word-wrapping* pada suatu `longtable` dapat dilihat pada Tabel 3.2.

**Tabel 3.2:** Contoh Tabel: Perbandingan metode pemodelan *access control*

Kategori	Model A	Model B	Model C
Latar belakang	Memodelkan struktur RBAC dalam perangkat lunak	Ekstensi dari RBAC sehingga bisa mendukung <i>constraint</i> berdasarkan properti subjek, objek, dan lingkungan	Memodelkan seluruh aspek keamanan dari sebuah <i>secure system</i>
Cakupan	Struktur eksplisit	Struktur eksplisit dengan <i>usage awareness</i>	Aspek-aspek keamanan generik dengan detail struktur bersifat implisit

**Tabel 3.2:** Contoh Tabel: Perbandingan metode pemodelan *access control* (sambungan)

Kategori	Model A	Model B	Model C
Format <i>diagram</i>	<i>Class diagram</i>	<i>Use case diagram</i> dan <i>sequence diagram</i>	RBAC pada <i>activity diagram</i>

Kode yang menyusun Tabel 3.2 terlihat pada Kode 3.5.

```

1 \begin{longtable}{|p{0.14\textwidth}|p{0.26\textwidth}|p{0.25\textwidth}|p{0.25\textwidth}|}
2   \caption{Contoh Tabel: Perbandingan metode pemodelan \f{access control}}
3   \label{tab:cellAlignmentWrapping} \\
4   \hline
5   \multicolumn{1}{|C{0.14\textwidth}|}{\bo{Kategori}}
6   &
7   \multicolumn{1}{C{0.26\textwidth}}{\bo{Model A}}
8   &
9   \multicolumn{1}{C{0.25\textwidth}}{\bo{Model B}}
10  &
11  \multicolumn{1}{C{0.25\textwidth}}{\bo{Model C}} \\
12  \hline
13  \endfirsthead % batas akhir header yang akan muncul di halaman pertama
14  \caption[]{}{Contoh Tabel: Perbandingan metode pemodelan \f{access control}
15  (sambungan)} \\
16  \hline
17  \multicolumn{1}{|C{0.14\textwidth}|}{\bo{Kategori}}
18  &
19  \multicolumn{1}{C{0.26\textwidth}}{\bo{Model A}}
20  &
21  \multicolumn{1}{C{0.25\textwidth}}{\bo{Model B}}
22  &
23  \multicolumn{1}{C{0.25\textwidth}}{\bo{Model C}} \\
24  \hline
25  \endhead
26  Latar \newline~belakang &
27  Memodelkan struktur RBAC dalam perangkat lunak &
28  Ekstensi dari RBAC sehingga bisa mendukung \f{constraint} berdasarkan properti
29  subjek, objek, dan lingkungan &
30  Memodelkan seluruh aspek keamanan dari sebuah \f{secure system} \\
31  \hline
32  Cakupan &
33  Struktur eksplisit &
34  Struktur eksplisit dengan \f{usage awareneess} &
35  Aspek-aspek keamanan generik dengan detail struktur bersifat implisit \\
36  \hline
37  Format \newline\f{diagram} &
38  \f{Class diagram} &
39  \f{Use case diagram} dan \f{sequence diagram} &

```

```

39   RBAC pada \{activity diagram} \\
40   \hline
41   \end{longtable}

```

**Kode 3.5:** Kode untuk Tabel 3.2

## 3.7 Daftar Isi atau Daftar Konten Lainnya

### 3.7.1 Menambahkan Konten ke Daftar Isi/Lampiran Secara Manual

Terkadang ada kebutuhan untuk memasukan kata-kata tertentu kedalam Daftar Isi. Perintah `\addChapter` dapat digunakan untuk judul bab dalam Daftar Isi. Contohnya dapat dilihat pada berkas `thesis.tex`. Untuk judul lampiran, Anda bisa menambahkannya ke dalam Daftar Lampiran dengan menggunakan `\addappendix`. Kedua perintah ini akan menambahkan entri baru setingkat sebuah bab (*chapter*).

### 3.7.2 Menambahkan Daftar Konten *Custom*

Selain itu, jika dibutuhkan, Anda juga bisa menambahkan daftar objek dengan jenis atau tujuan tertentu ke dalam laporan Anda. Misalkan, Anda ingin membuat "Daftar Aturan Transformasi" khusus untuk grafik-grafik yang menggambarkan aturan *transpiling* antar bahasa pemrograman. Untuk menambahkan hal tersebut, Anda perlu melakukan tahapan berikut:

1. Buka berkas `_internals/uithesis.sty` pada bagian "Daftar Konten Custom". Terdapat contoh kode untuk membuat daftar konten *custom*, dengan nama "Daftar Sesuatu" dan nama objek "Sesuatu". Untuk mencobanya, *uncomment* kode tersebut. Ada lima perintah yang akan dibuat kode tersebut.
  - `\listof....name`: Nama daftar isi untuk jenis objek tersebut, contoh: `\listofthingname` yang akan mengembalikan teks "Daftar Sesuatu".
  - `\listof....`: Daftar isi untuk jenis objek tersebut, contoh: `\listofthing` yang akan menghasilkan Daftar Sesuatu, yaitu daftar konten objek-objek Sesuatu.
  - `\....`: Nama jenis objek tersebut, contoh: `\thing` yang akan mengembalikan teks "Sesuatu".
  - `\caption....`: Caption untuk jenis objek tersebut, contoh: `\captionthing` yang berfungsi sebagai *caption* dari objek yang masuk

kategori "Sesuatu".

- `\captionsource....`: Caption dengan sumber untuk jenis objek tersebut,  
contoh: `\captionsource{thing}` yang berfungsi sebagai *caption* dari objek yang masuk kategori "Sesuatu", beserta dengan sumbernya.
- `\caption....cont`: Caption sambungan untuk jenis objek tersebut,  
contoh: `\caption{thing}cont.`
- `\captionsource....cont`: Caption sambungan dengan sumber untuk jenis objek tersebut,  
contoh: `\captionsource{thing}cont.`

Perintah `\caption{thing}cont` dan `\captionsource{thing}cont` bisa digunakan jika suatu objek "Sesuatu" ini merupakan tabel yang berlaku lintas halaman.

