



UNIVERSITAS INDONESIA

**EKSPERIMEN *END-TO-END* UNTUK *KAFKA RESOURCE MANAGEMENT* PADA PENELITIAN “EFFICIENT TOPIC PARTITIONING OF APACHE KAFKA FOR HIGH-RELIABILITY REAL-TIME DATA STREAMING APPLICATIONS” DI LAB CSL
FAKULTAS ILMU KOMPUTER UI**

LAPORAN KERJA PRAKTIK

**BRYAN RAIHAN 'ILMAN
2106704351**

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
AGUSTUS 2024**



UNIVERSITAS INDONESIA

**EKSPERIMEN *END-TO-END* UNTUK *KAFKA RESOURCE MANAGEMENT* PADA PENELITIAN “EFFICIENT TOPIC PARTITIONING OF APACHE KAFKA FOR HIGH-RELIABILITY REAL-TIME DATA STREAMING APPLICATIONS” DI LAB CSL
FAKULTAS ILMU KOMPUTER UI**

LAPORAN KERJA PRAKTIK

Diajukan sebagai salah satu syarat kelulusan mata kuliah
Kerja Praktik

**BRYAN RAIHAN 'ILMAN
2106704351**

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
AGUSTUS 2024**

HALAMAN PERSETUJUAN DOSEN KERJA PRAKTIK

Laporan Kerja Praktik ini diajukan oleh:

Nama : Bryan Raihan 'Ilman
NPM : 2106704351
Program Studi : Ilmu Komputer
Judul Kerja Praktik : Eksperimen *End-To-End* untuk *Kafka Resource Management* pada penelitian “Efficient topic partitioning of Apache Kafka for high-reliability real-time data streaming applications” di lab CSL
Fakultas Ilmu Komputer UI

Telah berhasil diselesaikan laporan kerja praktik untuk Fakultas Ilmu Komputer dan dipresentasikan hasil kerja praktiknya sebagai persyaratan yang harus dipenuhi dalam mata kuliah Kerja Praktik.

DOSEN MATA KULIAH KERJA PRAKTIK

Ichlasul Affan, M.Kom.

Ditetapkan di : Depok

Tanggal : Agustus 2024

ABSTRAK

Nama : Bryan Raihan 'Ilman
Program Studi : Ilmu Komputer
Judul : Eksperimen *End-To-End* untuk *Kafka Resource Management* pada penelitian “Efficient topic partitioning of Apache Kafka for high-reliability real-time data streaming applications” di lab CSL Fakultas Ilmu Komputer UI
Pembimbing : Ichlasul Affan, M.Kom.

Streaming data secara *real-time* merupakan proses pengiriman data secara kontinu dari sumber ke penerima, yang sering digunakan dalam aplikasi yang membutuhkan respons cepat dan aliran data besar. Apache Kafka adalah platform yang digunakan untuk mengelola proses ini, di mana data dipartisi menjadi topik untuk memungkinkan pemrosesan paralel oleh berbagai konsumen. Kerja praktik ini berfokus pada penelitian optimisasi partisi topik Apache Kafka, sebagaimana dijelaskan dalam paper berjudul “Efficient Topic Partitioning of Apache Kafka for High-Reliability Real-Time Data Streaming Applications”. Pelaksana kerja praktik berperan sebagai *Research Assistant* di Computer Systems Laboratory (CSL) Fakultas Ilmu Komputer Universitas Indonesia, sebuah laboratorium yang fokus pada penelitian komputasi kinerja tinggi dan manajemen big data. Tugas utamanya adalah menjalankan eksperimen *end-to-end*, yang dalam konteks ini berarti mengamati aliran data dari produser hingga konsumen untuk mengukur latensi dari di seluruh proses. Metodologi yang digunakan dalam kerja praktik ini mencakup beberapa langkah penting. Di awal, pelaksana menyiapkan VM di Google Cloud Platform (GCP) dan menginstal semua dependensi yang dibutuhkan untuk eksperimen Kafka. Setelah itu, eksperimen dijalankan dengan satu VM yang bertindak sebagai klien Kafka dan satu lagi sebagai kluster Kafka. Eksperimen difokuskan pada pengujian algoritma BroMax dan BroMin untuk memantau perilaku partisi topik. Algoritma tersebut digunakan untuk secara matematis menentukan jumlah partisi yang optimal demi mencapai *throughput* maksimal. Selain itu, resource monitoring dilakukan secara rutin untuk melacak penggunaan CPU, RAM, dan *disk space* pada VM. Eksperimen lalu diulang pada infrastruktur UI (DGX dan BMKGCS), termasuk eksplorasi penggunaan Docker-in-Docker untuk menjalankan eksperimen tanpa akses root. Metodologi Scrum diterapkan dalam proyek ini dengan siklus *sprint* dua minggu. Hasil dari penelitian ini membantu kita memahami perilaku *latency* dan *throughput* pada berbagai *application constraints*.

