

# Analisis Pandangan Relevansi Tugas Besar

## III IF2211 Strategi Algoritma terhadap

### Silabusnya



Naufarrel Zhafif Abhista

13523149

# Pendahuluan

Laporan ini menyajikan analisis terhadap "Tugas Besar 3" (selanjutnya disebut Tubes 3) untuk mata kuliah IF2211 Strategi Algoritma, sebagaimana yang dirinci dalam dokumentasi spesifikasi tahun 2021. Evaluasi ini bertujuan untuk membedah struktur tugas, persyaratan teknis, dan kriteria penilaian untuk mengukur efikasi pendidikannya. Analisis ini disusun dalam tiga bagian utama: pertama, pengujian keselarasan tugas dengan kurikulum formal dan capaian pembelajaran; kedua, penilaian kuantitatif dan kualitatif terhadap beban kerjanya berdasarkan standar 3 SKS (Satuan Kredit Semester); dan ketiga, evaluasi desain tugas untuk kolaborasi kelompok yang efektif dan adil. Tinjauan ini akan dibingkai melalui berbagai lensa, termasuk teori kurikulum, praktik rekayasa perangkat lunak, dan lanskap teknologi kontemporer, terutama pengaruh kecerdasan buatan generatif.

# Daftar Isi

<b>Analisis Pandangan Relevansi Tugas Besar III IF2211 Strategi Algoritma terhadap Silabusnya...</b>	<b>1</b>
Pendahuluan.....	2
Daftar Isi.....	3
Bagian 1	
Analisis Relevansi Kurikulum dan Keselarasan Capaian Pembelajaran.....	4
1.1. Pemetaan Langsung Konsep Inti ke Silabus.....	4
1.2. Pemenuhan Capaian Pembelajaran yang Ditetapkan.....	4
1.3 Sintesis Pengetahuan Prasyarat.....	5
Tabel 1: Pemetaan Persyaratan Tugas Terhadap Capaian Pembelajaran IF2211.....	6
Bagian 2	
Penilaian Kuantitatif dan Kualitatif Beban Kerja Tugas.....	8
2.1 Dekonstruksi Komponen Proyek.....	8
2.2 Perbandingan dengan Standar SKS.....	9
2.3 Analisis Faktor Ekstrinsik yang Mempengaruhi Beban Kerja.....	9
Bagian 3	
Evaluasi.....	11
3.1. Modularitas Arsitektur dan Paralelisasi.....	11
3.2. Potensi Ketidakeimbangan Beban Kerja dan Strategi Mitigasi.....	11
3.3 Ketergantungan Antar Modul dan Tantangan Integrasi.....	11
Kesimpulan.....	12

## Bagian 1

# Analisis Relevansi Kurikulum dan Keselarasan Capaian Pembelajaran

Bagian ini mengevaluasi sejauh mana tugas-tugas inti dan teknologi yang dipersyaratkan dalam proyek ini dapat melayani tujuan pembelajaran yang telah ditetapkan untuk mata kuliah IF2211, sebagaimana diuraikan dalam silabus.

### 1.1. Pemetaan Langsung Konsep Inti ke Silabus

Spesifikasi tugas menunjukkan keselarasan yang jelas dan tidak ambigu dengan kurikulum mata kuliah. Silabus IF2211 secara eksplisit mencantumkan "*String matching + regular expression* (Regex)" sebagai topik ke-11 dalam "Lingkup Bahasan". Sejalan dengan itu, spesifikasi Tubes 3 mewajibkan penggunaan "algoritma pencocokan string KMP, Boyer-Moore, dan Regex" untuk implementasi *backend* aplikasi.

Hubungan langsung ini memastikan bahwa mahasiswa dinilai berdasarkan materi yang telah diajarkan secara formal, yang merupakan landasan dari asesmen pendidikan yang valid. Proyek ini bukan tugas yang arbitrer, melainkan sebuah aplikasi yang ditargetkan untuk memberikan pengalaman praktis pada unit studi yang spesifik dan terencana. Lebih jauh, proyek ini mendorong mahasiswa untuk melampaui pemahaman teoretis semata dengan menuntut mereka untuk mengimplementasikan dan menerapkan algoritma-algoritma ini untuk memecahkan masalah yang nyata dan relevan bagi mereka, yakni mengelola tenggat waktu akademik melalui sebuah *chatbot*.