2. Untuk membuat daftar baru dengan nama berbeda, terdapat tiga frasa yang perlu diubah dari kode tersebut. Misalkan, Anda ingin membuat "Daftar Aturan Transformasi", maka Anda harus mengganti:
  - "Sesuatu" menjadi "Aturan Transformasi" untuk mengubah nama jenis objek,
  - `thing` menjadi `transformationrule` untuk mengubah tipe objek dalam L<sup>A</sup>T<sub>E</sub>X, dan
  - `loth` (akronim dari "list of things") menjadi `lotr` (singkatan dari "list of transformation rules") untuk mengubah ekstensi berkas *auxiliary* yang digunakan untuk menyimpan daftar objek tersebut.
3. Kemudian, Anda bisa menampilkan daftar konten *custom* yang baru Anda buat tersebut dengan mengikuti contoh kode yang ada di *thesis.tex*.
4. Gunakan `\caption....` dan `\captionsource....` untuk memberikan *caption* pada suatu objek (gambar/persamaan/tabel/kode) sekaligus menambahkannya ke dalam daftar objek tersebut.
5. Silakan definisikan sendiri konvensi label dan *cross-reference* yang menurut Anda cocok untuk jenis objek tersebut.

Misal: `\label{rule:....}` dan `\transformationrule~\ref{rule:....}`

Contoh kode untuk membuat daftar konten *custom*, dalam kasus ini Daftar Aturan Transformasi, dapat dilihat pada Kode 3.6.

```

1 % nama jenis objek
2 \newcommand{\transformationrule}{Aturan Transformasi}
3 \newcommand{\listoftransformationrulename}{Daftar \transformationrule}
4 % mendefinisikan daftar isi suatu jenis objek
5 \newlistof{transformationrule}{lotr}{\listoftransformationrulename}

```

```

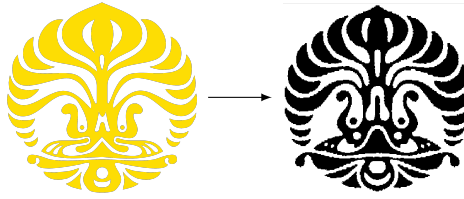
6 % mengatur penomoran suatu jenis objek
7 \counterwithin{transformationrule}{chapter}
8
9 \newcommand{\captiontransformationrule}[1]
10 {
11     % increment nomor caption
12     \refstepcounter{transformationrule}
13     % tambah caption
14     \caption*{\textbf{\thetransformationrule~\thetransformationrule:}~#1}
15     % tambah caption ke daftar isi
16     \addcontentsline{lotr}{transformationrule}
17     {\protect\numberline{\thetransformationrule}{\ignorespaces #1}}\par
18 }
19
20 \newcommand*{\captiontransformationrulecont}[2]{
21     % tambah caption sambungan
22     \caption*{\textbf{\thetransformationrule~\thetransformationrule:}~#1 (sambungan)}
23 }
24
25 \newcommand{\captionsourcetransformationrule}[2]
26 {
27     % increment nomor caption
28     \refstepcounter{transformationrule}
29     % tambah caption dengan sumber
30     \caption*{\textbf{\thetransformationrule~\thetransformationrule:}~#1\par
31         \footnotesize\textbf{Sumber:} #2\par}
32     % tambah caption ke daftar isi
33     \addcontentsline{lotr}{transformationrule}
34     {\protect\numberline{\thetransformationrule}{\ignorespaces #1}}\par
35 }
36
37 \newcommand*{\captionsourcetransformationrulecont}[2]{
38     % tambah caption sambungan dengan sumber
39     \caption*{\textbf{\thetransformationrule~\thetransformationrule:}~#1 (sambungan)\par
40         \footnotesize\textbf{Sumber:} #2\par}
41 }
42
43 \renewcommand\cftttransformationruleindent{0pt}
44 \renewcommand\cftttransformationrulenumwidth{50pt} % sesuaikan lebar ini agar penomoran
    tidak menimpa judul konten
45 \renewcommand\cftttransformationruleaftersnum{.}
46 \renewcommand\cftttransformationrulepresnum{\thetransformationrule~}

```

**Kode 3.6:** Kode Definisi untuk Daftar Aturan Transformasi di `_internals/uithesis.sty`

Dengan definisi yang telah diberikan di Kode 3.6, Anda bisa membuat objek ”Aturan Transformasi” dengan menggunakan fungsi *caption* seperti `\captiontransformationrule` atau `\captionsourcetransformationrule`. Contoh penggunaan *caption* tersebut da-

pat dilihat pada Aturan Transformasi 3.1.



**Aturan Transformasi 3.1:** Makara berwarna ke hitam-putih

### 3.8 Membuat Variabel atau Perintah Baru

Dalam  $\text{\LaTeX}$ , Anda bisa menambahkan variabel atau perintah baru yang dapat membantu penulisan laporan Anda. Sebenarnya variabel dalam  $\text{\LaTeX}$  merupakan perintah, namun tanpa argumen, contohnya adalah `\kucing`. Variabel dapat menyimpan suatu nilai teks. Sedangkan, suatu perintah pada  $\text{\LaTeX}$  sifatnya dapat menerima argumen dan mengolah argumen tersebut sesuai dengan kode yang didefinisikan di dalamnya. Contoh dari penggunaan perintah adalah `\section{Membuat Variabel atau Perintah Baru}`.

Ada dua perintah yang dapat digunakan untuk membuat variabel baru, yaitu:

- `\Var`

Digunakan untuk membuat variabel baru, namun setiap kata yang diberikan akan diproses dahulu menjadi huruf kapital.

Contoh: jika perintahnya adalah `\Var{\kucingBesar}{Areng}`, ketika perintah `\kucingBesar` dipanggil, yang akan muncul adalah ARENG.

- `\var`

Digunakan untuk membuat variabel baru tanpa mengubah *case* dari teks.

Contoh: jika perintahnya adalah `\var{\kucingKecil}{Areng}`, ketika perintah `\kucingKecil` dipanggil, yang akan muncul adalah Areng.

Membuat variabel baru sebaiknya dilakukan pada berkas `config/settings.tex`. Beberapa variabel yang terkait dengan metadata skripsi seperti judul, tanggal pengesahan, nama penulis, dsb. juga telah tersedia dalam `config/settings.tex` untuk dikonfigurasi.

Selain membuat variabel baru, membuat perintah baru dalam kasus tertentu diperlukan dalam melakukan *formatting*. Terdapat dua perintah untuk membuat suatu perintah baru yang nantinya bisa menerima argumen, yaitu:

- `\newcommand`

Digunakan untuk membuat perintah yang benar-benar baru. Beberapa contohnya adalah:

- `\newcommand{\sumber}[2]{\textbf{\#1: } \texttt{\#2}}` akan membuat perintah `\sumber` yang menerima dua argumen dan akan mencetak tulisan dengan format tertentu. Sehingga, ketika perintah `\sumber{Disadur dari}{Cimung}` dipanggil, yang akan muncul adalah **Disadur dari: Cimung**.
- `\newcommand{\kucing}[0]{Uyik}` akan membuat perintah `\kucing`, tanpa argumen. Ketika perintah `\kucing` dipanggil, yang akan muncul adalah Uyik.

