



# Testing Untuk Menghasilkan Software Berkualitas

Testing & Implementasi

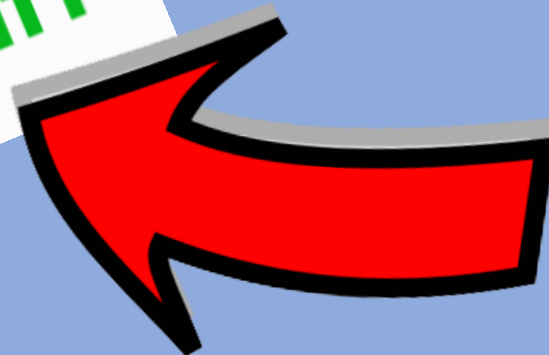
Elkin Rilvani

Pertemuan 2

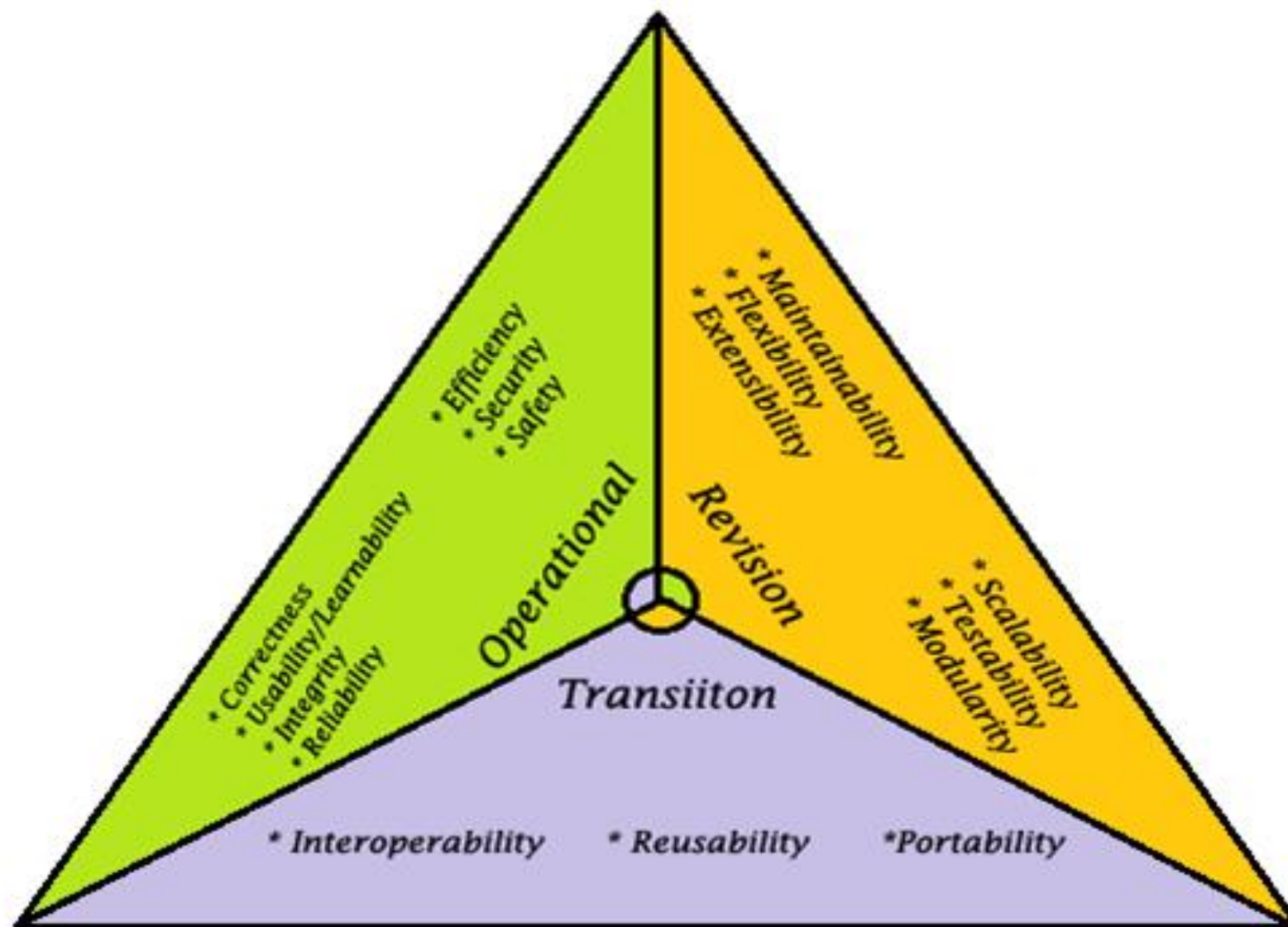
[elkinrilvani@gmail.com](mailto:elkinrilvani@gmail.com)



Mencerminkan seberapa baik atau sesuai dengan desain yang diberikan, berdasarkan persyaratan atau spesifikasi fungsional.

