

CONTOH PEMAHAMAN TENTANG PRINSIP TESTING

Nama : Argya Rijal Rafi
Kelas : D4 Teknik Informatika
NPM : 1204062

1. Functional Testing

- Pengujian Fungsional pada User Interface: Pengujian ini dilakukan untuk memastikan bahwa pengguna dapat berinteraksi dengan antarmuka pengguna (UI) aplikasi atau sistem dengan mudah dan sesuai dengan harapan. Contohnya, pengujian ini dapat meliputi pengujian kejelasan tampilan, pengujian fungsionalitas tombol, atau pengujian navigasi.
- Pengujian Fungsional pada Fungsionalitas: Pengujian ini dilakukan untuk memastikan bahwa aplikasi atau sistem dapat melakukan fungsionalitas yang diharapkan. Contohnya, pengujian ini dapat meliputi pengujian input data, pengujian output data, dan pengujian fungsionalitas lainnya.
- Pengujian Fungsional pada Integrasi: Pengujian ini dilakukan untuk memastikan bahwa aplikasi atau sistem dapat berintegrasi dengan aplikasi atau sistem lainnya dengan baik. Contohnya, pengujian ini dapat meliputi pengujian integrasi API, pengujian integrasi database, atau pengujian integrasi sistem lainnya.
- Pengujian Fungsional pada Kinerja: Pengujian ini dilakukan untuk memastikan bahwa aplikasi atau sistem dapat menangani beban kerja dengan baik. Contohnya, pengujian ini dapat meliputi pengujian kinerja pada waktu yang tepat, pengujian kinerja pada jumlah data yang besar, atau pengujian kinerja pada kecepatan akses.
- Pengujian Fungsional pada Keamanan: Pengujian ini dilakukan untuk memastikan bahwa aplikasi atau sistem aman dan terlindungi dari ancaman keamanan seperti hacking atau peretasan. Contohnya, pengujian ini dapat meliputi pengujian enkripsi data, pengujian autentikasi dan otorisasi, atau pengujian keamanan lainnya.

2. Non-functional Testing

- **Pengujian Kinerja:** Pengujian ini dilakukan untuk memastikan bahwa aplikasi atau sistem dapat menangani beban kerja dan volume transaksi dengan baik, serta memberikan respons yang cepat. Contohnya, pengujian ini dapat meliputi pengujian kinerja pada waktu yang tepat, pengujian kinerja pada jumlah data yang besar, atau pengujian kinerja pada kecepatan akses.
- **Pengujian Keandalan:** Pengujian ini dilakukan untuk memastikan bahwa aplikasi atau sistem dapat berjalan dengan baik dalam jangka waktu yang lama dan tidak mengalami kegagalan atau kerusakan yang tidak diinginkan. Contohnya, pengujian ini dapat meliputi pengujian stabilitas aplikasi, pengujian pengembalian data yang hilang, atau pengujian kemampuan pemulihan sistem setelah kegagalan.
- **Pengujian Kemampuan:** Pengujian ini dilakukan untuk memastikan bahwa aplikasi atau sistem dapat memenuhi persyaratan dan spesifikasi yang ditetapkan oleh bisnis atau pengguna. Contohnya, pengujian ini dapat meliputi pengujian kemampuan skala aplikasi, pengujian kemampuan penanganan beban besar, atau pengujian kemampuan respons pada permintaan pengguna yang banyak.
- **Pengujian Keamanan:** Pengujian ini dilakukan untuk memastikan bahwa aplikasi atau sistem aman dan terlindungi dari ancaman keamanan seperti hacking atau peretasan. Contohnya, pengujian ini dapat meliputi pengujian keamanan pada jaringan, pengujian enkripsi data, atau pengujian autentikasi dan otorisasi.
- **Pengujian Interoperabilitas:** Pengujian ini dilakukan untuk memastikan bahwa aplikasi atau sistem dapat berintegrasi dengan sistem atau aplikasi lainnya dengan baik. Contohnya, pengujian ini dapat meliputi pengujian interoperabilitas pada berbagai platform atau sistem operasi, pengujian interoperabilitas pada aplikasi pihak ketiga, atau pengujian interoperabilitas pada perangkat keras yang berbeda.

