

BAB I

PENDAHULUAN

1.1. Latar Belakang

Persaingan bisnis di era globalisasi ini semakin ketat. Perusahaan-perusahaan harus terus berinovasi dan menjaga kualitas produk mereka agar tetap kompetitif di pasar. Salah satu aspek yang paling menentukan keberhasilan atau kegagalan suatu perusahaan adalah kualitas produk yang dihasilkan. Kualitas produk yang baik tidak hanya mempengaruhi kepuasan pelanggan tetapi juga berkontribusi terhadap citra perusahaan dan loyalitas pelanggan. *Quality Assurance* (QA) menjadi salah satu elemen kunci dalam memastikan produk memenuhi standar kualitas yang ditetapkan [2].

Dalam industri teknologi informasi, khususnya dalam pengembangan perangkat lunak, QA memegang peranan yang sangat penting. Proses pengujian yang baik dapat mengidentifikasi dan memperbaiki kesalahan sebelum produk diluncurkan ke pasar. Namun, tantangan yang dihadapi adalah bagaimana memastikan proses QA berjalan efisien dan efektif. Dengan kompleksitas aplikasi yang semakin meningkat, pengujian manual sering kali tidak cukup untuk menjamin kualitas dan memerlukan waktu yang lama [1]. Di sinilah metode *automation testing* menjadi relevan. *Automation testing* dapat membantu mempercepat proses pengujian, mengurangi kesalahan manusia, dan memastikan cakupan pengujian yang lebih luas[7].

Salah satu *tools* yang digunakan dalam *automation testing* adalah Appium. Appium adalah *framework open-source* yang memungkinkan pengujian aplikasi *mobile* berbasis Android dan iOS. Kelebihan Appium adalah kemampuannya untuk menjalankan pengujian di berbagai *platform* dengan satu set kode yang sama. Appium mendukung berbagai bahasa pemrograman seperti *Java*, *Python*, *Ruby*, dan lain-lain[8]. Dalam konteks PT. Global Loyalty Indonesia, penggunaan

Appium diharapkan dapat meningkatkan efisiensi proses QA dan memastikan kualitas aplikasi mobile yang dihasilkan[10].

Pemilihan bahasa pemrograman juga menjadi faktor penting dalam *automation testing*. Java adalah salah satu bahasa yang paling banyak digunakan dalam pengembangan perangkat lunak dan *automation testing*. Java menawarkan stabilitas, performa yang baik, dan dukungan komunitas yang luas. Dengan menggunakan Java dalam *automation testing*, tim QA dapat memanfaatkan berbagai *library* dan *framework* yang tersedia untuk meningkatkan efisiensi pengujian[9]. Java juga dikenal mudah dipelajari dan digunakan, sehingga memudahkan tim dalam mengembangkan dan memelihara *script* pengujian[6].

Tanpa *automation testing*, proses QA sering kali menjadi repetitif dan memakan waktu. Pengujian manual membutuhkan sumber daya manusia yang besar dan rawan kesalahan. Misalnya, pengujian regresi yang harus dilakukan setiap kali ada perubahan pada aplikasi. Dengan *automation testing*, proses pengujian dapat diotomatisasi sehingga mengurangi beban kerja tim QA dan memastikan hasil yang konsisten[5]. *Automation testing* juga memungkinkan pengujian dilakukan di luar jam kerja, sehingga waktu pengujian dapat lebih fleksibel dan efisien[4].

Meskipun *automation testing* menawarkan banyak keuntungan, implementasinya juga memiliki tantangan tersendiri. Salah satu tantangan terbesar adalah investasi awal yang diperlukan untuk mengembangkan dan memelihara *script* pengujian otomatis. Selain itu, *tools automation testing* seperti Appium memerlukan keahlian teknis yang tinggi dan pemahaman mendalam tentang *framework* dan bahasa pemrograman yang digunakan[3]. Tim QA harus dilatih secara khusus untuk dapat menggunakan *tools* ini secara efektif. Selain itu, *automation testing* juga tidak selalu dapat menggantikan pengujian manual, terutama dalam kasus pengujian eksploratif yang membutuhkan intuisi dan kreativitas manusia[4].