### 1.2. Pemenuhan Capaian Pembelajaran yang Ditetapkan

Desain tugas ini secara holistik dirancang untuk memenuhi keempat Capaian Pembelajaran (Student Outcome) yang diadaptasi dari kriteria ABET, sebagaimana tercantum dalam silabus. Keempat capaian tersebut adalah: (1) Menganalisis masalah komputasi yang kompleks dan menerapkan prinsip-prinsip komputasi untuk mengidentifikasi solusi; (2) Merancang, mengimplementasikan, dan mengevaluasi solusi berbasis komputasi untuk memenuhi serangkaian kebutuhan komputasi; (3) Berkomunikasi secara efektif dalam berbagai konteks profesional; dan (4) Menerapkan teori ilmu komputer dan fundamental pengembangan perangkat lunak untuk menghasilkan solusi berbasis komputasi.

Tugas ini berhasil menyentuh setiap capaian tersebut. Merancang pola *Regex* dan logika *chatbot* secara keseluruhan untuk memahami input bahasa alami secara langsung menjawab Capaian 1. Membangun aplikasi web *full-stack* dari awal hingga akhir adalah perwujudan dari Capaian 2. Kewajiban untuk menulis laporan lima bab yang komprehensif, membuat *README.md* yang jelas, dan memproduksi video demonstrasi secara langsung melatih Capaian 3. Terakhir,

keseluruhan proses implementasi algoritma KMP/Boyer-Moore dari prinsip-prinsip teoretis menjadi kode yang berfungsi adalah aplikasi nyata dari Capaian 4. Hubungan sistematis antara persyaratan tugas dan capaian pembelajaran ini dirinci dalam Tabel 1.

### 1.3 Sintesis Pengetahuan Prasyarat

Mata kuliah ini mensyaratkan mahasiswa telah mengambil IF2120 Matematika Diskrit dan IF2110 Algoritma dan Struktur Data. Desain Tubes 3 secara implisit menuntut mahasiswa untuk memanfaatkan fondasi pengetahuan ini. Tugas inti untuk mengimplementasikan KMP dan Boyer-Moore secara langsung dibangun di atas konsep analisis algoritma, struktur data (seperti larik dan tabel), dan analisis kompleksitas yang diajarkan di IF2110. Untuk IF2120 Matematika Diskrit, belum terlihat hubungan jelas antara tugas besar ini dengan mata kuliah tersebut.

Selain itu, persyaratan untuk memilih antara mekanisme persistensi data berbasis basis data atau berbasis berkas teks kemungkinan besar terhubung dengan mata kuliah tentang basis data. Hal ini menjadikan proyek ini sebagai integrator pengetahuan yang efektif di seluruh kurikulum semester 4, mendorong mahasiswa untuk melihat bagaimana berbagai domain ilmu komputer saling berhubungan dalam penciptaan produk perangkat lunak yang fungsional.

Mata kuliah ini, meskipun berjudul "Strategi Algoritma", mencakup banyak persyaratan yang berada dalam domain rekayasa perangkat lunak. Spesifikasi tugas menuntut pembangunan aplikasi web, pembuatan program yang modular, penulisan README.md yang detail, kepatuhan pada struktur folder tertentu, dan mandat bahwa semua anggota kelompok memahami seluruh basis kode.

Secara esensial, meskipun fokus eksplisit IF2211 adalah pada paradigma algoritmik seperti *Greedy*, *Dynamic Programming*, dan *Backtracking*, proyek Tubes 3 ini memaksa mahasiswa untuk terlibat dalam siklus hidup pengembangan perangkat lunak yang lengkap dalam skala kecil. Mereka harus melakukan analisis kebutuhan (memahami spesifikasi), perancangan (arsitektur *chatbot*), implementasi (pengkodean), pengujian (memastikan semua fitur berfungsi), dan dokumentasi (laporan, README). Aktivitas-aktivitas ini secara langsung memetakan pada Capaian Pembelajaran ABET untuk merancang, mengimplementasikan, dan mengevaluasi solusi, serta berkomunikasi secara efektif.

