

RESEARCH ARTICLE

## Pengembangan Backend untuk Efisiensi Pengelolaan dan Penyebaran Informasi di Himpunan Mahasiswa Teknik Komputer

Jaisy Malikulmulki Arasy, Anggunmeka Luhur Prasasti\* and Astri Novianty

Fakultas Teknik Elektro, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia

\*Corresponding author: [anggunmeka@telkomuniversity.ac.id](mailto:anggunmeka@telkomuniversity.ac.id)

### Abstrak

Dalam era digital, teknologi informasi memiliki peran yang sangat signifikan dalam menunjang berbagai aspek kehidupan, termasuk di lingkungan akademik. Himpunan Mahasiswa Teknik Komputer (HMTK) di Telkom University menghadapi tantangan dalam pengelolaan dan penyebaran informasi melalui media sosial yang kurang efisien, seperti kurangnya konsistensi dalam penyebaran informasi terbaru, kesulitan menjangkau seluruh anggota mahasiswa, dan penyebaran informasi yang tersebar di berbagai *platform* tanpa integrasi yang baik. Oleh karena itu, diperlukan solusi berupa aplikasi My-HMTK yang berfokus pada pengembangan aplikasi yang dapat mendukung keterlibatan mahasiswa dalam kegiatan himpunan. Aplikasi ini dirancang untuk mengintegrasikan berbagai informasi terkait kegiatan, acara, dan perkembangan jurusan dalam satu *platform* yang terstruktur dan mudah diakses. Dengan demikian, diharapkan dapat meningkatkan keterlibatan dan komunikasi antar mahasiswa secara signifikan, yang memiliki dampak positif pada peningkatan penyebaran informasi, keterikatan sesama mahasiswa, dan tujuan akademis. Selain itu, komunikasi yang lebih efektif dan terpusat melalui aplikasi ini akan memastikan bahwa informasi penting dapat disampaikan dengan cepat dan tepat kepada seluruh mahasiswa. Dampaknya, penyebaran informasi menjadi lebih efisien dan akurat, mengurangi risiko informasi yang tidak sampai atau terlambat diterima.

**Key words:** Teknologi Informasi, Integrasi Informasi, Partisipasi Akademik, Pengelolaan Informasi.

### Pendahuluan

Himpunan mahasiswa adalah sebuah organisasi yang terdiri dari sekelompok mahasiswa. Himpunan Mahasiswa Teknik Komputer (HMTK) di Telkom University, Bandung, berfungsi sebagai wadah bagi mahasiswa program studi Teknik Komputer untuk memperoleh pengalaman berorganisasi. Tujuannya adalah untuk mengembangkan mahasiswa menjadi individu yang mampu menyeimbangkan pengetahuan dan keterampilan. Organisasi ini juga memberikan kesempatan bagi mahasiswa untuk meningkatkan kemampuan kepemimpinan, kerja sama, dan komunikasi, yang dapat bermanfaat bagi perkembangan pribadi dan kemampuan sosial mereka. Jurusan Teknik Komputer di Telkom University merupakan bagian dari Fakultas Teknik Elektro. Fokus studi di jurusan ini meliputi teknologi komputer, perangkat keras, dan perangkat lunak. Cakupan materinya termasuk perancangan, pengembangan, dan manajemen sistem komputer dalam berbagai aplikasi. Seiring dengan pertumbuhan populasi komputer, perkembangan teknologi informasi, khususnya komputer dan perangkatnya, akan berdampak pada penggunaannya [1].

Di era modern ini, kita dituntut untuk memahami dan bijak dalam menggunakan teknologi, terutama media sosial. Media sosial telah menjadi *platform* utama untuk komunikasi dan pencarian informasi. Definisi dari media sosial adalah media online yang memungkinkan

penggunaannya untuk berpartisipasi, berbagi, dan membuat konten seperti blog pribadi, jejaring sosial, wiki, forum, dan dunia virtual [2]. Hal ini memfasilitasi pertukaran informasi yang efisien antar pengguna, baik secara individu maupun kelompok. Jejaring sosial juga berfungsi sebagai ruang bagi komunitas dengan minat dan keahlian serupa untuk saling terhubung [3]. Untuk meningkatkan partisipasi mahasiswa dalam kegiatan akademik dan non-akademik, media sosial dapat menjadi solusi cepat sebagai wadah komunitas mahasiswa. Data menunjukkan bahwa dari sekitar 143 juta pengguna internet di Indonesia, hampir 90% adalah pengguna media sosial [2].

### Tinjauan Pustaka

#### Spesifikasi Produk

Terdapat beberapa dasar penentuan spesifikasi pengembangan aplikasi untuk mendukung keterlibatan mahasiswa dalam kegiatan Jurusan Teknik Komputer. Penentuan tujuan utama dari aplikasi terfokus pada meningkatkan keterlibatan mahasiswa dalam kegiatan yang relevan dengan jurusan seperti kemudahan akses, antarmuka pengguna yang intuitif dan fungsionalitas yang mendukung kegiatan akademis dan non akademis mahasiswa juga menjadi prioritas. Selanjutnya,

dalam menentukan spesifikasi teknis, penting untuk mempertimbangkan *platform* atau lingkungan tempat aplikasi akan diimplementasikan, misalnya, apakah aplikasi ini akan berbasis web, *mobile*, atau keduanya. Keamanan dan integritas data juga harus diintegrasikan secara cermat, mengingat aplikasi akan menangani informasi mahasiswa dan kegiatan jurusan. Selain itu, spesifikasi kinerja seperti respon-time, kapasitas pengguna, dan skalabilitas aplikasi perlu diperhatikan untuk memastikan bahwa aplikasi dapat berjalan dengan optimal ketika digunakan oleh sejumlah besar mahasiswa.

