

Nama : Aura Muhammad

NIM : 231111072

1. Hal apa sajakah yang harus dikelola dalam manajemen perangkat lunak?

Dalam manajemen perangkat lunak, ada beberapa hal penting yang harus dikelola agar proyek berjalan lancar dan hasilnya sesuai harapan. Di antaranya:

- **Waktu dan Jadwal:** Kapan proyek dimulai dan kapan harus selesai.
- **Biaya/Anggaran:** Berapa dana yang dibutuhkan untuk menyelesaikan proyek.
- **Sumber Daya Manusia:** Tim pengembang, tester, manajer, dan lainnya.
- **Kualitas:** Memastikan perangkat lunak sesuai standar dan bebas dari bug.
- **Risiko:** Mengantisipasi kemungkinan masalah atau hambatan yang bisa terjadi.
- **Perubahan:** Menyesuaikan jika ada perubahan permintaan dari klien/pengguna.
- **Dokumentasi:** Mencatat semua proses dan keputusan selama pengembangan.

2. Mengapa harus dilakukan pengujian perangkat lunak?

Pengujian sangat penting untuk memastikan bahwa perangkat lunak **berjalan sesuai fungsinya, bebas dari kesalahan (bug)**, dan memberikan **pengalaman yang baik bagi pengguna**. Tanpa pengujian, bisa saja software terlihat baik di permukaan, tapi menyimpan banyak kesalahan di dalamnya. Pengujian juga bisa menghemat biaya perbaikan di masa depan.

3. Hal apa saja yang harus diuji dalam perangkat lunak?

Yang perlu diuji antara lain:

- **Fungsi-fungsi utama:** Apakah fitur-fitur bekerja sesuai yang diharapkan.
- **Antarmuka pengguna (UI/UX):** Apakah tampilannya rapi dan mudah digunakan.
- **Keamanan:** Apakah data pengguna aman dari serangan atau kebocoran.
- **Kinerja (Performance):** Apakah aplikasi cepat, tidak lemot, dan stabil.
- **Kompatibilitas:** Apakah aplikasi berjalan di berbagai perangkat/sistem operasi.
- **Integrasi:** Apakah bagian-bagian aplikasi bisa bekerja sama dengan baik.

4. Jelaskan perbedaan pengujian black box dan white box!

- **Black Box Testing:** Pengujian dilakukan **tanpa mengetahui isi kode program**. Fokusnya pada **apa yang dilakukan** perangkat lunak, bukan bagaimana cara kerjanya.
 - Contoh: User mengisi form login → klik login → berhasil atau tidak?
- **White Box Testing:** Pengujian dilakukan **dengan melihat langsung isi kode program**. Fokusnya pada **logika, alur, dan struktur kode**.
 - Contoh: Mengecek apakah semua cabang *if-else* di fungsi tertentu dijalankan.

5. Jelaskan tentang pengujian unit!

Pengujian unit (unit testing) adalah **pengujian terhadap bagian terkecil dari kode program**, biasanya berupa **fungsi atau metode** dalam suatu modul. Tujuannya untuk memastikan bahwa setiap unit kode bekerja sesuai yang diharapkan secara mandiri.

6. Jelaskan tentang pengujian integrasi!

Pengujian integrasi adalah pengujian untuk memastikan bahwa **beberapa unit atau modul yang sudah digabungkan** bisa bekerja sama dengan benar. Karena kadang-kadang, modul yang bekerja baik secara individu belum tentu berjalan baik saat dikombinasikan.

7. Jelaskan tentang pengujian smoke integration!

Smoke testing adalah **pengujian awal yang sangat sederhana**, untuk mengecek apakah software bisa dijalankan dan fungsi-fungsi utamanya berjalan. Istilahnya, “apakah aplikasinya bisa menyala dulu?” Kalau gagal di tahap ini, pengujian lebih lanjut ditunda.

8. Jelaskan tentang pengujian regression integration!

Regression testing dilakukan untuk memastikan bahwa **fitur-fitur lama tetap bekerja** setelah ada perubahan kode, seperti penambahan fitur baru atau perbaikan bug. Tujuannya agar perubahan tidak menyebabkan kerusakan (bug baru) pada bagian lain yang sebelumnya sudah berfungsi baik.