- `\renewcommand`

Digunakan untuk mendefinisikan ulang perintah yang sudah ada. Contohnya adalah, jika sudah ada perintah `\sumber` yang menerima dua argumen, maka Anda bisa mendefinisikan ulang seperti ini: `\renewcommand{\sumber}{\textbf{\#1: } \texttt{\#2}}`. Sehingga, ketika perintah `\sumber{Disadur dari}{Cimung}` dipanggil, yang akan muncul adalah **Disadur dari: Cimung**.

Membuat perintah baru sebaiknya dilakukan pada berkas `uithesis.sty`. Berkas `uithesis.sty` adalah berkas khusus pengatur *styling* untuk tugas akhir ini. Berkas itu berisikan semua konfigurasi yang dibutuhkan untuk membuat dokumen L<sup>A</sup>T<sub>E</sub>X ini menjadi sesuai dengan Peraturan Rektor, termasuk perintah-perintah baru.

Jika perubahan ini dirasa penting untuk disertakan dalam template, silakan lakukan *fork* repositori Git template ini di <https://gitlab.com/ichlaffterlalu/latex-skripsi-ui-2017>, lalu lakukan *merge request* perubahan Anda terhadap *branch* master.

### 3.9 Pengaturan *Header* dan *Footer*

*Template* ini menggunakan *library* `fancyhdr` untuk mengatur *header* dan *footer*. Konfigurasi `fancyhdr` pada *template* ini terdiri dari empat profil, yaitu `empty`, `plain`, `first-pages`, dan `standard`. Profil `standard` merupakan profil standar untuk konten laporan, yaitu tulisan "Universitas Indonesia" di sisi kanan *footer*. Profil `first-pages` merupakan profil untuk konten depan laporan seperti abstrak, kata pengantar, dsb., yang mengharuskan nomor halaman di tengah *footer*. Profil `plain` dalam *template* ini akan selalu digunakan untuk halaman pertama pada setiap bab atau bagian (termasuk daftar isi, abstrak, dsb.), apapun jenis profil yang seharusnya digunakan pada bagian tersebut. Sedangkan, profil `empty` artinya tidak ada *header* dan *footer* sama sekali.

Konfigurasi profil dapat dilakukan dengan menggunakan `\pagestyle{nama-profil}`. Konfigurasi berlaku seterusnya dari halaman tersebut hingga ada konfigurasi profil berikutnya. Sedangkan untuk mendefinisikan sendiri isi *header* dan *footer* dapat dilakukan dengan perintah `\fancyhead[...]{...}` atau `\fancyfoot[...]{...}`. Contohnya, `\fancyhead[LO,RE]{Meong}` akan memberikan teks "Meong" di sisi kiri *header* untuk halaman ganjil (*odd*), dan di sisi kanan *header* untuk halaman genap (*even*).

### 3.9.1 Konfigurasi Satu Halaman per Lembar

Peraturan laporan tugas akhir di Universitas Indonesia tahun 2017 mensyaratkan pencetakan bolak-balik. Secara *default*, *template* ini juga sudah menggunakan konfigurasi bolak-balik. Namun, jika diperlukan, Anda dapat mengatur *header* dan *footer* ketika konfigurasi pencetakannya satu halaman per lembar. Penomoran halaman akan selalu dilakukan di bagian tengah pada *footer*. Oleh karena itu, dari bagian abstrak sampai akhir konten, cukup gunakan profil `first-page`. Kemudian, atur profil `plain` agar sama dengan profil `first-page`. Kemudian, hapus semua perintah `\clearchapter`, `\setoddevenheader`, `\naiveoddclearchapter`, dan `\naiveevenclearchapter` dalam berkas `thesis.tex`.

### 3.9.2 Konfigurasi untuk Submisi ke UI-ana

Berdasarkan peraturan terkini terkait pengumpulan naskah digital ke UI-ana, *header* dan *footer* perlu dihapus. Berikut ini adalah tahapan untuk mengatur hal tersebut:

1. Buka berkas `uithesis.sty`, lalu cari semua baris perintah `\fancypagestyle`. Hapus semua baris perintah tersebut.
2. Ubah isi dari perintah `\setoddevenheader` menjadi `\fancypagestyle{empty}`.
3. Di bagian akhir berkas `uithesis.sty`, tambahkan kode sebagai berikut:

```
1 \fancypagestyle{empty}{\fancyhead[L]{} \fancyhead[C]{} \fancyhead[R]{}
  \fancyfoot[L]{} \fancyfoot[C]{} \fancyfoot[R]{} }
2
```

4. Buka berkas `thesis.tex`, lalu cari semua baris perintah `\fancypagestyle` dan `\pagestyle{...}`. Hapus semua baris perintah tersebut.

## BAB 4

### ANALISIS DAN PEMBAHASAN

Bab ini menjelaskan tentang struktur dari *template* tugas akhir ini. Dengan memahami struktur *template*, pekerjaan Anda akan menjadi lebih terarah karena Anda tahu di mana Anda harus melakukan sesuatu.

#### **@todo**

Sejatinya bab ini digunakan untuk membahas inti dari penelitian Anda. Sesuaikan saja dengan kebutuhan Anda: misalkan bab empat Anda adalah penjelasan terkait implementasi sistem.

#### **4.1 thesis.tex**

Berkas `thesis.tex` berisi seluruh berkas  $\text{\LaTeX}$  yang dibaca, jadi bisa dikatakan sebagai berkas utama. Dari berkas ini kita dapat mengatur bab apa saja yang ingin kita tampilkan dalam dokumen.

#### **4.2 Direktori config**

Direktori `config` berisi berkas-berkas yang menyimpan konfigurasi variabel dan istilah-istilah yang bisa dimodifikasi sesuai dengan kebutuhan tugas akhir.

##### **4.2.1 settings.tex**

Berkas `settings.tex` berguna untuk mempermudah pembuatan beberapa template standar. Anda diminta untuk menuliskan judul laporan, nama, NPM, dan hal-hal lain yang dibutuhkan untuk pembuatan template.

##### **4.2.2 istilah.tex**

Berkas `istilah.tex` digunakan untuk mencatat istilah-istilah yang digunakan. Fungsinya hanya untuk memudahkan penulisan. Pada beberapa kasus, ada kata-kata yang harus selalu muncul dengan tercetak miring atau tercetak tebal. Anda juga bisa menggunakan berkas ini untuk mencatat istilah atau akronim khusus yang perlu dimunculkan di Daftar Istilah. Penggunaan lebih lanjut terkait berkas `_internals/istilah.tex` untuk menyim-

pan istilah atau akronim ada di Subbab 3.3. Dengan menjadikan kata-kata tersebut sebagai sebuah perintah  $\text{\LaTeX}$  tentu akan mempercepat dan mempermudah pengerjaan laporan.