Dalam hal fungsionalitas, aplikasi sebaiknya mencakup fitur-fitur yang dapat meningkatkan keterlibatan mahasiswa, seperti jadwal kegiatan, pengumuman, forum diskusi, dan kemungkinan integrasi dengan *platform* pembelajaran online yang mungkin digunakan oleh mahasiswa Teknik Komputer. Pendekatan pengembangan perangkat lunak seperti metode pengujian yang terencana dan iteratif juga harus dijelaskan secara rinci untuk memastikan kualitas dan keberhasilan implementasi aplikasi. Dasar penentuan spesifikasi ini akan menjadi landasan yang kokoh untuk merancang dan mengembangkan aplikasi yang memenuhi kebutuhan mahasiswa Teknik Komputer dengan keterlibatan mahasiswa.

### Platform

*Platform* merujuk pada suatu kerangka kerja atau infrastruktur yang menyediakan lingkungan untuk pengembangan dan pelaksanaan aplikasi atau layanan. Secara umum, *platform* memberikan fondasi bagi pengguna atau pengembang untuk membangun, mengintegrasikan, dan mengelola berbagai jenis perangkat lunak atau aplikasi. *Platform* menciptakan kerangka kerja yang konsisten dan efisien untuk pengembangan aplikasi, memfasilitasi inovasi dan interaksi yang lebih baik. Contoh *platform* melibatkan Android dan iOS untuk perangkat *mobile*, serta AWS atau Microsoft Azure untuk layanan komputasi awan.

### Platform Web

*Platform* informasi dan aplikasi website sudah semakin besar di masa kini. Kemudahan akses dan keringanan beban penggunaan menjadi salah satu alasan banyaknya teknologi ini digunakan dari berbagai kalangan. Di aplikasi yang menggunakan website, pengguna dapat mengaksesnya di mana pun sementara *software* memerlukan instalasi terlebih dahulu [4]. Dalam pengembangannya, web sendiri terdiri dari backend dan *front-end* sebagai dua pembagian utama. Pada pengembangan *back-end*, diatur mengenai bagaimana informasi dapat berkomunikasi antar server atau basis data terhadap *front-end* dengan membuat *endpoint-endpoint* yang dapat dipanggil. Salah satu contoh *framework* untuk pengembangan *back-end* adalah FastAPI yang merupakan salah satu *framework* yang bersifat *open source* untuk membangun aplikasi web berbasis API (*Application Programming Interface*) secara cepat dengan menggunakan bahasa pemrograman *Python*.

Pada sisi pengembangan *front-end*, terdapat tampilan antarmuka bagaimana web akan dilihat oleh penggunaannya seperti navigasi, desain, tombol, grafik, dan sebagainya. Secara sederhana, *front-end* hanya terdiri dari HTML (*HyperText Markup Language*) dan CSS (*Cascading Style Sheets*), namun terdapat juga *framework-framework* yang dapat mendukung pengembangan. Salah satu *framework* yang dapat mendukung pengembangan ini adalah TailwindCSS yang merupakan sebuah proyek *open source* yang menawarkan kemudahan dalam menampilkan CSS dengan simpel serta mendukung website yang dinamis dan responsif.

### Sistem Manajemen Basis Data Relasional

Perangkat lunak DBMS (*Database Management System*) merupakan salah satu *platform* yang digunakan untuk melakukan pengelolaan dan pengaturan dalam suatu basis data [5]. Salah satu sistem DBMS yang

banyak digunakan adalah yang bertipe relasional di mana sistem ini memungkinkan penyimpanan data dalam sebuah tabel yang terdiri dari baris dan kolom. Penggunaan RDBMS mencakup berbagai sektor, termasuk keuangan, kesehatan, pendidikan, dan pemerintahan. Sistem ini memanfaatkan SQL (*Structured Query Language*) sebagai bahasa standar untuk menjalankan operasi seperti pembuatan, pembacaan, pembaruan, dan penghapusan data. SQL memungkinkan pengguna untuk mengeksekusi query yang kompleks, menggabungkan tabel, serta memastikan integritas data melalui penggunaan kunci primer dan kunci asing. Beberapa contoh perangkat lunak DBMS relasional yang banyak digunakan, seperti :

### PostgreSQL

PostgreSQL dikenal dengan dukungannya untuk berbagai tipe data yang kompleks, serta kemampuan dalam menangani operasi dengan efisiensi tinggi dan menjaga integritas data melalui transaksi ACID (*Atomicity, Consistency, Isolation, Durability*). PostgreSQL sering digunakan dalam aplikasi yang memerlukan skalabilitas dan keandalan tinggi. Dukungan komunitas yang aktif dan terus berkembang juga memungkinkan PostgreSQL untuk terus diperbarui dengan fitur-fitur terbaru dan perbaikan keamanan.

### MySQL

MySQL mendukung berbagai sistem operasi, termasuk Windows, Linux, dan macOS, serta dapat diintegrasikan dengan berbagai bahasa pemrograman, seperti PHP, *Python*, dan Java. Fitur replikasi MySQL memungkinkan data untuk disalin ke beberapa server, yang meningkatkan ketersediaan dan pemulihan bencana. Selain itu, MySQL memiliki arsitektur yang modular dan dapat diperluas dengan *plugin*, memungkinkan pengembang untuk menyesuaikan fungsionalitas sesuai dengan kebutuhan spesifik.

### SQLite

SQLite adalah RDBMS yang ringan dan *self-contained*, yang menyimpan semua data dalam satu file basis data. SQLite sering digunakan dalam aplikasi *mobile*, *embedded systems*, dan aplikasi *desktop* karena ukurannya yang kecil dan kemampuannya untuk beroperasi tanpa memerlukan server terpisah. SQLite tidak memerlukan konfigurasi yang rumit, dan memiliki *footprint* yang kecil, menjadikannya *ideal* untuk aplikasi dengan sumber daya terbatas. Dari masing-masing contoh RDBMS yang sering digunakan pada aplikasi, berikut merupakan perbandingan untuk mendukung pengembangan aplikasi.

