

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring berjalannya waktu setiap individu mulai melibatkan teknologi di dalam kegiatan sehari-hari. Hal ini dikarenakan pengaruh dan peranan teknologi informasi terhadap kehidupan manusia sangatlah penting [2]. Teknologi informasi muncul sebagai akibat semakin merebaknya globalisasi. Waktu demi waktu, teknologi pun mulai membawa perubahan yang signifikan dalam berbagai bidang, khususnya dalam bisnis [3]. Beberapa perusahaan bahkan mulai membuat aplikasi untuk menunjang proses bisnis perusahaannya. Aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi *user* dengan menggunakan kemampuan komputer [4]. Tidak terkecuali PT Astra Sedaya Finance yang telah mengembangkan Astra Credit Companies (ACC) sejak tahun 1994. Astra Credit Companies sendiri memiliki *core* pada proses bisnis *leasing* (kredit) dengan mengelompokkannya ke dalam 3 produk, yaitu Multiguna untuk peminjaman uang dengan jaminan BPKB mobil, *New Car* untuk kredit mobil baru, dan *Used Car* untuk kredit untuk mobil bekas/*second*.

Berdasarkan uraian di atas, salah satu produk yang menyokong pertumbuhan *profit* di Astra Credit Companies adalah produk Multiguna, atau biasa dikenal dengan MGU. Beberapa cara telah dilakukan perusahaan untuk mendapatkan lebih banyak lagi peng-aplikasian MGU. Salah satu cara terbaik adalah menawarkan produk MGU kepada pelanggan dan mantan pelanggan dari Astra Credit Companies itu sendiri. Cara ini disebut dengan MGU WIN. Dahulu, data MGU WIN diberikan dengan menggunakan *file* Excel, dan dikirim melalui jaringan lokal perusahaan. Namun, seiring berjalannya waktu, cara pengiriman

file Excel ini dirasa tidak efektif karena pihak Departemen MGU dan pejabat di *Head Office* tidak bisa memantau aktivitas penawaran dan penjualan oleh *sales* di cabang. Maka, terbentuklah ide untuk menjadikan proses penawaran produk MGU WIN ke bentuk digital. Ide ini pun direalisasikan sehingga lahirlah sebuah aplikasi *mobile* yang bernama ACC Leads. Selain dapat menerima data MGU WIN, pada *enhancement* terbaru, ACC Leads juga mendapat penambahan fitur, dimana ACC Leads dapat menerima data-data pengajuan ACC Cash dan pengajuan secara digital yang berasal dari banyak *source* seperti YUNA, Facebook, Moxa, dan lainnya

Tentunya, setelah *enhancement* tersebut, ACC Leads harus melalui beberapa tahapan sebelum nantinya aplikasi tersebut dirilis dengan versi terbarunya untuk dipakai oleh pihak internal perusahaan. Salah satu tahapan yang harus dilalui adalah pengujian untuk mengetahui kelayakan aplikasi tersebut digunakan oleh *user*. Selain untuk menguji kelayakan, pengujian tersebut juga berfungsi untuk melihat *bugs* yang ada pada aplikasi ACC Leads. Hal ini sangatlah penting karena aplikasi yang memiliki cacat/kesalahan dapat menimbulkan kerugian. Kerugian yang diakibatkan-pun beragam dan dapat memberikan dampak yang signifikan pada semua *stakeholder* [5]. Pengujian juga sebaiknya dapat menemukan kesalahan yang tidak disengaja [6]. Salah satu kesalahan yang sering terjadi adalah aplikasi masih bisa menerima masukan, memproses data, dan berfungsi dengan baik saat menerima masukan yang tidak diharapkan. Contohnya adalah aplikasi tidak mengeluarkan *warning* pada saat terdapat kesalahan *input* tipe data dan aplikasi masih bisa memproses masukan alfabet pada *field* nomor telepon yang seharusnya menggunakan angka. Hal-hal seperti ini tentunya perlu menjadi perhatian khusus bagi ACC dimana ACC Leads harusnya bisa mencegah hal-hal tersebut terjadi mengingat tingginya kemungkinan terjadinya *human error* dalam penginputan data. Namun sayangnya,

pengujian yang ada pada saat ini lebih berfokus kepada fungsi-fungsi besar ACC Leads sehingga pengujian untuk mengetahui respon aplikasi tersebut saat menerima inputan yang tidak diharapkan, baik berupa tipe data ataupun panjang data yang tidak diharapkan perlu dilakukan. Selain itu, penting bagi setiap *quality control* untuk mencari bug sebanyak-banyaknya dalam pengujian sebuah aplikasi. Hal ini bertujuan agar tidak terdapat celah sedikitpun yang menyebabkan aplikasi tidak dapat berjalan dengan *maximal* [7].