3. Structural Testing

- **Pengujian Unit:** Pengujian ini dilakukan untuk menguji setiap unit kode program secara terpisah untuk memastikan bahwa setiap unit berfungsi dengan baik dan dapat bekerja dengan unit yang lainnya. Contohnya, pengujian ini dapat meliputi pengujian fungsi kalkulator atau pengujian fungsi log-in.
- **Pengujian Integrasi:** Pengujian ini dilakukan untuk menguji bagaimana unit-unit program yang berbeda dapat berinteraksi dan bekerja bersama untuk mencapai tujuan yang sama. Contohnya, pengujian ini dapat meliputi pengujian integrasi database atau pengujian integrasi antarmuka pengguna.
- **Pengujian Sistem:** Pengujian ini dilakukan untuk menguji sistem secara keseluruhan setelah unit-unit program diuji dan diintegrasikan bersama. Tujuannya adalah untuk memastikan bahwa sistem berfungsi dengan baik dan sesuai dengan persyaratan yang telah ditetapkan. Contohnya, pengujian ini dapat meliputi pengujian sistem pengelolaan gudang atau pengujian sistem manajemen transaksi.
- **Pengujian Fungsional:** Pengujian ini dilakukan untuk menguji fungsionalitas setiap bagian dari kode program. Tujuannya adalah untuk memastikan bahwa setiap fungsi program bekerja dengan baik dan sesuai dengan persyaratan fungsional yang telah ditetapkan. Contohnya, pengujian ini dapat meliputi pengujian fungsi pencarian atau pengujian fungsi perhitungan.
- **Pengujian Non-fungsional:** Pengujian ini dilakukan untuk menguji aspek non-fungsional dari kode program seperti kinerja, keamanan, dan keandalan. Tujuannya adalah untuk memastikan bahwa kode program dapat beroperasi dengan baik dan sesuai dengan persyaratan non-fungsional yang telah ditetapkan. Contohnya, pengujian ini dapat meliputi pengujian kinerja jaringan atau pengujian keamanan aplikasi.

4. Testing related to Change

- **Regression Testing:** Pengujian regresi adalah jenis pengujian yang dilakukan untuk memastikan bahwa perubahan pada perangkat lunak tidak mempengaruhi fungsionalitas yang sudah ada. Dalam pengujian ini, semua fungsi atau fitur yang sudah ada sebelumnya diuji ulang setelah perubahan diterapkan. Contohnya, jika dilakukan perubahan pada sistem penjualan online, maka pengujian regresi akan dilakukan untuk memastikan bahwa fungsi pembayaran, proses pemesanan, dan fitur lainnya masih berfungsi dengan baik.
- **Acceptance Testing:** Pengujian penerimaan adalah jenis pengujian yang dilakukan untuk memastikan bahwa sistem yang diperbarui atau diubah sesuai dengan persyaratan yang ditetapkan oleh pengguna. Tujuan dari pengujian ini adalah untuk memastikan bahwa perubahan yang dilakukan sesuai dengan kebutuhan pengguna dan tidak mengganggu fungsionalitas yang sudah ada. Contohnya, pengujian penerimaan dapat dilakukan untuk memastikan bahwa penambahan fitur baru pada sistem manajemen inventaris tidak mengganggu proses manajemen inventaris yang sudah ada.
- **Impact Analysis:** Analisis dampak adalah jenis pengujian yang dilakukan untuk mengevaluasi dampak dari perubahan pada sistem atau perangkat lunak. Tujuan dari pengujian ini adalah untuk memastikan bahwa perubahan tidak memiliki dampak negatif pada fungsionalitas sistem atau perangkat lunak. Contohnya, jika dilakukan perubahan pada sistem manajemen gaji, maka analisis dampak akan dilakukan untuk memastikan bahwa perubahan tersebut tidak mengganggu sistem penggajian yang sudah ada.
- **Smoke Testing:** Pengujian asap atau Smoke testing adalah jenis pengujian yang dilakukan untuk memastikan bahwa sistem yang diperbarui atau diubah dapat berjalan dengan baik. Dalam pengujian ini, hanya dilakukan pengujian dasar untuk memastikan bahwa sistem dapat berjalan tanpa ada masalah besar. Contohnya, Smoke testing dapat dilakukan untuk memastikan bahwa situs web masih dapat diakses setelah dilakukan perubahan pada server.

