

## Reading Material

### Menyusun Bug Report - Mengenal Bug



## READING

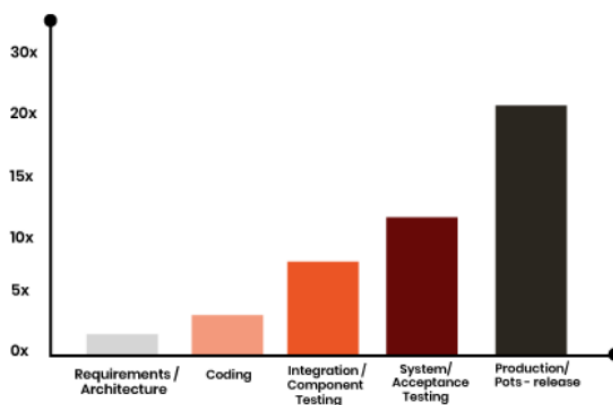
Semua aplikasi berjalan pada kode dalam berbagai bahasa baik itu Java, .Net, Python, atau lainnya. Sebagian besar kode ini ditulis oleh developer. Ini juga berarti rentan terhadap kesalahan. Kesalahan ini disebut **Bug**. Penyimpangan apa pun dari perilaku aplikasi yang diharapkan dalam hal fungsionalitas, hasil perhitungan, navigasi, atau keseluruhan tampilan dan nuansa dapat dianggap sebagai cacat atau bug dalam aplikasi atau perangkat lunak tersebut.

Perangkat lunak bebas bug adalah impian setiap developer, dan impian ini hanya dapat diwujudkan melalui pengujian menyeluruh. Berbagai jenis teknik digunakan untuk menemukan jumlah bug maksimum sebelum aplikasi atau produk mencapai pelanggan.

### 1. Apa itu Bug ?

**Bug** dapat disebut sebagai kesalahan atau anomali dalam sistem yang menyebabkan perilaku yang tidak biasa dan output yang tidak valid dalam suatu sistem. Kesalahan seperti ini sebagian besar disebabkan oleh manusia dan dapat sangat mempengaruhi fungsionalitas perangkat lunak.

Relative cost to fix bugs, based on time of detection



## 2. Apa itu **Bug Fixing** ?

Anomali yang mencegah perangkat lunak bekerja sesuai dokumen SRS dapat diperbaiki melalui proses yang disebut perbaikan bug. Akan ada tim pengujian perangkat lunak untuk memeriksa perangkat lunak secara menyeluruh dan melaporkan bug yang ditemukan kepada developer sehingga mereka dapat memperbaikinya.

## 3. Apa tantangan paling umum yang dihadapi saat **Debugging** (Mencari tahu penyebab atau cara mendapatkan Bug)?

- Debugger berhenti bekerja atau ada masalah dengannya dan Anda tidak menyadarinya
- Kesalahan logis sulit diperbaiki
- Data yang tidak disortir
- Masalah pembuatan log yang dalam
- Kesalahan tata bahasa
- Ketidakmampuan untuk melakukan debugging secara realtime

## 4. Apa saja jenis-jenis **Bug**?

Ada beberapa jenis bug yang ditemukan dalam aplikasi. Berikut jenisnya :

### a. **Functional Errors**

Masalah apa pun yang berhubungan dengan fungsionalitas aplikasi, dikatakan sebagai kesalahan fungsional. Misalnya. ketika Anda memasukkan beberapa data ke dalam aplikasi dan menekan tombol "Simpan", data Anda harus disimpan ke database aplikasi. Data harus dapat digunakan kembali pada tahap selanjutnya juga. Jika sistem tidak menyimpan data atau

membuat kesalahan saat menyimpan, ini dapat dianggap sebagai cacat fungsional.

**b. Logical Errors**

Kesalahan logis terutama dikaitkan dengan logika kode. Logika yang ditulis oleh developer mungkin tidak berfungsi seperti yang diharapkan sehingga menghasilkan output yang salah. Contoh klasik dari hal ini adalah pembagian dengan 0 atau menetapkan nilai ke variabel yang salah, atau kesalahan serupa lainnya di tingkat pengkodean. Kesalahan logis ini dapat dengan mudah dihindari dengan melakukan pengecekan dan penelusuran kode dengan tim (Debugging).