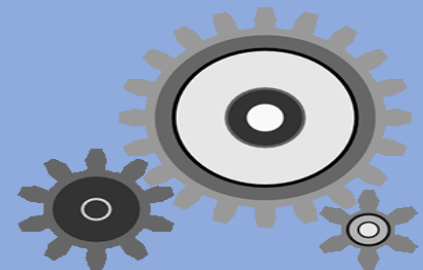


# TRIANGLE QUALITY SOFTWARE



# OPERASIONAL

- **Correctness (Kebenaran):** Perangkat lunak yang kita buat harus memenuhi semua spesifikasi yang dinyatakan oleh pelanggan.
- **Usability/Learnability (Kegunaan/Mudah dipelajari) :** Jumlah usaha atau waktu yang dibutuhkan untuk mempelajari bagaimana menggunakan software harus sedikit. Hal ini membuat perangkat lunak user-friendly bahkan untuk tidak menguasai IT.
- **Integrity :** Perangkat lunak berkualitas seharusnya tidak memiliki efek samping. (Pop-Up, Instalasi software tambahan).
- **Reliability (Keandalan):** Produk perangkat lunak seharusnya tidak memiliki cacat. Tidak hanya itu, seharusnya tidak gagal saat eksekusi.



# OPERASIONAL

- **Efficiency** : Perangkat lunak ini harus memanfaatkan ruang penyimpanan secara efektif dan menjalankan perintah sesuai kebutuhan waktu yang diinginkan.
- **Security** : Perangkat lunak seharusnya tidak memiliki efek buruk pada data / perangkat keras.
- **Safety** : Perangkat lunak seharusnya tidak berbahaya bagi lingkungan / kehidupan.





# ReVISION

- **Maintainability:** Pemeliharaan perangkat lunak harus mudah untuk semua jenis pengguna.
- **Fleksibilitas:** Perubahan dalam perangkat lunak harus mudah dilakukan.
- **Extensibility:** Seharusnya mudah untuk meningkatkan fungsi yang dilakukan olehnya.
- **Skalabilitas:** Ini harus sangat mudah untuk meng-upgrade untuk lebih banyak pekerjaan (atau untuk lebih banyak pengguna).
- **Testability:** Menguji perangkat lunak harus mudah.
- **Modularity :** Jika perangkat lunak dibagi menjadi bagian independen terpisah yang dapat dimodifikasi, diuji secara terpisah, ia memiliki modularitas tinggi.



# TRANSITION

- **Interoperability:** Interoperabilitas adalah kemampuan perangkat lunak untuk bertukar informasi dengan aplikasi lain dan memanfaatkan informasi secara transparan.
- **Reusability :** Jika didapat menggunakan kode perangkat lunak dengan beberapa modifikasi untuk tujuan yang berbeda maka kita sebut perangkat lunak agar dapat digunakan kembali.
- **Portability:** Kemampuan perangkat lunak untuk melakukan fungsi yang sama di semua lingkungan dan platform, menunjukkan portabilitasnya.





**SYARAT  
TESTING**







**Syarat  
Teknis  
Testing**

**Orang Yang  
Melakukan  
Testing**

**Menerapkan  
Prinsip  
Testing**

A 3D yellow figure stands on a white, cloud-like base, holding a large rectangular sign. The figure is simple, with a round head and no facial features.

## **Syarat Teknis Testing**

**Penguji Harus Memahami Software Yang diujikan**

**Menentukan Rencana Testing dan Hasil yang diharapkan**

**Adanya dokumentasi hasil Testing yang telah dilakukan**

**Memaksimalkan Oprasioanal Software bekerja dengan baik**



## Menerapkan Prinsip Testing

Testing merupakan pekerjaan yang kreatif dan sulit.

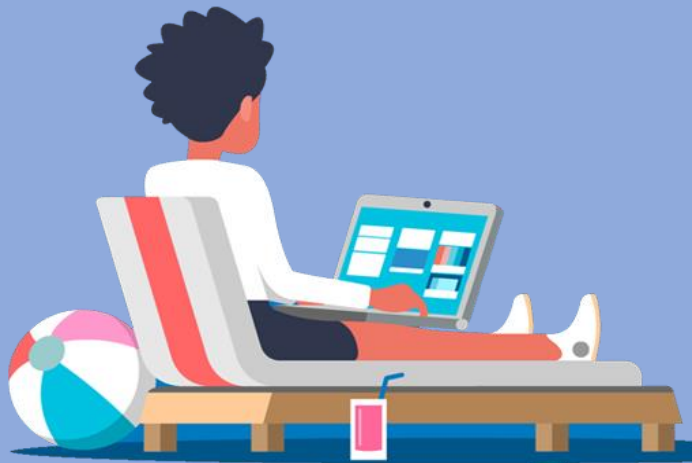
Tetap berpegang pada definisi Testing adalah proses eksekusi program dengan maksud menemukan kesalahan.

Uji kasus yang mewakili kondisi masukan yang tidak terduga dan tidak valid tampaknya memiliki hasil deteksi kesalahan yang lebih tinggi

Periksa secara seksama hasil setiap tes.