### 4.2.3 `references.bib`

Berkas `references.bib` berisi seluruh daftar referensi yang digunakan dalam laporan. Anda bisa membuat model daftar referensi lain dengan menggunakan BibTeX. Untuk menambahkan referensi dengan format BibTeX, Anda bisa mengisi berkas `references.bib`. Untuk merujuk pada salah satu referensi yang ada, gunakan perintah `\cite`, e.g. `\cite{book:sample}` yang akan akan memunculkan Surahmad and Sebedu (2020). Informasi lebih lanjut mengenai referensi bisa dilihat di Subbab 4.2.3. Untuk mempelajari bibtex lebih lanjut, silahkan buka <http://www.bibtex.org/Format>.

## 4.3 Direktori `_internals`

Direktori `_internals` berisi halaman-halaman dan *styling* yang tidak perlu diubah untuk penggunaan normal dari template ini. *Styling* bisa diubah jika diperlukan untuk menyesuaikan beberapa fitur template dengan kebutuhan tugas akhir, atau untuk menyesuaikan dengan aturan terbaru yang dirilis oleh Universitas Indonesia.

### 4.3.1 `hype.indonesia.tex`

Berkas `hype.indonesia.tex` berisi cara pemenggalan beberapa kata dalam bahasa Indonesia.  $\text{\LaTeX}$  memiliki algoritma untuk memenggal kata-kata sendiri, namun untuk beberapa kasus algoritma ini memenggal dengan cara yang salah. Untuk memperbaiki pemenggalan yang salah inilah cara pemenggalan yang benar ditulis dalam berkas `hype.indonesia.tex`.

### 4.3.2 `uithesis.sty`

Berkas `uithesis.sty` berisi konfigurasi inti dari *layoutting* untuk *template* ini. Secara umum, Anda tidak perlu mengubah apapun pada berkas ini. Akan tetapi, untuk kasus-kasus lanjutan, seperti menambahkan daftar konten *custom* atau menyalakan dukungan terhadap *multi-language*, Anda bisa mengubahnya secara langsung pada `uithesis.sty`. Salah satu contohnya adalah ketika Anda ingin mendefinisikan daftar suatu jenis objek baru, seperti yang dicontohkan pada Subbab 3.7.2. Atau bisa juga ketika Anda ingin

mengganti tipe referensi, seperti yang dicontohkan pada Subbab 3.2.3. Jika Anda memiliki feedback maupun ingin berkontribusi terhadap perbaikan *layout*, selama ke arah yang sesuai dengan ketentuan Peraturan Rektor UI terkait format Tugas Akhir, Anda bisa mengubah berkas ini dan berkas lainnya yang terkait lalu membuat Merge Request di repositori. Keterangan lebih lanjut terkait cara kontribusi dapat dilihat di berkas `README.md` dan `CONTRIBUTING`.

#### 4.4 Direktori `src/00-frontMatter`

Direktori `src/00-frontMatter` berisi bagian depan yang memuat halaman-halaman administratif untuk laporan ilmiah Anda. Sedangkan direktori `src/99-backMatter` berisikan berkas-berkas lampiran. Berikut adalah daftar berkas yang tersedia di `src/00-frontMatter`:

1. `pernyataanOrisinalitas.tex` untuk halaman pernyataan orisinalitas. Berlaku untuk semua tipe dokumen kecuali Laporan Kerja Praktik dan Kampus Merdeka.
2. `pengesahanKP.tex` untuk halaman pengesahan spesifik tipe dokumen Laporan Kerja Praktik.
3. `pengesahanMBKM.tex` untuk halaman pengesahan spesifik tipe dokumen Kampus Merdeka.
4. `pengesahanSidang.tex` untuk halaman pengesahan sidang. Berlaku untuk semua tipe dokumen kecuali laporan ilmiah mahasiswa S3 (Disertasi), Laporan Kerja Praktik, dan Kampus Merdeka.
5. `pengesahanSidangS3.tex` untuk halaman pengesahan sidang khusus mahasiswa S3.
6. `kataPengantar.tex` untuk kata pengantar. Berlaku untuk semua tipe dokumen kecuali Laporan Kerja Praktik dan Kampus Merdeka.
7. `persetujuanPublikasi.tex` untuk halaman persetujuan publikasi karya intelektual. Berlaku untuk semua tipe dokumen kecuali Laporan Kerja Praktik dan Kampus Merdeka.
8. `abstrak.tex` untuk halaman abstrak berbahasa Indonesia.
9. `abstract.tex` untuk halaman abstrak berbahasa Inggris.

Umumnya, Anda hanya perlu mengisi bagian-bagian seperti Abstrak dan Kata Pengantar. Berkas sisanya berisi kode yang akan menghasilkan halaman-halaman terkait secara otomatis, sehingga hanya bisa diubah jika diperlukan penyesuaian, misal ukuran *line*

*spacing.*

#### 4.5 Direktori `src/01-body`

Direktori ini berisi isi laporan yang Anda tulis. Setiap nama berkas e.g. `bab1.tex` merepresentasikan bab dimana tulisan tersebut akan muncul. Sebagai contoh, kode dimana tulisan ini dibuat berada dalam berkas dengan nama `bab4.tex`. Ada enam buah berkas yang telah disiapkan untuk mengakomodir enam bab dari laporan Anda, diluar bab kesimpulan dan saran. Jika Anda tidak membutuhkan sebanyak itu, silahkan hapus kode dalam berkas `thesis.tex` yang memasukan berkas  $\text{\LaTeX}$  yang tidak dibutuhkan; contohnya perintah `\include{bab6.tex}` merupakan kode untuk memasukan berkas `bab6.tex` kedalam laporan.

## BAB 5

### KASUS-KASUS KHUSUS

Awalnya, *template* ini hanya digunakan untuk Tugas Akhir (Skripsi) mahasiswa S1 di Fakultas Ilmu Komputer, Universitas Indonesia. Seiring berkembangnya kegiatan pendidikan dan kemahasiswaan di lingkup Fakultas Ilmu Komputer hingga tingkat universitas, penyusun *template* menyadari ada kasus-kasus lain yang bisa menggunakan format Tugas Akhir UI. Beberapa di antaranya adalah tesis S2, disertasi S3, dan laporan kegiatan/kerja praktik. Oleh karena itu, perlu ada penjelasan terkait berbagai kasus penggunaan (*use case*) untuk *template* L<sup>A</sup>T<sub>E</sub>X ini, dan bagaimana cara pengguna bisa memanfaatkan *template* untuk kasus tersebut.

