BAB I
PENDAHULUAN

1.1 Latar Belakang Penelitian

Perangkat lunak seluler memiliki batasan pada sumber daya yaitu CPU dan memori (Amalfitano dkk., 2020; Cruz & Abreu, 2019; Mannan dkk., 2016). Menurut Amalfitano dkk (dalam Toffalini dkk., 2019) dalam investigasinya bahwa kode program yang buruk yang berdampak penurunan kinerja khususnya memori terhadap proses berjalannya perangkat lunak seluler dalam jangka waktu yang lama. Selain itu, penurunan kinerja pada CPU diakibatkan adanya kode program yang buruk (Venters dkk., 2018). Berdasarkan masalah kode program tersebut, perlu adannya investigasi lebih lanjut terkait kode program dan pengaruh non-fungsional khususnya CPU dan memori terhadap perangkat seluler.

Kode program yang buruk disebabkan buruknya praktik menulis kode dan implementasi yang dapat menyebabkan pemeliharaan dalam jangka panjang (Habchi dkk., 2021; Mannan dkk., 2016). Penyebab adanya praktik menulis kode dan implementasi yang buruk karena kurangnya kesadaran dan kurangnya pengalaman pengembangan serta mengembangkan perangkat lunak dengan cepat karena tekanan tenggat waktu (Hecht dkk., 2016; Rasool & Arshad, 2017). Oleh karena itu kode program menjadi tantangan pada pengembang.

Tantangan tersebut dapat ditemui dalam pemeliharaan perangkat lunak (*software maintenance*), upaya yang dilakukan dalam pemeliharaan yaitu melakukan tinjauan dan modifikasi serangkaian kode program secara manual (Stefanović dkk., 2020). Menurut Stefanović dkk (dalam Brian & Jacob, 2007, hlm. 57) estimasi yang dibutuhkan untuk tinjauan manual membutuhkan waktu yang sangat lama dan tidak memungkinkan untuk melakukan tinjauan di setiap baris kode. Berdasarkan hal tersebut, tinjauan manual tersebut perlu adanya pengganti.

Salah satu perangkat lunak yang umum digunakan sebagai pendukung pemeliharaan perangkat lunak adalah *Automatic Static Analysis Tools* (ASATs) sebagai analisis kode statis. ASATs digunakan untuk menganalisis kode program secara otomatis dan menyediakan perbaikan saran modifikasi pada serangkaian kode (Marcilio dkk., 2020). Selain itu, ASATs dirancang untuk mengidentifikasi

berbagai kekurangan tanpa mengeksekusi kode program (Nikolic dkk., 2021). Salah satu ASATs yang populer digunakan bernama SonarQube yang mendukung bahasa pemrograman khususnya Java dan Kotlin. SonarQube digunakan untuk pemeriksaan kualitas kode yang dapat memberikan laporan rinci mengenai masalah pada kode program. Laporan tersebut mencakup *bugs*, *code smells*, *security vulnerabilities*, *code duplication* (De Andrade Gomes dkk., 2017; Marcilio dkk., 2019; Mitra dkk., 2022; Stefanović dkk., 2020).

Selama pemeliharaan perangkat lunak masih terdapat keputusan yang buruk oleh pengembang dengan mengabaikan pelanggaran aturan yang menyebabkan masalah yang bertentangan sehingga menyebabkan dampak negatif pada pemeliharaan perangkat lunak, salah satu pelanggaran yang merusak pemeliharaan perangkat lunak adalah *code smells* (Etemadi Someoliayi dkk., 2022; Marcilio dkk., 2019). Penyebab adanya *code smells* karena gaya pemrograman yang buruk, kurangnya dokumentasi, dan kompleksitas yang sangat tinggi (De Andrade Gomes dkk., 2017). Selain itu *code smells* berdampak pada kebutuhan non-fungsional, salah satunya penggunaan sumber daya CPU dan memori (Hecht dkk., 2016). Faktanya, perangkat lunak harus mempertimbangkan kinerja dan optimasi sumber daya sebagai faktor kesuksesan perangkat lunak seperti CPU dan Memori (Alkandari dkk., 2021; Habchi dkk., 2019; Kwan Kim, 2017; Oliveira dkk., 2018). Apabila terdapat fenomena perangkat lunak yang memiliki kelemahan terkait penggunaan sumber daya dengan cepat, maka akan melahirkan masalah baru terhadap pengguna yang dapat beralih atau menghapus pemasangan perangkat lunak dari perangkat (Oliveira dkk., 2018). Dalam survei penelitian membuktikan bahwa 75,2% karena kinerja berlebihan sehingga perangkat lunak tidak berjalan (*crashes*) dan 43,6% konsumsi memori yang besar, sehingga pengguna menghapus pemasangan perangkat lunak seluler (Huseynov, 2020; Ickin dkk., 2017). Namun penelitian serupa mengenai *code smells* terhadap sumber daya masih sedikit dan belum sepenuhnya tereksplorasi dan masih menggunakan perangkat virtual (*emulator*) (Alkandari dkk., 2021). Dalam penelitian Guerra-Manzanares & Välbe, (2022) menegaskan bahwa perilaku *emulator* dan perangkat fisik berbeda. Oleh karena itu, perlu adanya eksplorasi *code smells* yang berdampak pada sumber daya dan memperbaikinya secara efektif serta memastikan pada perangkat fisik.

