



Materi 01

# Quality Assurance Fundamental

Fachrul Pralienka Bani Muhamad  
fachrul.pbm@polindra.ac.id

# Outlines

1. Pengujian produk
2. Quality assurance (QA) vs Quality control (QC)
3. QA skills and mindset
4. Black box testing
5. White box testing
6. Grey box testing

# Capaian Pembelajaran

Setelah materi ini disampaikan, mahasiswa diharapkan mampu:

- Menjelaskan pentingnya pengujian perangkat lunak
- Menjelaskan perbedaan QA dan QC
- Menjelaskan prinsip dan pola pikir QA
- Menjelaskan 3 metode pengujian, black box, white box, dan grey box



# 1. Pengujian Produk

# Cacat produk



# Cacat produk

- Semua produk berpeluang untuk cacat
- Agar produk minim dari cacat, serta sesuai dengan kebutuhan bisnis dan kebutuhan pengguna, maka perlu dilakukan pengujian
- Kecacatan produk tersebut disebabkan oleh 5 faktor utama:
  - a. Manusia (Man)
  - b. Bahan baku (Material)
  - c. Mesin (Machine)
  - d. Metode (Method)
  - e. Lingkungan (Environment)

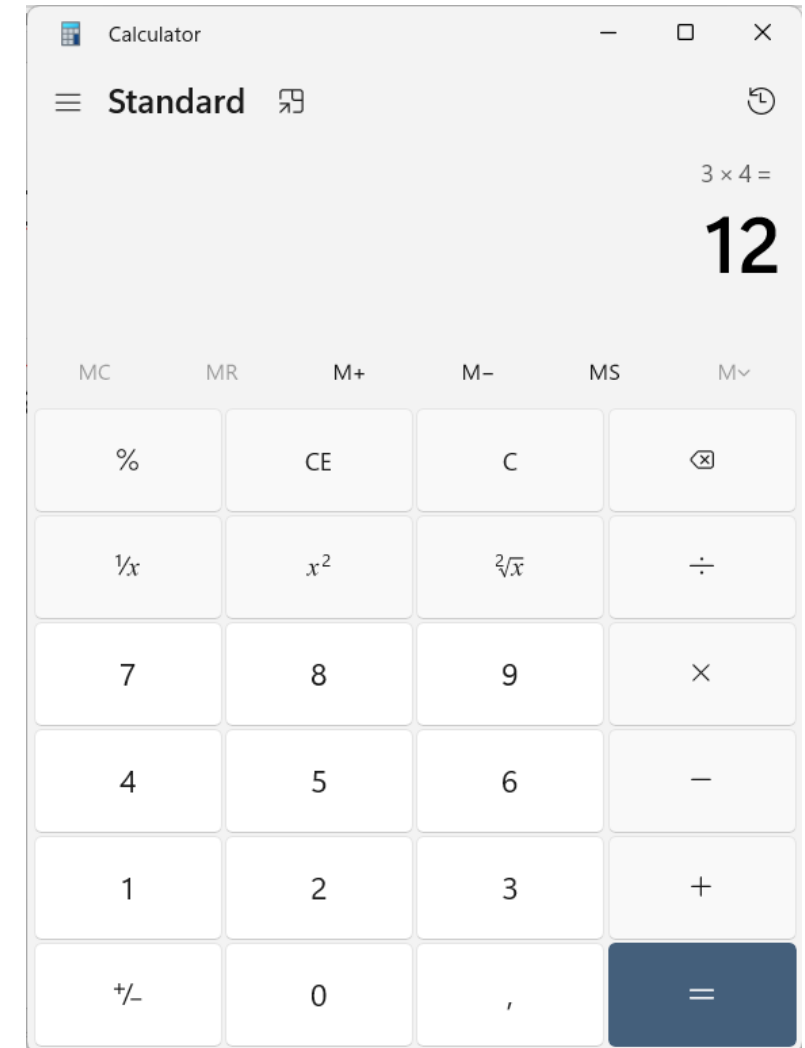


# Definisi Pengujian Perangkat Lunak

Serangkaian aktivitas yang dilakukan untuk mengevaluasi, memvalidasi, dan memverifikasi komponen-komponen serta fungsi dari suatu perangkat lunak berdasarkan hasil yang diinginkan