Black box testing merupakan salah satu metode pengujian yang cukup sering digunakan. *Black box testing* dianalogikan seperti halnya kita hanya melihat tampilan luar suatu benda tanpa apa yang ada di baliknya [8]. Analogi ini sama seperti pengujian yang akan dilakukan, yaitu hanya menguji fungsionalitas suatu aplikasi tanpa perlu mengetahui proses apa yang terjadi di belakangnya sampai fungsi tersebut bisa berjalan [8]. Pengujian *black box* ini-pun terdiri atas beberapa teknik, diantaranya adalah *Boundary Value Analysis*, *Equivalence Partitioning*, *Comparison Testing*, *Robustness Testing*, *Sample Testing* dan lain-lain [9].

Boundary Value Analysis (BVA) adalah salah satu teknik pengujian dalam metode *black box testing*. Teknik ini melakukan pengujian pada batas atas dan batas bawah nilai yang di-*input* pada sistem. Contohnya jika masukan adalah senilai a , maka pengujian yang akan dilakukan adalah $(a-1)$, (a) , dan $(a+1)$ [9]. Masukan-masukan ini nantinya diharapkan dapat memberikan *feedback* apakah aplikasi tetap akan memproses data ataupun sebaliknya jika masukan yang dimasukkan tidak sesuai dengan apa yang diharapkan.

Equivalence Partitioning (EP) adalah salah satu teknik pengujian dalam metode *black box testing* yang dengan mengelompokkan inputan-inputan ke dalam partisi tertentu sesuai jenisnya [10]. Pengelompokkan

ini sangat berguna untuk mengetahui apakah aplikasi masih dapat menerima masukan dari tipe data yang tidak sesuai. *Equivalence Partitioning* juga berorientasi untuk membuat beberapa kondisi berdasarkan kasus uji yang diperoleh. *Test case* digunakan untuk menghasilkan uji coba aplikasi untuk diuji. Hasil yang didapat dari kondisi ini adalah nilai yang benar/*valid* atau tidak benar/tidak *valid*. *Input-an* pada *EP* dapat berupa nilai *range*, nilai numerik, atau *boolean* [10].

Umumnya, terdapat 2 jenis pengujian yang sering digunakan, yaitu *manual testing* dan *automation testing* [11]. *Manual testing* sendiri membutuhkan waktu lebih lama dari *automation testing*. Secara keseluruhan, *automation testing* umumnya lebih efektif dan lebih baik dalam mengungkap kesalahan [1]. *Automation testing* sendiri biasanya dilakukan dengan bantuan sebuah *tool* yang bisa menghasilkan *test case* sesuai spesifikasi program [12].

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan diatas, maka permasalahan yang dirumuskan adalah :

1. Pengujian yang dilakukan secara *manual* cenderung membutuhkan waktu yang lebih lama.
2. Belum adanya pengujian aplikasi *mobile ACC Leads* untuk melihat apakah aplikasi *ACC Leads* memberikan *warning* dan mencegah *user* ketika menginputkan masukan berupa tipe dan panjang data yang tidak diharapkan.

1.3 Pertanyaan Penelitian

Berdasarkan uraian di atas, adapun pertanyaan penelitian yang sesuai adalah :

1. Bagaimanakah cara lain agar pengujian dapat berjalan dengan lebih cepat ?
2. Bagaimanakah pengujian yang tepat untuk mengetahui apakah aplikasi ACC Leads memberikan *warning* dan mencegah *user* ketika menginputkan masukan berupa tipe dan panjang data yang tidak diharapkan ?

1.4 Tujuan

Tujuan dari penelitian ini adalah :

1. Menerapkan *automation testing* sebagai pengganti *manual testing* yang biasanya cenderung membutuhkan waktu yang lebih lama.
2. Membuat pengujian pada aplikasi ACC Leads untuk mengetahui apakah ACC Leads memberikan *warning* dan mencegah *user* ketika menginputkan masukan berupa tipe dan panjang data yang tidak diharapkan.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Penelitian ini hanya akan menguji aplikasi ACC Leads pada *platform mobile* dengan sistem operasi Android.
2. *Scope* yang akan diuji pada penelitian ini hanyalah fitur yang bisa menerima masukan (baik berupa angka maupun huruf) pada aplikasi *mobile* ACC Leads.
3. Pengujian dalam penelitian ini hanya akan menggunakan dua teknik dalam *Black box testing*, yaitu teknik *Boundary Value Analysis* dan *Equivalence Partitioning*.

4. Pengujian pada *Boundary Value Analysis* tidak akan dilakukan pada *field* yang tidak memiliki batas panjang maksimal inputan / *no maximal length*
5. Penelitian hanya akan didasarkan pada pengujian yang dilakukan saat penelitian saja.