Appium memiliki beberapa keunggulan dibandingkan dengan tools *automation testing* lainnya. Pertama, Appium mendukung pengujian lintas *platform*, yang memungkinkan satu set script pengujian dapat digunakan untuk aplikasi di berbagai sistem operasi, seperti Android dan iOS. Kedua, Appium menggunakan API *WebDriver* yang sudah standar di industri, sehingga memudahkan integrasi dengan *tools* lain dan memanfaatkan pengetahuan yang sudah ada di tim QA. Ketiga, sebagai *framework open-source*, Appium tidak memerlukan biaya lisensi, yang dapat mengurangi biaya operasional perusahaan[8].

Penggunaan bahasa pemrograman Java dalam Appium juga menawarkan beberapa keunggulan. Java adalah bahasa yang *robust* dan memiliki performa tinggi, yang penting dalam pengujian aplikasi *mobile* yang kompleks. Java memiliki ekosistem yang luas dengan banyak *library* dan *framework* yang dapat digunakan untuk mendukung *automation testing*. Dukungan komunitas yang luas juga berarti bahwa tim QA dapat dengan mudah menemukan solusi dan bantuan untuk masalah yang dihadapi selama pengujian[6]. Java juga menawarkan dukungan yang baik untuk paralel *testing*, yang dapat meningkatkan efisiensi dan kecepatan pengujian[1].

Proses pengujian manual seringkali melibatkan tugas-tugas repetitif yang membosankan dan memakan waktu. Misalnya, pengujian regresi yang harus dilakukan setiap kali ada pembaruan pada aplikasi. Tugas-tugas ini tidak hanya membosankan tetapi juga rawan kesalahan manusia, yang dapat mengakibatkan kualitas produk yang tidak konsisten[5]. Dengan *automation testing*, tugas-tugas repetitif ini dapat diotomatisasi, sehingga tim QA dapat fokus pada pengujian yang lebih kompleks dan kritis. Hal ini tidak hanya meningkatkan efisiensi tetapi juga memastikan kualitas yang lebih konsisten dan andal[4].

Dalam implementasinya di PT. Global Loyalty Indonesia, Appium telah berhasil meningkatkan efisiensi dan efektivitas proses QA. Dengan menggunakan Appium, tim QA di PT. Global Loyalty Indonesia dapat menjalankan pengujian

otomatis di berbagai perangkat dan sistem operasi dengan satu set *script* pengujian yang sama. Hal ini tidak hanya menghemat waktu tetapi juga memastikan bahwa aplikasi *mobile* yang dihasilkan memiliki kualitas yang konsisten di berbagai *platform*. Penggunaan Java dalam pengembangan *script* pengujian memungkinkan tim QA untuk memanfaatkan berbagai *library* dan *framework* yang tersedia, sehingga meningkatkan efisiensi pengujian[10].

Implementasi *automation testing* menggunakan Appium dan Java telah memberikan dampak positif yang signifikan pada PT. Global Loyalty Indonesia. Efisiensi proses QA meningkat secara drastis, yang memungkinkan tim QA untuk melakukan lebih banyak pengujian dalam waktu yang lebih singkat. Kualitas aplikasi *mobile* yang dihasilkan meningkat, yang berkontribusi terhadap kepuasan pelanggan dan loyalitas pengguna. *Automation testing* juga membantu mengurangi biaya operasional dengan mengurangi kebutuhan akan pengujian manual yang intensif sumber daya.

Untuk memastikan perbaikan berkelanjutan dalam proses QA, PT. Global Loyalty Indonesia perlu terus mengembangkan dan memperbarui script pengujian otomatis mereka. Pelatihan dan pengembangan tim QA harus menjadi prioritas untuk memastikan mereka memiliki keahlian dan pengetahuan yang diperlukan untuk menggunakan *tools automation testing* secara efektif. PT. Global Loyalty Indonesia juga perlu terus memantau dan mengevaluasi kinerja proses QA mereka untuk mengidentifikasi area yang memerlukan perbaikan dan mengambil tindakan yang tepat[8].

