



UNIVERSITAS INDONESIA

**ANALISIS PERFORMA BERBAGAI ALGORITMA
FUNDAMENTAL DALAM JAVA, PYTHON, PHP, DAN RUST**

SKRIPSI

**ANIS FARHAN ITSNAINI
1906285541**

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

BULAN TAHUN



UNIVERSITAS INDONESIA

**ANALISIS PERFORMA BERBAGAI ALGORITMA
FUNDAMENTAL DALAM JAVA, PYTHON, PHP, DAN RUST**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer

**ANIS FARHAN ITSNAINI
1906285541**

**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 : Anis Farhan Itsnaini

NPM : 1906285541

Tanda Tangan :

Tanggal : Tanggal Bulan Tahun

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Anis Farhan Itsnaini

NPM : 1906285541

Program Studi : Ilmu Komputer

Judul Skripsi : Analisis Performa Berbagai Algoritma Fundamental
dalam Java, Python, PHP, dan Rust

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 : Dr. Ari Wibisono, S.Kom., M.Kom ()

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^AT_EX untuk membuat dokumen tugas akhir.

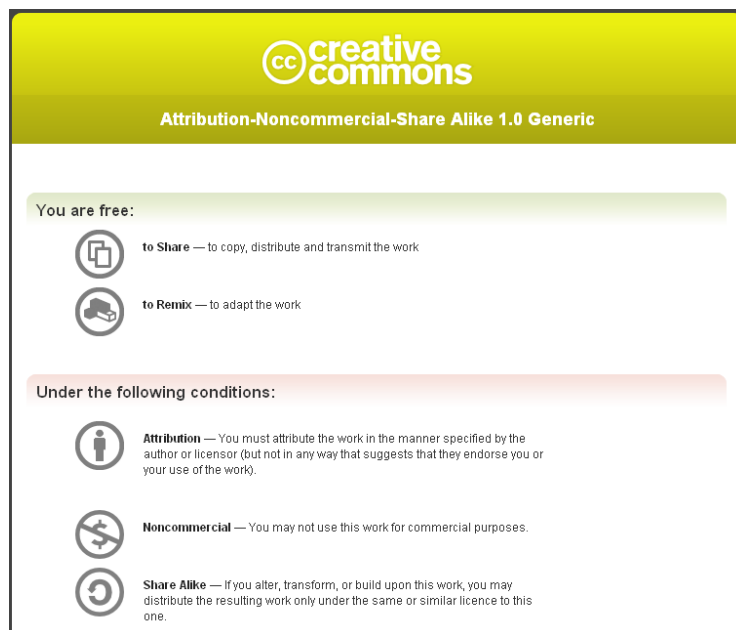
@todo

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.



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. 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 \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

Anis Farhan Itsnaini

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

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

Nama : Anis Farhan Itsnaini
NPM : 1906285541
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:

Analisis Performa Berbagai Algoritma Fundamental dalam Java, Python, PHP, dan Rust

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

(Anis Farhan Itsnaini)

ABSTRAK

Nama : Anis Farhan Itsnaini
Program Studi : Ilmu Komputer
Judul : Analisis Performa Berbagai Algoritma Fundamental dalam
Java, Python, PHP, dan Rust
Pembimbing : Dr. Ari Wibisono, S.Kom., M.Kom

Isi abstrak.

Kata kunci:

Keyword satu, kata kunci dua

ABSTRACT

Name : Anis Farhan Itsnaini
Study Program : Computer Science
Title : Performance Analysis of Various Fundamental Algorithms in Java,
Python, PHP, and Rust
Counselor : Dr. Ari Wibisono, S.Kom., M.Kom

Abstract content.

Key words:

Keyword one, keyword two

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE PROGRAM

DAFTAR LAMPIRAN

| | |
|--|----|
| Lampiran 1. CHANGELOG | 26 |
| Lampiran 2. Judul Lampiran 2 | 29 |

BAB 1

PENDAHULUAN

Pada bab ini, akan dijelaskan tentang latar belakang dan permasalahan yang diselesaikan pada penelitian ini.

1.1 Latar Belakang

- Pentingnya algoritma dalam pengembangan perangkat lunak.
- Pengaruh bahasa pemrograman terhadap performa algoritma.
- Masalah umum: bahasa berbeda = hasil performa berbeda.
- Relevansi untuk dunia industri & akademik.

1.2 Rumusan Masalah

- Bagaimana performa beberapa algoritma klasik berbeda saat diimplementasikan pada bahasa pemrograman yang berbeda?
- Apa saja faktor yang mempengaruhi performa algoritma dalam konteks pengembangan perangkat lunak?