1.6 Manfaat Penelitian

Penelitian ini juga membawa manfaat sebagai berikut :

1. Bagi keilmuan atau perkembangan ilmu

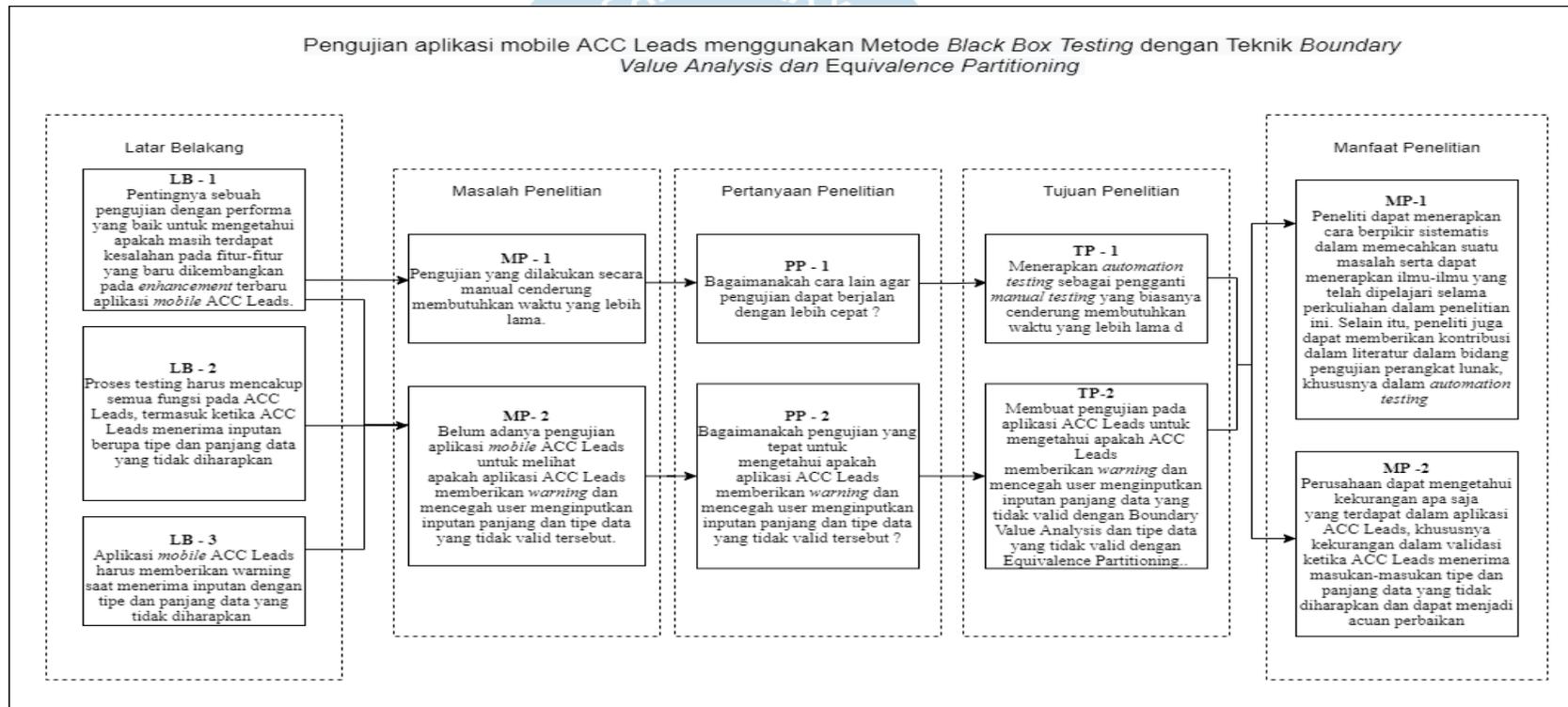
Peneliti dapat menerapkan cara berpikir sistematis dalam memecahkan suatu masalah serta dapat menerapkan ilmu-ilmu yang telah dipelajari selama perkuliahan dalam penelitian ini. Selain itu, peneliti juga dapat memberikan kontribusi dalam literatur dalam bidang pengujian perangkat lunak, khususnya dalam *automation testing*.

2. Bagi organisasi/perusahaan/dll

Perusahaan dapat mengetahui kekurangan apa saja yang terdapat dalam aplikasi ACC Leads, khususnya kekurangan dalam validasi ketika ACC Leads menerima masukan-masukan yang tidak diharapkan serta dapat menjadi rujukan perbaikan

1.7 Bagan Keterkaitan

Gambar 1.1 merupakan diagram keterkaitan antara latar belakang, masalah, pertanyaan penelitian, tujuan penelitian, dan manfaat penelitian.



Gambar 1.1 Bagan Keterkaitan