Kata kunci:

Fog Computing, Data Streaming, Topic Partitioning, Distributed Systems

ABSTRACT

Name : Bryan Raihan 'Ilman
Study Program : Computer Science
Title : End-To-End Experiments for Kafka Resource Management in the research "Efficient topic partitioning of Apache Kafka for high-reliability real-time data streaming applications" di lab CSL Fakultas Ilmu Komputer UI
Counselor : Ichlasul Affan, M.Kom.

Real-time data streaming is the process of continuously sending data from a source to a receiver, often used in applications that require quick responses and large data flows. Apache Kafka is the platform used to manage this process, where data is partitioned into topics to enable parallel processing by multiple consumers. This internship focused on researching the optimization of Apache Kafka topic partitioning, as outlined in the paper titled "Efficient Topic Partitioning of Apache Kafka for High-Reliability Real-Time Data Streaming Applications." The intern served as a Research Assistant at the Computer Systems Laboratory (CSL) of the Faculty of Computer Science, Universitas Indonesia, a laboratory focused on high-performance computing and big data management research. The primary task was to conduct end-to-end experiments, which in this context means observing the flow of data from producers to consumers to measure latency throughout the process. The methodology used in this internship included several key steps. Initially, the intern set up VMs on Google Cloud Platform (GCP) and installed all necessary dependencies for Kafka experiments. After that, experiments were run with one VM acting as a Kafka client and another as the Kafka cluster. The experiments focused on testing the BroMax and BroMin algorithms to monitor the behavior of topic partitioning. These algorithms were used to mathematically determine the optimal number of partitions to achieve maximum throughput. Additionally, resource monitoring was regularly conducted to track CPU, RAM, and disk space usage on the VMs. The experiments were then repeated on UI infrastructure (DGX and BMKGCS), including the exploration of using Docker-in-Docker to run experiments without root access. The Scrum methodology was applied in this project, with two-week sprint cycles. The results of this research help us understand the behavior of latency and throughput under various application constraints.

Keywords:

Fog Computing, Data Streaming, Topic Partitioning, Distributed Systems

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN DOSEN KERJA PRAKTIK	ii
ABSTRAK	iii
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
DAFTAR KODE PROGRAM	viii
DAFTAR LAMPIRAN	ix
 BAB 1 PENDAHULUAN	 1
1.1 Proses Pencarian Kerja Praktik	1
1.2 Tempat Kerja Praktik	2
1.2.1 Profil Tempat Kerja Praktik	2
1.2.2 Posisi Penempatan Pelaksana Kerja Praktik dalam Struktur Organisasi	2
 BAB 2 ISI	 4
2.1 Latar Belakang Pekerjaan	4
2.2 Deskripsi Pekerjaan	4
2.3 Tinjauan Pustaka	5
2.3.1 Apache Kafka	5
2.3.2 Docker Containerization	5
2.3.3 Apache Zookeeper	6
2.3.4 Distributed System	6
2.4 Membuat Tabel	6
2.5 Keterkaitan Teori Dengan Penelitian	9
 BAB 3 PENUTUP	 11
3.1 Membuat Persamaan Matematis	11
3.1.1 Satu Persamaan	11
3.1.2 Lebih dari Satu Persamaan	12
3.2 Mengubah Tampilan Teks	12
3.3 Menambahkan Kode Program	13
3.4 Memberikan Catatan	14
3.5 <i>Layoutting</i> Tingkat Lanjut	14
3.5.1 Menambahkan Tabel/Gambar Panjang secara Lanskap	14
3.5.2 <i>Alignment</i> dan <i>Word Wrapping</i> pada Tabel	18
3.6 Melakukan <i>Cross-Reference</i> ke Suatu Bagian dalam Laporan	19
3.7 Menggunakan BibTeX	20
3.7.1 Menambahkan Referensi	20
3.7.2 Melakukan Sitasi pada Konten Tugas Akhir	22
3.7.3 Mengubah Format Referensi/Sitasi	23
3.8 Daftar Isi atau Daftar Konten Lainnya	24
3.8.1 Menambahkan Konten ke Daftar Isi/Lampiran Secara Manual	24

3.8.2	Menambahkan Daftar Konten <i>Custom</i>	24
3.9	Memasukan PDF	25
3.10	Membuat Variabel atau Perintah Baru	30
3.11	Pengaturan <i>Header</i> dan <i>Footer</i>	31
3.11.1	Konfigurasi Satu Halaman per Lembar	32
3.11.2	Konfigurasi untuk Submisi ke UI-ana	32
DAFTAR REFERENSI		33

