BAB II

PROSEDUR KERJA

2.1. Deskripsi Penugasan Kerja

Deskripsi penugasan kerja yang dilakukan selama menjalankan Studi Independen di PT. Marka Kreasi Perdada (*Alterra Academy*) adalah sebagai berikut :

1. *Quality Engineer / Quality Assurance*
    a. Mampu memahami dan mengetahui apa dasar penguijan perangkat lunak
    b. Mampu mengetahui hal-hal yang diperlukan dalam membangun *software*
    c. Mampu melakukan pengujian pada perangkat lunak berbasis web *automation* dan manual
    d. Mmampu melakukan pengujian pada *software* berbasis *mobile automation* dan manual
    e. Mampu melakukan pengujian pada *software* berbasis REST API *automation* dan manual
    f. Mampu melakukan *load testing*
    g. Mampu menggunakan *tracking tools*
    h. Mampu menggunakan *test case management tools*
2. *Proffesional Skills*
    a. Mampu menerapkan *Time and Task Management*
    b. Mampu menerapkan *Teamworks & Collaboration*
    c. Mampu menerapkan *Basic Leadership Skills*
    d. Mampu menerapkan *Communication Skills*
    e. Mampu menerapkan *Landscape* peluang karir di bidang IT
    f. Mampu menerapkan pembuatan *Building Resume and CV*
    g. Mampu menerapkan *Public Speaking and Presentation Skills*
    h. Mampu menerapkan Persiapan *Interview* Untuk *Software Engineer*
    i. Mampu menerapkan *Understanding The Hiring Process* (Hak-hak sebagai Tenaga Kerja & Cara Negosiasi Gaji)

2.2. Teori Dasar Pendukung

1. *Software Testing*

Testing merupakan proses menjalankan atau eksekusi suatu program dengan tujuan utama untuk mencari atau menemukan kesalahan (*error*). Apabila tidak didapati adanya kesalahan pada saat testing dilakukan, bukan berarti aplikasi yang dibuat sudah sangat bagus. Tetapi ada kemungkinan teknik *testing* yang digunakan tidak cukup baik untuk mendeteksi kesalahan. *Testing* yang sukses adalah testing yang dapat mengungkap semua kesalahan yang belum pernah terjadi sebelumnya. *Test case* yang baik adalah *test case* yang memiliki kemungkinan besar untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya. Sehebat apapun *testing* yang dilakukan, tidak ada *software* yang bebas dari kesalahan [4].

2. Web *Testing*

Web *Testing* adalah teknik yang efektif untuk memastikan kualitas aplikasi Web. Meskipun demikian, teori dan metodologi tes tradisional sulit diterapkan karena kekhususan dan kompleksitas aplikasi Web. Menguji aplikasi Web tidak hanya untuk melakukan tugas-tugas penting, tetapi juga untuk faktor-faktor seperti keamanan, skalabilitas, dan keter-gantungan tinggi pada keluaran web browser dan cara pengguna berinteraksi web browser.[5]

Beberapa tipe web *testing* termasuk pengujian *fungsionalitas*, pengujian beban dan *stress*, *browser rendering* dan pengujian kegunaan. Pengujian fungsionalitas digunakan untuk me-nguji komponen web, memastikan bahwa situs web menjalankan fungsi yang diharapkan.[5]

3. *End-To-End Testing*

End-to-end testing adalah sebuah metodologi yang digunakan untuk menguji apakah *flow* aplikasi bekerja sebagaimana yang dirancang dari awal hingga selesai. *End-to-end testing* merupakan salah satu teknik *testing* yang harus dilakukan oleh perusahaan pengembang web, karena

jika telah melewati *end-to-end testing*, secara umum dipandang telah menjamin interaksi user dengan halaman web, dan sudah sesuai dengan kebutuhan *user*. [6]