1.3 Tujuan Penelitian

- Menganalisis dan membandingkan performa algoritma dalam bahasa pemrograman yang berbeda

1.4 Batasan Masalah

- Algoritma yang diuji: Sorting, Matrix Multiplication, BFS/DFS, Dijkstra, dll (8 total)
- Bahasa pemrograman: Java, Python, PHP, Rust
- Metrik evaluasi: waktu eksekusi, konsumsi memori (opsional: CPU usage)
- Eksperimen dilakukan dalam lingkungan sistem yang dikontrol

1.5 Manfaat Penelitian

- Memberikan referensi bagi developer/peneliti terkait pilihan bahasa dalam implementasi algoritma
- Menjadi acuan untuk pembelajaran pemrograman dan optimasi performa

1.6 Metodologi Penelitian

- Implementasi algoritma di 4 bahasa
- Uji performa pada input beragam ukuran
- Bandingkan hasil berdasarkan metrik tertentu

1.7 Sistematika Penulisan

- Bab 1: Pendahuluan
- Bab 2: Tinjauan Pustaka / Studi literatur
- Bab 3: Metodologi Penelitian
- Bab 4: Implementasi
- Bab 5: Hasil & Evaluasi

BAB 2

STUDI LITERATUR

@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".

2.1 Algoritma yang digunakan

- Sorting (Bubble Sort): logika, kompleksitas
- Matrix Multiplication: metode naive $O(n^3)$
- Graph Traversal (DFS & BFS): konsep & aplikasi
- Dijkstra: shortest path
- Knapsack/Coin Change: dynamic programming
- String Matching: naive/KMP
- Path-finding: konsep umum
- Huffman Coding: algoritma kompresi berbasis frekuensi

2.2 Bahasa Pemrograman yang digunakan

- Java: bahasa compiled, OOP kuat
- Python: interpreted, sintaks sederhana, banyak dipakai di AI
- PHP: scripting web, masih digunakan luas
- Rust: sistem modern, fokus ke performa dan safety

2.3 Studi Terdahulu (Literature Review)

- Penelitian serupa tentang perbandingan performa algoritma
- Studi benchmark bahasa pemrograman
- Relevansi penelitian dengan topik skripsi kamu

2.4 Ringkasan Literatur

- Tabel ringkasan hasil penelitian terdahulu dan posisi penelitian kamu

BAB 3

METODOLOGI PENELITIAN

Bab ini menjelaskan tahapan yang dilakukan selama proses penelitian.

3.1 Pendekatan Penelitian

Penelitian ini menggunakan pendekatan **kuantitatif dan eksperimental**. Tujuan utama dari penelitian ini adalah untuk membandingkan kinerja berbagai algoritma fundamental ketika diimplementasikan dalam bahasa pemrograman yang berbeda, yaitu Java, Python, PHP, dan Rust. Pengujian dilakukan dengan mengimplementasikan setiap algoritma dalam keempat bahasa tersebut, lalu mengukur performa eksekusinya berdasarkan metrik waktu proses (*execution time*) dan penggunaan memori (*memory usage*).

3.2 Tujuan Eksperimen

Tujuan dari eksperimen ini adalah untuk:

- perbandingan performa waktu dan memori antara bahasa pemrograman dalam menjalankan algoritma tertentu
- kelebihan dan kekurangan dari masing-masing bahasa pemrograman dalam konteks algoritma yang diuji
- Mengetahui bagaimana performa algoritma berubah tergantung bahasa pemrograman yang digunakan.
- Mengamati dampak perbedaan paradigma, interpreter/kompiler, dan sistem manajemen memori dari masing-masing bahasa terhadap eksekusi algoritma.
- Memberikan rekomendasi dan insight terkait pemilihan bahasa dalam konteks kebutuhan algoritmik tertentu.

3.3 Algoritma yang Diuji

1. **Bubble Sort** - Algoritma pengurutan sederhana.
2. **Matrix Multiplication (Naive)** - Operasi numerik pada matriks.
3. **Trapezoidal Rule** - Pendekatan numerik untuk menghitung integral.
4. **Depth-First Search (DFS)** dan **Breadth-First Search (BFS)** - Algoritma penelusuran

graf.

5. **0/1 Knapsack dan Coin Change** - Algoritma *dynamic programming*.
6. **Naive String Matching** - Pencocokan pola sederhana pada string.
7. **Dijkstras Algorithm** - Algoritma pencarian jalur terpendek.
8. **Huffman Coding** - Algoritma kompresi berbasis pohon biner.