Berdasarkan konteks permasalahan di atas, *code smells* adalah pelanggaran kode program yang perlu dianalisis dan dapat membuat keterbatasan pada konsumsi penggunaan CPU dan memori, sehingga hal tersebut dapat menjadi motivasi untuk mengidentifikasi *code smells* pada perangkat lunak Android untuk menjadi target penelitian. Analisis *code smells* dilakukan menggunakan SonarQube dengan menganalisis kode program dan memperbaiki dengan memodifikasi kode program secara manual. Setelah analisis *code smells* dilanjutkan dengan *refactoring* tunggal dan kumulatif untuk menurunkan intensitasnya dengan memperhatikan kuantitas *code smells*. Hasil dari *refactoring* akan menghasilkan versi perangkat lunak yang akan dibandingkan dengan perangkat lunak yang belum dilakukan *refactoring*. Selain itu, hasil perbandingan akan diteliti lebih lanjut mengenai konsumsi penggunaan CPU dan memori menggunakan ADB dengan ObreusDroid.

1.2 Rumusan Masalah Penelitian

Berdasarkan penjelasan latar belakang yang disampaikan, terdapat beberapa rumusan masalah dalam penelitian ini antara lain:

1. Bagaimana dampak *refactoring* dengan *Automatic Static Analysis Tools* (ASATs) terhadap intensitas *code smells*? (**RQ1**)
2. Apakah *refactoring code smells* dapat mengurangi konsumsi penggunaan CPU pada perangkat lunak Android? (**RQ2**)
3. Apakah *refactoring code smells* dapat mengurangi konsumsi penggunaan memori pada perangkat lunak Android? (**RQ3**)

1.3 Tujuan Penelitian

Adapun tujuan penelitian berdasarkan rumusan masalah yang disampaikan dalam penelitian ini antara lain:

1. Mengidentifikasi dampak *refactoring* dengan *Automatic Static Analysis Tools* (ASATs) terhadap intensitas *code smells*.
2. Mengidentifikasi konsumsi penggunaan CPU perangkat lunak Android setelah dilakukan *refactoring code smells.*
3. Mengidentifikasi konsumsi penggunaan memori perangkat lunak Android setelah dilakukan *refactoring code smells*.

1.4 Manfaat Penelitian

Adapun manfaat penelitian yang diperoleh setelah melakukan penelitian ini antara lain:

1. Hasil dari penelitian ini secara akademis dapat dijadikan acuan penelitian terkait *refactoring* terhadap *code smells* mengenai perbedaan *code smells* yang dianalisis dan konsumsi penggunaan CPU dan memori yang masih belum signifikan.
2. Hasil dari penelitian ini secara praktis dapat dijadikan sebagai acuan *refactoring* secara efektif untuk pengembang apabila perangkat lunak memiliki *code smells*, sehingga perangkat lunak pada pengguna menjadi konsisten sesuai artefak.
3. Hasil dari penelitian ini dapat dijadikan sebagai acuan pemeliharaan perangkat lunak secara singkat untuk meningkatkan kinerja perangkat lunak dalam lingkup perusahaan yang memiliki bidang teknologi sehingga dapat menghemat biaya, tenaga, dan waktu.