# Contoh Pengujian Perangkat Lunak

- Misalnya pengujian Aplikasi kalkulator dalam melakukan perhitungan  $3 \times 4$  yang hasilnya sama dengan 12

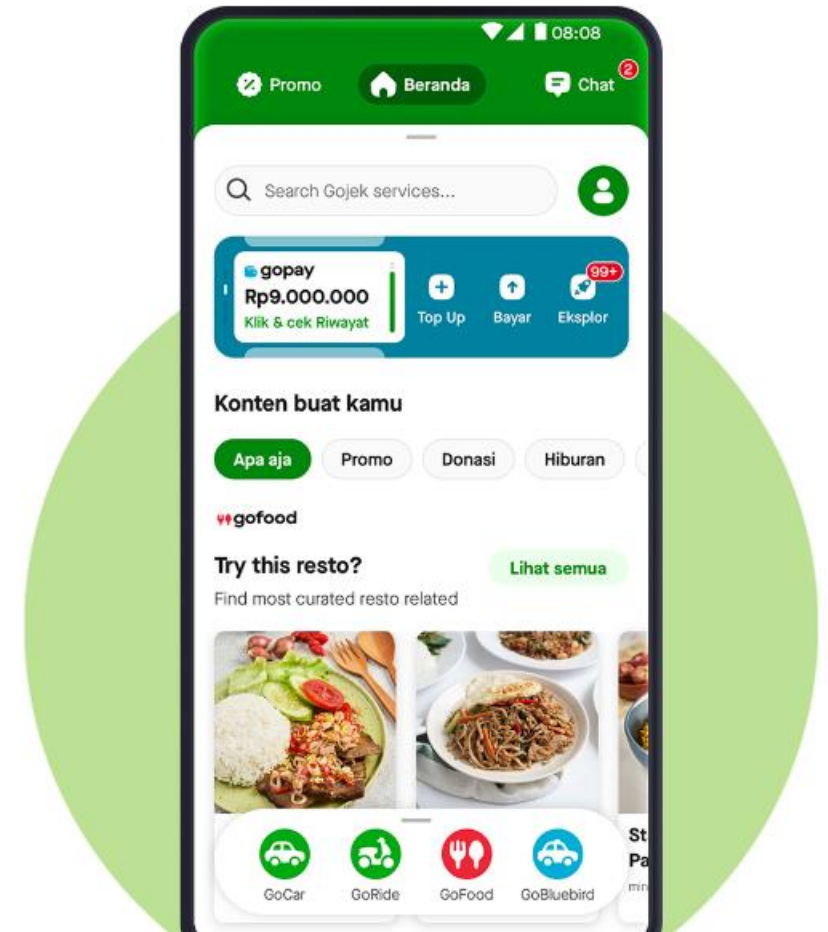




# Contoh Pengujian Perangkat Lunak

## Tujuan Aplikasi Ojek Online:

- Dapat memesan **makanan** dari suatu resto hingga diantarkan ke lokasi pemesan
- Dapat memesan **kendaraan** untuk jemput antar penumpang
- Dapat memberikan **informasi** biaya jasa driver, dan lokasi driver
- Dapat **menghubungi** driver saat melakukan pemesanan



# Contoh Kegagalan Perangkat Lunak



Kecelakaan yang disebabkan kegagalan perangkat lunak:

- **4 Juni 1996**

Roket Ariane 5 yang diluncurkan oleh European Space Agency karena kesalahan konversi tipe data

<https://www.idntimes.com/science/experiment/pri-145/fakta-ariane-roket-kelas-berat-eropa-c1c2-1#:~:text=4.%20Kegagalan%20peluncuran%20Ariane%205&text=Kesalahan%20yang%20paling%20fatal%20terjadi,r obek%20pada%20ketinggian%204%20kilometer.>

