

Testing

White-box testing

- Juga dikenal dengan **clear box testing, glass box testing, transparent box testing**, dan **structural testing**.
- white-box testing, kita membuat test cases dengan melihat source code untuk mencari adanya kesalahan pada program. White-box testing dilakukan oleh Software Engineer karena membutuhkan pengetahuan tentang programming dan implementasinya. Hal-hal yang biasa diuji dalam test cases seperti Loops(while or for loop), Decision Making (if statement or switch statement) atau data structure.

Keuntungan White-box testing

1. Sebagai Software engineer yang memiliki akses ke source code, hal ini menjadi sangat mudah untuk melakukan skenario pengujian secara efektif.
2. Membantu Software engineer untuk mengoptimalkan source code.
3. Baris kode yang tidak efisien dapat dihilangkan agar mencegah bugs pada program.

Kerugian White-box testing

1. Karena dibutuhkan Software engineer yang berpengalaman dalam White-box testing sehingga mengeluarkan biaya tambahan.
2. Terkadang sangat sulit melihat setiap baris kode untuk mencari bugs pada program yang akan diuji.

Black-box testing

- Black-box testing merupakan kebalikan dari White-box testing dimana Software Tester tidak memiliki akses source code atau mengetahui implementasi dari program tersebut untuk mencari adanya kesalahan pada program dan juga tidak diharuskan memiliki pengetahuan tentang programming dan implementasinya.
- Ketika melakukan Black-box testing, tester akan berinteraksi dengan user interface yang menyediakan input dan memeriksa outputnya, juga menguji performa program atau menguji function-function yang tidak bekerja dengan benar.

Keuntungan Black-box testing

1. Cocok dan efisien untuk source code dengan skala besar.
2. Menguji program dari sudut pandang user.
3. Software tester dalam jumlah yang banyak dapat menguji program tersebut tanpa harus memiliki pengetahuan tentang programming.

Kerugian Black-box testing

1. Software tester hanya menjalankan beberapa skenario pengujian yang dipilih.
2. Pengujian yang tidak efisien karena Software tester memiliki pengetahuan yang terbatas tentang program.
3. Pengujian yang tidak spesifik karena Software tester tidak memiliki akses ke source code.

Proses Black-Box Testing



Contoh Black Box Testing

No	Skenario	Test Case	Hasil yang diharapkan	Status
1.	Berhasil Remove Album dari cart dengan jumlah Album masing-masing nama 1 buah	User menghapus Album yang diinginkan "Remove from cart"	Sistem menghapus Album dari daftar cart	Pass
2	Berhasil Remove Album dari cart dengan jumlah Album lebih dari 1 buah	User menghapus Album yang diinginkan "Remove from cart"	Sistem mengurangi jumlah Album yang dihapus	Failed

Pengujian Usabilitas SUS

System Usability Scale (SUS)

The System Usability Scale (SUS) provides a “quick and dirty”, reliable tool for measuring the usability. It consists of a 10 item questionnaire with five response options for respondents; from Strongly agree to Strongly disagree. Originally created by John Brooke in 1986, it allows you to evaluate a wide variety of products and services, including hardware, software, mobile devices, websites and applications.

Benefits of using a SUS

SUS has become an industry standard, with references in over 1300 articles and publications. The noted benefits of using SUS include that it:

- Is a very easy scale to administer to participants
- Can be used on small sample sizes with reliable results
- Is valid – it can effectively differentiate between usable and unusable systems

Pengujian Usabilitas SUS

Considerations when using a SUS

If you are considering using a SUS, keep the following in mind:

- The scoring system is somewhat complex
- There is a temptation, when you look at the scores, since they are on a scale of 0-100, to interpret them as percentages, they are not
- The best way to interpret your results involves “normalizing” the scores to produce a percentile ranking
- SUS is not diagnostic - its use is in classifying the ease of use of the site, application or environment being tested

Pengujian Usabilitas SUS

The System Usability Scale

When a SUS is used, participants are asked to score the following 10 items with one of five responses that range from Strongly Agree to Strongly disagree:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.