### Verifikasi

Dalam menjalani proses pengukuran dan verifikasi spesifikasi secara rinci untuk memastikan keselarasan antara rencana pengembangan aplikasi dan tujuan yang hendak dicapai, fokus utama terletak pada peningkatan keterlibatan mahasiswa dalam kegiatan Program Studi Teknik Komputer. Spesifikasi yang cermat, termasuk frekuensi partisipasi, jenis kegiatan, dan dampak yang diantisipasi, akan diukur dengan metode yang terperinci, seperti analisis data prototipe dan survei.

### Pengukuran Back-end

*Back-end* adalah pusat kontrol data dari sistem yang menghubungkan server dengan antarmuka dengan API. Salah satu aplikasi yang dapat digunakan untuk melakukan testing adalah Postman. Aplikasi ini menyediakan environment yang terpusat dan mudah digunakan untuk menguji, mendokumentasikan, dan berkolaborasi pada API. Dengan Postman, penggunaannya dapat dengan mudah mengirim permintaan HTTP ke API, menguji respons yang diterima, memeriksa header, parameter, dan *body* dari setiap respons, serta mengelola dan menyimpan

**Table 1.** Huruf Aksara Lontara Tertinggi dan Terendah

RDBMS	Kelebihan	Keterbatasan
PostgreSQL	a) Dukungan untuk tipe data kompleks b) Efisiensi tinggi dalam operasi c) Transaksi ACID (Atomicity, Consistency, Isolation, Durability) d) Replikasi dan sharding e) Dukungan komunitas yang aktif f) Project yang <i>open source</i>	a) Kurva belajar yang lebih tinggi b) Memerlukan lebih banyak waktu dan sumber daya untuk konfigurasi dan pemeliharaan
MySQL	a) Dukungan berbagai tipe penyimpanan (InnoDB, MyISAM) b) Integrasi dengan berbagai bahasa pemrograman c) Dukungan komunitas dan ekosistem yang luas	a) Keterbatasan dalam dukungan untuk transaksi kompleks dan pemrosesan paralel b) Beberapa fitur SQL yang mungkin tidak tersedia
SQLite	a) Ringan dan selfcontained b) Tidak memerlukan konfigurasi yang rumit c) Footprint kecil d) Mendukung sebagian besar fitur SQL standar	a) Kinerja lebih rendah b) Dukungan terbatas untuk operasi multipengguna c) Tidak ada manajemen pengguna

koleksi permintaan untuk digunakan ulang. Fitur pengujian otomatisnya memungkinkan pembuatan skrip pengujian yang kompleks, mempercepat pengujian fungsionalitas API secara menyeluruh. Postman juga menyediakan visualisasi yang jelas terhadap data yang dikirim dan diterima, mempermudah pemahaman dan *debugging*, serta memfasilitasi kolaborasi tim dalam mengembangkan API yang optimal dan responsif.

Untuk melakukan testing menggunakan Postman, dapat dilakukan beberapa permintaan seperti GET, POST, PUT, DELETE, sesuai dengan *endpoint* API yang telah dibuat. Kemudian tentukan parameter yang diperlukan untuk permintaan tersebut, dari *query parameters*, *headers*, maupun *body request*. Setelah pengiriman permintaan, *respons* yang diterima dari API dapat dievaluasi dengan memeriksa *status code*, data yang dikirim balik, dan header yang terkait.

## Metodologi Penelitian

### Desain Sistem

Dalam sub bab analisis dan pemilihan solusi untuk pengembangan aplikasi yang mendukung keterlibatan mahasiswa dalam kegiatan Program Studi Teknik Komputer, parameter-parameter kritis seperti tingkat kesesuaian, biaya, waktu, kehandalan, dan sumber daya manusia, alat, serta bahan harus diperhatikan. Tingkat kesesuaian menjadi fokus utama untuk memenuhi kebutuhan pengguna, sementara evaluasi biaya dan waktu mendukung keefisienan implementasi. Keandalan sistem menjadi parameter krusial untuk memastikan kelancaran keterlibatan mahasiswa. Mekanisme pemilihan solusi dapat dilakukan melalui metode *design matrix scoring* atau *engineering design matrix*, di mana solusi terpilih dapat berasal dari alternatif terbaik yang mencapai skor tertinggi atau melalui penggabungan beberapa alternatif solusi yang saling melengkapi, memastikan bahwa solusi yang diambil memenuhi seluruh kriteria dan spesifikasi yang telah ditetapkan.

### Back-end Framework

Dalam pemilihan *back-end framework*, perlu dipertimbangkan kemampuan *framework* dalam menyediakan struktur dasar untuk menangani permintaan HTTP, akses basis data, keamanan, dan manajemen sesi, sehingga mendukung pengembangan aplikasi yang efisien dan handal.

### Parameter

Analisis pada *framework back-end* ditujukan untuk melakukan perbandingan mendalam antara solusi-solusi dengan beberapa parameter berikut:

#### Kecepatan Respons (Response Time)

Parameter ini mengukur seberapa cepat *framework* mampu memberikan *respons* terhadap permintaan HTTP. Hal ini penting untuk mengevaluasi kinerja dan responsifitas dari aplikasi yang dibangun.

#### Penggunaan Memori (Memory Usage)

Parameter ini berguna dalam mengevaluasi efisiensi pengelolaan sumber daya oleh setiap *framework*. Melalui pengukuran penggunaan memori saat aplikasi berjalan, analisis ini memberikan gambaran tentang penggunaan memori setiap *framework*.

#### Dokumentasi (Documentation)

Kualitas dokumentasi resmi *framework* dan ukuran komunitas di sekitar *framework*. Hal ini mencakup kejelasan dokumentasi serta kemudahan akses untuk mendapatkan bantuan, tutorial, dan informasi lainnya.

