BAB I
PENDAHULUAN

1.1. Latar Belakang

Pada era digital saat ini, penerapan teknologi informasi telah menjadi bagian yang penting untuk mengembangkan organisasi yang mampu bersaing di pasar global. Menurut laporan *Business Research Insight*, ukuran pasar teknologi informasi (IT) global mencapai USD 10.524 miliar pada tahun 2023 dan diproyeksikan mencapai USD 26.925 miliar pada tahun 2032 dengan tingkat pertumbuhan *compounded annual growth rate* (CAGR) sebesar 11,0% selama periode tersebut. Revolusi industri telah mendorong berbagai perusahaan untuk mengintegrasikan teknologi informasi dalam mengoptimalkan proses bisnisnya dan memperluas jangkauan pasar. Untuk mencapai hal tersebut, diperlukan pengelolaan teknologi informasi dengan sistem yang mumpuni, berjalan dengan lancar, dan bebas dari bug. Salah satu penerapan teknologi informasi pada perusahaan berbasis teknologi adalah *website* yang digunakan sebagai media pemasaran dan sarana interaksi dengan pelanggan. Salah satu perusahaan yang telah menerapkan *website* sebagai media pemasaran dan sarana interaksi dengan pelanggan adalah PT. Generasi Anak Muda Berkarya.

PT. Generasi Anak Muda Berkarya merupakan sebuah perusahaan *startup* yang bergerak di bidang edukasi dan teknologi. *Startup* adalah perusahaan yang baru memulai operasi dan dirancang untuk berkembang dengan cepat [1]. Salah satu produk teknologi informasi yang dimiliki oleh perusahaan ini bernama HiColleagues. Produk ini merupakan sebuah *website* yang dirancang untuk menjadi platform media pembelajaran untuk mendukung individu dalam meningkatkan keterampilan teknis dan profesional mereka. Platform ini menawarkan berbagai kelas dan program pelatihan yang dirancang secara khusus sesuai dengan kebutuhan industri teknologi saat ini. Melalui platform ini, diharapkan setiap individu dapat berkembang menjadi talenta digital indonesia yang siap menghadapi tantangan dan kebutuhan industri teknologi masa kini.

Website HiColleagues telah digunakan berbagai jenis pengguna dan terbagi menjadi tiga layanan utama. Layanan pertama adalah *dashboard* yang digunakan untuk mengelola manajemen konten, *customer relationship management* (CRM), keuangan, program kelas, serta memonitor statistik bisnis

yang ada pada platform ini. Layanan kedua adalah *learning management system* (LMS) yang digunakan untuk pengguna agar dapat mengakses kelas dan materi pembelajaran yang telah terdaftar serta memantau aktivitas pembelajaran mereka di platform ini. Layanan terakhir adalah halaman utama yang digunakan untuk menampilkan program-program yang ada di HiColleagues yang digunakan untuk memberi informasi kepada pengujung platform ini mengenai kelas dan pelatihan terbaru yang tersedia untuk diikuti.

Namun selama masa pengembangan *website* HiColleagues, pengujian yang dilakukan hanya berfokus pada aspek fungsionalitas tanpa disertai dokumentasi skenario pengujian yang jelas dan terstruktur. Hal ini menyebabkan proses pengujian bersifat subjektif sehingga para *developer* hanya mengandalkan pengamatan pribadi untuk memastikan fitur berjalan sesuai harapan. Kondisi ini berpotensi menimbulkan celah yang memungkinkan terjadinya bug dan mengurangi keandalan *websit*e. Oleh karena itu, diperlukan pengujian yang lebih komprehensif untuk mengidentifikasi kelemahan sistem serta meningkatan kualitas dan keadalan sistem secara keseluruhan.

Secara umum, pengujian perangkat lunak terbagi kedalam tiga jenis pendekatan. Pendekatan pertama adalah *black box* yang pengujiannya berfokus pada aspek fungsionalitas sistem. Pada pendekatan ini, penguji tidak perlu memahami kode atau logika internal sistem. Penguji hanya memperhatikan *input* yang diberikan dan *output* yang dihasilkan serta mengevaluasi apakah sistem sudah berfungsi sesuai dengan yang diharapkan atau belum. Pendekatan kedua adalah *grey box* yang pengujianya dengan mengombinasikan pendekatan *white box* dan *black box* [2].. Pada pendekatan ini penguji memiliki pemahaman parsial pada struktur internal sistem. Pendekatan terakhir adalah *white box* yang pengujiannya berfokus pada pemeriksaan kode serta logika internal dari sistem. Pengujian ini dilakukan dengan menganalisis kode program dan merancang kasus uji berdasarkan struktur sistem tersebut [2]. Dengan pendekatan ini, penguji dapat memastikan bahwa kode internal telah sesuai desain, memenuhi kebutuhan fungsional, serta tidak memiliki celah bug.

