

**Makalah**  
**DevOps Tools Kubernetes**  
**Metodologi Desain Perangkat Lunak Praktik XIV**



Disusun Oleh:

Kelompok 2

5200411401 Maoren Ganesta F.K.B

5200411420 Teofilus Figo A N

5200411422 Agya Rahmani Divasasri

5200411455 Aldico Yogaswara

PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
2020/2021

## Daftar Isi

BAB I PENDAHULUAN .....	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	1
C. Tujuan Penelitian .....	1
Bab II .....	2
Pembahasan.....	2
A. Pengertian.....	2
C. Cara Kerja Kubernetes .....	2
D. Keunggulan Kubernetes.....	3
E. Kekurangan Kubernetes.....	4
BAB III CONTOH PENGGUNAAN TOOLS .....	5
A. Pengembangan Infrastruktur Analisis Data Heart Rate berbasis Microservices menggunakan Kubernetes.....	5
B. Alur Sistem Microservices .....	7
BAB IV PENUTUP .....	9
A. Kekurangan dan Kelebihan Metode Waterfall .....	9
B. Kekurangan dan Kelebihan Metode <i>Prototype</i> .....	10
C. Kekurangan dan Kelebihan Metode RAD.....	10
D. Kekurangan dan Kelebihan Metode Agile.....	11
E. Kelebihan dan Kekurangan DevOps.....	12

# BAB I

## PENDAHULUAN

### A. Latar Belakang

DevOps merupakan singkatan dari dua kata yaitu Development dan Operation. Di mana kedua kata tersebut bermakna menggabungkan proses development/pengembangan dari sebuah sistem/aplikasi dengan operation/operasional. Seperti yang disebutkan sebelumnya, DevOps adalah sebuah prinsip developer untuk mengkoordinasikan antar tim yaitu tim development dengan tim operations dengan efektif dan efisien. Pola pikir yang dibentuk oleh DevOps adalah koordinasi antar tim yang dapat dilakukan dengan cara singkat sehingga tidak membutuhkan banyak pertanyaan. Tim operation atau development cukup mengonfigurasi beberapa komponen yang dibutuhkan melalui prosedur yang dibuat.

Tentunya koordinasi yang diterapkan pada DevOps membutuhkan sebuah tools. Banyak tools yang bisa kamu gunakan, salah satunya adalah *Source Code Management* (SCM) yang biasa digunakan secara umum oleh tim development. Produk SCM yang paling terkenal adalah Git, ditemani oleh *Source Code Repository* (SCR) seperti GitHub, GitLab, Bitbucket, atau yang lainnya. Namun SCM saja tak cukup untuk mengomunikasikan antara tim development dengan tim operational.

Agar tim operational dapat mengetahui permasalahan yang terjadi, biasanya akan dihubungkan ke Product Management Software, seperti Jira. Melalui Product Management Software, tim *operation* dapat mengetahui berbagai permasalahan yang terjadi pada sistem/aplikasi. Sehingga antara pihak development dengan operational akan saling terhubung satu sama lain. Pada DevOps sendiri, banyak tools yang bisa dipilih seperti Maver, Kubernetes, Splunk, Github, Nagios, Jenkins, Docker, Selenium, Bamboo, Ansible, Gradle.

Pada Makalah berikut ini, akan ada penjabaran lebih jauh seputar salah satu tools yaitu Kubernetes. Mari kita berkenalan lebih dekat dengan DevOps tools Kubernetes agar anda paham konsep dan tujuan dari salah satu tools DevOps yang paling sering digunakan.

### B. Rumusan Masalah

1. Apa itu Kubernetes?
2. Apa saja komponen Kubernetes?
3. Bagaimana cara kerja Kubernetes?
4. Apa saja keunggulan Kubernetes?
5. Apa saja kekurangan Kubernetes

### C. Tujuan Penelitian

1. Mengetahui Kubernetes
2. Mengetahui komponen Kubernetes
3. Mengetahui cara kerja Kubernetes
4. Mengetahui keunggulan Kubernetes
5. Mengetahui kekurangan Kubernetes