#### Kesederhanaan Penggunaan (Ease of Use)

Menilai seberapa mudah untuk menggunakan, mempelajari, dan mengimplementasikan *framework* dalam proyek ini.

#### Mekanisme

Analisis pada *framework back-end* ditujukan untuk melakukan perbandingan mendalam antara solusi-solusi dengan beberapa mekanisme berikut:

#### Parameter Kecepatan Respons

Pada parameter ini, penilaian kecepatan ditentukan dengan menjalankan masing-masing *framework* dengan kode minimal untuk memberikan *respons 'Hello World!'* dengan metode GET. Masing-masing *framework* dijalankan secara bersamaan dan pada *port* yang berbeda-beda. *Framework* FastAPI berjalan pada port 8000, *Flask* berjalan

pada port 5000, sedangkan *framework* Django berjalan pada port 3000.

Untuk menentukan waktu secara terstruktur, kami menggunakan aplikasi Postman untuk memberikan nilai waktu yang ditentukan untuk melakukan request pada masing-masing *framework* di port-nya. Aplikasi ini dapat melakukan testing dengan melakukan *request* pada API dengan memberikan parameter-parameter yang diperlukan dan akan menghasilkan status *respons*, isi *respons*, dan juga waktu yang diperlukan untuk menerima *respons*.

### Parameter Penggunaan Memori

Parameter ini dievaluasi dengan mengimplementasikan kode yang mencari proses yang terkait dengan setiap *framework* yang berjalan pada port tertentu. Kode ini menggunakan modul *psutil* untuk mengidentifikasi proses-proses terkait dengan aplikasi yang berjalan pada masing-masing *port*. Penggunaan memori pada proses yang ditemukan diukur dengan menggunakan fungsi untuk mengambil informasi ukuran memori yang digunakan oleh proses tersebut. Skrip ini memberikan gambaran tentang seberapa efisien setiap *framework* dalam penggunaan memori saat dijalankan pada lingkungan yang sama.

### Parameter Dokumentasi

Penilaian untuk parameter dokumentasi didasarkan pada analisis laman dokumentasi dari setiap *framework*. Fitur-fitur yang dievaluasi mencakup tingkat kemudahan pemahaman, keberadaan komunitas yang aktif, ketersediaan contoh kode yang jelas, kelengkapan informasi, serta sumber daya terkait yang tersedia. Evaluasi ini melibatkan peninjauan setiap bagian dari dokumentasi, memperhatikan sejauh mana informasi disajikan, se jelas apa penjelasannya, kehadiran komunitas yang aktif untuk mendukung pengguna, serta ketersediaan contoh kode dan sumber daya lainnya yang dapat membantu pengguna memahami dan menggunakan *framework* tersebut.

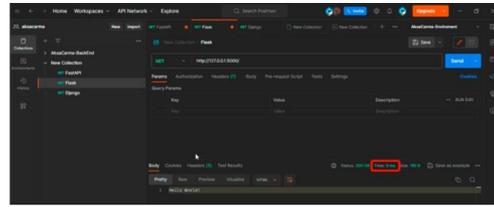
### Parameter Kesederhanaan Penggunaan

Parameter kesederhanaan penggunaan dievaluasi melalui penggunaan langsung dari setiap *framework*. Pengujian ini dilakukan dengan mencoba mengimplementasikan fitur-fitur dasar serta fungsionalitas yang umum digunakan pada masing-masing *framework*. Evaluasi mencakup pengamatan atas fitur-fitur yang menjadi kelebihan dalam penggunaan, seperti kemudahan implementasi, struktur yang terstruktur dengan baik, dan kejelasan dalam penulisan kode. Kelemahan dari masing-masing *framework* juga diamati seperti kompleksitas dalam penggunaan, kurangnya struktur yang sudah tersedia secara bawaan, serta kemungkinan tantangan yang mungkin dihadapi oleh pengguna baru saat menggunakan *framework* tersebut.

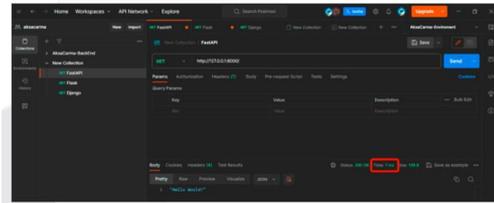
## Hasil dan Pembahasan

Pada pengukuran parameter kecepatan *respons* yang dibaca pada Postman, didapatkan hasil sebagai berikut.

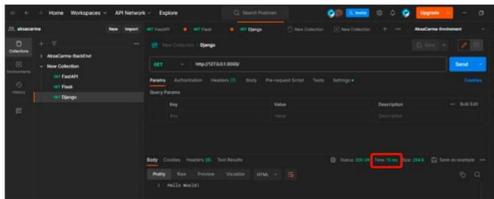
- Gambar 1 merupakan contoh pengukuran waktu kecepatan *respons* pada *Framework* FastAPI.
- Gambar 2 merupakan contoh pengukuran waktu kecepatan *respons* pada *Framework* Flask.
- Gambar 3 merupakan contoh pengukuran waktu kecepatan *respons* pada *Framework* Django.
- Gambar 4 merupakan hasil output skrip penggunaan memori pada *Framework* FastAPI, Flask, dan Django.



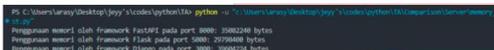
Gambar 1. Pengukuran Waktu Kecepatan *Respons* FastAPI



Gambar 2. Pengukuran Waktu Kecepatan *Respons* Flask



Gambar 3. Pengukuran Waktu Kecepatan *Respons* Django



Gambar 4. Hasil Output Skrip Penggunaan Memori *Framework*

Tabel 2 merupakan perbandingan *Back-end Framework* pada *Framework* FastAPI, Flask, dan Django.