# Contoh Kegagalan Perangkat Lunak



Kecelakaan yang disebabkan kegagalan perangkat lunak:

- **29 Oktober 2018**

Jatuhnya pesawat Lion Air JT 610 karena perangkat lunak yang mengontrol hidung pesawat (MCAS) salah dalam melakukan kalibrasi

*<https://www.kompas.com/tren/read/2022/10/29/082900165/hari-ini-dalam-sejarah--tragedi-jatuhnya-lion-air-jt-610-189-orang?page=all#:~:text=Investigasi%20penyebab%20jatuhnya%20Lion%20Air%20JT%20610&text=Dalam%20laporan%20tersebut%2C%20KNKT%20menyimpulkan,kurangnya%20dokumentasi%20tentang%20sistem%20pesawat.>*

# Faktor pentingnya pengujian

- **Trust** → kepercayaan pelanggan dari kualitas produk (kredibel)
- **Safety** (lives) → kesalahan pada sistem kritis berdampak pada hilangnya nyawa
- **Security** → menghindari kerugian dari pencurian data
- **Money** → menghemat biaya untuk perbaikan
- **Times** → menghemat waktu untuk perbaikan
- **Effort** → menghemat tenaga untuk perbaikan

# Peran QA

- Membangun kepercayaan customer terhadap produk
- Menjaga kualitas produk
- Mediator antara user, bisnis, produk, dan developer
- Mengkomunikasikan secara verbal dan tertulis



## 2. Quality Assurance (QA) vs Quality Control (QC)

# Perbedaan QA vs QC

- QA berfokus pada teknik yang berorientasi pada **proses**, tentang bagaimana perangkat lunak tsb dibuat atau disampaikan
- QC berfokus pada teknik yang berorientasi pada **produk** atau *final product*, hingga memastikan bahwa standar produk telah terpenuhi ketika dilakukan pengujian

