BAB I
PENDAHULUAN

I.1 Latar Belakang Masalah

Selama beberapa dekade ini teknologi *web* berkembang dengan cepat. Perkembangan ini mengakibatkan banyak bahasa pemrograman baru bermunculan yang sering dikenal dengan "*language war*". Bahasa pemrograman baru ini hadir dengan kualitas yang lebih baik sehingga menjadikan dunia *web development* berkembang cepat. Persaingan ini hadir dalam hal performa, keamanan, kesederhanaan sintaks, optimalisasi, dan kemampuan untuk berkembang di masa datang, sehingga banyak developer yang memigrasikan kode dari bahasa satu ke bahasa lain demi menjadikan program yang lebih baik dan cepat. Hal ini juga berlaku untuk aplikasi Crimezone.

Crimezone merupakan aplikasi *citizen journalism* pada *mobile* untuk melaporkan bentuk kriminalitas. Aplikasi ini tersedia dalam platform *android* dan *windows phone*. Berdasarkan wawancara yang telah dilakukan diperoleh informasi bahwa aplikasi ini sudah berjalan dengan baik, namun memiliki beberapa kekurangan dari segi sistem *backend*. Pertama, sistem *backend* saat ini belum bekerja dengan baik, dalam melayani akses pengguna. Hal ini dibuktikan dengan masih terjadinya kegagalan dalam mengirimkan data yang berupa *image stream* dengan kapasitas yang cukup besar. Kegagalan ini terjadi apabila proses uploadnya memakan waktu yang terlalu lama akan menyebabkan *request time out*. Kedua, sistem *backend* belum bekerja dengan cepat dan optimal yang dibuktikan dengan kinerja dari *web service* yang lambat dalam melayani *request* dan *response* data dari pengguna. Ketiga, sintaksis PHP yang digunakan belum mengikuti standar *update* terbaru. Salah satu contohnya yaitu, ketika ingin menghubungkan PHP ke *database* MySQL masih menggunakan *method* mysql. Padahal *method* mysql itu sendiri masih rentan terhadap berbagai jenis serangan *web* seperti misalnya SQL *Injection*.

Secara umum, untuk membangun sistem *backend* yang baik dan bagus ditentukan oleh bahasa pemrograman yang digunakan. Pemilihan bahasa pemrograman yang tepat akan menentukan tingkat kualitas sistem yang dibangun. Ada beberapa kriteria yang menentukan bahasa pemrograman baik dan bagus. Pertama, bisa menghasilkan kode yang lebih baik, dengan kode yang lebih baik akan menghasilkan performa yang cepat ketika dijalankan. Kedua, bisa mengatasi masalah *concurrency* dan mendukung *parallelism* sehingga menjadikan program *scalable* [1]. Ketiga, kode yang di tulis secara *default* memiliki keamanan yang tinggi dalam melindung terhadap berbagai jenis serangan *web* [2].

Oleh karena itu, berdasarkan permasalahan yang telah dijelaskan dan berdasarkan studi literatur, didapatkan solusi yaitu dengan memigrasikan kode pada *backend* Crimezone dari PHP ke Scala. Pemilihan Scala sebagai bahasa pemrograman akan meningkatkan performa sistem *backend* [3], memudahkan dalam menggunakan pemrograman *asynchronous*, *concurrency*, dan *parallelism* serta meningkatkan keamanan pada sistem [4]. Selain itu, masih banyak keuntungan lain yang akan diperoleh dengan menggunakan Scala dibandingkan menggunakan PHP.

I.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dirumuskan suatu masalah yaitu bagaimana memigrasikan kode pada *backend* Crimezone dari PHP ke Scala.

I.3 Maksud dan Tujuan

Maksud dari penelitian ini adalah memigrasikan kode pada *backend* Crimezone dari PHP ke Scala dengan tujuan sebagai berikut:

1. Meningkatkan performa sistem *backend* dalam melayani akses pengguna.
2. Memudahkan dalam menggunakan pemrograman yang mendukung *asynchrnous*, *concurrency*, dan *parallelism*.
3. Meningkatkan keamanan sistem *backend* sehingga tidak rentan terhadap berbagai serangan *web*.

I.4 Batasan Masalah

Adapun batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Pengembangan perangkat lunak hanya difokuskan untuk sistem *backend*.
2. Perangkat lunak menggunakan basis data yang telah tersedia pada pembangunan perangkat lunak sebelumnya.
3. Pendekatan analisis yang digunakan dalam migrasi perangkat lunak ini menggunakan *Structure Analysis and Design* (SAD) dan *Object Oriented Analysis and Design* (OOAD).
4. Perangkat lunak menggunakan JSON (*Javascript Object Nation*) sebagai media pertukaran data dengan *server*.
5. Pada penelitian ini, keamanan sistem yang di-*handle* yaitu pada *server*. Di sistem baru, sistem *backend* dan *frontend* akan dipasang *Cloudflare* yang berguna meningkatkan keamanan sistem dari berbagai jenis serangan *web*.

I.5 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini adalah metode penelitian komparatif. Metode ini bertujuan untuk membandingkan persamaan dan perbedaan dua atau fakta-fakta dan sifat-sifat objek yang di teliti, sehingga dapat menentukan mana yang lebih baik atau mana yang sebaiknya dipilih. Dalam pelaksanaannya, metodologi penelitian ini terdiri dari metode pengumpulan data dan metode pembangunan perangkat lunak itu sendiri.

I.5.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan adalah sebagai berikut:

1. Studi Literatur

Pengumpulan data yang dilakukan dengan cara mempelajari dan memahami berbagai literatur ilmiah yang bersumber dari jurnal, buku, *proceedings*, *thesis*, dan bacaan lainnya yang dapat dijadikan acuan dalam penelitian ini seperti konsep *software migration*, *software maintenance*, metode *reverse enginnering*, scala, *concurrency*, *paralleism*, *asynchronous*, *non-blocking*, teknologi *cloud*, dan JSON.

