A vibrant collage of illustrations related to software engineering and technology. It includes a large blue target with a yellow arrow hitting the bullseye, a person working on a laptop, a stack of books, a person carrying a large purple coffee cup, a person working on a large screen displaying a grid, and various other icons like databases and charts.

RPL101

# Pengantar Rekayasa Perangkat Lunak

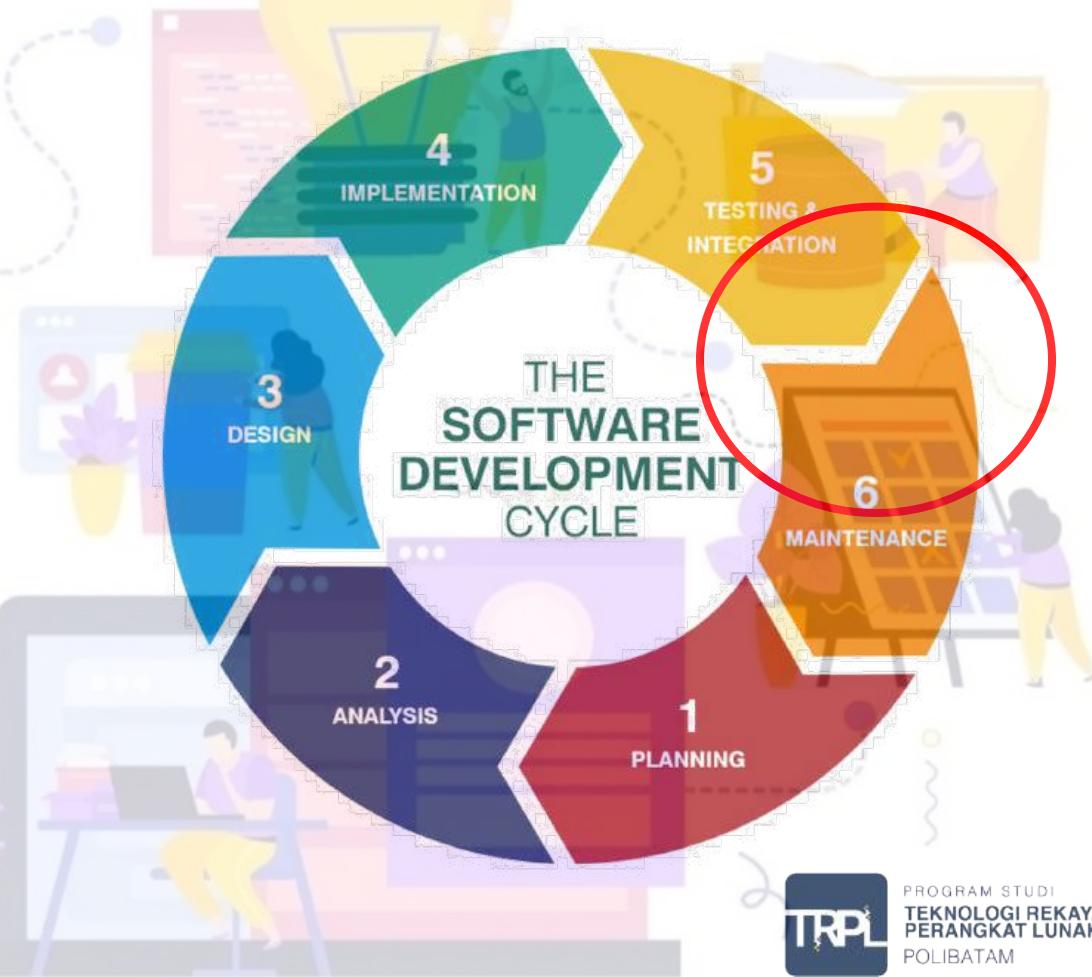
## *Introduction to Software Engineering*

Semester Ganjil Tahun Ajaran 2025-2026



PROGRAM STUDI  
**TEKNOLOGI REKAYASA**  
**PERANGKAT LUNAK**  
POLIBATAM

## *6. Deployment*



# Latar Belakang

Karena aplikasi dikembangkan untuk sponsor dan pengguna tertentu, aplikasi tersebut akhirnya harus berjalan di sumber daya perangkat keras dan lingkungan mereka, yang kami sebut lingkungan sponsor (atau pengguna/target/konsumen). Jadi, ketika tim pengembang merasa puas dan siap untuk "merilis" aplikasi, aplikasi tersebut harus dipindahkan dari lingkungan produsen dan diterapkan di lingkungan konsumen agar dapat digunakan oleh pengguna.