Dengan demikian, cakupan pedagogis tugas ini lebih luas dari yang tersirat pada judul mata kuliahnya. Tugas ini berfungsi sebagai "kurikulum tersembunyi" untuk rekayasa perangkat lunak, menyediakan jembatan praktis yang krusial antara mata kuliah algoritma teoretis dan proyek pengembangan perangkat lunak yang lebih terapan di semester-semester saati itu, maupun berikutnya.

**Tabel 1: Pemetaan Persyaratan Tugas Terhadap Capaian Pembelajaran IF2211**

Capaian Pembelajaran	Persyaratan Tugas Terkait	Justifikasi dan Penjelasan	Bukti Pemuenuhan
Menganalisis masalah komputasi yang kompleks dan menerapkan prinsip-prinsip komputasi untuk mengidentifikasi solusi.	Mendeteksi perintah pengguna yang tidak <i>exact matching</i> ; mengekstrak tanggal, kode mata kuliah, dan topik dari kalimat bahasa alami menggunakan <i>Regex</i> ; merancang logika untuk berbagai kueri (misalnya, "deadline 3 minggu ke depan").	Mahasiswa harus memecah masalah "memahami input pengguna" menjadi sub-masalah yang dapat diselesaikan secara komputasi. Ini memerlukan analisis mendalam tentang struktur kalimat dan perancangan pola <i>Regex</i> yang kompleks untuk mengekstrak informasi secara akurat.	Kode implementasi modul <i>parser</i> dan logika <i>Regex</i> ; kemampuan aplikasi untuk menangani berbagai format input seperti yang dicontohkan dalam spesifikasi.
Merancang, mengimplementasikan, dan mengevaluasi solusi berbasis komputasi untuk memenuhi serangkaian kebutuhan komputasi.	"Aplikasi yang dibuat berbasis web (wajib)"; "Program harus modular"; "Membuat suatu <i>database</i> sederhana atau menyimpannya dalam bentuk struktur data sendiri"; implementasi 8 fitur wajib.	Mahasiswa harus merancang arsitektur lengkap aplikasi web, mulai dari <i>frontend</i> hingga <i>backend</i> dan persistensi data. Mereka kemudian mengimplementasikannya menggunakan tumpukan teknologi pilihan dan secara implisit mengevaluasi fungsionalitasnya terhadap 8 fitur yang disyaratkan.	Kode sumber yang diserahkan; aplikasi web yang berfungsi; struktur folder dan modularitas program; skema basis data atau format berkas teks.

Berkomunikasi secara efektif dalam berbagai konteks profesional.	"Laporan (soft copy)" dengan 5 bab yang ditentukan (Deskripsi, Teori, Analisis, Implementasi, Kesimpulan); "README selengkap mungkin"; (Bonus) "membuat video aplikasi".	Persyaratan dokumentasi yang ketat memaksa mahasiswa untuk menjelaskan proses teknis mereka (analisis, desain, implementasi) secara jelas dan terstruktur kepada audiens (asisten, dosen). Ini mensimulasikan praktik dokumentasi profesional dalam industri perangkat lunak.	Dokumen laporan yang diserahkan; berkas README.md di repositori; video demonstrasi di YouTube (jika dikerjakan).
Menerapkan teori ilmu komputer dan fundamental pengembangan perangkat lunak untuk menghasilkan solusi berbasis komputasi.	"Aplikasi (backend) harus menggunakan algoritma pencocokan <i>string</i> KMP, Boyer-Moore, dan Regex"; "Pencocokan <i>string</i> ... tidak dilakukan secara <i>exact matching</i> "; (Bonus) "memanfaatkan <i>Levenshtein distance</i> ".	Mahasiswa harus menerjemahkan konsep teoretis algoritma <i>string matching</i> (misalnya, tabel LPS untuk KMP, tabel <i>bad-character</i> untuk Boyer-Moore) dari buku teks atau materi kuliah menjadi implementasi kode yang berfungsi dan mengintegrasikannya ke dalam sistem yang lebih besar.	Implementasi fungsi-fungsi algoritma KMP dan Boyer-Moore dalam kode sumber; penggunaan Regex untuk validasi dan ekstraksi data; kebenaran fungsional dari fitur pencarian dan deteksi perintah.

## Bagian 2

### Penilaian Kuantitatif dan Kualitatif Beban Kerja Tugas

Bagian ini akan menyajikan rincian terperinci tentang perkiraan beban kerja proyek untuk menentukan kesesuaiannya dengan nilai kredit 3 SKS yang dialokasikan.

