

BAB II

LANDASAN TEORI

A. Definisi *Testing*

Pengertian *testing* menurut para ahli :

1. Menurut Myers 1979

Testing merupakan proses eksekusi program ataupun sistem secara intens untuk menemukan *error*.

2. Menurut Hetzel 1983

Menurutnya *testing* merupakan aktivitas yang digunakan untuk dapat melakukan evaluasi suatu parameter ataupun kemampuan dari program atau sistem serta memastikan apakah sudah memenuhi kebutuhan ataupun hasil yang diharapkan.

3. Menurut standar ANSI/ IEEE 1059

Testing merupakan proses menganalisa sesuatu entitas aplikasi untuk mendeteksi perbandingan antara keadaan yang terdapat dengan keadaan yang diinginkan (*defects/ error/ bugs*) serta mengevaluasi fitur- fitur dari entitas *software*.

Untuk memastikan kualitas mutu aplikasi, *testing* harus dilakukan yaitu dengan menguji apakah sistem data yang dihasilkan sesuai dengan *testing* yang telah dilakukan untuk memastikan kualitas. Proses menganalisa suatu entitas aplikasi untuk mendeteksi perbandingan antara keadaan yang terdapat dengan keadaan yang diinginkan serta mengevaluasi fitur-fitur dari entitas *software*. [1]

B. Tujuan *Testing*

Beberapa tujuan *testing* adalah sebagai berikut :

1. Memastikan aplikasi berjalan sebagaimana mestinya.
2. Mendeteksi terjadinya *error* serta memvalidasi apakah sudah memenuhi keinginan *user*.
3. Melakukan pengecekan/pengetesan entitas-entitas, termasuk *software*, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan.
4. Melakukan validasi untuk dapat melihat kebenaran sistem, apakah proses yang telah ditulis dalam spesifikasi adalah apa yang sebenarnya.

5. Mendeteksi adanya *error*, *testing* untuk mendeteksi kesalahan secara insentif, yaitu menentukan apakah suatu hal tersebut terjadi bilamana tidak seharusnya terjadi. [1]

C. Prinsip *Testing*

Adapun prinsip *testing* sebagai berikut :

1. Tidak dapat terpenuhi secara lengkap

Karena banyaknya kombinasi *test* yang amat besar dan luas tingkat kompleksitas serta alur program yang cukup luas membuat *testing* tidak bisa dilakukan untuk keseluruhan aplikasi.

2. *Testing* merupakan pekerjaan yang memiliki tingkat kesulitan tinggi

Agar mampu melakukan *testing* dengan baik dibutuhkan pemahaman yang cukup baik terhadap sistem itu sendiri, dibutuhkan kreatifitas, pengetahuan bisnis, pengetahuan *testing*, dan metodologi *testing* agar mampu melakukan *testing* dengan baik dan benar.

3. Mencegah terjadinya kesalahan.

Dalam proses *testing*, tidak hanya dilakukan proses pengembangan semata, hasil dari *testing* nantinya akan dijadikan pengembangan dan masukan untuk versi aplikasi selanjutnya.

4. Penanganan resiko

Munculnya penambahan biaya yang diakibatkan oleh kegagalan aplikasi yang diakibatkan cacatnya aplikasi menjadi salah satu sumber daya yang dibutuhkan untuk melakukan proses *testing* yang dapat diukur dari skala prioritas, kompleksitas, dan kesulitan *testing* yang ada sehingga meminimalisir resiko yang akan terjadi.

5. Perencanaan yang matang

Untuk menjaga kesesuaian penggunaan sumber daya yang ada dan jadwal proyek yang telah ditetapkan serta untuk menjaga alur dari proses *testing* agar nantinya tidak menyimpang dari tujuan awal yang telah ditetapkan.

6. Membutuhkan Independensi

Untuk melakukan *testing* sebaiknya dilakukan oleh pihak eksternal, tester independen juga dikatakan sebagai pengamat yang tidak biasa dan mengukur kualitas software secara akurat untuk hasil yang maksimal. [1]

D. Manfaat *Testing*

Dengan adanya proses *testing* melalui beberapa tahap pengetesan terhadap *system* akan dapat :

1. Meningkatkan kepercayaan, tingkat resiko yang dapat diterima.
2. Menyediakan informasi yang dapat mencegah terulangnya *error*.
3. Menyediakan informasi untuk mendeteksi *error* secara dini.
4. Mencari *error* dan kelemahan/keterbatasan sistem.
5. Mencari sejauh apa kemampuan dari sistem.
6. Menyediakan informasi untuk kualitas produk *software*. [1]

E. Definisi Kualitas

Definisi lain yang ditemukan dalam beberapa literatur, mendefinisikan *testing* sebagai pengukuran kualitas software. Apa yang dimaksud dengan kualitas? Sama halnya dengan *testing*, pengertian kualitas bagi tiap praktisi dapat berbeda-beda, karena kualitas memang merupakan suatu hal yang subyektif dan abstrak. Berikut ini beberapa definisi sederhana tentang kualitas :