Setiap algoritma dipilih karena mewakili kategori yang berbeda dan relevan untuk uji performa lintas bahasa pemrograman.

3.4 Bahasa Pemrograman

Empat bahasa pemrograman yang digunakan dalam penelitian ini adalah:

- **Java** - Bahasa pemrograman berorientasi objek dengan tipe statis dan dikompilasi ke dalam bytecode.
- **Python** - Bahasa scripting tingkat tinggi yang dinamis dan diinterpretasi.
- **PHP** - Bahasa scripting server-side yang umum digunakan untuk pengembangan web.
- **Rust** - Bahasa pemrograman sistem yang modern, dikompilasi, dan berorientasi pada keamanan memori.

Pemilihan keempat bahasa ini bertujuan untuk membandingkan berbagai paradigma (compiled vs interpreted, static vs dynamic typing) serta keunggulan dan kelemahan masing-masing dalam konteks implementasi algoritma.

3.5 Tools dan Lingkungan Pengujian

Eksperimen dilakukan dalam lingkungan yang terkontrol, dengan spesifikasi sebagai berikut:

- perangkat keras

@todo

Harusnya GCP

Segera minta Pak Ari !!!

- perangkat lunak:

- bahasa dan versi
- editor dan alat bantu

3.6 Skenario Pengujian

Pembuatan *testcase* dilakukan dengan menggunakan *random generator*. Pengukuran akan dilakukan terhadap:

- waktu eksekusi
- penggunaan memori

3.7 Pengumpulan dan Analisis Data

Data yang diperoleh dari hasil eksekusi akan dicatat dalam tabel. Selanjutnya akan divisualisasikan dalam bentuk grafik untuk memudahkan analisis. Data dibandingkan untuk melihat tren performa antar bahasa.

Langkah - langkah analisis:

1. Menghitung rata-rata waktu eksekusi dari setiap algoritma dalam setiap bahasa.
2. Menyajikan hasil dalam bentuk tabel dan grafik (*bar chart / line chart*).
3. Melakukan interpretasi hasil, khususnya jika ada hasil yang tidak konsisten atau menunjukkan perbedaan signifikan.

BAB 4

IMPLEMENTASI

Bab ini menjelaskan tentang implementasi dari kode-kode yang dipilih pada tugas akhir ini.

@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 Sorting - Bubble Sort

4.1.1 Deskripsi Algoritma

Bubble Sort merupakan algoritma pengurutan sederhana yang efektif untuk dataset kecil. Algoritma ini bekerja dengan cara melakukan iterasi berulang pada larik data dan menukar pasangan elemen bersebelahan jika berada dalam urutan yang salah. Proses ini diulang hingga seluruh elemen berada dalam urutan yang benar. Kompleksitas waktu dalam kasus terburuk dan rata-rata adalah $O(N^2)$, sedangkan kompleksitas ruang adalah $O(1)$.

4.1.2 Pseudocode

```
procedure bubbleSort(arr, n):  
  for i from 0 to n-1:  
    for j from 0 to n-i-1:  
      if arr[j] > arr[j+1]:  
        swap arr[j] and arr[j+1]
```

4.1.3 Contoh Input-Output

Input:

```
6  
3 2 1 5 6 4
```

Output:

```
1 2 3 4 5 6
```

4.2 Matrix Multiplication - Naive**4.2.1 Deskripsi Algoritma**

Matrix Multiplication adalah proses mengalikan dua matriks untuk menghasilkan matriks baru. Jika matriks A berukuran $m \times n$ dan matriks B berukuran $n \times p$, maka hasil kali matriks $C = A * B$ akan berukuran $m \times p$. Proses ini melibatkan penjumlahan produk elemen baris dari matriks A dengan elemen kolom dari matriks B. Kompleksitas waktu untuk metode naive adalah $O(N^3)$, sedangkan kompleksitas ruang adalah $O(N^2)$.

