

BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi informasi dan komunikasi membawa perubahan besar dalam berbagai aspek kehidupan manusia. Salah satu dampak yang paling terlihat adalah meningkatnya penggunaan internet baik secara global maupun lokal. Menurut Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), jumlah pengguna internet di Indonesia pada tahun 2024 telah mencapai 79,5% dari total penduduk di Indonesia [1], meningkat 1,4% dari tahun sebelumnya.

Dibalik pertumbuhan yang signifikan ini, ancaman dalam dunia teknologi juga semakin meningkat. Kasus peretasan dan kegagalan sistem kini menjadi ancaman serius. Contoh kasus peretasan pada Bank Syariah Indonesia (BSI) pada tahun 2023 yang dilakukan kelompok peretas LockBit. Dalam kasus tersebut, LockBit mengaku berhasil mengambil jutaan data nasabah BSI dan meminta tebusan sebesar Rp 295,61 miliar [2]. Pada tahun 2024, ribuan perangkat Windows di seluruh dunia mengalami *blue screen* masal yang berdampak pada sejumlah layanan, seperti penerbangan, perbankan, hingga kesehatan di berbagai negara [3]. Hal ini menunjukkan pentingnya antisipasi terhadap kegagalan sistem.

Sebagai perusahaan yang bergerak dibidang perbankan, PT XYZ memiliki risiko terhadap serangan siber dan kegagalan sistem. Hal tersebut menjadi ancaman serius terhadap keamanan data, ketersediaan layanan, serta mempengaruhi kepercayaan pengguna atau nasabah. Bahkan, gangguan kecil sekalipun berdampak besar mengingat perbankan merupakan salah satu pendukung stabilitas ekonomi.

Saat terjadi kegagalan sistem dalam studi kasus PT XYZ, biasanya pengguna akan menghubungi penanggungjawab sistem tersebut, kemudian melaporkan kegagalan tersebut. Tim pengembang umumnya akan bertanya apa

yang dilakukan pengguna tersebut. Masalah muncul ketika pengguna lupa dan/atau sulit menjelaskan apa yang telah dilakukan hingga kegagalan sistem terjadi. Hal ini dibuktikan dengan wawancara dengan salah satu karyawan PT XYZ mengatakan “Kalo ada *error* biasanya kita cek ke loggingnya sih, soalnya kadang user tuh lupa mereka habis ngapain aja sampai bisa muncul *error* itu”.

Saat ini PT XYZ sudah menerapkan sistem *logging*. Data *logging* dicatat di *server* tempat aplikasi tersebut berada. Namun, untuk melihat *logging* di *server* sulit dilakukan karena menggunakan tampilan berbasis terminal dan akses yang terbatas. Selain itu, data *log* tidak tersimpan di basis data yang membuat data *log* hilang ketika *server restart*.

Untuk mencegah terjadinya hal-hal yang tidak diinginkan dalam dunia IT, dapat dilakukan dengan manajemen insiden. Manajemen insiden merupakan proses yang digunakan tim IT untuk merespons gangguan layanan yang tidak direncanakan [4]. Tujuan dari manajemen insiden adalah mengembalikan operasi layanan IT secepat mungkin dengan dampak seminimal mungkin.

Salah satu upaya yang dapat dilakukan untuk mencegah insiden adalah menggunakan sistem pemantauan (*monitoring*) aplikasi. Terdapat beberapa contoh layanan untuk melakukan *monitoring* seperti Grafana, Prometheus, dan Kibana [5]. Layanan tersebut biasanya menyediakan fitur *alerting* ketika terjadi anomali pada suatu aplikasi. Dengan adanya sistem *logging* dan *alerting*, hal ini dapat memastikan apa yang terjadi pada suatu aplikasi.

Aplikasi pemantauan tidak hanya membantu dalam mendeteksi serta mencegah terjadinya insiden, tetapi juga dapat memberikan wawasan yang lebih baik tentang bagaimana pengguna berinteraksi dengan sistem. Pengalaman pengguna dapat dioptimalkan dengan memanfaatkan data yang disimpan. Hal tersebut dapat membantu memastikan kesehatan dari sistem yang digunakan.