1. Menurut CROSBY : Kualitas adalah pemenuhan terhadap kebutuhan.
2. Menurut ISO-8402 : Kualitas adalah keseluruhan dari fitur yang menjadikan produk dapat memuaskan atau dipakai sesuai kebutuhan dengan harga yang terjangkau.
3. Menurut W.E. Perry : Kualitas adalah pemenuhan terhadap standar.
4. Menurut R. Glass: Kualitas adalah tingkat kesempurnaan.
5. Menurut J. Juran: Kualitas adalah tepat guna. [2]

F. Hubungan *Testing* dan Kualitas

Definisi software berkualitas adalah software yang bebas *error* dan *bug* secara obyektif, tepat waktu dan dana, sesuai dengan kebutuhan atau keinginan dan dapat dirawat (*maintainable*). Pengertian kata obyektif adalah suatu proses pembuktian yang terstruktur, terencana dan tercatat / terdokumentasi dengan baik. Pendekatan yang obyektif sangat diperlukan karena kualitas adalah suatu hal yang tidak nyata dan subyektif. *Testing* membuat kualitas dapat dilihat secara obyektif, karena *testing* merupakan pengukuran dari kualitas *software*. Dengan kata lain *testing* berarti pengendalian kualitas (*Quality Control-*

QC), dan QC mengukur kualitas produk, sedangkan jaminan kualitas (*Quality Assurance-QA*) mengukur kualitas proses yang digunakan untuk membuat produk berkualitas. *Testing* tidak dapat memastikan kualitas *software*, namun dapat memberikan kepercayaan atau jaminan terhadap *software* dalam suatu tingkat tertentu. Karena *testing* merupakan pembuktian dalam suatu kondisi terkendali, dimana *software* difungsikan sebagaimana yang diharapkan pada *test case* yang digunakan. [2]

G. Usability Testing

Usability testing adalah sebuah metode untuk mengevaluasi *user experience* terhadap *software* ataupun *website* yang dibuat. Umumnya, metode ini dilakukan oleh para *UX developer* dengan melibatkan beberapa *user* (pengguna) tertentu untuk diteliti bagaimana proses mereka selama berinteraksi dengan *website*. Dengan melakukan *usability testing*, pengembang bisa mengukur kualitas *website* untuk menciptakan *user experience* yang baik dan mengadopsi sistem navigasi yang mudah. Metode *usability testing* memerlukan aturan tertentu untuk memastikan hasilnya benar-benar valid.

Manfaat dari *usability testing* :

1. Mengatasi masalah internal

Usability testing menawarkan pemeriksaan yang tidak biasa, akurat, dan langsung berkaitan dengan pengalaman pengguna *website*.

2. Proses mudah dan cepat

3. Mengetahui setiap aktivitas pengunjung

Sudah cukup jelas, kalau *usability testing* akan membantu pengembang untuk tahu apa saja yang dilakukan pengguna selama berada di dalam *website* dan mengapa mereka melakukan hal tersebut.

4. Mengecek kekurangan *website*

Usability testing memungkinkan pengembang untuk mengetahui apa saja kekurangan atau kesalahan yang terdapat di dalam *website*. Sehingga, biaya yang dikeluarkan selama merancang *website* pun menjadi tidak sia-sia.

5. Strategi mengembangkan bisnis

Metode ini juga bermanfaat pada pertumbuhan bisnis. Karena *user experience* merupakan salah satu pendukung untuk membuat produk yang dibuat laku di pasaran. [3]

H. Cara Melakukan *Usability Testing*

Berikut langkah-langkah saat melakukan *usability testing* :

1. Tentukan bagian *website* yang akan diuji

Langkah awal yang perlu dipersiapkan adalah menentukan bagian *website* mana yang akan diuji. Cobalah untuk pertimbangkan kira-kira fitur apa atau desain mana yang masih menjadi kendala utama pada *website*.

2. Siapkan prototipe

Prototipe adalah bentuk awal dari *website* yang belum melewati fase penyempurnaan. Dengan adanya prototipe, pengembang bisa lebih mudah untuk mencari tahu bagaimana respon dari calon pengguna mengenai fitur *website* yang dibuat.

3. Buat parameter keberhasilan

Tujuannya untuk menilai elemen mana saja yang perlu dikoreksi dan sejauh mana fitur *website* sudah cukup memenuhi kebutuhan pengguna.

4. Mempersiapkan skenario pengujian

5. Menentukan calon pengguna yang akan diuji

6. Memulai Tahap Pengujian

7. Analisis hasil data pengujian

Setelah mengumpulkan hasil data pengujian, buatlah laporan secara rinci untuk dijadikan bahan diskusi antara pengembang dengan tim. Bangun *mindset* bahwa akan ada banyak hasil temuan yang mungkin saja tidak akan sesuai dengan yang direncanakan. [3]