DAFTAR GAMBAR

Gambar 1.1.	Struktur Organisasi Pusat Penelitian Fakultas Ilmu Komputer UI . . .	2
Gambar 2.1.	<i>User interface</i> dari <i>website</i> https://www.tablesgenerator.com/	7
Gambar 2.2.	Keterkaitan konsep hasil studi literatur terhadap penelitian	10

DAFTAR TABEL

Tabel 2.1.	Contoh Tabel	7
Tabel 2.2.	Contoh Tabel Panjang	8
Tabel 2.3.	An Example of Rows Spanning Multiple Columns	8
Tabel 2.4.	An Example of Columns Spanning Multiple Rows	9
Tabel 2.5.	An Example of Spanning in Both Directions Simultaneously	9
Tabel 3.1.	Contoh Tabel: Data Kasus COVID-19 di Asia, 14 September 2020 . .	15
Tabel 3.2.	Contoh Tabel: Perbandingan metode pemodelan <i>access control</i>	18

DAFTAR KODE PROGRAM

Kode 3.1.	Kode sampel Java	13
Kode 3.2.	Kode sampel Python	13
Kode 3.3.	Daftar referensi di <code>references.bib</code>	20

DAFTAR LAMPIRAN

Lampiran 1. CHANGELOG	35
Lampiran 2. Judul Lampiran 2	38

BAB 1

PENDAHULUAN

Pada bab ini dijelaskan mengenai tempat pelaksanaan kerja praktik. Pertama dijelaskan mengenai proses pencarian kerja praktik. Kemudian, dijelaskan mengenai profil tempat pelaksanaan kerja praktik. Selain itu, dijelaskan juga mengenai posisi penempatan di tempat kerja praktik.

1.1 Proses Pencarian Kerja Praktik

Proses pencarian kerja praktik dimulai dengan pelaksana kerja praktik meminta izin kepada dosen pembimbing akademik untuk mengisi liburan musim panas dengan menjadi *Research Assistant* di salah satu laboratorium di Fakultas Ilmu Komputer Universitas Indonesia. Dosen pembimbing akademik mendukung penuh keputusan ini dan menekankan bahwa ini adalah cara yang produktif untuk memanfaatkan liburan dan penting bagi pengembangan keahlian teknis serta kemampuan analisis. Dengan dorongan tersebut, pelaksana kerja praktik mulai mencari laboratorium yang sesuai untuk melanjutkan penelitian.

Pelaksana kerja praktik mengajukan lamaran untuk menjadi *Research Assistant* di salah satu lab di Fakultas Ilmu Komputer UI. Proses seleksi dimulai dengan wawancara *online* yang dilakukan bersama dosen peneliti dari laboratorium yang dituju. Pada saat wawancara, dosen peneliti menjelaskan beberapa topik penelitian yang sedang berlangsung di laboratorium tersebut, termasuk topik-topik yang berfokus pada *computer networks protocols*, *high-performance computing*, dan *real-time distributed systems*. Pelaksana kerja praktik memutuskan untuk terlibat dalam penelitian terkait *real-time distributed systems*, sebuah topik yang sangat relevan dengan pengelolaan data streaming dalam skala besar.

Setelah memilih topik ini, pelaksana kerja praktik mengikuti proses administrasi lebih lanjut untuk secara resmi diterima sebagai *Research Assistant* di lab tersebut.

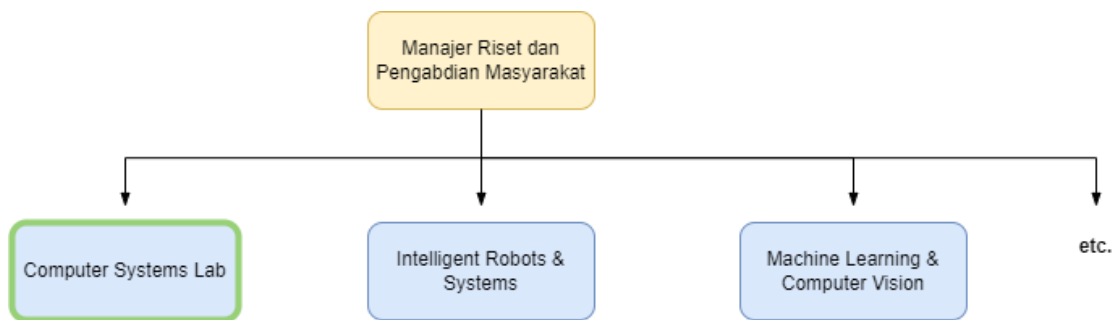
1.2 Tempat Kerja Praktik

Bagian ini menjelaskan tempat pelaksanaan kerja praktik secara rinci, meliputi profil laboratorium, struktur organisasi, dan penempatan pelaksana kerja praktik selama periode kerja praktik.