Untuk menjawab tantangan ini, dilakukan penelitian untuk membangun aplikasi pemantauan terintegrasi dengan contoh kasus PT XYZ. Penelitian ini diharapkan dapat menyelesaikan masalah utama seperti visualisasi data log,

pencarian data log, bahkan membantu menemukan penyebab *error*. Penelitian ini berjudul “**Pembangunan Aplikasi Pemantauan Aktivitas Pengguna Terhadap API (*Application Programming Interface*) Berbasis Web (Studi Kasus PT XYZ)**”.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, PT XYZ menghadapi beberapa tantangan dalam implementasi pemantauan sistem. Pertama, perusahaan masih bergantung pada layanan pihak ketiga yang memiliki keterbatasan fungsional dan kurang sesuai dengan kebutuhan operasional. Kedua, ketidakmampuan layanan tersebut untuk melakukan pemantauan multi-aplikasi secara simultan, dimana hanya satu aplikasi yang dapat dipantau dalam satu waktu. Selain itu, sistem pemantauan API yang ada belum menyajikan antarmuka yang *user-friendly* karena masih berbasis terminal dengan akses terbatas pada *server*. Kondisi ini diperparah dengan penempatan sistem *logging* di *server* yang membuat proses pelacakan data lebih sulit.

C. Pertanyaan Penelitian

Pertanyaan penelitian yang diperoleh berdasarkan latar belakang dan rumusan masalah penelitian adalah sebagai berikut.

1. Bagaimana membangun aplikasi pemantauan aktivitas pengguna terhadap API berbasis web yang sesuai dengan kebutuhan PT XYZ dan mudah digunakan?
2. Bagaimana cara menyimpan data *request log* dari aplikasi Django Rest Framework?

D. Batasan Masalah

Penelitian yang diperoleh berdasarkan latar belakang penelitian adalah sebagai berikut.

1. Penelitian ini menggunakan aplikasi internal di PT XYZ sebagai studi kasus.

2. Sistem yang dibangun berbasis *website* yang bekerja optimal pada tampilan desktop.
3. Obyek yang dipantau adalah *log* dari *request* yang diterima *server*.
4. Bahasa pemrograman yang mendukung untuk menyimpan data request log adalah Django.
5. Basis data yang digunakan adalah PostgreSQL.
6. Sistem notifikasi menggunakan email dari layanan Google dengan batas 200 email per hari.
7. Dataset yang digunakan untuk memperoleh lokasi dari *ip address* disimpan secara lokal bersama kode aplikasi dengan data terakhir diperbarui 30 November 2024.
8. Akses pengguna terbagi menjadi dua, yaitu *Admin* dan *Guest*.
9. Data yang disimpan adalah *endpoint*, *body*, *headers*, *method*, *ip address*, *user agent*, *city*, *country name*, *country code*, *response time*, *status code*, *error message* dan *created at*.

E. Tujuan Penelitian

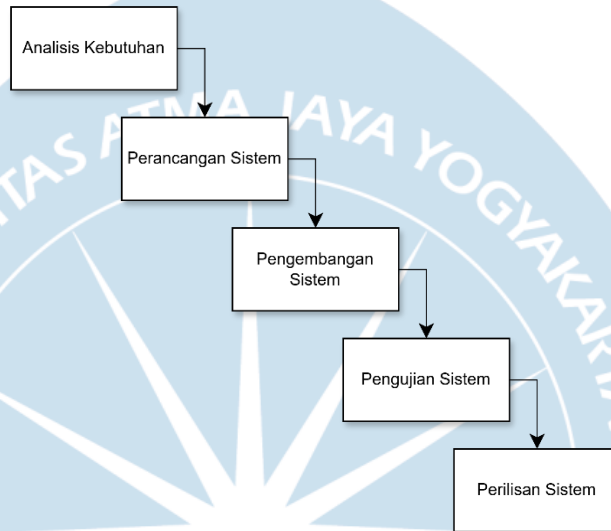
Tujuan dari penelitian pembangunan aplikasi pemantauan aktivitas pengguna terhadap API berbasis *website* adalah sebagai berikut.

