

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam pengembangan perangkat lunak, pengujian adalah proses yang sangat penting karena secara langsung mempengaruhi kualitas akhir sebuah perangkat lunak, sehingga harus dilakukan dengan pendekatan yang terstruktur dan sistematis (Gutama dkk., 2019). Pengujian perangkat lunak umumnya dibagi menjadi 2 kategori utama, yaitu black-box dan white-box, pada pengujian white-box, terdapat salah satu metode pengujian yang dikenal sebagai pengujian alur kontrol (control flow testing) yang sangat bergantung pada pemahaman tentang alur dalam kode program (Putri dkk., 2024). Alur kontrol menunjukkan bagaimana instruksi dijalankan, bercabang atau berulang di dalam program, pemahaman ini penting untuk memastikan bahwa program berfungsi dengan benar dan bebas dari kesalahan.

Control Flow Graph (CFG) adalah metode yang digunakan untuk memodelkan alur kontrol dalam kode program (Shiddiq, 2022) di mana setiap blok instruksi atau pernyataan digambarkan sebagai simpul (node) dan hubungan alur eksekusinya sebagai garis penghubung (edge). CFG memungkinkan pemahaman, analisis, dan penelusuran alur dalam kode dengan lebih mudah, serta menjadi metode yang sering digunakan dalam tahap unit test untuk menghasilkan test case yang tepat. Setiap jalur atau cabang dalam CFG mewakili skenario eksekusi yang mungkin terjadi dalam kode, yang mana hal ini sangat membantu dalam pengujian program khususnya pengujian alur kontrol.

Namun membuat dan menganalisis CFG secara manual pada kode program sangat memakan waktu dan berpotensi menimbulkan kesalahan (Pamela dkk., 2021). Biasanya, proses ini melibatkan pembacaan kode program secara langsung dan membayangkan jalur eksekusi dari kode program yang ingin dianalisis, hal ini menciptakan kebutuhan akan alat bantu yang dapat membangun dan memvisualisasikan CFG secara otomatis beserta jalurnya. Aplikasi visualisasi CFG tidak hanya mempercepat proses analisis, tetapi juga meningkatkan akurasi dalam memahami alur kontrol program, baik dalam pengembangan perangkat lunak

maupun dalam aktivitas seperti pembelajaran pemrograman.

Melalui penelitian ini, aplikasi yang dikembangkan diharapkan mampu menjadi solusi untuk membantu pembuatan test case dalam pengujian unit khususnya pengujian alur kontrol. Dengan menghasilkan CFG secara otomatis, mengidentifikasi jalur eksekusi dan kemampuan untuk menguji test case yang dimasukkan. Selain itu, aplikasi ini juga diharapkan dapat bermanfaat untuk proses pembelajaran pemrograman, dengan mengetahui alur eksekusi dalam sebuah program.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah:

- 1) Bagaimana merancang dan mengembangkan aplikasi yang mampu memvisualisasikan Control Flow Graph secara otomatis, termasuk semua jalur eksekusi yang mungkin dilewati?
- 2) Bagaimana memastikan aplikasi yang dikembangkan dapat membantu dalam pembuatan test case untuk kebutuhan pengujian unit?
- 3) Bagaimana aplikasi yang dikembangkan dapat menguji jalur-jalur eksekusi berdasarkan test case yang diinputkan?

1.3 Batasan Masalah

Agar ruang lingkup penelitian tidak terlalu luas, batasan masalah yang diterapkan adalah sebagai berikut:

1. Aplikasi ini hanya akan menganalisis alur kontrol pada kode program yang ditulis dalam bahasa pemrograman Python.
2. Aplikasi ini hanya dirancang untuk membantu analisis kode dalam pengujian alur kontrol pada sebuah method.
3. Aplikasi akan dibatasi pada pemetaan alur kontrol dasar, yaitu: pernyataan kondisional (if-else), perulangan (for dan while), dan percabangan lainnya yang umum dalam struktur program.
4. Visualisasi yang ditampilkan hanya berupa node dan edge yang terhubung, visualisasi yang lebih kompleks tidak akan diimplementasikan.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Penelitian ini bertujuan untuk mengembangkan aplikasi yang secara otomatis memvisualisasikan Control Flow Graph (CFG) dan menampilkan jalur yang dilewati oleh program sehingga bisa mempercepat proses pembuatan test case dalam pengujian unit khususnya untuk alur kontrol.

1.4.2 Manfaat

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Mempermudah analisis kode program dengan menyediakan alat visualisasi yang membantu memahami alur eksekusi kode.
2. Mempercepat pembuatan test case karena mempermudah prosesnya dengan cara menyediakan CFG secara otomatis beserta jalur dan juga *Cyclomatic Complexity* untuk menentukan jumlah dari jalur pengujian.
3. Menguji jalur yang ditemukan dengan menggunakan *test case* yang diinputkan.
4. Mengurangi potensi kesalahan dalam analisis alur kontrol dengan menampilkan jalur eksekusi kode yang sulit dilacak secara manual.
5. Mendukung kegiatan pembelajaran dengan menyediakan alat bantu untuk memahami alur eksekusi dari sebuah program.