1.2.1 Profil Tempat Kerja Praktik

Kerja praktik dilaksanakan di Computer Systems Laboratory (CSL) yang berlokasi di Fakultas Ilmu Komputer, Universitas Indonesia. CSL merupakan pusat penelitian yang fokus pada pengembangan teknologi komputasi kinerja tinggi, manajemen big data, dan aplikasi berbasis cloud. Laboratorium ini mendukung berbagai penelitian di bidang teknologi informasi, mencakup topik-topik seperti manajemen cluster, sistem terdistribusi, dan optimasi performa komputasi. CSL tidak hanya menjadi tempat bagi penelitian akademik, tetapi juga berperan dalam kolaborasi lintas disiplin untuk menghasilkan inovasi di dunia teknologi.

1.2.2 Posisi Penempatan Pelaksana Kerja Praktik dalam Struktur Organisasi



Gambar 1.1: Struktur Organisasi Pusat Penelitian Fakultas Ilmu Komputer UI

Dijelaskan pada Gambar 1.1 bahwa Computer Systems Laboratory (CSL) berada di bawah supervisi Manajer Riset dan Pengabdian Masyarakat, bersama dengan beberapa pusat penelitian lainnya, seperti Intelligent Robots & Systems (IRoS) dan Machine Learning & Computer Vision (MLCV). Struktur organisasi di CSL bersifat flat, di mana setiap penelitian dilakukan secara independen oleh masing-masing tim atau individu. Namun ada dosen peneliti yang bertanggung jawab memantau seluruh proyek untuk memastikan semua penelitian berjalan sesuai standar akademik.

Pelaksana kerja praktik ditempatkan dalam tim yang mengerjakan penelitian manajemen cluster Kafka, yang berfokus pada optimasi performa sistem distribusi data real-time. Sebagai *Research Assistant*, pelaksana terlibat langsung dalam berbagai tahap eksperimen, mulai dari penyiapan infrastruktur hingga analisis hasil.

BAB 2

ISI

Pada bab ini dijelaskan secara rinci mengenai detail pekerjaan yang dilakukan oleh peserta kerja praktik selama program kerja praktik berlangsung. Latar belakang pekerjaan hingga hasil pekerjaan dijelaskan pada bab ini.

2.1 Latar Belakang Pekerjaan

Pelaksana kerja praktik terlibat dalam penelitian terkait manajemen kluster Kafka di Computer Systems Laboratory (CSL) Fakultas Ilmu Komputer Universitas Indonesia. Penelitian ini berfokus pada optimalisasi partisi topik untuk meningkatkan performa sistem streaming data real-time berbasis Kafka. Proyek ini merupakan bagian dari penelitian bertajuk *Efficient Topic Partitioning of Apache Kafka for High-Reliability Real-Time Data Streaming Applications*. Tugas pelaksana kerja praktik adalah menjalankan eksperimen yang bertujuan untuk menguji algoritma BroMax dan BroMin, yang digunakan untuk menentukan jumlah partisi optimal dalam kluster Kafka guna memaksimalkan throughput.

2.2 Deskripsi Pekerjaan

Pelaksana kerja praktik bertanggung jawab atas konfigurasi dan pengelolaan eksperimen end-to-end. Proses dimulai dengan penyiapan VM di Google Cloud Platform (GCP) dan infrastruktur Universitas Indonesia (DGX dan BMKGCS), yang digunakan untuk menjalankan simulasi kluster Kafka. Tugas utama meliputi instalasi dependencies, pengaturan Kafka broker, serta analisis hasil eksperimen untuk mengukur performa sistem, khususnya throughput dan latensi, saat algoritma BroMax dan BroMin diterapkan.

Pekerjaan dilakukan secara hybrid dari pukul 08.00 a.m. WIB hingga pukul 12.00 p.m. WIB, dengan kehadiran di lab hanya pada hari Jumat, sedangkan hari-hari lainnya bekerja secara remote. Meskipun demikian, pekerjaan tetap mengikuti jadwal sprint yang ditetapkan dalam kerangka metodologi Scrum, memastikan target pekerjaan tercapai dalam setiap periode. Hasil eksperimen dianalisis dan dilaporkan kepada dosen pembimbing untuk evaluasi lebih lanjut.

2.3 Tinjauan Pustaka

Penelitian yang dilakukan selama kerja praktik ini melibatkan sejumlah konsep penting dalam pengelolaan sistem distribusi data real-time dengan menggunakan Apache Kafka dan teknologi pendukung lainnya. Konsep-konsep ini menjadi fondasi utama dalam penelitian dan pengembangan yang dilakukan.