#### @todo

Sejatinya bab ini digunakan untuk membahas inti penelitian Anda. Bab lima pada tugas akhir S1 umumnya merupakan pembahasan analisis dari penelitian. Namun, sekali lagi, sesuaikan dengan kebutuhan Anda. Tesis atau disertasi tentunya berbeda dengan skripsi.

### 5.1 Tugas Akhir Individu S1, Proposal Tesis, dan Tesis S2

Tugas Akhir Individu di Fakultas Ilmu Komputer Universitas Indonesia berlaku sama dengan Tugas Akhir atau Skripsi mahasiswa S1 di fakultas lain di Universitas Indonesia. Proposal Tesis dan Tesis (di beberapa jurusan disebut Karya Akhir) di Fakultas Ilmu Komputer Universitas Indonesia juga berlaku sama dengan Tesis mahasiswa S2 di fakultas lain di Universitas Indonesia. Format yang digunakan untuk semua fakultas juga sama, mengacu ke Keputusan Rektor Universitas Indonesia nomor 2143/SK/R/UI/2017 tentang Pedoman Teknis Penulisan Tugas Akhir Mahasiswa Universitas Indonesia. Sejak versi 2.0.0, *template* ini sudah mengacu ke Keputusan Rektor UI tersebut. Pada versi tersebut juga dukungan untuk cetak skripsi atau tesis bolak-balik sudah tersedia. Tidak ada perubahan khusus yang perlu dilakukan terhadap konfigurasi *template* untuk Tugas Akhir untuk Mahasiswa S1 atau Proposal Tesis dan Tesis untuk Mahasiswa S2. Anda bisa mengikuti tahapan berikut untuk memulai penulisan Anda:

1. Buka `config/settings.tex`. Terdapat lima bagian yang perlu dilengkapi:

- **Judul dokumen:** Anda bisa memasukkan judul dalam bahasa Indonesia dan bahasa Inggris di sini.
  - **Tipe dokumen:** Pada variabel `\type`, cukup tuliskan "Skripsi" atau "Tugas Akhir", sesuaikan dengan aturan dari Fakultas masing-masing. Isi variabel `\jenjang` dengan "Sarjana" atau "Magister". Kosongkan variabel lainnya yang tidak relevan (jangan dihapus).
  - **Informasi penulis:** Karena pada kasus ini, tugas akhir Anda bersifat individu, cukup isi variabel `\penulisSatu` dengan nama Anda, `\npmSatu` dengan NPM Anda, `\programSatu` dengan nama program studi Anda dalam bahasa Indonesia, dan `\studyProgramSatu` dengan nama program studi Anda dalam bahasa Inggris. Untuk variabel lain mohon agar tetap dikosongkan (namun jangan dihapus) sehingga *template* bisa mendeteksi bahwa Anda akan menuliskan skripsi individu.
  - **Informasi dosen pembimbing dan penguji:** Pada umumnya, dosen pembimbing skripsi di UI terdiri dari satu atau dua orang dosen, dan penguji skripsi di UI terdiri dari dua orang dosen. Silakan isi variabel yang relevan dan kosongkan variabel lainnya (namun jangan dihapus).
  - **Informasi lain:** Anda bisa melihat komentar di setiap variabel untuk mengetahui apa yang harus diisi di setiap variabel.
  - **Judul setiap bab:** Silakan isi variabel yang ada untuk judul setiap bab. Jika ada bab yang ingin ditambahkan sebelum bab kesimpulan (misal: bab 6, bab 7), Anda dapat membuat variabel baru, contohnya: `\Var{\bab6}{Analisis Pendapat Pengguna Aplikasi}`.
  - Bagian lainnya seperti "Capitalized Variables" tidak perlu dimodifikasi. Variabel-variabel tersebut menunjang fungsi-fungsi khusus di *template*, salah satunya adalah versi *all caps* dari judul skripsi di halaman judul.
2. Setelah mengisi konfigurasi, Anda bisa periksa halaman-halaman awal dokumen. Jika terdapat ketidaksesuaian pada ukuran atau jarak antar elemen, Anda bisa mengatur melalui berkas-berkas yang ada di `src/00-frontMatter`. Halaman pengesahan sidang yang dipakai di format Tugas Akhir Individu ada di `src/00-frontMatter/pengesahanSidang.tex`. Silakan perbesar atau perkecil ukuran yang ada pada kode `\vspace*{...}`, untuk menyesuaikan *spacing*. Tahapan ini akan berguna terutama jika judul tugas akhir Anda cukup panjang sehingga beberapa teks ada yang terlempar ke halaman berikutnya. Jika ada perubahan kode yang sig-

nifikan, Anda bisa mengusulkan ke penyusun *template*. Keterangan lebih lanjut terkait cara kontribusi dapat dilihat di berkas `README.md` dan `CONTRIBUTING`.

3. Anda juga bisa mengatur beberapa hal sebagai berikut:
  - Pelajari cara sitasi dengan melihat Subbab 3.2 dan cara melakukan *cross-reference* dengan melihat Subbab 3.1. Kedua fitur tersebut merupakan fitur yang sangat penting dalam penulisan skripsi menggunakan  $\text{\LaTeX}$ .
  - Jika fakultas Anda memerlukan format sitasi selain APA (yang menjadi *default* di tingkat universitas), silakan baca Subbab 3.2.3.
  - Jika Anda membutuhkan support untuk selain tulisan alfabet, silakan baca Subbab ??.
  - Jika Anda membutuhkan penulisan notasi matematis, silakan baca Subbab 2.5. Jika Anda membutuhkan penulisan kode program, silakan baca Subbab 2.6.
4. Di akhir penulisan, Anda perlu memeriksa ulang tulisan Anda secara lebih teliti untuk memaksimalkan penggunaan kertas, sebisa mungkin hindari *unused space*. Selain itu, perhatikan juga pemenggalan yang dilakukan  $\text{\LaTeX}$  apakah sudah sesuai atau belum. Jika ada pemenggalan yang kurang sesuai, silakan tambahkan di `_internals/hypeindonesia.tex` dan *request* untuk kontribusi. Keterangan lebih lanjut terkait cara kontribusi dapat dilihat di berkas `README.md` dan `CONTRIBUTING`.