Pada pendekatan *white box* terdiri dari beragam teknik pengujian, di antaranya adalah *data flow testing* yang digunakan untuk mendeteksi kesalahan yang berkaitan dengan pengunaan variabel dalam kode internal, *control flow*

testing digunakan untuk menguji berbagai kemungkinan jalur eksekusi yang ada dalam kode internal, *loop testing* digunakan untuk menguji bahwa perulangan yang ada dalam kode internal sudah berjalan dengan benar atau belum, *branch testing* digunakan untuk menguji kondisi pengambilan keputusan yang ada dalam kode program sudah berjalan dengan benar atau belum, dan *basis path testing* digunakan untuk menguji setiap pernyataan yang ada pada kode program dieksekusi setidaknya satu kali.

Dibandingkan dengan teknik lainnya, teknik *basis path testing* ini lebih sesuai karena dapat menghasilkan *test case* yang lengkap, mengevaluasi logika program secara mendalam, serta menemukan jalur-jalur penting yang perlu diuji [3]. Pada penerapan *basic path testing* menggunakan empat komponen utama. Komponen pertama adalah *flowgraph* yang digunakan sebagai representasi visual dari alur logika kode program. Komponen kedua adalah *cylomatic complexity* yang digunakan untuk menghitung jumlah jalur pengujian dalam kode program. Komponen ketiga adalah *independent path* yang digunakan untuk menentukan jalur-jalur yang wajib diuji minimal satu kali. Komponen terakhir adalah *deriving test case* yang digunakan untuk memastikan kesesuaian fungsionalitas berdasarkan *independent path* yang telah diidentifikasi. Berbeda dengan penelitian lainnya yang hanya menggunakan teknik *basis path testing* secara konvesional, pada penelitian ini pembuatan diagram *flowgraph* dan perhitungan *cyclomatic complexity* dilakukan secara otomatis.

Berdasarkan permasalahan tersebut, tujuan dari penelitian ini adalah melakukan pengembangan *tools* automatisasi untuk pembuatan diagram *Flowgraph* dan perhitungan *cyclomatic complexity* guna mendukung pengujian *white box* dengan teknik *basis path* pada aplikasi berbasis bahasa Golang, mengevaluasi hasil pengujian *white box* menggunakan teknik *basis path* yang dilakukan dengan bantuan *tools* automatisasi pembuatan diagram *flowgraph* dan perhitungan *cyclomatic complexity* pada *website* HiColleagues, dan memberikan rekomendasi perbaikan pada *website* HiColleagues berdasarkan temuan dari pengujian yang telah dilakukan.

1.2. Rumusan Masalah

Berdasarkan latar belakang dari permasalahan yang diangkat maka dapat dirumuskan pokok-pokok permasalahannya sebagai berikut:

1. Bagaimana merancang dan mengembangkan *tools* automatisasi pembuatan diagram *flowgraph* dan perhitungan *cyclomatic complexity* guna mendukung pengujian *white box* dengan teknik *basis path* pada aplikasi berbasis bahasa Golang?
2. Apa saja hasil pengujian *white box* menggunakan teknik *basis path* dengan bantuan *tools* automatisasi pembuatan diagram *flowgraph* dan perhitungan *cyclomatic complexity* yang diperoleh pada *website* HiColleagues?
3. Apa saja rekomendasi perbaikan untuk *website* HiColleagues berdasarkan temuan dari pengujian yang telah dilakukan?

1.3. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah untuk merancang dan mengembangkan *tools* automatisasi pembuatan diagram *flowgraph* dan *cyclomatic complexity* pada aplikasi berbasis bahasa Golang untuk mendukung pengujian *white box* dengan *basis path* pada *website* HiColleagues.

1.4. Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini adalah dari perancangan dan pengembangan *tools* automatisasi untuk pembuatan diagram *flowgraph* dan *cyclomatic complexity* pada aplikasi berbasis bahasa Golang untuk mendukung pengujian *white box* dengan *basis path* pada *website* HiColleagues memberikan kemudahan dan mempercepat proses pembuatan diagram *flowgraph* serta perhitungan *cyclomatic complexity* sehingga mendukung pengujian *white box* dengan *basis path* dengan lebih efisien dan akurat.

1.5. Batasan Masalah

Adapun ruang batasan masalah dalam penelitian adalah sebagai berikut:

1. Penelitian yang dilakukan berfokus pada pengujian internal *website* HiColleagues.

2. Pengujian hanya mencakup fitur yang telah selesai dikembangkan pada *website* HiColleagues dan tidak mencakup fitur yang masih dalam tahap pengembangan.
3. Pengujian yang dilakukan berada pada REST API *website* HiColleagues yang kategori filenya *usecase*.
4. *Tools* otomatisasi yang dikembangkan menggunakan bahasa Golang.
5. Otomatisasi hanya pada pembuatan *flowgraph* dan perhitungan *cylomatic complexity* (CC), penentuan *indepependent path* dan *deriving test case* masih dilakukan secara konvesional.
6. Hasil akhir dari penelitian ini terdiri dari hubungan nilai *cylomatic complexity* (CC) dengan tingkat resiko dalam prosedur pada sistem *website* HiColleagues serta rekomendasi perbaikan yang ditemukan dari hasil pengujian.
7. *Tools* otomatisasi yang dikembangkan berbasis *dekstop application.*

Halaman ini sengaja dikosongkan