2.3.1 Apache Kafka

Apache Kafka adalah platform open-source yang dirancang untuk menangani streaming data secara real-time dengan throughput tinggi dan latensi rendah. Kafka memungkinkan integrasi data yang cepat dan efisien melalui model distribusi data yang skalabel. Dalam sistem ini, data dibagi menjadi topik-topik yang dipartisi, sehingga memungkinkan pemrosesan paralel yang lebih efisien di seluruh kluster (Raptis et al., 2024). Partisi pada topik Kafka sangat penting karena memungkinkan distribusi beban kerja yang merata antar broker, sehingga meningkatkan kemampuan sistem untuk menangani data dalam jumlah besar tanpa menimbulkan bottleneck (Raptis et al., 2024). Selain itu, dengan optimalisasi jumlah partisi yang tepat, throughput sistem dapat dimaksimalkan, sehingga menjaga kinerja tinggi meskipun volume data yang diproses sangat besar (Raptis et al., 2024).

2.3.2 Docker Containerization

Docker Containerization adalah teknologi yang memungkinkan aplikasi dijalankan dalam container yang terisolasi, sehingga dapat memisahkan lingkungan eksekusi aplikasi dari sistem host. Dalam konteks kerja praktik ini, Docker digunakan untuk mengelola dan menjalankan kluster Kafka pada berbagai infrastruktur, termasuk Google Cloud Platform dan sistem internal. Docker memungkinkan konsistensi lingkungan, meminimalkan konflik konfigurasi, serta meningkatkan portabilitas sistem distribusi data real-time yang digunakan selama eksperimen (Kul et al., 2024). Dengan Docker, seluruh proses deployment Kafka menjadi lebih efisien karena setiap node dalam kluster dapat dijalankan dengan konfigurasi yang seragam, mengurangi kesalahan operasional dan mempercepat penyebaran aplikasi (Kul et al., 2024).

2.3.3 Apache Zookeeper

Apache Zookeeper adalah layanan koordinasi terpusat yang digunakan untuk mengelola metadata dan menjaga konsistensi di dalam klaster Kafka. Zookeeper bertugas mengatur eleksi pemimpin partisi, menyinkronkan data antar broker, serta memastikan bahwa setiap perubahan dalam klaster Kafka ditangani dengan cepat dan efisien (Pham et al., 2014). Dalam konteks Kafka, Zookeeper memainkan peran penting dalam memastikan bahwa sistem tetap fault-tolerant. Jika terjadi kegagalan pada salah satu broker, Zookeeper secara otomatis memilih broker lain untuk menjadi pemimpin, menjaga kestabilan sistem selama proses pemrosesan data real-time berlangsung (Pham et al., 2014). Zookeeper juga menyimpan informasi konfigurasi mengenai topik dan partisi Kafka, memastikan bahwa produser dan konsumen dapat berinteraksi dengan sistem secara konsisten (Pham et al., 2014).

2.3.4 Distributed System

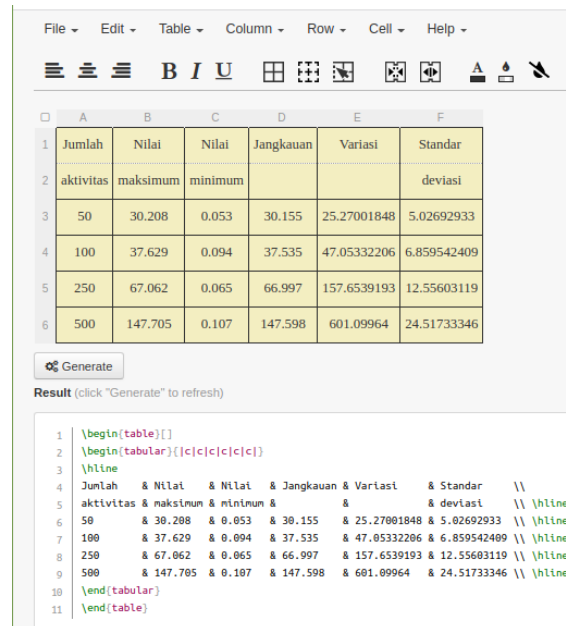
Distributed System adalah sistem yang terdiri dari beberapa node yang terhubung dan bekerja bersama untuk mencapai tujuan tertentu. Dalam konteks Kafka, sistem terdistribusi ini memastikan bahwa data dapat diolah secara paralel dan skalabel di berbagai broker yang tersebar dalam klaster. Keunggulan dari sistem terdistribusi seperti Kafka adalah kemampuannya untuk tetap berfungsi meskipun terjadi kegagalan pada satu atau beberapa node dalam klaster (Ledmi et al., 2018). Dalam sistem terdistribusi Kafka, partisi topik didistribusikan ke berbagai broker, sehingga meningkatkan ketersediaan data dan mempercepat proses pengiriman data dari produser ke konsumen (Ledmi et al., 2018). Sistem ini juga memungkinkan replikasi data secara otomatis, di mana salinan data disimpan di beberapa broker untuk meningkatkan keandalan dan redundansi sistem (Ledmi et al., 2018).