### Matrix Scoring

*Matrix scoring* merujuk pada suatu metode penilaian atau evaluasi yang menggunakan matriks atau tabel untuk memberikan skor atau nilai pada berbagai elemen atau kriteria tertentu. Dari hasil analisis *matrix scoring* di atas, setiap *framework*, yakni FastAPI, Flask, dan Django, memiliki keunggulan dan kelemahan masing-masing dalam parameter-parameter yang dievaluasi. FastAPI mendominasi dalam Kecepatan *Respons* dan Penggunaan Memori dengan skor tertinggi, menunjukkan kinerja yang sangat baik dalam kedua aspek tersebut, sementara itu, Flask menunjukkan keseimbangan yang baik antara Kecepatan *Respons*, Penggunaan Memori, dan Kesederhanaan Penggunaan dengan skor yang kompetitif. Django, di satu sisi, menunjukkan skor yang cukup baik dalam Dokumentasi, namun menunjukkan tantangan dalam Penggunaan Memori dan Kesederhanaan Penggunaan. Dengan *matrix scoring* ini, tergambar dengan jelas bahwa setiap *framework* memiliki kelebihan dan kekurangan, yang bisa menjadi pertimbangan utama dalam pemilihan *framework* yang sesuai dengan kebutuhan proyek yang sedang dikerjakan. Setelah dievaluasi dengan kebutuhan yang paling sesuai dengan aplikasi ini, FastAPI berada sebagai pilihan terbaik. Alasannya FastAPI menunjukkan performa yang unggul dalam kecepatan dan penggunaan memori. Dokumentasinya yang jelas serta kemudahan penggunaannya juga menjadi kelebihan yang signifikan.

Table 2. Perbandingan *Back-end Framework*

Parameter	FastAPI	Flask	Django
Kecepatan Respons	7 ms	9 ms	15 ms
Penggunaan Memori	35.082.240 bytes	29.798.400 bytes	39.604.224 bytes
Dokumentasi	Kelebihan: a) Dokumentasi yang jelas b) Komunitas yang berkembang c) Pembaruan yang cepat dan konsisten d) Contoh kode dan hasil running e) Terdapat fitur live chat f) Tersedia tutorial dan sumber daya Kekurangan: Belum banyak <i>framework</i> lain	Kelebihan: a) Dokumentasi yang baik b) Informasi setiap fungsi yang lengkap c) Komunitas yang besar d) Tersedia tutorial yang tersusun dengan rapi sehingga agak sulit dibaca	Kelebihan: a) Dokumentasi yang luas b) Komunitas yang kuat c) Tersedia tutorial dan sumber daya Kekurangan: Kompleksitas yang lebih tinggi
Kesederhanaan Penggunaan	Kelebihan: a) Deklaratif dan jelas b) Dokumentasi otomatis c) Pembuatan sederhana dapat dilakukan hanya dalam 5 baris Kekurangan: d) Membutuhkan pengalman <i>Python</i> lebih terkait penggunaan <i>decorators</i> dan tipe data	Kelebihan: a) Ringkas dan minimalis b) Fleksibilitas yang tinggi Kekurangan : Kurang nya struktur bawaan sehingga terkadang memerlukan upaya ekstra dalam menetapkan struktur yang jelas untuk proyek yang besar dan kompleks.	Kelebihan: a) Memiliki banyak fitur bawaan dan struktur yang sudah terdefinisi dengan baik b) ORM yang kuat Kekurangan: a) Kompleksitas yang lebih tinggi b) Suatu proyek dapat membuat hingga belasan file hanya untuk tugas sederhana

Table 3. *Matrix Scoring*

Framework	Kecepatan Respons	Penggunaan Memo	Dokumentasi	Kesederhanaan Penggunaan	Rerata
FastAPI	9	9	8	7	8,25
Flask	8	7	7	8	7,5
Django	6	5	7	5	5,75

Berdasarkan hasil evaluasi ini, keputusan diambil untuk menggunakan FastAPI sebagai *framework* utama dalam pengembangan aplikasi ini.

### Implementasi Sistem

Dalam rancangan solusi ini, beberapa alternatif usulan solusi dipertimbangkan untuk mengoptimalkan pengembangan aplikasi yang mendukung keterlibatan mahasiswa dalam kegiatan Program Studi Teknik Komputer. Wujud akhir solusi dari implementasi yang diberikan berupa *software* yang melibatkan beberapa tahapan yang penting untuk mewujudkan *prototype* solusi secara riil. Pengujian dilakukan secara menyeluruh untuk memastikan bahwa aplikasi berfungsi sesuai yang diharapkan, mulai dari fungsionalitas dasar hingga kinerja yang responsif dan efisien. Pengujian unit, integrasi, dan uji coba fungsional dilakukan secara berulang untuk mengidentifikasi dan memperbaiki bug atau masalah lainnya. Proses pengembangan dilakukan dengan mengimplementasikan logika, pemrosesan data, dan interaksi antara *front-end* dan *back-end*.

### Back-End

*Back-end* adalah bagian dari sebuah aplikasi atau sistem yang tidak langsung diakses oleh pengguna tetapi berfungsi sebagai otak dari aplikasi tersebut atau bisa disebut proses perancangan *system* yang dilakukan di belakang layar dari sebuah website ataupun aplikasi [6]. *Back-end* mengelola data, memproses permintaan dari *front-end*,

dan mengatur logika bisnis dari sebuah aplikasi. Ini mencakup *database*, server, dan semua proses yang terjadi di belakang layar untuk memastikan aplikasi berjalan dengan baik.