## 5.2 Tugas Akhir Kelompok S1

Beberapa fakultas, salah satunya Fakultas Ilmu Komputer Universitas Indonesia (sejak tahun 2022) mengizinkan pengerjaan skripsi secara berkelompok paling banyak 3 (tiga) orang. Format yang digunakan juga mengacu ke Keputusan Rektor Universitas Indonesia nomor 2143/SK/R/UI/2017 tentang Pedoman Teknis Penulisan Tugas Akhir Mahasiswa Universitas Indonesia, namun ada penyesuaian di beberapa hal. Sejak versi 2.1.3, *template* ini mendukung *format* Tugas Akhir kelompok dengan menyesuaikan bagian depan dari *template*. Untuk memanfaatkan *format* tersebut, silakan ikuti tahapan berikut:

1. Buka `config/settings.tex`. Isi variabel pada bagian "**Informasi Penulis**" untuk penulis pertama, kedua dan ketiga secara berurutan. Misal: `\penulisSatu` untuk nama penulis pertama, `\penulisDua` untuk nama penulis kedua, dan `\penulisTiga` untuk nama penulis ketiga. Pastikan Anda mengisi data secara lengkap pada variabel yang sesuai. Jika kelompok Anda hanya terdiri dari 2 (dua) orang, maka variabel-variabel data penulis ketiga harus dikosongkan (namun jangan dihapus). *Template*

akan menyesuaikan *format* sesuai dengan jumlah anggota kelompok di skripsi Anda.

2. Setelah mengisi konfigurasi, Anda bisa periksa halaman-halaman awal dokumen. Jika terdapat ketidaksesuaian pada ukuran atau jarak antar elemen, Anda bisa mengatur melalui berkas-berkas yang ada di `src/00-frontMatter`. Halaman pengesahan sidang yang dipakai di format Tugas Akhir Kelompok ada di `src/00-frontMatter/pengesahanSidang.tex`. Silakan perbesar atau perkecil ukuran yang ada pada kode `\vspace*{...}`, untuk menyesuaikan *spacing*. Tahapan ini akan berguna terutama jika judul tugas akhir Anda dan data kelompok Anda cukup panjang sehingga beberapa teks ada yang terlempar ke halaman berikutnya. Jika ada perubahan kode yang signifikan, Anda bisa mengusulkan ke penyusun *template*. Keterangan lebih lanjut terkait cara kontribusi dapat dilihat di berkas `README.md` dan `CONTRIBUTING`.

### 5.3 Laporan Ilmiah dan Disertasi S3

Disertasi S3 dan laporan-laporan lain yang diwajibkan untuk jenjang S3 juga menggunakan format sesuai Keputusan Rektor Universitas Indonesia nomor 2143/SK/R/UI/2017 tentang Pedoman Teknis Penulisan Tugas Akhir Mahasiswa Universitas Indonesia, namun ada penyesuaian di beberapa hal. Salah satu penyesuaian yang perlu dilakukan adalah istilah pembimbing yang berganti menjadi Promotor, Kopromotor. Jumlah penguji juga lebih banyak, bisa mencapai 6 orang dosen penguji. Sejak versi 2.1.2, *template* ini mendukung *format* disertasi dengan menyesuaikan bagian depan dari *template*. Untuk memanfaatkan *format* tersebut, silakan ikuti tahapan berikut:

1. Buka `config/settings.tex`.
  - Pada bagian **”Tipe Dokumen”**, variabel `\type` bisa diisi dengan **”Disertasi”** atau tipe dokumen lainnya. Variabel `\jenjang` wajib diisi dengan **”Doktor”**.
  - Pada bagian **”Informasi Pembimbing dan Penguji”**, isi nama lengkap dan gelar Promotor pada variabel `\pembimbingSatu`, dan Kopromotor pada variabel `\pembimbingDua` (jika kopromotor ada dua orang, variabel `\pembimbingTiga` bisa diisi). Untuk penguji, Anda bisa mengisi secara berurutan dari `\pengujiSatu` hingga `\pengujiEnam`.

Konfigurasi untuk dokumen laporan ilmiah S3 tidak mendukung format Tugas Akhir Kelompok.

2. Setelah mengisi konfigurasi, Anda bisa periksa halaman-halaman awal dokumen. Jika terdapat ketidaksesuaian pada ukuran atau jarak antar elemen, Anda bisa mengatur melalui berkas-berkas yang ada di `src/00-frontMatter`. Halaman pengesahan sidang yang dipakai di format laporan ilmiah S3 ada di `src/00-frontMatter/pengesahanSidangS3.tex`. Jika "Halaman Pengesahan" menjadi dua halaman, hal tersebut adalah lumrah. Jika ada hal yang tidak lumrah, silakan perbesar atau perkecil ukuran yang ada pada kode `\vspace*{...}`, untuk menyesuaikan *spacing*. Jika ada perubahan kode yang signifikan, Anda bisa mengusulkan ke penyusun *template*. Keterangan lebih lanjut terkait cara kontribusi dapat dilihat di berkas `README.md` dan `CONTRIBUTING`.

## 5.4 Laporan Kerja Praktik

Mata kuliah Kerja Praktik umumnya ditawarkan bagi individu sebagai mata kuliah bernilai SKS untuk mempresentasikan dan mendokumentasikan pekerjaan magang di industri melalui laporan karya ilmiah. Laporan Kerja Praktik di Fakultas Ilmu Komputer UI (dan sebagian fakultas yang menyediakan mata kuliah Kerja Praktik) juga menggunakan format sesuai Keputusan Rektor Universitas Indonesia nomor 2143/SK/R/UI/2017 tentang Pedoman Teknis Penulisan Tugas Akhir Mahasiswa Universitas Indonesia, namun ada penyesuaian di beberapa hal. Salah satu penyesuaian yang perlu dilakukan adalah halaman persetujuan yang berbeda karena Kerja Praktik tidak memerlukan sidang. Selain itu, ada beberapa halaman yang tidak diperlukan seperti Pernyataan Orisinalitas dan Persetujuan Publikasi. Sejak versi 2.1.2, *template* ini mendukung *format* laporan kerja praktik dengan menyesuaikan bagian depan dari *template*. Untuk memanfaatkan *format* tersebut, silakan ikuti tahapan berikut:

1. Buka `config/settings.tex`.
    - Pada bagian "**Tipe Dokumen**", variabel `\type` wajib diisi dengan "Laporan Kerja Praktik". Variabel `\jenjang` wajib diisi dengan "Sarjana".
    - Pada bagian "**Informasi Pembimbing dan Penguji**", isi nama lengkap dan gelar dosen kelas Kerja Praktik pada `\pembimbingSatu`, dan kosongkan semua variabel lain pada bagian tersebut (namun jangan dihapus).
- Konfigurasi untuk Laporan Kerja Praktik tidak mendukung format Tugas Akhir Kelompok.

2. Setelah mengisi konfigurasi, Anda bisa periksa halaman-halaman awal dokumen. Jika terdapat ketidaksesuaian pada ukuran atau jarak antar elemen, Anda bisa mengatur melalui berkas-berkas yang ada di `src/00-frontMatter`. Halaman persetujuan yang dipakai di format Laporan Kerja Praktik ada di `src/00-frontMatter/pengehasilanKP.tex`. Silakan perbesar atau perkecil ukuran yang ada pada kode `\vspace*{...}`, untuk menyesuaikan *spacing*. Jika ada perubahan kode yang signifikan, Anda bisa mengusulkan ke penyusun *template*. Keterangan lebih lanjut terkait cara kontribusi dapat dilihat di berkas `README.md` dan `CONTRIBUTING`.