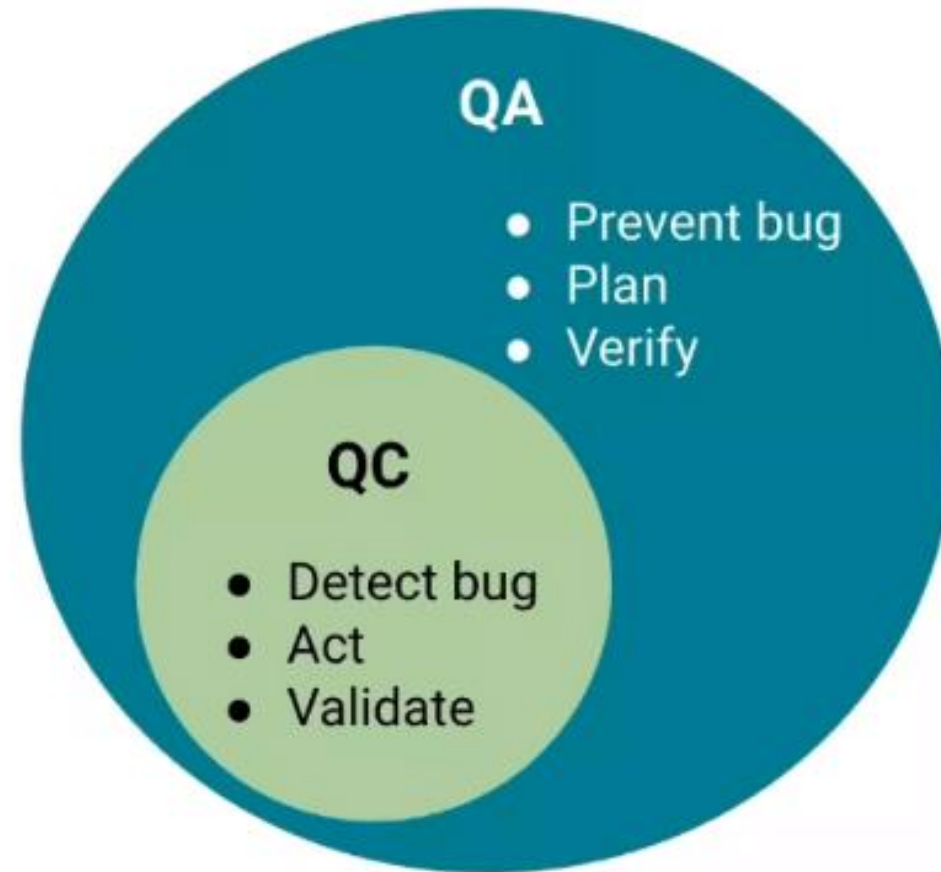


# Perbandingan QA vs QC

Quality Assurance (QA)	Quality Control (QC)
Memastikan bahwa telah dilakukan hal yang benar dengan cara yang benar	Memastikan bahwa hasil dari apa yang dilakukan telah sesuai dengan ekspektasi
Berfokus untuk membangun kualitas dan mencegah terjadinya cacat	Berfokus untuk menguji kualitas dan mendeteksi cacat
Berurusan dengan proses	Berurusan dengan produk
Dilakukan pada siklus pengembangan perangkat lunak	Dilakukan pada siklus pengujian perangkat lunak
QA adalah proses preventif (pencegahan)	QC adalah proses korektif (perbaikan)



# Perbedaan Utama





# 3. QA skills and Mindset

# QA Mindset & Principles

- Pengujian dilakukan untuk **menunjukkan adanya cacat**
- Pengujian dilakukan **seawal mungkin**
- Pengujian yang **lengkap tidak mungkin** dilakukan
- Pengujian dilakukan berdasarkan **konteks (domain)**
- *Defect* (cacat) perangkat lunak perlu **dikelompokkan** (identifikasi)
- *Pesticide paradox* → pestisida lama kelamaan membuat hama kebal
- *Absence of error fallacy* → merasa tidak error (asumsi)

# QA Skills

1. Kemampuan **memperhatikan** detail produk
2. Kemampuan manajemen **waktu**
3. Kemampuan **multitasking**
4. Kemampuan **penyelesaian** masalah
5. Kemampuan **komunikasi** tertulis dan lisan
6. Kemampuan untuk **bekerja** sebagai bagian dari **tim**
7. Pengetahuan tentang **standar kualitas**
8. Kemampuan memahami **proses bisnis** produk

# Fase Pengujian Perangkat Lunak

## Requirement analysis

- Tinjauan umum persyaratan (kebutuhan) pengguna
- Memahami persyaratan yang penting untuk pengujian produk

## Test case design and development

- Digunakan untuk menemukan apa saja persyaratan komponen yang diperlukan
- Spesifikasi desain diuji dan diperiksa lalu dibuat tinjauan atau review dari spesifikasi pengujian



# QA Testing Methods

- Black box testing
- White box testing
- Grey box testing



# 4. Black Box Testing

# Pendahuluan Black Box Testing

Pengujian yang dilakukan tanpa mengetahui proses logik kode program

## Kelebihan

- Tidak membutuhkan hardskill basic pemrograman
- Dapat dilakukan dari sudut pandang pengguna (tepat sasaran)
- Cepat dan sederhana

## Kekurangan

- Kemungkinan terjadi bug yang aman dari sudut pandang pengguna, tetapi akan bermasalah pada program maupun database di kemudian hari
- Berpotensi pada terlewatnya suatu bagian proses pengujian akibat dari konsentrasi yang terbatas
- Dibutuhkan pengetahuan produk secara rinci, proses bisnis, beserta alur pengguna secara keseluruhan (mendalam)



# Contoh Black Box Testing

**Equivalence Partitioning (EP):** Pembagian input data berdasarkan spesifikasi input yang dibutuhkan perangkat lunak

Misalnya program kalkulator hanya menerima bilangan genap, maka data input akan dibagi menjadi beberapa kelompok data, yaitu:

- Bilangan **genap**: expected program berjalan lancar
- Bilangan **ganjil**: expected program menampilkan pesan error “khusus bilangan genap”
- Angka **nol**: expected program menampilkan error “tidak boleh nol”
- **Bukan** angka: expected program menampilkan error “input hanya angka”

# Contoh Black Box Testing

**Boundary Value Analysis (BVA):** Teknik pengujian untuk memeriksa batasan-batasan input dari suatu perangkat lunak

Misalnya input tanggal khusus antara [10,20] atau 10 diterima tetapi 20 tidak, maka data input dapat dibagi menjadi:

- Input angka 9 untuk memeriksa batas bawah yang **tidak diterima**
- Input angka 10 untuk memeriksa batas bawah yang **diterima**
- Input angka 19 untuk memeriksa batas atas yang **diterima**
- Input angka 20 untuk memeriksa batas atas yang **tidak diterima**

# Contoh Black Box Testing

Teknik lainnya:

- Cause-effect graphing techniques
- Orthogonal array testing
- All-pairs testing
- Fuzz testing
- State testing (UI testing)



# 5. White Box Testing

# Pendahuluan White box Testing

- Proses pengujian untuk mengetahui cara kerja internal perangkat lunak
- Bertujuan untuk menjamin operasi-operasi telah sesuai dengan spesifikasi yang telah ditentukan sebelumnya

# Pro & Cons White box Testing

## Kelebihan

- Lebih rinci (detail)
- Lebih mudah dilakukan otomasi
- Lebih cepat mengetahui bug (karena diuji sebelum ada GUI)

## Kekurangan

- Partial source code testing
- Kemungkinan pengulangan pengujian yang sama antara penguji dan pengembang
- Tidak semua kemungkinan flow input output dijalankan

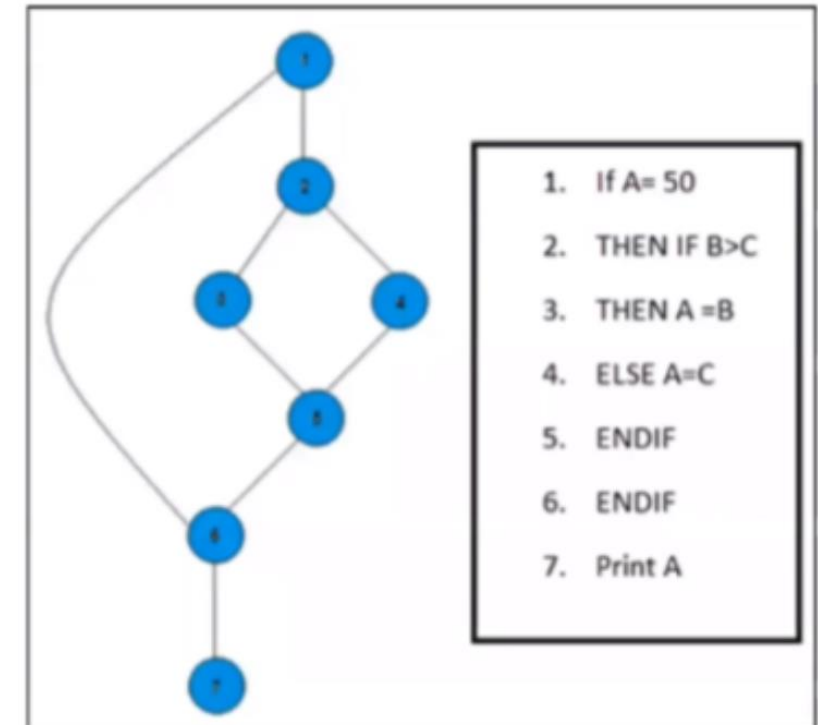
# Contoh White Box Testing

## Basis path

- Pembuatan alur yang memungkinkan dari sebuah program mulai dari input hingga output