1. Membangun aplikasi berbasis web untuk melakukan pemantauan aktivitas pengguna terhadap API yang mudah digunakan.
2. Menambahkan sistem untuk menyimpan data *request log* untuk RESTful API yang menggunakan Django Rest Framework.

F. Metode Penelitian

Pembangunan aplikasi pemantauan aktivitas pengguna terhadap API berbasis web dengan studi kasus PT XYZ dilakukan dengan menerapkan metode pengembangan Waterfall. Waterfall merupakan salah satu kerangka kerja dalam SDLC yang memiliki pendekatan manajemen secara berurutan. Dalam metode

Waterfall setiap fase harus diselesaikan sebelum melanjutkan ke fase berikutnya. Metode ini memiliki kelebihan seperti proses yang lebih terstruktur dan terorganisasi. Adapun metode Waterfall yang diterapkan dapat dilihat pada Gambar 1.1.



Gambar 1. 1 Metode Waterfall

1. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengetahui kebutuhan fungsional dan non-fungsional dari aplikasi yang akan dibangun. Proses analisis dilakukan dengan melakukan wawancara sesuai kebutuhan pengguna di PT XYZ.

2. Perancangan Sistem

Perancangan sistem dilakukan untuk menghasilkan rancangan mendetail sebagai panduan dasar saat pengembangan sistem. Perancangan sistem mencakup pembuatan arsitektur sistem, basis data, dan desain antarmuka pengguna.

3. Pengembangan Sistem

Pengembangan sistem dilakukan untuk mewujudkan rancangan sistem menjadi kode dengan bahasa pemrograman dan teknologi yang sesuai. Adapun

bahasa pemrograman yang digunakan dalam kasus ini adalah Django Rest Framework dan Vue Js.

4. Pengujian Sistem

Pengujian sistem merupakan tahapan untuk memastikan aplikasi yang telah dikembangkan telah sesuai dengan rancangan sistem. Pengujian dilakukan secara menyeluruh terhadap kebutuhan fungsional dan non-fungsional.

5. Perilisan Sistem

Perilisan atau *Deployment* merupakan tahapan akhir dalam metode *waterfall*. Proses ini memungkinkan aplikasi dipersiapkan untuk dapat digunakan baik di *server* lokal maupun online untuk pengujian.

G. Sistematika Penulisan

Sistematika penulisan tugas akhir disusun dalam beberapa bab sebagai berikut.

BAB I PENDAHULUAN

Pendahuluan membahas latar belakang masalah, rumusan masalah, batasan penelitian, tujuan penelitian, metode penelitian, serta sistematika penulisan yang digunakan dalam pembangunan *website* pemantauan aktivitas pengguna terhadap API.

BAB II TINJAUAN PUSTAKA

Tinjauan membahas beberapa penelitian serupa atau terkait dengan pembuatan tugas akhir yang dilakukan. Bagian ini bertujuan sebagai dasar perbandingan terhadap *website* yang akan dibuat.

BAB III LANDASAN TEORI

Landasan teori membahas teori-teori yang digunakan dalam pembangunan *website* tugas akhir. Teori-teori yang ada kemudian digunakan sebagai dasar ilmu dalam pembuatan tugas akhir ini.

BAB IV ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan sistem membahas analisis sistem, lingkup masalah, perspektif produk, fungsi produk, kebutuhan antarmuka *website*, perancangan arsitektur, dan perancangan antarmuka *website*.

BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM

Implementasi dan pengujian sistem membahas implementasi sistem, implementasi antarmuka sistem, pengujian fungsionalitas, dan hasil pengujian sistem terhadap pengguna.

BAB VI PENUTUP

Penutup membahas kesimpulan dan saran dari sistem yang dibangun dengan tujuan sebagai acuan perkembangan sistem dimasa yang akan datang.

DAFTAR PUSTAKA

Daftar pustaka berisi urutan pustaka yang digunakan dalam tugas akhir. Daftar pustaka ditulis menggunakan format IEEE.

LAMPIRAN

Lampiran berisi dokumen pendukung dalam penelitian.