Terdapat dua jenis aplikasi:

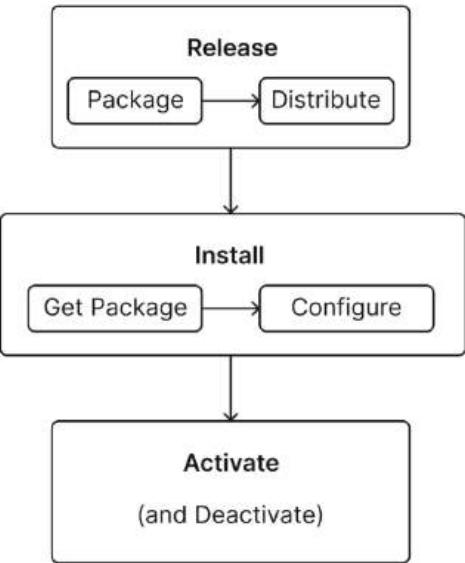
1. aplikasi mandiri yang dapat diinstal dan berjalan di komputer pengguna
2. aplikasi web, dimana *front-end* aplikasi berjalan di dalam browser (yang berjalan di komputer pengguna), dan *back-end* berjalan di server tertentu di lingkungan lain, dan basis data berjalan di server basis data yang dapat dihubungkan dengan server *back-end*.

# Proses Deployment

Proses deployment merupakan aktivitas yang dilakukan sehingga aplikasi dapat digunakan oleh pengguna

Proses deployment terdiri dari beberapa aktivitas:

- Rilis/*release*: Menyiapkan dan mengemas aplikasi untuk ditransfer
- Instalasi/*install*: Memindahkan package ke lingkungan target dan melakukan konfigurasi
- Aktivasi/*activate*: Memulai aplikasi untuk penggunaan yang sebenarnya



Tantangan utama dalam melakukan deployment adalah mengelola kebergantungan/*dependencies* antar komponen (komponen aplikasi bergantung pada komponen lain dalam aplikasi atau di lingkungan). Jika *dependencies* tidak tersedia, maka saat komponen tersebut dijalankan di lingkungan target, komponen tidak akan berfungsi dengan baik.

# Release and Packaging

- Rilis adalah tindakan memindahkan aplikasi dari tahap pengembangan ke tahap penggunaannya.
  - Rilis aplikasi perangkat lunak melibatkan penyediaan versi final atau pembaruan/versi terbaru perangkat lunak kepada pengguna untuk diinstal dan digunakan.
  - Sebuah aplikasi dianggap siap untuk dirilis setelah fitur/persyaratannya telah diimplementasikan, semua skrip pengujian telah berhasil dijalankan, dan rencana pengujian sistem menyeluruh telah diselesaikan dengan memuaskan.
- Fase rilis terutama mempersiapkan aplikasi untuk ditransfer dari lingkungan produsen ke lingkungan konsumen.
  - Selama pengemasan/*packaging*, sumber daya yang diperlukan agar aplikasi perangkat lunak dapat beroperasi dengan benar dikumpulkan dan biasanya dikemas dalam satu berkas yang dapat dengan mudah ditransfer ke lingkungan target dan diinstal di sana.
  - Sumber daya tersebut mencakup berkas kode aplikasi serta berkas sumber daya lainnya (berisi gambar, data, kode akses, dll.) yang dibutuhkan oleh kode selama eksekusi. Paket tersebut biasanya juga terdiri dari daftar dependensi eksternal yang menjadi rujukan kode aplikasi.

# Packaging

## Aplikasi mandiri

- *Packaging* terdiri dari penggabungan semua kode untuk suatu aplikasi, beserta resource yang digunakan oleh kode tersebut (data, gambar, skrip, dll.), ke dalam sebuah "paket", yaitu satu file tunggal. Paket ini kemudian dapat ditransfer ke pengguna akhir melalui berbagai mekanisme distribusi seperti email, unduhan, App Store, dll.
- Paket ini berisi semua berkas dan informasi yang diperlukan untuk menginstal dan menjalankan aplikasi di komputer pengguna. Paket ini juga dapat berfungsi sebagai penginstal aplikasi melalui alat yang sesuai.

## Aplikasi web

- Front-end code (HTML/JavaScript) berjalan di browser
- Back-end code berjalan di server
- Database berjalan di database server
- *Packaging* meliputi pengaturan semua komponen berserta dependencies-nya
- Dapat ditransfer sebagai file archive atau di-clone dari repositories

# Instalasi dan Aktivasi

Tahapan instalasi terdiri dari dua sub-tahapan:

- Transfer paket dari host ke target
- Konfigurasi aplikasi sampai siap digunakan

Setelah instalasi selesai, aplikasi dapat diaktifasi oleh pengguna.

## Aplikasi mandiri

- Ekstrak file dari paket
- Jalankan setup scripts atau ikuti petunjuk manual
- Seting *environment variables*
- Jalankan paket aplikasi

## Aplikasi web

- Memerlukan setup web server, application server, dan database server
- Konfigurasi *network routing* dan nama domain
- Instal dependencies dan konfigurasi hubungan antar komponen
- Lebih kompleks karena arsitektur multi-component

# Jenis-jenis Deployment

## 1. On-Premises Deployment

Pada On-premise, deployment aplikasi atau sistem di-deploy di dalam infrastruktur lokal suatu organisasi atau perusahaan. Oleh karena itu, ia memerlukan perawatan infrastruktur hardware dan software secara langsung.