4.2.2 Pseudocode

```
function matrixMultiply(A, B):
    for i from 0 to A.rows-1:
        for j from 0 to B.cols-1:
            C[i][j] = 0
            for k from 0 to A.cols-1:
                C[i][j] += A[i][k] * B[k][j]

    return C
```

4.2.3 Contoh Input-Output**Input:**

```
matrix A size:
2 3
matrix A:
1 2 3
4 5 6
```

```

matrix B size:
3 2
matrix B:
1 1
1 2
2 3

```

Output:

```

result matrix:
9 14
21 32

```

4.3 Integral Modeling - Trapezoidal Rule Integration

4.3.1 Deskripsi Algoritma

Trapezoidal Rule adalah metode numerik untuk menghitung integral dari fungsi. Metode ini bekerja dengan membagi area di bawah kurva menjadi beberapa trapezoid, kemudian menjumlahkan luas trapezoid tersebut untuk mendapatkan perkiraan nilai integral. Kompleksitas waktu dari metode ini adalah $O(n)$, di mana n adalah jumlah subinterval (dengan asumsi perhitungan nilai fungsi dilakukan dalam $O(1)$), dan kompleksitas ruangnya adalah $O(1)$.

4.3.2 Pseudocode

```

function trapezoidalRule(f, a, b, n):
    h = (b - a) / n
    sum = f(a) + f(b)
    for i from 1 to n-1:
        sum += 2 * f(a + i * h)
    return sum * h / 2

```

4.3.3 Contoh Input-Output

Input:

10

```
f(x) = x^2  
a = 0  
b = 1  
n = 1000
```

Output:

```
0.333333
```

4.4 Graph Traversal - Depth-First Search (DFS)

4.4.1 Deskripsi Algoritma

Depth-First Search (DFS) adalah algoritma pencarian yang digunakan untuk menjelajahi graf atau pohon. DFS bekerja dengan cara menjelajahi sejauh mungkin di sepanjang cabang sebelum melakukan backtrack. Algoritma ini menggunakan struktur data stack (atau rekursi) untuk melacak jalur yang sedang dieksplorasi. Kompleksitas waktu DFS adalah $O(V E)$, di mana V adalah jumlah vertex dan E adalah jumlah edge. Kompleksitas ruangnya adalah $O(V)$ untuk menyimpan status kunjungan.

4.4.2 Pseudocode

```
procedure DFS(graph, start):  
    create a stack S  
    push start onto S  
    mark start as visited  
  
    while S is not empty:  
        vertex = pop S  
        for each neighbor of vertex:  
            if neighbor is not visited:  
                mark neighbor as visited  
                push neighbor onto S
```


4.4.3 Contoh Input-Output

Pada contoh ini algoritma DFS digunakan untuk menghitung luas pulau dari koordinat (x, y) yang diberikan. Dimana 1 menandakan daratan dan 0 menandakan lautan. **Input:**

```
grid size:
5 5
grid:
1 1 0 0 0
1 1 0 1 0
0 0 0 1 1
0 0 1 1 1
1 1 0 0 0
start point:
1 1
```

Output:

```
area of island: 4
```

4.5 Dynamic Programming - Coin Change

4.5.1 Deskripsi Algoritma

Contoh masalah algoritma dynamic programming diberikan dalam bentuk Coin Change Problem. Masalah ini bertujuan untuk menemukan jumlah minimum koin yang dibutuhkan untuk mencapai nilai tertentu dengan denominasi koin yang diberikan. Kompleksitas waktu adalah $O(N \cdot M)$, di mana N adalah jumlah nilai yang ingin dicapai dan M adalah jumlah jenis koin. Kompleksitas ruangnya adalah $O(N)$.

4.5.2 Pseudocode

```
function coinChange(coins, amount):
    dp = array of size (amount + 1) initialized to infinity
    dp[0] = 0
```

```

for i from 1 to amount:
    for each coin in coins:
        if i - coin >= 0:
            dp[i] = min(dp[i], dp[i - coin] + 1)

return dp[amount] if dp[amount] != infinity else -1

```

4.5.3 Contoh Input-Output

Input:

```

coins:
1 3 4
amount:
6

```

Output:

```

minimum number of coins: 2

```

4.6 String Matching - Naive

4.6.1 Deskripsi Algoritma

Naive String Matching adalah algoritma yang digunakan untuk mencari substring dalam string utama dengan cara membandingkan setiap karakter dari substring dengan karakter yang sesuai dalam string utama. Algoritma ini bekerja dengan iterasi melalui string utama dan memeriksa kecocokan karakter satu per satu. Kompleksitas waktu dalam kasus terburuk adalah $O(N \cdot M)$, di mana N adalah panjang string utama dan M adalah panjang substring. Kompleksitas ruangnya adalah $O(1)$.