#### 2.1 Dekonstruksi Komponen Proyek

Proyek ini menuntut implementasi 8 fitur inti, antarmuka pengguna berbasis web, logika *backend* dengan algoritma spesifik (KMP, BM, Regex), persistensi data, laporan komprehensif, dan beberapa fitur bonus. Estimasi waktu yang terperinci dapat dipecah menjadi beberapa fase:

- **Implementasi Algoritma Inti (12 jam)**  
Meneliti dan mengimplementasikan KMP dan Boyer-Moore dari awal, termasuk membangun tabel pra-pemrosesan yang diperlukan (misalnya, larik LPS, tabel *bad-character/good-suffix*). Bagian ini sulit secara konseptual.
- **Logika Regex dan Parsing (4 jam)**  
Merancang, menulis, dan menguji ekspresi reguler yang kompleks yang diperlukan untuk mengurai input bahasa alami untuk tanggal, kode mata kuliah, jenis tugas, dan perintah.
- **Backend dan Logika Aplikasi (12 jam)**  
Menyiapkan kerangka kerja web (misalnya, Flask, Django), membuat *endpoint* API, mengelola status aplikasi, dan mengintegrasikan modul algoritma/parsing.
- **Pengembangan Frontend (4 jam)**  
Membuat antarmuka obrolan yang sederhana namun fungsional menggunakan HTML/CSS/JavaScript. Meskipun bukan fokus utama, ini tetap membutuhkan upaya yang tidak sepele.
- **Persistensi Data (4 jam)**  
Mengimplementasikan skema basis data sederhana atau mekanisme pemuatan/penyimpanan berkas.
- **Pengujian dan Integrasi (2 jam)**  
Memastikan semua komponen bekerja sama dengan lancar dan menguji dengan berbagai input pengguna seperti yang ditunjukkan dalam contoh.
- **Dokumentasi dan Pelaporan (8 jam)**  
Menulis laporan terperinci sesuai spesifikasi, membuat tangkapan layar, dan mempersiapkan demo.

Berdasarkan rincian ini, **total estimasi beban kerja** untuk proyek ini berada di kisaran **40–50 jam**.

#### 2.2 Perbandingan dengan Standar SKS

Di ITB, 1 SKS setara dengan:

- 50 menit pembelajaran sinkron



- 50 menit pembelajaran asinkron
- 50 menit pembelajaran mandiri

yang dicapai dalam kurun waktu satu minggu. Ini berarti total upaya yang diharapkan per minggu untuk 3 SKS adalah 450 menit. Untuk alokasi tugas besar, kita asumsikan 50 menit pembelajaran sinkron tidak dihitung. Sehingga, untuk satu minggu, terdapat 300 menit alokasi pengerjaan tugas besar per minggu, alias 5 jam per minggu. Dengan waktu pengerjaan 3 minggu, diasumsikan terdapat 15 jam waktu untuk mengerjakannya. Namun, ada kalanya periode kekosongan tugas besar (yang terjadi di awal semester, sekitar tiga minggu). Dengan begitu, karena terdapat tiga tugas besar, satu minggu tersebut dapat “dipindahkan”/dialokasikan untuk pengerjaan tugas besar. Sehingga, total waktu untuk mengerjakan tugas besar ini adalah **20 jam**. Jelas, ini sangat kurang dari 40 jam. Hal ini menunjukkan bahwa (a) proyek ini dimaksudkan untuk menjadi yang paling menuntut di antara ketiganya, atau (b) beban kerja keseluruhan untuk komponen proyek mata kuliah ini sangat tinggi.