## 2. Cloud Deployment

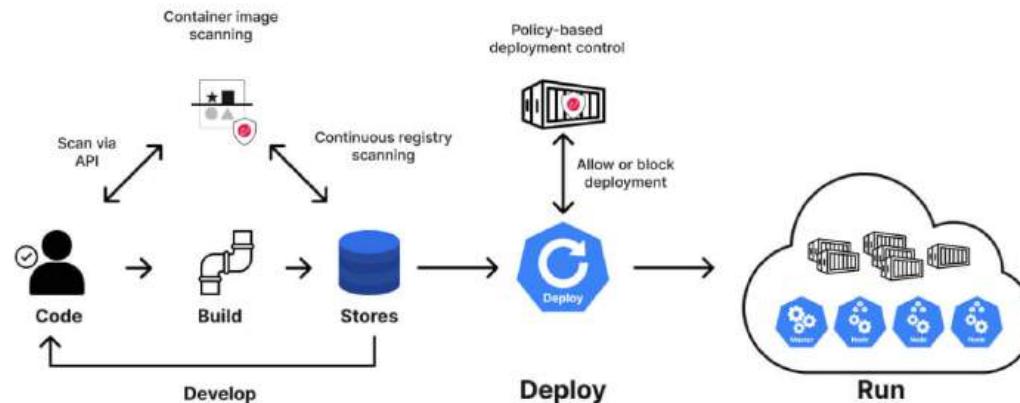
Aplikasi atau sistem di-deploy di lingkungan cloud. Biasanya kita akan menyewa/membayar infrastruktur atau layanan yang disediakan oleh penyedia layanan cloud, seperti Amazon Web Services (AWS), Microsoft Azure, atau Google Cloud.



# Jenis-jenis Deployment

## 3. Container Deployment

Aplikasi di-deploy di dalam container yang independen dan portabel sehingga dapat dijalankan secara konsisten di berbagai lingkungan yang berbeda, baik di lingkungan cloud maupun on-premises. Contohnya, deploying aplikasi microservices dalam wadah Docker di lingkungan Kubernetes.

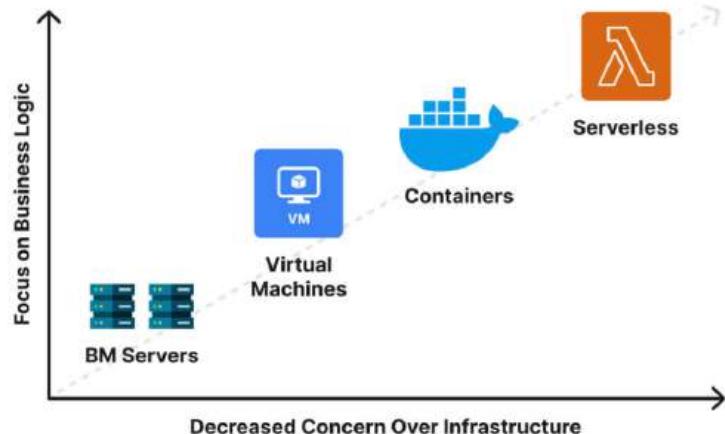


# Jenis-jenis Deployment

## 4. Serverless Deployment

Aplikasi dijalankan tanpa perlu mengelola infrastruktur server secara eksplisit. Tujuannya adalah agar developer lebih fokus pada pengembangan dan implementasi fungsional kode tanpa memikirkan tentang infrastruktur di baliknya.

Umumnya pengguna hanya membayar berdasarkan penggunaan yang sebenarnya, bukan kapasitas server yang disediakan. Contohnya, menggunakan layanan FaaS (Function as a Service) seperti AWS Lambda atau Google Cloud Functions untuk mengeksekusi kode.

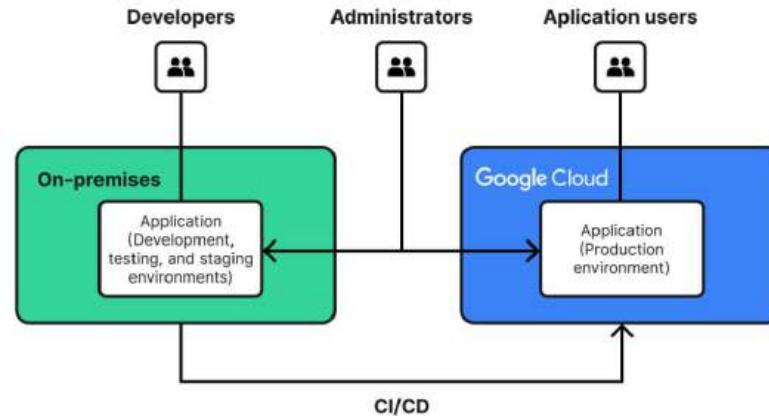


# Jenis-jenis Deployment

## 5. Hybrid Deployment

Kombinasi dari dua atau lebih jenis deployment dapat disebut juga sebagai hybrid deployment. Memungkinkan organisasi untuk memanfaatkan kelebihan dari setiap jenis deployment sesuai dengan kebutuhan mereka.

Misalnya, beberapa aplikasi dapat di-deploy di lingkungan cloud, sementara yang lain tetap dijalankan on-premises untuk alasan keamanan atau kepatuhan terhadap regulasi.



Selesai. 😊