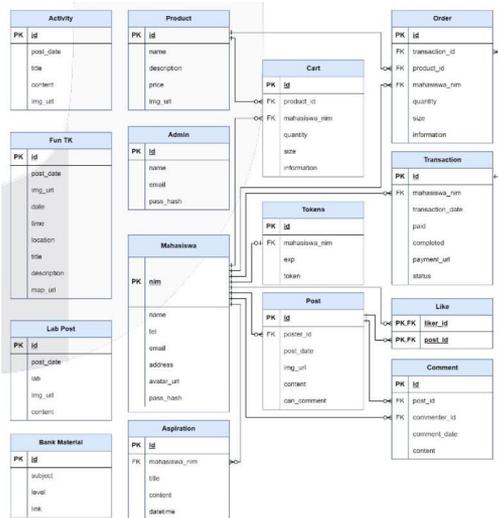
### FastAPI

FastAPI adalah sebuah *framework back-end* yang digunakan untuk membangun aplikasi web menggunakan bahasa pemrograman *Python*. *Framework* ini terkenal karena kombinasi kecepatan eksekusi tinggi, validasi data yang ketat berkat pemanfaatan *type hints*, serta sistem dokumentasi otomatis yang sangat baik. FastAPI banyak digunakan untuk membangun aplikasi yang membutuhkan kinerja tinggi dan validasi data yang ketat dalam lingkungan yang efisien.

### Analisis Pengerjaan Implementasi Sistem

#### Basis Data

Aplikasi ini memanfaatkan implementasi dari manajemen basis data PostgreSQL dengan membuat tabel relasi dengan banyak entitas. Pendekatan ini memungkinkan penyimpanan data secara terstruktur dan pengelolaan yang efisien melalui kueri yang kompleks. Aplikasi dapat mengatur informasi terkait satu sama lain, memudahkan pencarian, pembaruan, dan penghapusan data dengan menggunakan tabel relasi. Dukungan dari manajemen basis data yang memperkuat kinerja dan skalabilitas aplikasi, memberikan fleksibilitas dan persiapan untuk skala yang lebih besar. Keterhubungan antara entitas atau relasi antar entitas



Gambar 5. Diagram Entity Relationship

dapat digambarkan dengan diagram *entity* relationship seperti berikut. Basis data ini terdiri dari empat belas entitas yang dibuat untuk mendukung pengembangan aplikasi dari semua fitur yang memerlukan data dinamis. Entitas yang sudah dideklarasikan pada basis data adalah sebagai berikut:

1. Entitas Mahasiswa

Entitas Mahasiswa berisikan data seluruh mahasiswa yang sudah terdaftar di aplikasi dengan atribut nim sebagai identifikasi, name untuk penyimpanan nama, tel untuk penyimpanan nomor telepon, email untuk penyimpanan alamat email, pass\_hash untuk penyimpanan *password hashing* dari pengguna, avatar\_url untuk penyimpanan foto profil, dan address untuk menyimpan alamat pengguna yang dibutuhkan untuk pengiriman barang.

2. Entitas Admin

Entitas Admin berisi data seluruh admin yang didaftarkan secara manual dengan atribut id sebagai identifikasi, email untuk penyimpanan alamat email admin, name untuk penyimpanan nama, dan pass\_hash untuk menyimpan hash dari *password* admin.

3. Entitas Fun TK

Entitas Fun TK berisi data mengenai kegiatan atau aktivitas yang dilakukan bersama anggota Himpunan Mahasiswa Teknik Komputer (HMTK). Entitas ini memiliki atribut id sebagai identifikasi, post\_date untuk tanggal unggahan, img\_url untuk menyimpan URL gambar, date untuk tanggal kegiatan, time untuk waktu kegiatan, location untuk lokasi kegiatan, title untuk judul kegiatan, description untuk deskripsi kegiatan, dan map\_url untuk URL peta lokasi kegiatan.

4. Entitas Activity

Entitas Activity berfungsi sebagai portal berita terkini, artikel, atau pengumuman bagi mahasiswa Himpunan Mahasiswa Teknik Komputer (HMTK). Portal ini hanya bisa diunggah oleh seorang admin. Entitas ini memiliki atribut id sebagai identifikasi, post\_date untuk tanggal unggahan, title untuk judul artikel atau pengumuman, content untuk isi berita, artikel, ataupun pengumuman, dan img\_url untuk URL gambar terkait.

5. Entitas Aspiration

Entitas Aspiration berisi umpan balik atau aspirasi dari anggota himpunan untuk Himpunan Mahasiswa Teknik Komputer (HMTK). Entitas ini memiliki atribut id sebagai identifikasi, mahasiswa\_nim

sebagai referensi ke entitas Mahasiswa, title untuk judul aspirasi, content untuk isi aspirasi, dan datetime untuk tanggal dan waktu aspirasi diunggah.

6. Entitas Tokens

Entitas Tokens digunakan untuk menyimpan token sementara yang digunakan oleh pengguna untuk me-reset kata sandi. Token ini akan dikirim ke email terdaftar milik pengguna. Entitas ini memiliki atribut id sebagai identifikasi, mahasiswa\_nim sebagai referensi ke entitas Mahasiswa, exp untuk waktu kedaluwarsa token, dan token untuk menyimpan nilai token.

7. Entitas Post

Entitas Post berisi postingan mahasiswa Himpunan Mahasiswa Teknik Komputer (HMTK). Semua mahasiswa bisa mengunggah post di sini. Entitas ini memiliki atribut id sebagai identifikasi, poster\_id sebagai referensi ke entitas Mahasiswa, post\_date untuk tanggal unggahan, img\_url untuk URL gambar, content untuk isi postingan, dan can\_comment untuk status apakah postingan bisa dikomentari atau tidak.

8. Entitas Comment

Entitas Comment digunakan untuk menyimpan komentar pada postingan. Entitas ini memiliki atribut id sebagai identifikasi, post\_id sebagai referensi ke entitas Post, commenter\_id sebagai referensi ke entitas Mahasiswa, comment\_date untuk tanggal komentar, dan content untuk isi komentar yang diberikan.

