

# MEKANISME BUG FIXING

Oleh: Bety Etikasari

## Tujuan Pembelajaran:

1. Memahami tentang bug
2. Mengetahui penyebab yang memungkinkan berakibat terjadinya bug
3. Mengetahui jenis-jenis bug
4. Mengetahui pengelompokan priority level of bug
5. Memahami bug life cycle
6. Mengetahui upaya untuk menghindari terjadinya bug

# Apa itu Bug?

- **Bug** can be defined as the abnormal behavior of the software. No software exists without a bug. The elimination of bugs from the software depends upon the efficiency of testing done on the software. A bug is a specific concern about the quality of the Application under Test (AUT).
- **Bug** adalah sebuah kesalahan atau cacat pada sistem baik itu Software maupun Hardware yang menimbulkan sistem berjalan secara tidak normal bahkan sampai tidak dapat digunakan. Bug sendiri dapat di klasifikasikan sesuai dengan tingkat kerusakan yang ditimbulkan, sehingga pengklasifikasian bug tersebut dapat memberikan informasi kepada developer untuk lebih memprioritaskan Bug mana dulu yang akan dibenahi.



# PENYEBAB BUG PADA SOFTWARE

- Komunikasi yang kurang baik
- Kerumitan Software
- Tidak adanya log perubahan kode
- Deadline yang terlalu singkat
- Kesalahan logis pada desain software
- Aplikasi Third-Party

# JENIS-JENIS BUG

## Functional error

- mencakup masalah fungsionalitas sebuah program. *Bug* jenis ini bervariasi, mulai dari tombol yang tidak dapat di-klik hingga masalah pada kegunaan aplikasi itu sendiri

## Performance defects

- kecepatan, stabilitas, *response time*, dan penggunaan sumber daya dari sebuah *software*

## Usability defects

- masalah pada kode software, desain UI yang terlalu rumit

## Compatibility error

- merujuk pada masalah *software* yang tidak dapat berjalan sebagaimana mestinya dalam situasi tertentu

## Security error

- sistem keamanan *software*, besar atau kecil akan membuka celah untuk *cracking* dan mencuri data atau merusak *software*

## Syntax error

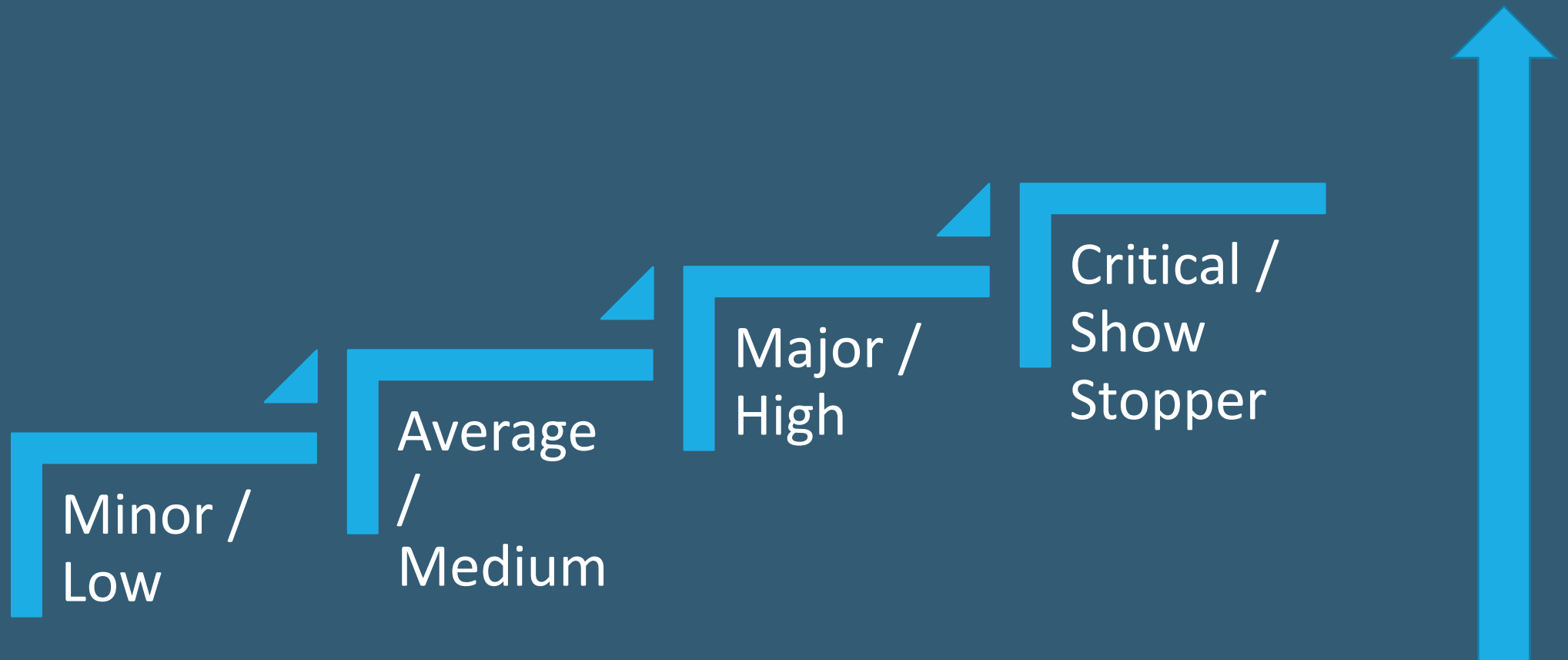
- software tidak dapat di-compile/source code error