Quality Assurance memainkan peran yang krusial dalam memastikan kualitas produk dan proses di PT. Global Loyalty Indonesia. Dengan adopsi metode *automation testing* menggunakan Appium dan bahasa pemrograman Java, diharapkan efisiensi dan efektivitas proses QA dapat ditingkatkan. Laporan ini akan menguraikan secara rinci bagaimana metode ini diterapkan, tantangan yang dihadapi, dan solusi yang diusulkan untuk perbaikan berkelanjutan. Dengan demikian, diharapkan laporan ini dapat memberikan panduan dan rekomendasi

yang bermanfaat bagi PT. Global Loyalty Indonesia dalam upaya meningkatkan kualitas produk dan daya saing di pasar[1].

1.2. Maksud dan Tujuan Kerja Magang

Sebagai mahasiswa magang di PT Global Loyalty Indonesia dengan peran *Quality Assurance Intern*, terdapat maksud dan tujuan yang menjadi landasan utama dari pengalaman magang ini. Maksud dan tujuan tersebut dirumuskan untuk memberikan arah yang jelas dan memastikan bahwa waktu yang dihabiskan selama magang memberikan manfaat yang maksimal bagi pengembangan keterampilan dan pemahaman di bidang *Quality Assurance*.

1.2.1 Maksud Kerja Magang

- a. Pengalaman Praktis: Maksud utama dari magang ini adalah untuk mendapatkan pengalaman praktis dalam lingkungan kerja nyata di industri loyalitas. Melalui tugas-tugas sebagai *Quality Assurance Intern*, diharapkan dapat memperdalam pemahaman tentang proses pengujian perangkat lunak, pemantauan kualitas, dan peningkatan efisiensi operasional.
- b. Penerapan Teori dalam Praktek: Magang ini memungkinkan penerapan konsep dan teori yang dipelajari di bangku kuliah ke dalam situasi kerja sehari-hari. Hal ini memberikan kesempatan untuk melihat bagaimana prinsip-prinsip *Quality Assurance* diterapkan dalam konteks bisnis nyata.
- c. Peningkatan Keterampilan: Melalui magang ini, diharapkan dapat meningkatkan keterampilan dalam melakukan pengujian perangkat lunak, analisis kebutuhan pengguna, dan evaluasi kualitas produk. Hal ini akan membantu dalam pengembangan kemampuan praktis yang diperlukan untuk bekerja di bidang *Quality Assurance* di masa depan.

1.2.2 Tujuan Kerja Magang

- a. Memahami Proses *Quality Assurance*: Salah satu tujuan utama dari magang ini adalah untuk memahami secara mendalam proses *Quality*

Assurance, termasuk metodologi pengujian, pelaporan *bug*, dan evaluasi kualitas produk secara menyeluruh.

- b. Berpartisipasi dalam Proyek-Proyek: Tujuan lainnya adalah untuk berpartisipasi aktif dalam proyek-proyek *Quality Assurance* yang sedang berlangsung di perusahaan. Dengan demikian, diharapkan dapat memberikan kontribusi yang berarti dan memperluas wawasan tentang praktik terbaik dalam industri.
- c. Meningkatkan Kemampuan Analisis: Tujuan lainnya adalah untuk meningkatkan kemampuan dalam menganalisis kebutuhan pengguna, mengidentifikasi masalah potensial, dan mengusulkan solusi yang efektif. Hal ini akan membantu dalam mengembangkan kemampuan analitis yang diperlukan dalam pekerjaan *Quality Assurance*.
- d. Memperluas Jaringan Profesional: Salah satu tujuan tambahan adalah untuk memperluas jaringan profesional di industri loyalitas dan *Quality Assurance*. Dengan berinteraksi dengan berbagai profesional di perusahaan, diharapkan dapat membangun hubungan yang berkelanjutan dan mendukung perkembangan karir di masa depan.

Dengan menetapkan tujuan dan sasaran yang jelas selama proses magang pada divisi *Quality Assurance* di PT GLI, pengalaman ini akan memberikan landasan yang kuat untuk pengembangan karir di bidang jaminan kualitas dan memastikan keuntungan berkelanjutan dalam mengembangkan keterampilan dan pengetahuan.