9. Jelaskan tentang pengujian top-down integration!

Top-down integration testing dilakukan dengan cara menguji **modul dari level atas (utama)** dulu, lalu secara bertahap menguji modul-modul bawahan yang terhubung. Modul yang belum tersedia biasanya digantikan dengan **stub** (semacam pengganti sementara).

10. Jelaskan tentang pengujian bottom-up integration!

Kebalikan dari top-down. Bottom-up integration testing dilakukan dengan menguji **modul dari level bawah terlebih dahulu**, baru kemudian naik ke modul yang lebih atas. Modul yang belum tersedia biasanya digantikan dengan **driver** (semacam pemicu sementara).

Berikut adalah dokumentasi ringkas **jenis-jenis pengujian perangkat lunak** (software testing) beserta penjelasannya dalam format yang mudah dibaca dan bisa kamu simpan:

Dokumentasi Jenis-Jenis Software Testing

1. Black Box Testing

- Menguji **fungsi** tanpa melihat kode program.
- Fokus pada **input dan output**.
- Cocok untuk menguji sistem secara keseluruhan.

2. White Box Testing

- Pengujian dengan **melihat isi kode program**.
- Fokus pada **alur logika dan struktur internal** kode.
- Biasanya dilakukan oleh developer.

3. Unit Testing

- Menguji **unit terkecil dari aplikasi** (fungsi/metode).
- Dilakukan secara terisolasi.
- Biasanya dilakukan oleh developer.

4. Incremental Integration Testing

- Menguji modul **sedikit demi sedikit**, lalu digabungkan bertahap.

- Untuk mendeteksi **bug antar modul** sedini mungkin.

5. Integration Testing

- Menguji bagaimana **modul bekerja saat digabungkan**.
- Tujuannya agar tidak terjadi kesalahan komunikasi antar modul.

6. Functional Testing

- Menguji apakah **fungsi-fungsi dalam sistem** berjalan sesuai spesifikasi.
- Biasanya berbasis black box testing.

7. System Testing

- Menguji **keseluruhan sistem** sebagai satu kesatuan.
- Termasuk fungsi, kinerja, keamanan, dan lainnya.

8. End-to-End Testing

- Menguji **aliran dari awal hingga akhir** penggunaan sistem.
- Tujuannya untuk meniru perilaku nyata pengguna.

9. Sanity Testing

- Pengujian cepat untuk memastikan **perubahan kecil tidak merusak fitur utama**.
- Dilakukan sebelum regresi testing.

10. Regression Testing

- Menguji apakah fitur lama **masih berfungsi setelah ada perubahan** (fitur baru/bugfix).

11. Acceptance Testing

- Menguji apakah sistem **sesuai kebutuhan pengguna**.
- Biasanya dilakukan oleh **user atau klien**.

12. Load Testing

- Menguji sistem saat **dibebani banyak pengguna/data** dalam kondisi normal.
- Menilai **kapasitas maksimum sistem**.

13. Stress Testing

- Menguji sistem dalam **kondisi ekstrem atau melebihi batas kapasitas**.

- Tujuannya melihat **titik kegagalan** sistem.

14. Performance Testing

- Menguji **kecepatan, stabilitas, dan respons** sistem.
- Bisa termasuk load dan stress testing.

15. Usability Testing

- Menguji apakah aplikasi **mudah digunakan dan dipahami** pengguna.
- Fokus pada antarmuka dan pengalaman pengguna (UX).

16. Install/Uninstall Testing

- Menguji apakah **instalasi dan penghapusan aplikasi** berjalan lancar.
- Termasuk pemeriksaan file dan konfigurasi yang terbentuk.

17. Recovery Testing

- Menguji apakah sistem bisa **pulih setelah crash/error**.
- Penting untuk sistem yang harus **tetap berjalan stabil**.

18. Security Testing

- Menguji sistem terhadap **kerentanan keamanan**, seperti peretasan, akses ilegal, dll.

19. Compatibility Testing

- Menguji apakah aplikasi **berjalan baik di berbagai perangkat, OS, dan browser**.

20. Comparison Testing

- Membandingkan sistem dengan **versi sebelumnya** atau **produk kompetitor**.
- Untuk mengetahui peningkatan atau kelebihan.

21. Alpha Testing

- Dilakukan oleh **tim internal pengembang**.
- Bertujuan mendeteksi bug sebelum diuji oleh user luar.

22. Beta Testing

- Dilakukan oleh **pengguna nyata (eksternal)**.
- Untuk mendapatkan feedback sebelum dirilis resmi.