**c. Calculation Errors**

Seperti namanya, bug ini muncul karena kesalahan perhitungan atau formula pada saat pengkodean. Itu juga bisa disebabkan oleh melewati parameter yang salah ke fungsi. Beberapa alasan umum lainnya untuk kesalahan perhitungan termasuk memilih algoritma yang salah, ketidakcocokan dalam tipe data, nilai data yang salah mengalir dari sistem lain, atau bahkan melakukan hardcoding beberapa nilai dalam kode.

**d. Validation Message Errors**

Kesalahan yang disebabkan oleh pesan validasi yang salah atau hilang disebut sebagai kesalahan pesan validasi. Misalnya, saat memasukkan data atau format yang salah, seperti di bidang angka (umur) Anda memasukkan karakter seperti nama, seharusnya muncul pesan kesalahan yang sesuai. Demikian pula, kapan pun Anda ingin menyimpan beberapa data, konfirmasi penyimpanan atau pesan kegagalan penyimpanan harus

ditampilkan. Dalam semua kasus tersebut, pop-up pesan positif dan negatif. Jika tidak ditampilkan, maka dianggap error.

**e. Cosmetic Errors**

Kesalahan kosmetik adalah kesalahan yang tidak memengaruhi aplikasi atau fungsinya secara langsung. Itu juga merupakan masalah kecil yang diperbaiki dan diterapkan dengan prioritas paling kecil. Beberapa contoh kesalahan kosmetik adalah kesalahan ejaan, masalah perataan, variasi warna, dan lainnya.

**f. Workflow Errors**

Kesalahan alur kerja juga disebut kesalahan navigasi. Seperti namanya, mereka merujuk pada masalah navigasi saat bepergian bolak-balik melalui aplikasi. Halaman yang akan ditampilkan saat tombol “Berikutnya” di klik atau saat tombol “Kembali” di klik, harus seperti yang diharapkan. Setiap ketidakcocokan di halaman yang sebenarnya dianggap sebagai kesalahan alur kerja.

**g. Integration Errors**

Kesalahan integrasi mengacu pada kesalahan yang timbul dari ketidakcocokan data atau lainnya selama interaksi antara beberapa sistem dan modul. Kesalahan ini hanya dapat diidentifikasi selama pengujian integrasi. Dalam kebanyakan kasus, kesalahan integrasi ini disebabkan oleh bagaimana data dari satu modul dikonsumsi oleh modul lain, atau terkadang data dapat diubah dalam aliran sistem, atau mungkin ada ketidakcocokan tipe data, dll.

#### **h. Memory Leaks**

Kesalahan umum lainnya yang biasanya ditemukan selama pengujian yang ketat dan berkelanjutan terkait dengan kebocoran memori. Dalam kasus tersebut, kinerja aplikasi mulai memburuk secara drastis setelah jangka waktu tertentu. Ini umumnya karena penggunaan memori yang terus menerus tanpa melepaskannya setelah digunakan. Kebocoran memori sangat sulit ditemukan atau diperbaiki dan satu-satunya cara adalah memastikan standar pengkodean yang tepat dan praktik terbaik diikuti pada saat pengkodean.

#### **i. App Crashes**

App Crash adalah masalah yang sangat prioritas yang perlu diselesaikan sedini mungkin. Jelas dari namanya bahwa dalam kasus seperti itu aplikasi tiba-tiba ditutup dan semua data hilang. Ini bisa sangat mengganggu pelanggan akhir, dapat disebabkan oleh beberapa alasan termasuk kegagalan panggilan API, waktu halaman habis(timeout), sistem upstream atau downstream down, atau lainnya.

#### **j. Security Errors**

Di dunia internet saat ini, keamanan adalah hal yang paling penting. Bug keamanan sangat kritis. Mereka dapat terkait dengan keamanan data pengguna, penyamaran data dan preferensi pengguna, data keuangan atau kesehatan, keamanan berdasarkan hak istimewa (seperti halaman admin yang hanya dapat diakses oleh admin aplikasi), dan banyak lagi. Masalah-masalah ini jika ditemukan dalam produksi dapat merugikan aplikasi itu sendiri karena pelanggan akan kehilangan kepercayaan pada sistem.