## 5.5 Laporan Kegiatan Merdeka Belajar Kampus Merdeka

Program Merdeka Belajar Kampus Merdeka<sup>1</sup> merupakan program *flagship* dari Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbud) Republik Indonesia yang bertujuan untuk memberikan peluang mahasiswa mendapatkan pengalaman belajar di luar kampus. Terdapat banyak pilihan program Kampus Merdeka yang tersedia bagi mahasiswa UI, beberapa di antaranya adalah Magang Bersertifikat, Studi Independen Bersertifikat (termasuk Program Bangkit<sup>2</sup>), dan beberapa program lain di tingkat UI seperti *Build Your Own Course* (BYOC). Pada akhir program, mahasiswa diminta menyusun laporan dengan format yang disediakan untuk Kemendikbud, yang tentunya hanya tersedia untuk program dan jalur yang dikelola Kemendikbud. Beberapa program seperti BYOC dan jalur yang diselenggarakan UI seperti Kampus Merdeka Mandiri tidak memiliki akses ke template Kemendikbud. Di Fakultas Ilmu Komputer, laporan MBKM yang tidak melewati jalur yang dikelola Kemendikbud menggunakan laporan akhir layaknya Laporan Kerja Praktik yang formatnya menggunakan aturan Keputusan Rektor Universitas Indonesia nomor 2143/SK/R/UI/2017 tentang Pedoman Teknis Penulisan Tugas Akhir Mahasiswa Universitas Indonesia, dengan beberapa penyesuaian. Salah satu penyesuaian yang perlu dilakukan adalah halaman persetujuan yang berbeda karena Kampus Merdeka tidak memerlukan sidang, namun berbeda dengan Kerja Praktik, laporan kegiatan Kampus Merdeka membutuhkan persetujuan dari mitra. Sejak versi 2.1.3, *template* ini mendukung *format* laporan kerja praktik dengan menyesuaikan bagian depan dari *template*. Untuk memanfaatkan *format* tersebut, silakan ikuti tahapan berikut:

1. Buka `config/settings.tex`.

---

<sup>1</sup><https://kampusmerdeka.kemdikbud.go.id/>

<sup>2</sup><https://www.dicoding.com/programs/bangkit>

- Pada bagian **”Tipe Dokumen”**, variabel `\type` wajib diisi dengan **”Kampus Merdeka”**. Variabel `\jenjang` wajib diisi dengan **”Sarjana”**. Variabel `\kampusMerdekaType` wajib diisi dengan tipe kegiatan atau jalur yang diambil, misal: Magang, Studi Independen, Bangkit, dsb. Jika program memiliki mitra, variabel `\partnerPosition` wajib diisi dengan jabatan yang dimiliki perwakilan mitra yang akan menandatangani laporan Anda. Jika program memiliki mitra, variabel `\partnerInstance` wajib diisi dengan instansi, perusahaan, atau program yang menjadi tempat kerja perwakilan mitra yang akan menandatangani laporan Anda.
- Pada bagian **”Informasi Pembimbing dan Penguji”**, isi nama lengkap dan gelar dosen penanggungjawab program Kampus Merdeka yang diambil (untuk mahasiswa Fasilkom UI) atau Pembimbing Akademik (untuk fakultas lain) pada `\pembimbingSatu`. Kemudian, isi nama lengkap perwakilan penyelia atau manajer dari mitra tempat kegiatan pada `\pembimbingDua`. Jika program tidak memiliki mitra (misalkan BYOC), kosongkan variabel `\pembimbingDua`. Kosongkan semua variabel lain pada bagian tersebut (namun jangan dihapus).

Konfigurasi untuk Kampus Merdeka tidak mendukung format Tugas Akhir Kelompok.

2. Setelah mengisi konfigurasi, Anda bisa periksa halaman-halaman awal dokumen. Jika terdapat ketidaksesuaian pada ukuran atau jarak antar elemen, Anda bisa mengatur melalui berkas-berkas yang ada di `src/00-frontMatter`. Halaman persetujuan yang dipakai di format Laporan Kerja Praktik ada di `src/00-frontMatter/pengesahanMBKM.tex`. Silakan perbesar atau perkecil ukuran yang ada pada kode `\vspace*{...}`, untuk menyesuaikan *spacing*. Jika ada perubahan kode yang signifikan, Anda bisa mengusulkan ke penyusun *template*. Keterangan lebih lanjut terkait cara kontribusi dapat dilihat di berkas `README.md` dan `CONTRIBUTING`.



## **BAB 6**

### **PENUTUP**

Pada bab ini, Penulis akan memaparkan kesimpulan penelitian dan saran untuk penelitian berikutnya.

#### **6.1 Kesimpulan**

Berikut ini adalah kesimpulan terkait pekerjaan yang dilakukan dalam penelitian ini:

##### **1. Poin pertama**

Penjelasan poin pertama.

##### **2. Poin kedua**

Penjelasan poin kedua.

Tulis kalimat penutup di sini.

#### **6.2 Saran**

Berdasarkan hasil penelitian ini, berikut ini adalah saran untuk pengembangan penelitian berikutnya:

1. Saran 1.

2. Saran 2.



## DAFTAR REFERENSI

Clark, J. (2010). Introduction to L<sup>A</sup>T<sub>E</sub>X.

Heinz, C., Moses, B., and Hoffmann, J. (2024). *The Listings Package*, 1.10c edition.

Surahmad, A. and Sebedu, B. P. (2020). *Buku Test*. Universitas Antah Berantah, Depok.

Universitas Indonesia (2017). *Pedoman Teknis Penulisan Tugas Akhir Mahasiswa Universitas Indonesia*. Depok, revisi edition.



## DAFTAR ISTILAH

**L<sup>A</sup>T<sub>E</sub>X** Sebuah *mark up language* yang didesain khusus untuk karya tulis ilmiah. 1, 2, 6, 7, 11, 15, 19, 20, 23, 26, 28, 29, 33, 34, 45, 47, 48, 50, 51, 53, 54, 56

**PDF** *Portable Document Format*. 28, 34, 36



# LAMPIRAN



## Lampiran 1: CHANGELOG

**@todo**

Silakan hapus lampiran ini ketika Anda mulai menggunakan *template*.