2.4 Metodologi Pekerjaan

2.5 Membuat Tabel

Tabel pada Latex dapat dibuat dengan bantuan *website* seperti <https://www.tablesgenerator.com/>. Dengan menggunakan *website* ini, maka pembuatan tabel akan menjadi lebih mudah. *User interface* dari <https://www.tablesgenerator.com/>

dapat dilihat pada Gambar 2.1.



Gambar 2.1: 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 tabel sederhana dapat dilihat pada 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
jumlah	25	30

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`. Contohnya adalah sebagai berikut:

Tabel 2.2: Contoh Tabel Panjang

	kol 1	kol 2
--	-------	-------

Tabel 2.2: Contoh Tabel Panjang (sambungan)

	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

Ada jenis tabel lain yang dapat dibuat dengan \LaTeX berikut beberapa diantaranya. Contoh-contoh ini bersumber dari <http://en.wikibooks.org/wiki/LaTeX/Tables>

Tabel 2.3: An Example of Rows Spanning Multiple Columns

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

Tabel 2.4: An Example of Columns Spanning Multiple Rows

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

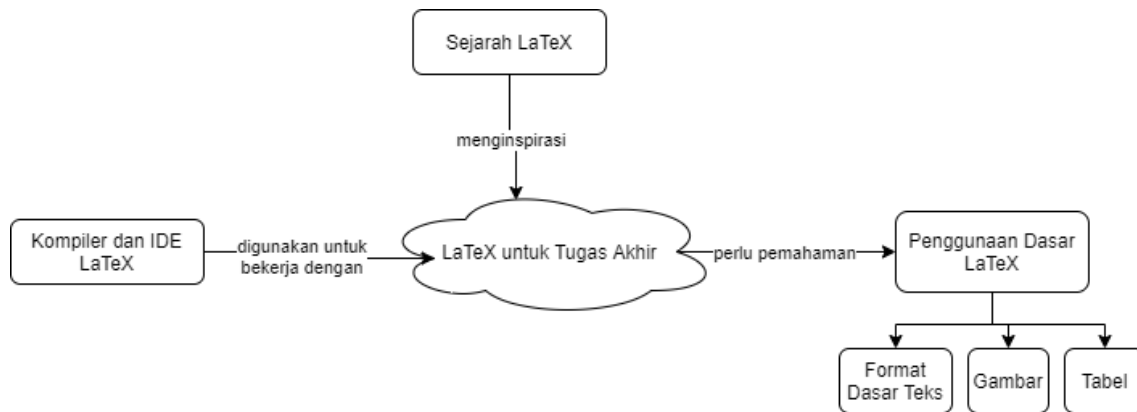
Tabel 2.5: An Example of Spanning in Both Directions Simultaneously

		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

2.6 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 2.2: Keterkaitan konsep hasil studi literatur terhadap penelitian

@todo

Jelaskan Gambar 2.2 di sini. Setiap gambar pada tugas akhir butuh penjelasan. Gambar hadir untuk mempermudah membaca memahami konteks, tetapi tidak bisa berdiri sendiri tanpa penjelasan. Terkait gambar, Anda juga bisa mengatur skalanya. Gambar kali ini lebarnya 0,8x dari lebar teks halaman.

BAB 3

PENUTUP

Bab ini menjelaskan tentang hal-hal *advanced* dalam L^AT_EX. Hal ini mencakup bagaimana cara menulis persamaan matematis di L^AT_EX, 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 Membuat Persamaan Matematis

Di L^AT_EX, 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 L^AT_EX juga perlu latihan dan lihai memahami dokumentasi.

3.1.1 Satu Persamaan

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1} \quad (3.1)$$

Persamaan 3.1 diatas adalah persamaan garis. Persamaan 3.1 dan 3.2 sama-sama dibuat dengan perintah `\align`. Perintah ini juga dapat digunakan untuk menulis lebih dari satu persamaan.

$$\underbrace{|\overline{ab}|}_{\text{pada bola } |\overline{ab}| = r} = \sqrt[2]{(x_b - x_a)^2 + (y_b - y_a)^2 + ||(z_b - z_a)^2} \quad (3.2)$$

3.1.2 Lebih dari Satu Persamaan

$$|\bar{a} * \bar{b}| = |\bar{a}| |\bar{b}| \sin \theta \quad (3.3)$$

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

Pada Persamaan 3.3 dapat dilihat beberapa baris menjadi satu bagian dari Persamaan 3.3. Sedangkan dibawah ini dapat dilihat bahwa dengan cara yang sama, Persamaan 3.4, 3.5, dan 3.6 memiliki nomor persamaannya masing-masing.

$$\int_a^b f(x) dx + \int_b^c f(x) dx = \int_a^c f(x) dx \quad (3.4)$$

$$\lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = 0 \quad \text{jika pangkat } f(x) < \text{pangkat } g(x) \quad (3.5)$$

$$a^{m^{a^n \log b}} = b^{\frac{m}{n}} \quad (3.6)$$

3.2 Mengubah Tampilan Teks

Beberapa perintah yang dapat digunakan untuk mengubah tampilan adalah:

- \f

Merupakan alias untuk perintah \textit, contoh *contoh hasil tulisan*.