## 2.3 Analisis Faktor Ekstrinsik yang Mempengaruhi Beban Kerja

Penilaian beban kerja tidak dapat dilepaskan dari konteks teknologi dan keterampilan prasyarat. Ada dua faktor utama yang secara signifikan mengubah perhitungan beban kerja aktual yang dialami oleh mahasiswa. Pertama, terdapat *overhead* teknologi tambahan yang tidak secara langsung terkait dengan tujuan pembelajaran inti. Sebagian besar dari estimasi beban kerja (Pengembangan *Backend* dan *Frontend*, sekitar 30–40 jam) melibatkan teknologi pengembangan web (seperti Flask, JavaScript, dll.) yang bukan merupakan bagian dari kurikulum "Strategi Algoritma" IF2211. Tujuan pembelajaran utama adalah algoritma itu sendiri: KMP, Boyer-Moore, dan aplikasi *Regex*. Namun, untuk mendemonstrasikan pemahaman mereka, mahasiswa pertama-tama harus mengatasi rintangan teknis dalam membangun aplikasi web yang fungsional—sebuah **keahlian yang diasumsikan tetapi tidak diajarkan oleh mata kuliah ini**. Hal ini menciptakan "pajak teknologi" atau beban kognitif. Mahasiswa mungkin menghabiskan waktu yang tidak proporsional untuk men-debug konfigurasi server atau *event listener* JavaScript, yang mengalihkan fokus mereka dari konsep algoritmik inti. Terdapat risiko pedagogis berupa **disonansi**, yakni ketika upaya yang diinvestasikan oleh mahasiswa tidak selaras sempurna dengan hasil pembelajaran yang dituju dari mata kuliah spesifik ini. Keterampilan praktis yang diperoleh memang berharga untuk pendidikan mereka secara keseluruhan, tetapi mengurangi fokus pada strategi algoritma dalam konteks IF2211.

Kedua, dan yang lebih transformatif, adalah keberadaan kecerdasan buatan (AI) generatif. Tugas ini dirancang pada tahun 2021, sebelum adopsi luas alat-alat AI canggih seperti ChatGPT dan GitHub Copilot. Seorang mahasiswa saat ini akan menggunakan *Large Language Model* (LLM) untuk menghasilkan seluruh kode *boilerplate* aplikasi web (pengaturan Flask/Django, HTML/CSS/JS untuk antarmuka obrolan, kode koneksi basis data). Ini dapat

mengurangi "overhead teknologi tambahan" dari 12 jam menjadi kurang dari 3 jam. Hal ini secara dramatis mengubah perhitungan beban kerja, berpotensi membawa total upaya lebih dekat ke kisaran 40–50 jam yang diharapkan, membuatnya lebih sesuai dengan nilai SKS-nya.

Namun, keberadaan AI adalah pedang bermata dua. Alat yang sama juga dapat menghasilkan implementasi algoritma KMP dan Boyer-Moore yang nyaris sempurna, serta merancang pola *Regex* yang diperlukan. Ini menciptakan tantangan integritas akademik yang signifikan, karena memungkinkan mahasiswa untuk melewati proses belajar sama sekali. Oleh karena itu, metode penilaian menjadi sangat krusial. Penekanan pada "pemahaman terhadap cara kerja program" yang menyumbang 25% dari nilai, serta adanya demo langsung di hadapan asisten, menjadi benteng pertahanan yang sangat penting untuk memastikan bahwa AI digunakan sebagai alat produktivitas, bukan sebagai pengganti pembelajaran.

## Bagian 3

### Evaluasi

Bagian ini akan menganalisis arsitektur dan ruang lingkup proyek untuk menilai kesesuaiannya untuk tim yang terdiri dari 2–3 mahasiswa.

#### 3.1. Modularitas Arsitektur dan Paralelisasi

Desain proyek secara alami dapat dipecah menjadi beberapa modul yang berbeda: (1) Mesin Pencocokan String (KMP/BM), (2) *Parser* Perintah berbasis *Regex*, (3) *API Backend/Server Web*, (4) Antarmuka Pengguna *Frontend*, dan (5) Lapisan Persistensi Data.

Modularitas ini adalah kekuatan utama untuk kerja kelompok. Sebuah tim yang terdiri dari tiga orang dapat bekerja secara paralel dengan efektif:

- **Mahasiswa A (Spesialis Algoritma)** fokus pada implementasi algoritma KMP dan Boyer-Moore dalam modul yang mandiri.
- **Mahasiswa B (Spesialis Backend/Logika)** menyiapkan server web, mendefinisikan API, menulis logika aplikasi inti, dan merancang pola *Regex*.
- **Mahasiswa C (Spesialis Frontend/Integrasi)** mengembangkan antarmuka obrolan dan bekerja dengan Mahasiswa B untuk mengintegrasikannya dengan *API backend*.

Alur kerja paralel ini memungkinkan penggunaan waktu yang efisien dan memanfaatkan kekuatan individu masing-masing anggota tim.