## Logic error

- penggunaan kode yang tidak sesuai sehingga menghasilkan *output* yang salah

- Saat melakukan testing sebagai seorang tester banyak sekali bug berkeliaran pada sistem itu, pada saat itu juga sebagai seorang tester harus memfilter mana bug yang sifatnya sangat berpengaruh besar terhadap sistem mana yang hanya punya pengaruh kecil.
- Ketika sebuah bug ditemukan, maka developer akan memperbaiki kesalahan tersebut. Proses pembersihan bug ini disebut proses **debugging**.
- Dengan memfilter jenis-jenis bug akan sangat berpengaruh pada kinerja developer untuk memprioritaskan bug yang akan diperbaiki terlebih dahulu sehingga kinerja dari sistem dan developer bisa selaras dengan waktu yang telah ditetapkan

# Priority Levels Of Bugs - Pengelompokan Bug



# MINOR / LOW

- Bug ini merupakan level terendah dari sebuah bug atau bisa dibilang terkecil tingkatnya, artinya Bug Minor/Low memiliki tingkat kesalahan paling rendah sehingga hal ini terkadang tidak terlalu berpengaruh pada jalannya sebuah sistem. Kebanyakan developer menganggap sepele bug jenis ini karena berkaitan dengan hal-hal kecil yang mungkin menurut mereka belum terlalu urgent untuk dikerjakan dalam waktu dekat.
- Contoh:
  - Penggunaan Tampilan yang tidak selaras atau tidak sesuai desain dari UI
  - Penggunaan Caption yang typo seperti “Sign Up” menjadi “Sing Pu”



# AVERAGE / MEDIUM

- Bug ini merupakan level menengah yang tingkat urgentnya bisa di anggap penting meski tidak terlalu berpengaruh besar terhadap jalannya sebuah sistem. Akan tetapi bug ini tidak bisa disepelekan begitu saja karena pada beberapa kasus bug jenis ini lama kelamaan akan menjadi hal yang harus segera dikerjakan, karena user akan semakin sadar bahwa ternyata hal ini mengganggu efektifitas mereka dalam menggunakan sistem yang dibuat
- Contoh:
  - Saat kita add sebuah data member, setelah itu kita lakukan save, biasanya akan keluar notifikasi sukses pada sistem akan tetapi data tersebut tidak muncul pada List Member
  - Saat kita klik button login yang harusnya masuk ke dashboard malah masuk kedalam menu register