1.3. Waktu dan Prosedur Pelaksanaan Kerja Magang

1.3.1 Waktu Pelaksanaan Kerja Magang

Program magang berlangsung selama setidaknya 640 jam, yang berhasil diselesaikan dalam rentang waktu 64 hari kerja dengan kisaran 10 jam kerja per hari. Diawali pada tanggal 19 Februari 2024 dan berakhir pada tanggal 19 Agustus 2024, jadwal magang dilaksanakan pada hari Senin hingga hari Jumat di kantor atau WFO (*Work from Office*), serta hari Sabtu dengan sistem Hybrid. PT. Global Loyalty Indonesia menjadi tempat pelaksanaan program kerja magang,

yang terletak di Alfa Tower, 29th Floor, Jl. Jalur Sutera Barat, Kav.7-9, Alam Sutera. Penempatan dilakukan di Departemen *Product Operation* dengan posisi magang di *Quality Assurance*. Durasi program magang ini dilaksanakan selama 6 bulan.

Selama waktu tersebut, waktu magang dibagi ke beberapa proyek berbeda pada setiap bulannya. Setiap proyek memiliki cakupan dan tujuan yang unik, memungkinkan untuk terlibat dalam berbagai aspek pengembangan perangkat lunak dan pengujian. Dengan demikian, didapatnya pengalaman yang beragam dan mendalam dalam industri teknologi informasi, serta kesempatan untuk memperluas pengetahuan dan keterampilannya di berbagai bidang terkait. Selain itu, berpartisipasi dalam proyek-proyek yang berbeda juga memungkinkan untuk bekerja dengan beragam tim dan memperluas jaringan profesionalnya. Hal ini menjadikan program kerja magang ini sebagai pengalaman yang berharga dan bervariasi dalam mempersiapkan diri untuk karir di dunia kerja. Timeline kerja magang dijabarkan pada tabel 1.1.



Tabel 1.1 Timeline Kerja Magang

Pekerjaan	Febuari		Maret				April				Mei			
	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Pengenalan Perusahaan, Mempelajari <i>user guide</i> , instalasi, dan mempelajari <i>query-query</i>	■													
Melakukan <i>full cycle rollout</i> aplikasi beserta fitur pada aplikasi AlfaGift	■													
Bergabung dengan SCRUM tim	■	■	■	■	■	■	■	■	■	■	■	■	■	■
<i>Project e-Service</i> (skenario, <i>testing</i> , <i>bug report</i> , dokumentasi)		■	■	■	■	■								
<i>Project Virtual Product</i> (skenario, <i>testing</i> , <i>bug report</i> , dokumentasi)					■		■	■	■	■	■	■	■	■

1.3.2 Prosedur Pelaksanaan Kerja Magang

Pelaksanaan kerja magang pertama-tama dimulai dengan proses eksplorasi pencarian tempat kerja. Dimana langkah awal dimulai dengan mengirimkan lamaran kerja kepada perusahaan yang sekiranya menarik, salah satunya PT. GLI. Dimana PT. GLI merespons dengan undangan seleksi, dimulai dengan tes psikologis dan dua tahap wawancara, pertama dengan HR dan kedua dengan pengguna dari divisi *Quality Assurance*. Pada 25 Januari 2024, penerimaan magang sebagai *Quality Assurance* diumumkan. Hari pertama kerja pada 19 Februari 2024 dimulai dengan pengenalan perusahaan, penjelasan kontrak magang, dan sambutan dari CEO. Setelah itu, diperkenalkan ke kantor yang terdiri dari 2 lantai, bertemu anggota divisi QA, dan mempelajari alat kerja serta membaca panduan pengguna aplikasi AlfaGift.

Peran atau kerjaan dari posisi QA meliputi bergabung dengan proyek sesuai arahan dari *Supervisor*, apabila sudah bergabung dalam proyek, kemudian mengikuti *sprint* proyek yang sedang berjalan untuk memahami proyek dan menerima alur dan *flow* serta tampilan pada Figma, membuat skenario pengujian berdasarkan sistem desain ataupun dapat dalam bentuk *storylist* yang ada, melakukan pengujian sesuai dengan skenario yang telah dibuat, melaporkan *bug* kepada divisi terkait, melakukan verifikasi *bug* yang telah diperbaiki, dan melakukan pembuatan dokumentasi pada proyek yang sebelumnya sudah selesai dikerjakan.