9. Entitas Like

Entitas Like digunakan untuk menyimpan informasi tentang siapa yang menyukai postingan tertentu. Entitas ini memiliki atribut liker\_id sebagai referensi ke entitas Mahasiswa dan post\_id sebagai referensi ke entitas Post.

10. Entitas Lab Post

Entitas Lab Post berisi berita terkini, informasi, ataupun pengu-muan terkait Laboratorium Teknik Komputer. Portal ini hanya bisa diunggah oleh seorang admin. Entitas ini memiliki atribut id sebagai identifikasi, post\_date untuk tanggal unggahan, lab untuk nama lab, img\_url untuk URL gambar, dan content untuk isi pengumuman.

11. Entitas Product

Entitas Product mencakup data produk-produk yang dijual oleh himpunan. Entitas ini memiliki atribut id sebagai identifikasi, name untuk nama produk, description untuk deskripsi produk, price untuk harga produk, dan img\_url untuk URL gambar produk.

12. Entitas Cart

Entitas Cart berisi data produk yang belum dipesan pengguna. Entitas ini memiliki atribut id sebagai identifikasi, product\_id sebagai referensi ke entitas Product, mahasiswa\_nim sebagai referensi ke entitas Mahasiswa, quantity untuk jumlah produk, size untuk ukuran produk, dan information untuk informasi tambahan terkait produk.

13. Entitas Order

Entitas ini serupa dengan entitas Cart, tetapi sudah masuk ke tahap transaksi. Entitas ini memiliki atribut id sebagai identifikasi, transaction\_id sebagai referensi ke entitas Transaction, product\_id sebagai referensi ke entitas Product, mahasiswa\_nim sebagai referensi ke entitas Mahasiswa, quantity untuk jumlah produk, size untuk ukuran produk, dan information untuk informasi tambahan terkait produk.

14. Entitas Transaction

Entitas Transaction berisi kumpulan order yang telah dilakukan. Entitas ini memiliki atribut id sebagai identifikasi, mahasiswa\_nim sebagai referensi ke entitas Mahasiswa, transaction\_date untuk tanggal transaksi, paid untuk status pembayaran, completed untuk status penyelesaian transaksi, payment\_url untuk URL pembayaran, dan status untuk status transaksi.

15. Entitas Bank Material



Gambar 6. Dokumentasi Seluruh Endpoint API

Entitas Bank Material berisi setiap mata kuliah yang dipelajari mahasiswa Himpunan Mahasiswa Teknik Komputer (HMTK). Entitas ini memuat informasi mengenai nama mata kuliah, pada tingkatan berapa mata kuliah tersebut diajarkan, serta tautan ke Google Drive yang berisikan file referensi mata kuliah terkait.

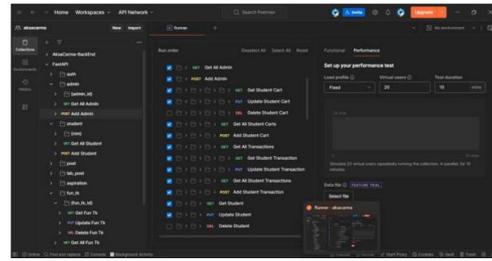
**Hasil Implementasi Akhir *Back-end* API**

Pengembangan aplikasi di sisi *back-end* yang sudah dibuat untuk digunakan pada aplikasi terealisasi pada 72 endpoint yang ter-host di internet. *Framework* FastAPI memberikan rute khusus untuk memberikan dokumentasi khusus untuk pengguna API melihat tampilan dan struktur dari setiap endpoint yang ada pada API. Dokumentasi yang diberikan memudahkan penggunaan dan penyesuaian API terhadap aplikasi yang dikembangkan. Laman dokumentasi ini dilengkapi dengan fitur untuk mengisi autentikasi pengguna, seluruh parameter yang digunakan pada setiap endpoint, serta bentuk respon yang akan diberikan oleh endpoint. Uji coba API juga dapat dilakukan pada dokumentasi tersebut tanpa harus melakukan *request* dari kode.

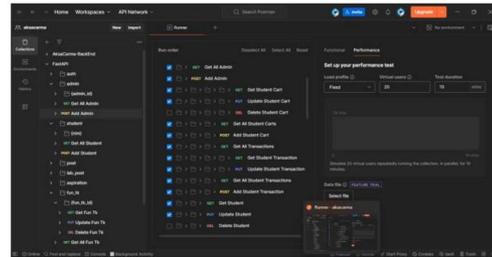
**Stress Test API**

Pengujian stress test pada aplikasi Postman dilakukan dengan cara impor seluruh endpoint yang digunakan dari *back-end* ke dalam sebuah Collection di Postman. Variabel acak kemudian dimasukkan pada parameter yang diperlukan dengan menggunakan variabel dinamis yang disediakan oleh Postman, seperti `$randomLoremSentences`, `$random-FirstName`, `$randomEmail`, `$randomPassword`, dan lain-lain. Variabel dinamis ini akan membuat simulasi data acak yang realistis dalam setiap permintaan. Setelah Collection disiapkan dengan endpoint dan variabel yang diperlukan, pengujian dijalankan menggunakan opsi performance test dengan konfigurasi berikut: Load profile diatur pada opsi Fixed untuk menentukan jumlah virtual user yang konstan selama pengujian; Virtual users diatur pada jumlah maksimum 20 user untuk mensimulasikan 20 pengguna yang menjalankan Collection secara paralel dan berulang selama durasi tes; dan Test duration diatur selama 10 menit. Dengan pengaturan ini, pengujian akan berjalan selama 10 menit dengan 20 pengguna berjalan secara bersamaan, sehingga memungkinkan evaluasi kinerja API di bawah kondisi beban tinggi dan memastikan bahwa sistem dapat menangani situasi kritis tanpa mengalami kegagalan total. Hasil dari stress test yang dilakukan setelah 10 menit adalah sebagai berikut: jumlah permintaan total yang diminta ke *back-end* API sejumlah 21.526 permintaan, keluaran bernilai 35,30 permintaan setiap detik, rata-rata waktu respon 396ms, dengan tingkat kesalahan 1,07%.