#### 3.2. Potensi Ketidakseimbangan Beban Kerja dan Strategi Mitigasi

Meskipun modular, tugas-tugas dalam proyek ini tidak setara dalam sifat atau tingkat kesulitannya. Mengimplementasikan KMP/Boyer-Moore (tugas Mahasiswa A) sangat menantang secara konseptual tetapi menghasilkan jumlah kode yang relatif kecil; ini membutuhkan pemahaman teoretis yang mendalam. Membangun *API backend* dan logika bisnis (tugas Mahasiswa B) memiliki kompleksitas sedang dan melibatkan jumlah pengkodean dan pengujian berbagai kasus yang signifikan. Sementara itu, membangun antarmuka pengguna *frontend* (tugas Mahasiswa C) seringkali kurang sulit secara konseptual tetapi bisa sangat memakan waktu, melibatkan pekerjaan HTML/CSS/JS yang berulang.

#### 3.3 Ketergantungan Antar Modul dan Tantangan Integrasi

Sistem yang dibangun adalah satu kesatuan. *Frontend* bergantung pada *API backend*, yang pada gilirannya bergantung pada modul algoritma dan *parsing*. Titik integrasi yang paling kritis adalah kontrak API antara *frontend* dan *backend*. Kelompok harus menyepakati struktur permintaan dan respons sejak awal. Hal ini menuntut komunikasi dan perencanaan, yang merupakan keterampilan lunak (*soft skills*) berharga yang diperkuat oleh proyek ini.

Fase integrasi akhir, di mana semua modul disatukan, sering kali menjadi tempat ditemukannya *bug* tersembunyi dan ketidakcocokan. Ini memberikan pengalaman belajar yang realistis dan berharga dalam pengembangan perangkat lunak, meniru tantangan yang dihadapi dalam proyek-proyek dunia nyata.

## Kesimpulan

**Dalam hal relevansi kurikuler,** tugas ini menunjukkan keselarasan. Tugas ini tidak hanya secara langsung menguji pemahaman materi silabus yang spesifik—yaitu *string matching* dan *regular expression*—tetapi juga berfungsi sebagai proyek integratif yang memaksa mahasiswa untuk menerapkan pengetahuan dari mata kuliah prasyarat. Lebih penting lagi, tugas ini secara efektif menyematkan "kurikulum tersembunyi" dalam rekayasa perangkat lunak, memberikan pengalaman praktis dalam siklus hidup pengembangan perangkat lunak dan secara komprehensif memenuhi semua capaian pembelajaran yang ditetapkan.

**Mengenai beban kerja,** analisis menunjukkan bahwa proyek ini sangat menuntut, dengan estimasi jam kerja yang melebihi alokasi standar untuk porsi tugas besar dalam mata kuliah 3 SKS. Ditambah lagi dengan beban kognitif akibat tidak adanya pengajaran mengenai *web development* pada semester tersebut, yang berpotensi menyebabkan disonansi. Namun, beban kerja yang tinggi ini dapat diredam secara signifikan oleh penggunaan alat AI generatif modern, yang dapat mengotomatiskan sebagian besar pekerjaan pengembangan web tambahan. Hal ini menciptakan dinamika baru di mana tantangan utama bergeser dari implementasi teknis ke integritas akademik. Keberhasilan tugas ini dalam mencapai tujuannya di era AI sangat bergantung pada ketatnya proses penilaian, terutama demo langsung dan evaluasi pemahaman konseptual.

**Dari perspektif kolaborasi,** proyek ini dirancang dengan baik. Strukturnya yang modular memungkinkan pembagian kerja yang efisien dan pengembangan secara paralel. Meskipun ada risiko ketidakseimbangan beban kerja karena sifat tugas yang berbeda (konseptual vs. implementasi), kebijakan yang mengharuskan setiap anggota untuk memahami seluruh proyek berfungsi sebagai mekanisme mitigasi yang kuat, mendorong komunikasi dan pembelajaran bersama.

Secara keseluruhan, Tugas Besar 3 IF2211 adalah instrumen pembelajaran yang baik. Meskipun menantang dan berpotensi membebani, tugas ini berhasil menjembatani kesenjangan antara teori algoritma abstrak dan aplikasi rekayasa perangkat lunak praktis.