*Template* versi terbaru bisa didapatkan di <https://gitlab.com/ichlaffterlalu/latex-skripsi-ui-2017>. Daftar perubahan pada *template* hingga versi ini:

- versi 1.0.3 (3 Desember 2010):
  - *Template* Skripsi/Tesis sesuai ketentuan *formatting* tahun 2008.
  - Bisa diakses di <https://github.com/edom/uistyle>.
- versi 2.0.0 (29 Januari 2020):
  - *Template* Skripsi/Tesis sesuai ketentuan *formatting* tahun 2017.
  - Menggunakan BibTeX untuk sitasi, dengan format *default* sitasi IEEE.
  - *Template* kini bisa ditambahkan kode sumber dengan *code highlighting* untuk bahasa pemrograman populer seperti Java atau Python.
- versi 2.0.1 (8 Mei 2020):
  - Menambahkan dan menyesuaikan tutorial dari versi 1.0.3, beserta cara kontribusi ke *template*.
- versi 2.0.2 (14 September 2020):
  - Versi ini merupakan hasil *feedback* dari peserta skripsi di lab *Reliable Software Engineering* (RSE) Fasilkom UI, semester genap 2019/2020.
  - BibTeX kini menggunakan format sitasi APA secara *default*.
  - Penambahan tutorial untuk `longtable`, agar tabel bisa lebih dari 1 halaman dan header muncul di setiap halaman.
  - Menambahkan tutorial terkait penggunaan BibTeX dan konfigurasi *header/footer* untuk pencetakan bolak-balik.
  - Label "Universitas Indonesia" kini berhasil muncul di halaman pertama tiap bab dan di bagian abstrak - daftar kode program.
  - *Hyphenation* kini menggunakan `babel Bahasa Indonesia`. Aktivasi dilakukan di `hype-indonesia.tex`.
  - Minor adjustment untuk konsistensi *license* dari *template*.
- versi 2.0.3 (15 September 2020):

- Menambahkan kemampuan orientasi *landscape* beserta tutorialnya.
- \captionsource telah diperbaiki agar bisa dipakai untuk longtable.
- Daftar lampiran kini telah tersedia, lampiran sudah tidak masuk daftar isi lagi.
- Nomor halaman pada lampiran dilanjutkan dari halaman terakhir konten (daftar referensi).
- Kini sudah bisa menambahkan daftar isi baru untuk jenis objek tertentu (custom), seperti: "Daftar Aturan Transformasi". Sudah termasuk mekanisme *captioning* dan tutorialnya.
- Perbaiki minor pada tutorial.
- versi 2.1.0 (8 September 2021):
  - Versi ini merupakan hasil *feedback* dari peserta skripsi dan tesis di lab *Reliable Software Engineering* (RSE) Fasilkom UI, semester genap 2020/2021.
  - Minor edit: "Lembar Pengesahan", dsb. di daftar isi menjadi all caps.
  - Experimental multi-language support (Chinese, Japanese, Korean).
  - *Support* untuk justifikasi dan word-wrapping pada tabel.
  - Penggunaan suffix "(sambungan)" untuk tabel lintas halaman. Tambahan support suffix untuk \captionsource.
- versi 2.1.1 (7 Februari 2022):
  - Update struktur mengikuti fork template versi 1.0.3 di <https://github.com/rkkautsar/edom/ui-thesis-template>.
  - *Support* untuk simbol matematis amsfonts.
  - Kontribusi komunitas terkait improvement GitLab CI, atribusi, dan format sitasi APA bahasa Indonesia.
  - Perbaiki tutorial berdasarkan perubahan terbaru pada versi 2.1.0 dan 2.1.1.
- versi 2.1.2 (13 Agustus 2022):
  - Modifikasi penamaan beberapa berkas.
  - Perbaiki beberapa halaman depan (halaman persetujuan, halaman orisinalitas, dsb.).
  - *Support* untuk lembar pengesahan yang berbeda dengan format standar, seperti Laporan Kerja Praktik dan Disertasi.
  - Kontribusi komunitas terkait kesesuaian dengan format Tugas Akhir UI, kelengkapan dokumen, perbaiki format sitasi, dan *quality-of-life*.
  - Perbaiki tutorial.
- versi 2.1.3 (22 Februari 2023):

- Dukungan untuk format Tugas Akhir Kelompok di Fasilkom UI.
- Dukungan untuk format laporan Kampus Merdeka Mandiri di Fasilkom UI.
- Minor *bugfix*: Perbaikan kapitalisasi variabel.
- Quality-of-Life: Pengaturan kembali `config/settings.tex`.
- Tutorial untuk beberapa *use case*.
- versi 2.2.0 (28 Agustus 2024):
  - Perbaikan format agar sesuai dengan format Tugas Akhir terbaru. Hal ini mencakup halaman judul, halaman pernyataan orisinalitas, header/footer, dan lampiran.
- versi 2.2.1 (16 Desember 2024):
  - *Bugfix*: isu *header* dan *footer* untuk halaman bolak-balik.
  - *Bugfix*: isu *auto-wrapping* pada kode yang tidak bisa berjalan sejak v2.2.0.
  - *Bugfix*: isu penomoran objek kustom yang tidak sesuai konvensi `[bab].[objek]`.
  - *Bugfix*: penomoran bab di Daftar Isi yang belum sesuai konvensi Tugas Akhir UI.
  - *Bugfix*: hal-hal lain pada *formatting* sesuai dengan permintaan dari Perpustakaan Fasilkom UI.
  - Perbaikan *formatting* untuk *landscape* dengan *library* `pdfscape`.
  - Perbaikan cara memasukkan sebuah persamaan ke dalam daftar persamaan.
  - Perbaikan penggunaan "saya" menjadi "kami" untuk dokumen-dokumen awal pada Tugas Akhir Kelompok.
  - Fitur baru: *Support* untuk *code highlighting* pada berbagai bahasa pemrograman yang tidak di-*support* secara *default* oleh *library listings*.
  - Fitur baru: *Support* untuk *glossary* (daftar istilah).
  - Perbaikan *major* pada tutorial, termasuk menampilkan contoh kode ke dalam PDF tutorial, dan pengaturan ulang subbab.

## Lampiran 2: Judul Lampiran 2

Lampiran hadir untuk menampung hal-hal yang dapat menunjang pemahaman terkait tugas akhir, namun akan mengganggu *flow* bacaan sekiranya dimasukkan ke dalam bacaan. Lampiran bisa saja berisi data-data tambahan, analisis tambahan, penjelasan istilah, tahapan-tahapan antara yang bukan menjadi fokus utama, atau pranala menuju halaman luar yang penting.

**Subbab dari Lampiran 2****@todo**

Isi subbab ini sesuai keperluan Anda. Anda bisa membuat lebih dari satu judul lampiran, dan tentunya lebih dari satu subbab.