- \bi

Contoh hasil tulisan.

- \bo

Contoh hasil tulisan.

- \m

Contoh hasil tulisan.

- \mc

Contoh hasil tulisan

- `\code`

Contoh hasil tulisan.

3.3 Menambahkan Kode Program

Pada \LaTeX , kode program seringkali disebut *listing*. Kita bisa memasukkan kode program (*listing*) ke dalam tugas akhir kita seperti kode Java seperti berikut:

```
1 package com.sample;
2 import java.util.List;
3
4 public interface Sample {
5     String sample();
6 }
```

Kode 3.1: Kode sampel Java

Syntax highlighting kini sudah bisa dilakukan secara otomatis oleh *library* yang ada di \LaTeX . Sudah tidak perlu lagi membuat skrip manual untuk menambahkan *syntax highlighting* sendiri. Cukup definisikan bahasa pemrograman yang digunakan, pada parameter `language=` di perintah `\lstinputlisting`.

Berikut ini adalah daftar bahasa pemrograman yang didukung *library listings*: ABAP, ACSL, Ada, Algol, Ant, Assembler, Awk, bash, Basic, C#, C++, C, Caml, Clean, Cobol, Comal, csh, Delphi, Eiffel, Elan, erlang, Euphoria, Fortran, GCL, 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, Objective C, OCL, Octave, Oz, Pascal, Perl, PHP, PL/I, Plasm, POV, Prolog, Promela, Python, R, Reduce, Rexx, RSL, Ruby, S, SAS, Scilab, sh, SHELXL, Simula, SQL, tcl, TeX, VBScript, Verilog, VHDL, VRML, XML, XSLT. (?)

Satu contoh lagi, sebuah kode bahasa pemrograman Python:

```
1 def sample(args):
2     print(args)
3     return args
```

Kode 3.2: Kode sampel Python

Anda juga bisa menambahkan *caption* untuk memberikan ringkasan tentang kode tersebut. Namun, jangan lupa untuk menjelaskan kode melalui paragraf, terutama pada bagian-bagian yang perlu penjelasan lebih. Penting bagi pembaca untuk memahami mengapa kode tersebut disertakan dalam laporan tugas akhir Anda.

3.4 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

3.5 Layoutting Tingkat Lanjut

3.5.1 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	.../IM 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	349222248
5	Pakistan	302020	539	6383	4	289806	377	5831	551	1362	29	13388	221741906
6	Turkey	291162		7056		258833		25273	1267	3445	83	100796	845222503
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	.../IM 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	.../IM 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

3.5.2 *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>

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

Kategori	Model A	Model B	Model C
Cakupan	Struktur eksplisit	Struktur eksplisit dengan <i>usage awareness</i>	Aspek-aspek keamanan generik dengan detail struktur bersifat implisit
Format diagram	<i>Class diagram</i>	<i>Use case diagram</i> dan <i>sequence diagram</i>	RBAC pada <i>activity diagram</i>

3.6 Melakukan *Cross-Reference* ke Suatu Bagian dalam Laporan

Dengan menggunakan \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 \LaTeX akan melakukannya secara otomatis. Selain itu, pada PDF yang dihasilkan oleh \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 3.6.
- `\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 3.3.

- `\label{fig:[...]}` untuk sebuah gambar.

Contoh: `\label{fig:testGambar}`

Cara referensi: `\pic~\ref{fig:testGambar}`

Hasil referensi: Gambar ??.

- `\label{tab:[...]}` untuk sebuah tabel.

Contoh: `\label{tab:Tabel1}`

Cara referensi: `\tab~\ref{tab:tab1}`

Hasil referensi: Tabel 2.1.

- Untuk sebuah kode sumber, label diletakkan sebagai argumen `\lstinputlisting` seperti: `\lstinputlisting[... , label=code:...]`.

Contoh: `\lstinputlisting[language=Python, caption=Kode sampel Python, label=code:python]`

Cara referensi: `\lst~\ref{code:python}`

Hasil referensi: Kode 3.2.

3.7 Menggunakan BibTeX

BibTeX adalah *library* dalam \LaTeX 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.