Testing harus independensi & menghasilkan suatu hasil objektif.

# Orang Yang Melakukan Testing



**INDEPENDENT**



 **Developer**

# CHARACTERISTIC



 Developer

Melakukan Testing tahap awal

Memahami Aplikasi yang diujikan

**BUT**

Tidak Menghasilkan Testing yang maksimal

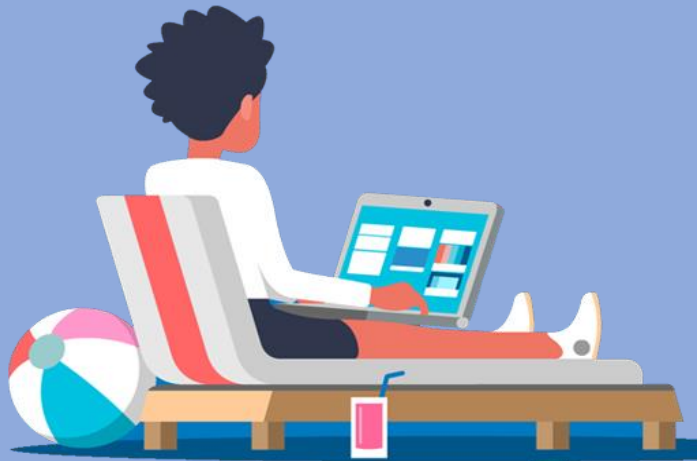
Ingin cepat selesai

**CHARACTERISTIC**

**Melakukan testing semua fitur dalam aplikasi**

**Tahap Testing tingkat lanjut**

**Tujuannya menghasilkan Software berkualitas**



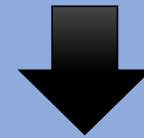
**INDEPENDENT**





*use***R!**

**Apakah User Melakukan Testing ?**



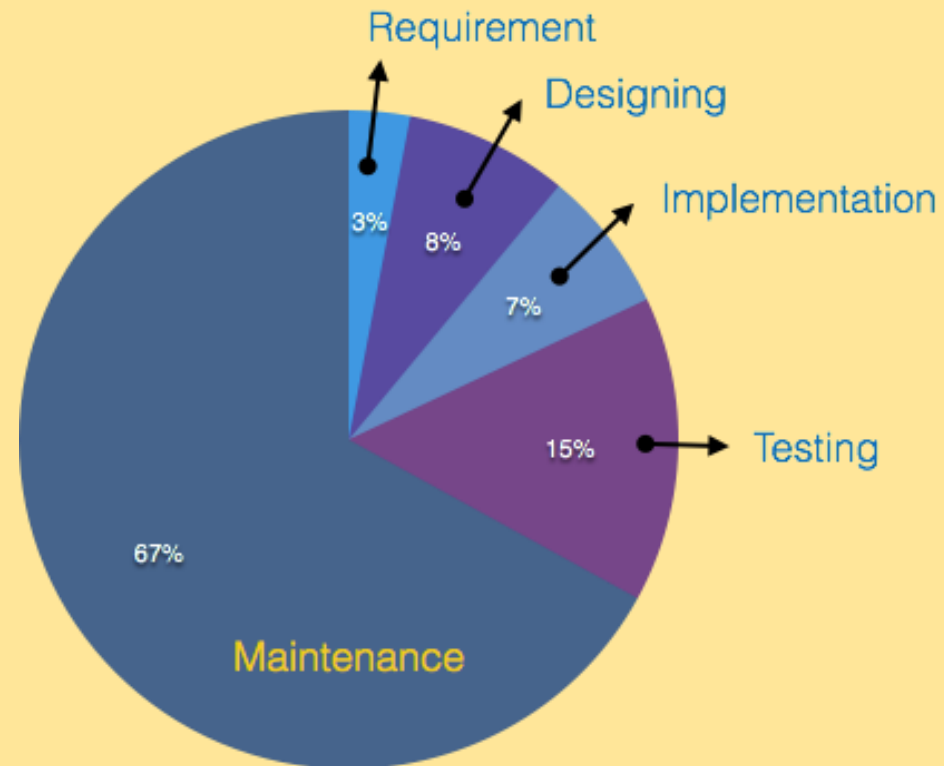
**User Mamakai atau  
Menggunakan Aplikasi**



**Pengujian Secara  
Tidak Langsung**



Of The Software





- Pengembangan software adalah bisnis yang sangat berisiko sehingga menghabiskan banyak waktu, tenaga dan uang. Tetapi biaya terbesar terjadi lama setelah pengembangan awal selesai.
- Dengan menerapkan praktik yang baik ini, dengan fokus pada pengembangan sistem yang memberikan apa yang benar-benar diinginkan pengguna, biasanya akan menghasilkan sistem yang memiliki kualitas pembuatan lebih tinggi dan 'sesuai untuk tujuan'.



- Dan biasanya menghasilkan sistem yang lebih mudah dipelihara dan direvisi sehingga mengurangi biaya operasional jangka panjang dan risiko yang terkait.

RISK