2. Wawancara

Pengumpulan data yang dilakukan dengan cara bertanya langsung yang ada kaitannya dengan topik yang diambil. Wawancara ini dilakukan kepada beberapa *backend* developer aplikasi Crimezone. Wawancara yang dilakukan lebih mengarah kepada performa dan kekurangan apa saja yang dimiliki aplikasi saat ini.

I.5.2 Metode Migrasi Perangkat Lunak

Dalam memigrasikan kode pada *backend* perangkat lunak ini akan menggunakan pendekatan *Reference Migration Processes* (ReMiP). ReMiP merupakan model proses umum yang menggambarkan aktivitas yang dibutuhkan sebagai tahapan dalam proses migrasi. Proses-proses yang terjadi di dalamnya adalah [6]:

1. *Requirement Analysis*

Tahap *requirement analysis* merupakan tahapan pertama di ReMiP. Pada tahap ini hanya dilakukan analisis kebutuhan non-fungsional karena memiliki dampak langsung terhadap pemilihan teknologi. Sedangkan analisis fungsional tidak perlu dilakukan karena fungsionalitas sistem tidak mengalami perubahan. Selain itu, kebutuhan peralatan migrasi diperiksa dan semua kebutuhan dikelola untuk mengatasi perubahan.

2. *Legacy Analysis*

Pada tahap ini dimulai dengan menganalisa *legacy system* kasar. *Legacy system* dinilai kualitas teknis dan nilai bisnisnya.

3. *Target Design*

Pada tahap ini, arsitektur, struktur data, dan antarmuka pengguna dari sistem baru didefinisikan. Pada tahap ini juga disertakan *artifact* sementara yang mungkin disertakan selama perubahan. Selain itu, *legacy artifact* dipetakan ke target *artifact* yang sesuai.

4. *Strategy Selection*

Pada tahap ini dilakukan pemilihan strategi migrasi yang akan digunakan. Strategi migrasi menjelaskan bagaimana *legacy system* akan dikonversi ke dalam sistem yang dimigrasikan. Strategi yang dipilih yaitu strategi *re-implementation*

dengan alasan sistem baru di kembangkan ulang berdasarkan ide-ide yang disediakan oleh *legacy software* tanpa mengubah fungsionalitas sistem yang ada. *Re-Implementation* dimulai dengan menganalisis *legacy system*, kebutuhan fungsional dan non-fungsional diambil.

5. *Implementation*

Pada tahap ini dilakukan implementasi desain sistem yang telah didefinisikan. Selain itu, *legacy system* dimigrasikan sesuai dengan strategi migrasi yang dipilih pada tahapan *strategy selection*.

6. *Test*

Salah satu tujuan utama dari migrasi perangkat lunak yaitu untuk membuat salinan fungsional *one-to-one* dari *system legacy*. Pada tahap ini, sistem baru akan diuji, apakah mendukung fungsi yang sama dengan sistem lama. Untuk melakukannya bisa menggunakan *unit test*.

7. *Deployment*

Tahap *deployment* merupakan tahap terakhir dalam tahapan migrasi perangkat lunak. Tahap ini berbicara bagaimana memperkenalkan sistem baru yang sudah dihasilkan dengan menggunakan strategi yang ada. Salah satu strategi yang bisa digunakan yaitu *big-bang strategy*. Strategi ini dilakukan dengan cara sistem baru di-*deploy* secara lengkap dan sekaligus menutup *legacy system* pada waktu yang bersamaan. Keuntungan menggunakan strategi ini yaitu cepat dan mengurangi biaya migrasi. Dari berbagai tahapan-tahapan tersebut, untuk lebih jelasnya dapat dilihat pada Gambar I.1.



Gambar I.1 Tahapan ReMiP [6]

I.6 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum mengenai penelitian yang dikerjakan. Sistematika penulisan dalam penelitian ini adalah:

BAB 1 PENDAHULUAN

Bab ini menguraikan latar belakang permasalahan, merumuskan inti permasalahan, mencari solusi atas masalah tersebut, mengidentifikasi masalah tersebut, menentukan maksud dan tujuan, kegunaan penelitian, pembatasan masalah, metode penelitian, dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab ini menguraikan bahan-bahan kajian, konsep dasar, dan teori dari para ahli yang berkaitan dengan penelitian. Meninjau permasalahan dan hal-hal yang berguna dari penelitian-penelitian dan sintesis serupa yang pernah dikerjakan sebelumnya dan menggunakannya sebagai acuan pemecahan masalah pada penelitian ini.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab 3 berisi tentang analisis sistem yang terdiri dari analisis masalah, analisis spesifikasi kebutuhan perangkat lunak, analisis kebutuhan non fungsional, analisis kebutuhan fungsional *legacy system*, analisis kebutuhan *class new system*, analisis *concurrency* dan *asynchronous*, dan analisis data. Perancangan sistem terdiri dari perancangan *class*, perancangan data, perancangan arsitektur menu, dan perancangan antarmuka.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini berisi tentang implementasi hasil dari analisis dan perancangan sistem, perancangan sistem ke dalam bentuk bahasa pemrograman, kebutuhan perangkat keras dan perangkat lunak yang diperlukan dalam membangun sistem serta pengujiannya.

BAB 5 KESIMPULAN DAN SARAN

Bab ini akan berisikan kesimpulan-kesimpulan dari penelitian, dan terdapat pula saran yang diberikan agar dapat digunakan di masa mendatang untuk perbaikan perangkat lunak menjadi lebih baik.