## Cyclomatic complexity

- Proses untuk mengukur kompleksitas suatu jalur (*path*) dan alur (*flow*) pada perangkat lunak





# 6. Grey Box Testing



# Pendahuluan Grey Box Testing

- Penggabungan proses black box dan white box testing
- Penguji memiliki akses pada kode program dan basis data
- Penguji dapat membuat rencana dan eksekusi pengujian yang lebih komprehensif

# Pro & Cons Grey Box Testing

## Kelebihan

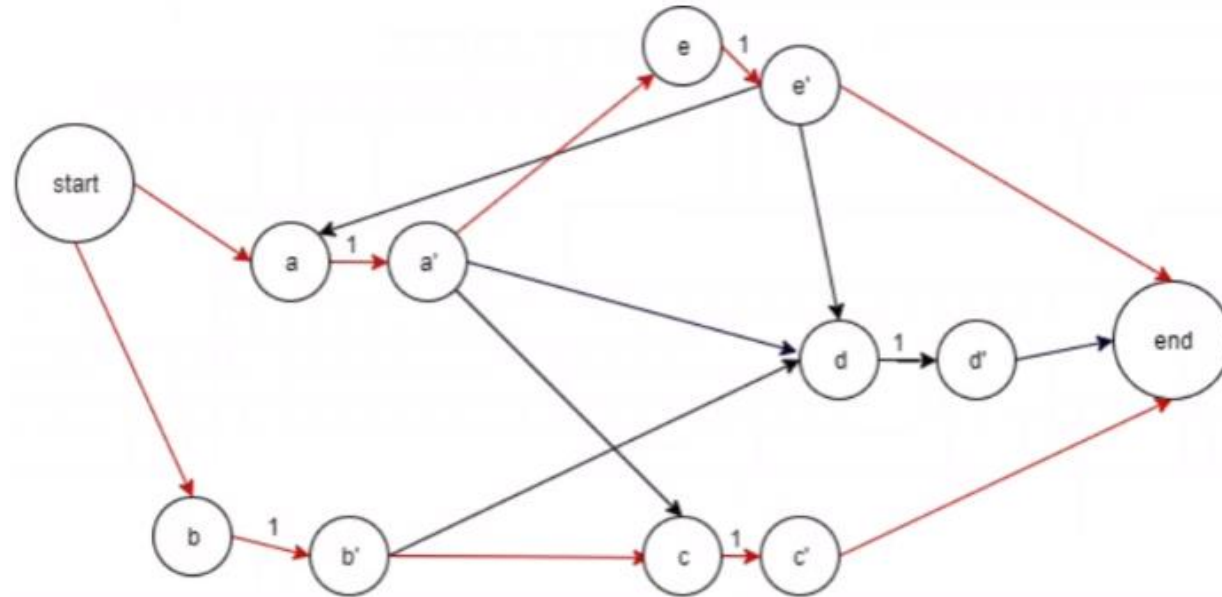
- Dimungkinkan untuk melakukan kombinasi metode pengujian
- Unbiased testing, tetap ada batasan antara pengembang (developer) and penguji (tester)

## Kekurangan

- Penguji perlu mengetahui pemrograman dari level basic hingga level advance
- Cenderung lebih kompleks dan mahal dibandingkan black atau white box testing
- Dibutuhkan waktu dan usaha yang lebih banyak terhadap perangkat lunak yang kompleks

# Masalah Pengujian

- *Complete testing*



- *Oracle problem (the source of truth)* → kehilangan informasi penting
- *Testing for product to work* → pengujian terhadap fungsionalitas utama saja



# Kesimpulan

# Kesimpulan

- Proses pengujian berlangsung di sepanjang siklus hidup pengembangan perangkat lunak
- Pengujian dilakukan baik secara statis (tanpa eksekusi source code) maupun dinamis (eksekusi source code)
- Proses pengujian mencakup perencanaan, persiapan, dan evaluasi produk perangkat lunak
- Proses memastikan terpenuhinya persyaratan / kebutuhan
- Proses mendeteksi cacat (defect)