1.5 Batasan Penelitian

Penelitian ini akan dilakukan dengan beberapa batasan untuk memfokuskan pada tujuan dan topik penelitian sebagai berikut:

1. Penelitian berfokus pada pemeliharaan perangkat lunak dengan meninjau kode program yang telah dianalisis menggunakan *Automatic Static Analysis Tools* (ASATs) dan dilakukan *refactoring* terkait *code smells* pada kode program perangkat lunak berorientasi objek atau *Object Oriented* (OO).
2. Penelitian menggunakan kumpulan perangkat lunak seluler Android (*corpus*) yang ditulis dalam bahasa Java atau Kotlin yang telah digunakan pada penelitian sebelumnya dan diukur dengan metrik-metrik penelitian sebelumnya.
3. Penelitian berfokus dengan membandingkan perangkat lunak seluler Android versi orisinal dengan yang telah dilakukan *refactoring* pada perangkat fisik.

1.6 Struktur Organisasi Skripsi

Sistematika penulisan yang berperan sebagai pedoman untuk peneliti, supaya dalam penulisan lebih sistematis dan terstruktur untuk mencapai tujuan akhir dari penelitian. Secara garis besar, sistematika penulisan yang digunakan

dalam penelitian ini mengacu pada Pedoman Penulisan Karya Ilmiah Universitas Pendidikan Indonesia Tahun 2019. Struktur penulisan terdiri dari lima bab, antara lain:

BAB I PENDAHULUAN

Bagian BAB I membahas mengenai subbab pada BAB I yaitu latar belakang penelitian, rumusan masalah penelitian, tujuan penelitian, manfaat penelitian, batasan penelitian, hipotesis penelitian, dan struktur organisasi skripsi. Latar belakang penelitian membahas mengenai topik penelitian dan bidang penelitian berdasarkan permasalahan dan fenomena serta motivasi terkait. Rumusan masalah penelitian menjelaskan poin-poin pertanyaan penelitian berdasarkan identifikasi masalah spesifik yang menjadi permasalahan dari penelitian sebelumnya dan dijadikan sebagai tujuan penelitian. Manfaat penelitian membahas kontribusi penelitian yang kemudian dijadikan sebagai hasil penelitian yang telah dilakukan. Batasan penelitian membahas mengenai ruang lingkup penelitian yang menjadi fokus penelitian. Hipotesis penelitian menjelaskan jawaban sementara terhadap rumusan masalah. Struktur organisasi skripsi menjelaskan secara ringkas mengenai isi dari setiap BAB yang dijelaskan.

BAB II KAJIAN PUSTAKA

Bagian BAB II menjelaskan mengenai beberapa penelitian yang serupa untuk menemukan perbedaan dari penelitian yang dilakukan saat ini (*state of the art*). Selain itu terdapat dasar teori yang berdasarkan teoritis yang menjadi acuan untuk pelaksanaan penelitian, mencakup teknologi yang berkaitan untuk digunakan selama penelitian berlangsung.

BAB III METODE PENELITIAN

Bagian BAB III menjelaskan tahapan yang bersifat prosedural, seperti desain penelitian, *corpus*, instrumen penelitian, prosedur penelitian, dan analisis data. Desain penelitian merepresentasikan aliran penelitian yang dilakukan berdasarkan referensi kerangka kerja metodologi penelitian yang sesuai. *Corpus* membahas mengenai kumpulan perangkat lunak yang digunakan dalam penelitian. Instrumen penelitian menjelaskan mengenai alat perangkat lunak dan metrik serta spesifikasi perangkat yang digunakan dalam penelitian. Prosedur penelitian menjelaskan tahapan penelitian yang dilakukan berdasarkan aliran penelitian yang

sudah pernah dilakukan oleh penelitian sebelumnya. Analisis data membahas mengenai formula-formula dan perangkat lunak untuk analisis statistik dalam penelitian.

BAB IV TEMUAN DAN PEMBAHASAN

Bagian BAB IV Temuan dan Pembahasan menjelaskan hasil yang diperoleh selama proses penelitian yang dilakukan, Bab ini akan menjawab rumusan masalah penelitian, tujuan penelitian, dan hipotesis serta hasil lain berdasarkan batasan penelitian yang dijelaskan secara rinci. Selain itu hasil dari penelitian dilakukan interpretasi yang dikaitkan dengan kajian pustaka.

BAB V SIMPULAN, IMPLIKASI, dan REKOMENDASI

Bagian BAB V Simpulan, Implikasi, dan Rekomendasi menjelaskan mengenai kesimpulan yang diperoleh dari interpretasi hasil dan temuan dari penelitian yang telah dilakukan. Bab ini akan menjelaskan dampak yang bermanfaat pada penelitian serupa. Selain itu dilanjutkan dengan saran penelitian untuk ke depannya.