Pelaksanaan *end-to-end testing* dapat menggunakan salah satu metode ataupun keduanya untuk kebutuhan *testing scenario* yang kompleks. Pada pelaksanaannya, *end-to-end testing* merupakan salah satu teknik *testing* yang bisa dieksekusi dengan manual *testing* ataupun dengan *automated software testing*.[6]

a. *Manual Testing*

Manual *testing* adalah sebuah teknik *testing* dimana *tester* menyiapkan *test cases* secara manual dan mengeksekusi *test cases* untuk mengidentifikasi *defect* di *software*. Secara umum penggunaan manual testing pada pelaksanaan *end-to-end testing* adalah melakukan eksekusi *test* dengan menjalankan software sesuai dengan skenario yang tertulis pada *test cases*. Selanjutnya membandingkan *output* yang keluar dari aplikasi dengan *output* yang diharapkan dari setiap *test cases*.[7]

b. *Automation Testing*

Automated software testing melibatkan pengembangan *test script* menggunakan *scripting languages* seperti *phyton*, *java script*, atau *tool command language* (TCL), sehingga *test case* dapat dieksekusi oleh komputer dengan minimal campur tangan manusia. *Automated software testing* adalah proses membuat sebuah program (*test script*) untuk mensimulasikan langkah-langkah *test case* manual dalam bahasa pemrograman apapun dengan bantuan *external automation helper tool* lainnya. Eksekusi *end-toend testing* menggunakan *automated software testing* membutuhkan konversi *test case* menjadi *test script*, yang setelah itu dilakukan *running test script* pada *automation tools* tersebut.[7]

4. *Test Case*

Pengujian dilaksanakan untuk melihat waktu pengerjaan *end-to-end testing* secara manual menggunakan *automation framework*. Selain mengetahui waktu pengerjaan, juga luaran dari penelitian ini dapat merepresentasikan faktor-faktor yang berpengaruh dalam pelaksanaan *end-to-end testing*. Karena penelitian berfokus pada *test exexution*, maka perancangan *test cases* menjadi proses terpenting.[8]

a. **STLC** (***Software Testing Life Cycle***)

Gambar 2.2 STLC[4].

Software Testing Life Cycle (STLC) didefinisikan sebagai rangkaian kegiatan yang sistematis dalam testing bertujuan untuk mengevaluasi apakah hasil aktual telah sesuai dengan hasil yang diharapkan serta memastikan bahwa program atau aplikasi yang dirancang tidak menghasilkan bug lagi.. Menurut banyak sumber memiliki pendekatan berbeda mengenai aktivitas atau *phase* yang harus dilewati pada STLC. Pada penelitian ini *phase* yang diikuti adalah *phase* utama STLC, yaitu terdiri dari *test planning*, *test design*, *test execution*, dan *test review*.[4]

1. *Requirement Analysis*

   Pada fase ini, *test team* mempelajari persyaratan dari sudut pandang pengujian untuk mengidentifikasi *requirements* yang akan diuji. *Requirements* dapat berupa fungsional atau non fungsional. Aktivitas – aktivitas yang dilakukan berupa mengidentifikasi jenis tes yang akan dilakukan, mengumpulkan

detail mengenai prioritas testing hingga melakukan identifikasi lingkungan pengujian.[9]

2. *Test Planning*

   Pada tahap ini, seorang manajer yang bertanggung jawab pada *testing*, akan menentukan perkiraan aktivitas dan biaya serta akan menyiapkan dan menyelesaikan *test plan*. Aktivitas yang dilakukan berupa persiapan *test plan* dan dokumen, pemilihan alat uji hingga perencanaan *resources* dan menentukan *roles*.[9]

3. *Test case development*

   Fase ini melibatkan pembuatan, verifikasi dan pengerjaan ulang *test cases* dan *test scripts*. *Test* data diidentifikasi dan ditinjau lalu dikerjakan ulang. Aktivitas yang dilakukan meliputi, membuat *test case, automation scripts* (jika ada), melakukan *review test case* dan *scripts*.[9]

4. *Test Environment Setup*

   Test environment adalah salah satu aspek penting dalam proses *testing*. Proses ini dapat dilakukan secara *parallel* dengan *test case development stage*. Tahap ini menentukan kondisi dari *hardware* dan *software* yang akan digunakan untuk melakukan *testing*.[9]

5. *Test Execution*

   Selama fase ini, testers akan melakukan pengujian berdasarkan *test plans* dan *test case* yang telah disiapkan sebelumnya. Apabila ditemukan *bug*, *bug* akan dilaporkan kepada tim *developers*. Aktivitas yang dilakukan pada fase ini meliputi, *execute test* sesuai dengan *test plan*, melakukan dokumentasi hasil tes dan melaporkan *bug* apabila ditemukan[9].