# MAJOR / HIGH

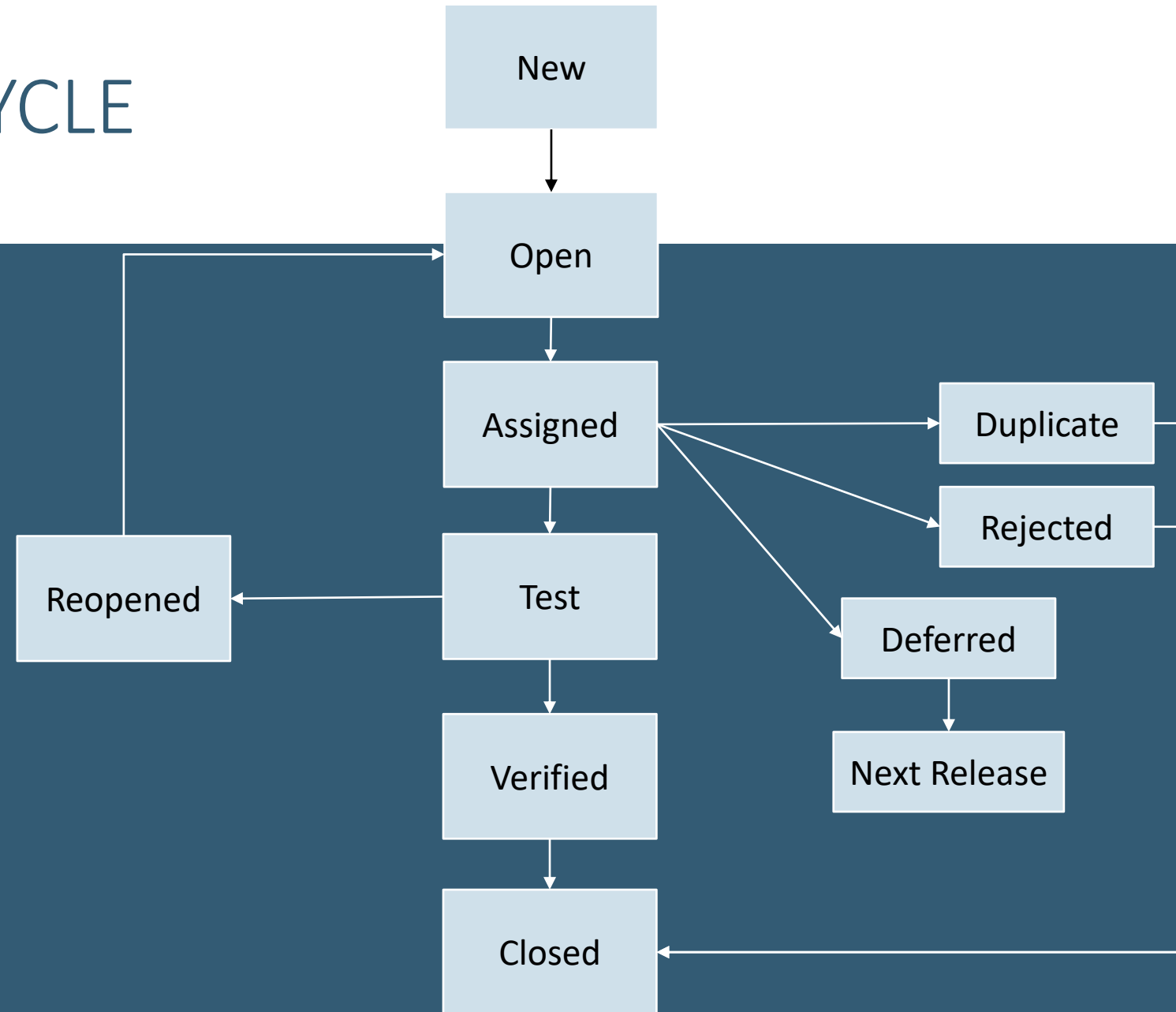
- Bug ini bisa dibilang sebagai bug Level Menengah Atas, peranan bug ini sangat terasa sekali terhadap fungsionalitas sistem. Tentu bug ini yang paling sering diperhatikan oleh Developer untuk segera dikerjakan, tingkat urgennya pun tidak main-main karena ketika bug ini ditemukan maka banyak pengaruhnya pada penggunaan user atau bahkan bisa jadi user tidak bisa memakai sistem sewajarnya
- Contoh:
  - Saat kita add member baru pada sistem kemudian kita save maka sistem terus menganggap ada kesalahan pada kolom tertentu sehingga tidak dapat di save padahal secara input yang kita lakukan sudah benar, maka hal ini akan membuat user sangat panik bahkan bisa-bisa dia stress karena user akan bertanya apa yang salah dari proses yang dilakukan.
  - Saat kita add member baru kemudian ketika klik save sistem memunculkan error tertentu yang isinya adalah bahasa mesin.

# CRITICAL / SHOW STOPPER

- Bug jenis ini merupakan bug dengan Level Tertinggi, pengaruhnya teramat besar pada sistem sehingga sistem benar-benar tidak bisa digunakan sama sekali. Biasanya apabila ada temuan Bug Critical pada sebuah sistem, maka developer akan membutuhkan waktu lebih banyak karena bisa saja terjadi kesalahan mulai dari alur, database dan hal-hal terkait lainnya yang membuat sistem benar-benar tidak bisa digunakan atau di akses oleh user. Akan tetapi perlu diketahui bahwa bug jenis ini sangat jarang ditemui pada sistem yang di Rancang dan disusun dengan baik dan benar sebelum adanya proses Development
- Contoh:

Sistem tidak dapat dibuka atau digunakan sama sekali, software yang sering mengalami hal ini seperti software dengan platform Desktop ataupun Mobile Apps.

# BUG LIFE CYCLE



## New

When the bug is posted for the first time, its state will be “NEW”. This means that the bug is not yet approved

## Open

After a tester has posted a bug, the lead of the tester approves that the bug is genuine and he changes the state as “OPEN”

## Assign

Once the lead changes the state as “OPEN”, he assigns the bug to corresponding developer or developer team. The state of the bug now is changed to “ASSIGN”

---

## Test

Once the developer fixes the bug, he has to assign the bug to the testing team for next round of testing. Before he releases the software with bug fixed, he changes the state of bug to “TEST”. It specifies that the bug has been fixed and is released to testing team

---

## Deferred

The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software

---

## Rejected

If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to “REJECTED”

---

## Duplicate

If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to “DUPLICATE”

---

## Verified

Once the bug is fixed and the status is changed to “TEST”, the tester tests the bug. If the bug is not present in the software, he approves that the bug is fixed and changes the status to “VERIFIED”

---

## Reopened

If the bug still exists even after the bug is fixed by the developer, the tester changes the status to “REOPENED”. The bug traverses the life cycle once again

---

## Closed

Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to “CLOSED”. This state means that the bug is fixed, tested and approved

---



# Upaya Yang Dapat Dilakukan Pada Proses Development Untuk Menghindari Bug

- Membangun alur komunikasi yang baik
- Melakukan testing
- Menjalankan program Bug Bounty
- Menggunakan Agile Methodology
- Menggunakan platform kolaborasi

SELAMAT BELAJAR