4.6.2 Pseudocode

```

function stringMatch(text, pattern):

```

```

n = length of text
m = length of pattern
for i from 0 to n - m:
    j = 0
    while j < m and text[i + j] == pattern[j]:
        j += 1
    if j == m:
        return i // found match at index i
return -1 // no match found

```

4.6.3 Contoh Input-Output

Input:

```

text: "hello world"
pattern: "world"

```

Output:

```

match found at index: 6

```

4.7 Path-Finding Algorithm - Dijkstra's Algorithm

4.7.1 Deskripsi Algoritma

Tuliskan penjelasan singkat tentang algoritma tersebut: cara kerja, kompleksitas waktu/ruang jika perlu, dan alasan penggunaannya secara umum. Algoritma Dijkstra adalah algoritma yang digunakan untuk menemukan jalur terpendek dari satu vertex ke semua vertex lainnya dalam graf berbobot. Algoritma ini bekerja dengan cara menginisialisasi jarak dari simpul awal ke semua simpul lain sebagai tak hingga, kemudian secara iteratif memperbarui jarak terpendek yang ditemukan. Kompleksitas waktu Algoritma Dijkstra adalah $O(V E \log V)$ dengan menggunakan priority queue, di mana V adalah jumlah vertex dan E adalah jumlah edge. Kompleksitas ruangnya adalah $O(V E)$.

4.7.2 Pseudocode

```

function dijkstra(graph, start):
    dist = array of size V initialized to infinity
    dist[start] = 0
    priorityQueue = min-heap containing (0, start)

    while priorityQueue is not empty:
        (currentDist, currentVertex) = extract-min(priorityQueue)
        for each neighbor of currentVertex:
            newDist = currentDist + weight(currentVertex, neighbor)
            if newDist < dist[neighbor]:
                dist[neighbor] = newDist
                update priorityQueue with (newDist, neighbor)

    return dist

```

4.7.3 Contoh Input-Output

Input:

```

vertex count: 5
edge count: 6
edges: // (u, v, weight)
1 2 2
1 3 4
2 3 1
2 4 7
3 5 3
4 5 1
start vertex: 1

```

Output:

```

shortest distances from source vertex 1:

```

```

vertex 1: 0
vertex 2: 2
vertex 3: 3
vertex 4: 7
vertex 5: 6

```

4.8 Data Compression - Huffman Coding

4.8.1 Deskripsi Algoritma

Tuliskan penjelasan singkat tentang algoritma tersebut: cara kerja, kompleksitas waktu/ruang jika perlu, dan alasan penggunaannya secara umum. Huffman Coding adalah algoritma kompresi data yang digunakan untuk mengurangi ukuran data dengan cara menggantikan karakter yang sering muncul dengan kode biner yang lebih pendek. Algoritma ini membangun Huffman tree berdasarkan frekuensi kemunculan karakter dalam data. Kompleksitas waktu untuk membangun pohon Huffman adalah $O(N \log N)$, di mana N adalah jumlah karakter unik, dan kompleksitas ruangnya adalah $O(N)$.

4.8.2 Pseudocode

```

function huffmanCoding(frequencies):
    create a priority queue Q
    for each character and frequency in frequencies:
        insert (frequency, character) into Q

    while Q has more than one element:
        (freq1, char1) = extract-min(Q)
        (freq2, char2) = extract-min(Q)
        newNode = createNode(freq1 + freq2, char1 + char2)
        insert newNode into Q

    return root of the Huffman tree

procedure generateCodes(node, code, codes):
    if node is a leaf:

```

```
        codes[node.character] = code
    return
    generateCodes(node.left, code + '0', codes)
    generateCodes(node.right, code + '1', codes)
```

4.8.3 Contoh Input-Output

Input:

text to compress: "aaaabbbccd"

Output:

Huffman Codes:

'a': 0

'b': 10

'c': 111

'd': 110

Compressed text:

"0000101010111111110"

BAB 5

HASIL EVALUASI

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^AT_EX 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 ?? dan cara melakukan *cross-reference* dengan melihat Subbab ?. Kedua fitur tersebut merupakan fitur yang sangat penting dalam penulisan skripsi menggunakan L^AT_EX.
 - Jika fakultas Anda memerlukan format sitasi selain APA (yang menjadi *default* di tingkat universitas), silakan baca Subbab ?.
 - Jika Anda membutuhkan support untuk selain tulisan alfabet, silakan baca Subbab ?. Jika Anda membutuhkan penulisan notasi matematis, silakan baca Subbab ?. Jika Anda membutuhkan penulisan kode program, silakan baca Subbab ?.
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 L^AT_EX 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 Merdeka¹ 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 Bangkit²), 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`.

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

²<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 ISTILAH

L^AT_EX 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.