6. *Test Cycle closure*

   Testing team akan melakukan pertemuan untuk membahas dan menganalisis hasil testing untuk megidentikasi strategi atau langkah yang harus diambil untuk diterapkan kedepannya dengan mengambil pelajaran dari hasil siklus testing pada periode tersebut. Aktivitas yang dilakukan meliputi melakukan,

menyiapkan test metrics, melakukan dokumentasi serta mempersiapkan laporan[9].

b. *Test Documentation*

Dokumen *test cases* yang digunakan untuk eksperimen ini telah melalui proses *test planning* dan *design*. Hasil dari *test planning* adalah dokumen *test plan* yang disusun berdasarkan acuan IEEE Std 829-1998 untuk *Software Test Documentation*, dan dokumen *test design* yang disusun berdasarkan sumber yang sama dengan *test plan*.[10]

1. Manual *Test Scenario* dan *Test Case*

Project Name : Capstone Project
Application Test : Katalon
Tester Name : Ghina Auliannisa Ramanda

NO	TEST CASE ID	FEATURE	PRE-CONDITION	TEST SCENARIO	TEST CASE	TEST STEPS	TEST DATA	EXPECTED RESULT	POSITIVE / NEGATIVE
1	TC/WTLA-01	LOGIN	User sudah berada pada login page	User ingin melakukan pengelasan fitur login pada web	User Login With Valid Data	1. User memasukkan Valid email 2. User memasukkan valid password 3. klik login	email : admin@gmail.com Password : passwordadmin	1. menampilkan inputan user 2. menampilkan inputan user 3. Berhasil melakukan login data	POSITIVE
2	TC/WTLA-02				User Login With Non-Existed data	1. User memasukkan email yang tidak terdaftar 2. User memasukkan password yang tidak 3. klik login	email : capstone26@gmail.com Password : 12345678	1. menampilkan inputan user 2. menampilkan inputan user 3. Tidak Berhasil melakukan login data dan terdapat pesan "Akun tidak ditemukan"	NEGATIVE
3	TC/WTLA-03				User Login With Invalid Email	1. User memasukkan invalid email 2. User memasukkan valid password 3. klik login	email : 12345678@gmail.com Password : passwordadmin	1. menampilkan inputan user 2. menampilkan inputan user 3. Tidak Berhasil melakukan login data dan terdapat pesan "Email/Password salah"	NEGATIVE
4	TC/WTLA-04				User Login With Invalid Password	1. User memasukkan Valid email 2. User memasukkan invalid password 3. klik login	email : admin@gmail.com Password : abc123456	1. menampilkan inputan user 2. menampilkan inputan user 3. Tidak Berhasil melakukan login data dan terdapat pesan "Email/Password salah"	NEGATIVE
5	TC/WTLA-05				User Login With Empty Email	1. User mengosongkan field email 2. User memasukkan valid password 3. klik login	email : null Password : passwordadmin	1. menampilkan inputan user 2. menampilkan inputan user 3. Tidak Berhasil melakukan login data dan terdapat pesan "Email is required"	NEGATIVE
6	TC/WTLA-06				User Login With Empty Password	1. User memasukkan valid email 2. User mengosongkan field password 3. klik login	email : admin@gmail.com Password : null	1. menampilkan inputan user 2. menampilkan inputan user 3. Tidak Berhasil melakukan login data dan terdapat pesan "Password is required"	NEGATIVE

Gambar 2.3 *Manual Test Case*

2. *Automation Test Scenario* dan *Test Case*

Gambar 2.4 *Automation Test Case*

5. Metode *Black Box Testing*

Blackbox testing adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pengujian

ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Tidak ada upaya untuk mengetahui kode program apa yang *output* pakai. Pada proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan oleh perusahaan[11]. *Black Box testing* dibagi menjadi 2 bagian, yaitu:

1. *Functional Testing*

   Jenis ini berkaitan dengan persyaratan fungsional atau spesifikasi aplikasi. Beberapa jenis pengujian fungsional utama yaitu *System Testing, Integration Testing, Smoke Testing, Regression Testing, sanity testing dan, User acceptance Testing.*[11]

2. *Non-Functional Testing*

   Terlepas dari fungsionalitas persyaratan ada beberapa aspek non-fungsional yang perlu diuji untuk meningkatkan kualitas dan kinerja aplikasi yaitu *Performace testing, Usability testing, Load testing, Compability testing, Scalability testing* dan *Stress testing*.[11]