3.7.1 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 3.3.

```
1 @article{its:docker,
```

```

2   author   = {Seda Kul and Sarper Kumcu and Ahmet Sayar},
3   year     = {2024},
4   month    = {Maret},
5   title    = {Docker Container-Based Framework of Apache Kafka Node Ecosystem: Vehicle Tracking System b
6   journal   = {International Journal of Intelligent Transportation Systems Research},
7   doi       = {https://doi.org/10.1007/s13177-024-00392-6},
8   publisher = {ITS Japan},
9 }
10
11 @article{comp:zookeeper,
12   author   = {Cuong Manh Pham and Victor Dogaru and Rohit Wagle and Chitra Venkatramani and Zbigniew Kal
13   year     = {2014},
14   month    = {Maret},
15   title    = {An evaluation of zookeeper for high availability in system S},
16   journal   = {ICPE '14: Proceedings of the 5th ACM/SPEC international conference on Performance engineer
17   doi       = {http://dx.doi.org/10.1145/2568088.2576801},
18   publisher = {Association for Computing Machinery},
19 }
20
21 @article{elsevier:kafka,
22   author   = {Theofanis P. Raptis and Claudio Cicconetti and Andrea Passarella},
23   year     = {2024},
24   month    = {Januari},
25   title    = {Efficient topic partitioning of Apache Kafka for high-reliability real-time data streaming
26   journal   = {Future Generation Computer Systems},
27   doi       = {https://doi.org/10.1016/j.future.2023.12.028},
28   publisher = {Elsevier},
29 }
30
31 @article{ieee:distributed,
32   author   = {Abdeldjalil Ledmi and Hakim Bendjenna and Sofiane Mounine Hemam},
33   year     = {2018},
34   month    = {Oktober},
35   title    = {Fault Tolerance in Distributed Systems: A Survey},
36   journal   = {2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)},
37   doi       = {http://dx.doi.org/10.1109/PAIS.2018.8598484},
38   publisher = {IEEE},
39 }

```

Kode 3.3: 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.

3.7.2 Melakukan Sitasi pada Konten Tugas Akhir

Berikut ini adalah contoh kalimat yang menggunakan sitasi:

”Kalimat menurut ? terdiri dari subjek, predikat, dan objek (?).”

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. 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.

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

3.8 Daftar Isi atau Daftar Konten Lainnya

3.8.1 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*).

3.8.2 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 `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 gambar/kode/tabel/persamaan yang masuk kategori "Sesuatu".
 - `\captionsource.....`: Caption dengan sumber untuk jenis objek tersebut, contoh: `\captionsourcething` yang berfungsi sebagai *caption* dari gambar/kode/tabel/persamaan yang masuk kategori "Sesuatu", beserta dengan sumbernya.
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 \LaTeX , 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:.....}`

3.9 Memasukan PDF

Untuk memasukan 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.

Untitled

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 dimasukan dalam dokumen laporan.

3.10 Membuat Variabel atau Perintah Baru

Dalam \LaTeX , Anda bisa menambahkan variabel atau perintah baru yang dapat membantu penulisan laporan Anda. Sebenarnya variabel dalam \LaTeX merupakan perintah, namun tanpa argumen, contohnya adalah `\kucing`. Variabel dapat menyimpan suatu nilai teks. Sedangkan, suatu perintah pada \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. 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{\textit{\texttt{\#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^AT_EX 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.

3.11 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*).

3.11.1 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 `\naiveevenlearchapter` dalam berkas `thesis.tex`.

3.11.2 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:


```
\fancypagestyle{plain}{\fancyhead[L]{} \fancyhead[C]{}
\fancyhead[R]{} \fancyfoot[L]{} \fancyfoot[C]{} \fancyfoot[R]{}}
```
4. Buka berkas `thesis.tex`, lalu cari semua baris perintah `\fancypagestyle` dan `\pagestyle{...}`. Hapus semua baris perintah tersebut.

DAFTAR REFERENSI

- Kul, S., Kumcu, S., and Sayar, A. (2024). Docker container-based framework of apache kafka node ecosystem: Vehicle tracking system by license plate recognition on surveillance camera feeds. *International Journal of Intelligent Transportation Systems Research*.
- Ledmi, A., Bendjenna, H., and Hemam, S. M. (2018). Fault tolerance in distributed systems: A survey. *2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*.
- Pham, C. M., Dogaru, V., Wagle, R., Venkatramani, C., Kalbarczyk, Z., and Iyer, R. (2014). An evaluation of zookeeper for high availability in system s. *ICPE '14: Proceedings of the 5th ACM/SPEC international conference on Performance engineering*.
- Raptis, T. P., Cicconetti, C., and Passarella, A. (2024). Efficient topic partitioning of apache kafka for high-reliability real-time data streaming applications. *Future Generation Computer Systems*.

LAMPIRAN

@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.

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.