Gambar 8 menunjukkan grafik garis dengan beberapa garis yang mewakili berbagai set data yang diplotkan seiring waktu. Pada sumbu horizontal, waktu ditunjukkan mulai dari 03:49:20 hingga 03:59:20, dan pada sumbu vertikal sebelah kiri, jumlah pengguna virtual ditampilkan dari 0 hingga 20. Sumbu vertikal sebelah kanan menunjukkan waktu *respons* dalam milidetik dari 0 hingga 2200 ms. Terdapat empat garis dengan warna berbeda yang mewakili 'Pengguna Virtual',



Gambar 7. Persiapan Performance Test di Aplikasi Postman



Gambar 8. Grafik Tren Waktu *Respons* Selama Durasi Pengujian



Gambar 9. Tingkat Permintaan yang Dikirim Per Detik Selama Durasi Pengujian

'Waktu *respons* rata-rata', 'Persentil ke-90', dan 'Persentil ke95'. Garis untuk pengguna virtual berwarna ungu, waktu *respons* rata-rata berwarna merah muda, persentil ke-90 berwarna biru, dan persentil ke-95 berwarna abu-abu.

Grafik ini relevan untuk menganalisis kinerja layanan web atau aplikasi dengan menunjukkan bagaimana layanan tersebut menanggapi lalu lintas (pengguna virtual) dan waktu *respons*nya selama periode tertentu. Gambar 9 adalah grafik garis dengan dua skala berbeda dan garis yang mewakili 'Pengguna Virtual' dan 'Throughput'. Sumbu horizontal menunjukkan waktu, mulai dari 03:49:20 hingga 03:59:20, menunjukkan bahwa data dikumpulkan selama interval sepuluh menit. Sumbu vertikal sebelah kiri diberi label 'VU 20' dan memiliki nilai dari 0 hingga 20, yang kemungkinan sesuai dengan jumlah pengguna virtual. Sumbu vertikal sebelah kanan diberi label '900req/s', menunjukkan permintaan per detik, dengan nilai dari 0 hingga 900.

**Kesimpulan**

Dari hasil pembahasan aplikasi My-HMTK yang sudah dilakukan, didapatkan wawasan bahwa kegunaan dan waktu kerja yang diperlukan dari keseluruhan aplikasi ini sudah tepat. Pengujian yang sudah dilakukan menunjukkan kecepatan API jika Berdasarkan pada hasil dan pembahasan aplikasi MyHMTK ini, menunjukkan bahwa tingkat kegunaan dan kecepatan sistem aplikasi MyHMTK dengan menggunakan questionnaire dapat dikategorikan sangat layak. Ketidak adanya kesalahan navigasi pada bagian *front end* yang di terangkan pada pengujian *Blacbox testing* dan dengan *presentase* skor *error* pada bagian *backend* mencapai 1,07%. Aplikasi MyHMTK dapat dinyatakan sudah dapat memenuhi tujuan pengguna dengan ke efisienan dan ke efektifan serta kepuasan pengguna.

**Table 4.** Permintaan Paling Lambat Teratas Berdasarkan Waktu Respons Rata-Rata

Rute	Rata-rata waktu respon (ms)	Per sentil ke-90 (ms)	Per sentil ke-95 (ms)	Per sentil ke-99 (ms)	Min (ms)	Max (ms)
Get All Posts	976	1.480	1.608	1.798	306	1.899
Get All Aspirations	926	1.449	1.581	1.976	281	2.556
Get All Activity	892	1.374	1.451	1.683	294	5.082
Get All Post Comments	739	1.203	1.340	1.501	281	1.590
Get All Student	709	1.116	1.200	1.428	288	1.654

### Daftar Pustaka

- Syaputra A. PELATIHAN MAINTENANCE KOMPUTER PADA HIMPUNAN MAHASISWA TEKNIK INFORMATIKA KOTA PAGAR ALAM. Pengabdian Kepada Masyarakat. 2020;03:75-81.
- Hermawansyah A. Analisis Profil dan Karakteristik Pengguna Media Sosial di Indonesia; 2022.
- Furqon A, Hermansyah D, Sari R, Sukma A, Akbar Y, Rakhmawati NA. ANALISIS SOSIAL MEDIA PEMERINTAH DAERAH DI INDONESIA BERDASARKAN RESPONS WARGANET ANALYSIS OF LOCAL GOVERNMENT SOCIAL MEDIA IN INDONESIA BASED ON NETIZEN RESPONSE;.
- Çetin E, Özdemir S. A Study on an Educational Website's Usability. *Procedia Soc Behav Sci.* 2013 July;83:683-8.
- Banerjee P, Kumar B, Singh A, Kumar R, Kumar R. Comparative performance analysis of optimized round robin scheduling (ORR) using dynamic time quantum with round robin scheduling using static time quantum in Real Time System. *International Journal of Engineering and Computer Science.* 2019 December;8(12):24890-3.
- Yanti SN, Rihyanti E. Rancang Bangun Rest API Aplikasi WeShare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan. *Jurnal Informatika Teknologi dan Sains.* 2021;4(1):8-14.
- Christie M, et al. An extensible Django-based web portal for Apache Airavata. In: *ACM International Conference Proceeding Series. Association for Computing Machinery;* 2020. p. 160-7.
- Sabrina A, Sufa AO, Ritonga DP, Siregar ERS. Pengenalan Konsep Dasar Dan Penggunaan Database Manajemen Sistem (Dbms). *Jurnal Sains Dan Teknologi (JSIT);*3(3).
- Duisebekova K, Khabirov R, Zholzhan A. DJANGO AS SECURE WEB-FRAMEWORK IN PRACTICE. 2021;1(116):275-81.