# Bab II

## Pembahasan

### A. Pengertian

Kubernetes merupakan platform open-source yang digunakan untuk melakukan manajemen workloads aplikasi yang dikontainerisasi, serta menyediakan konfigurasi dan otomatisasi. Kubernetes berada di dalam ekosistem yang besar dan berkembang cepat. Service, support, dan tools Kubernetes tersedia secara meluas. Kubernetes pertama kali dikembangkan oleh Google namun kini dikelola oleh Cloud Native Computing Foundation sebagai platform manajemen kontainer yang cukup populer. Kubernetes memiliki kemampuan untuk melakukan penjadwalan aplikasi, load balancing server dan peningkatan kapasitas kontainer secara otomatis. Tak heran, Kubernetes kini banyak digunakan untuk membangun microservices, yaitu aplikasi kecil yang menjadi pengembangan dari aplikasi besar dan saling terhubung satu sama lain. Beberapa perusahaan yang menggunakan microservices pada produk mereka di antaranya adalah Netflix, Amazon dan Apple. Inilah yang membuat Kubernetes menjadi semakin populer berkat kebutuhan penggunaan pada aplikasi modern

### B. Komponen Kubernetes

#### a. Cluster

Cluster adalah suatu kelompok berisi server fisik atau VPS untuk menjalankan Kubernetes. Ada dua jenis server yang dibutuhkan, yaitu master node dan worker node.

#### b. Object

Di dalam sebuah cluster, terdapat berbagai object, yaitu entitas yang merepresentasikan kondisi dari suatu cluster. Ada berbagai object yang ada pada sebuah cluster Kubernetes, yaitu:

- 1) Pod  
Pod merupakan objek terkecil di dalam cluster kubernetes yang terletak di dalam node. Fungsinya untuk menjalankan docker images yang membentuk sebuah kontainer.
- 2) Service  
Service adalah objek yang digunakan untuk mengarahkan request atau traffic ke beberapa Pod menggunakan IP address. Tujuannya agar Pod bisa diakses dari luar.
- 3) Volume  
Volume adalah objek yang berfungsi untuk penyimpanan data suatu kontainer. Letaknya di luar kontainer.
- 4) Namespace  
Namespace adalah objek untuk memisahkan resource atau environment cluster. Dengan namespace, pengguna dapat memisahkan tiap cluster project supaya tidak saling terganggu satu sama lain.

### C. Cara Kerja Kubernetes

Kubernetes dapat dianalogikan sebagai sebuah pertunjukan orkestra. Di dalam orkestra ada orang yang memainkan alat musik masing masing, pemain musik ini dapat diibaratkan sebagai kontainer docker yang bekerja di dalam cluster. Kemudian ada 1 peran lagi dalam orkestra yaitu seorang dirigen yang mengatur para pemain musik agar memainkan alat musik sesuai instruksi yang diberikan. Maka dirigen ini dapat diibaratkan sebagai Kubernetes yang mengatur aplikasi aplikasi agar berjalan sesuai perintah. Layaknya seorang dirigen musik, Kubernetes juga mengatur aspek dari aplikasi. Mulai dari mengelola workloads dari semua kontainer seperti kapan gitar dimainkan atau

drum mulai dipukul, hingga menambah skalabilitas kontainer seperti menentukan berapa pemain terompet yang membunyikan nada. Di dalam cluster terdapat otak inti Kubernetes yang bernama control plane. Fungsinya untuk mengekspos API Server yang menangani request internal dan eksternal. Semua request untuk menjalankan cluster disimpan di dalam database. Layaknya otak dari si dirigen yang merancang sistem orkestra, control plane Kubernetes mengelola worker nodes yang berisi kumpulan node atau server. Setiap node memiliki kubelet, yaitu aplikasi yang berjalan untuk berkomunikasi dengan control plane. Node yang dikelola oleh worker node ini berisi banyak kontainer atau musisi. Kita bisa membayangkan node seperti kelompok musisi perkusi di satu sisi dan kelompok musisi terompet di bagian lain. Di dalam kontainer itulah terdapat aplikasi-aplikasi yang kita jalankan ibarat alat musik yang dimainkan oleh para musisi di kelompok orkestra tersebut.

## **D. Keunggulan Kubernetes**

### **1) Service Discovery dan Load Balancing**

Fitur service discovery memudahkan untuk melacak kontainer dengan otomatis. Hal ini tentu sangat penting di dalam mengembangkan aplikasi microservices. Kubernetes dapat mengenali sebuah service berdasar DNS atau IP address server tersebut. Tak hanya itu, berkat Load Balancing pengelolaan trafik menjadi lebih mudah. Saat terjadi trafik yang cukup besar, Kubernetes mampu membagi beban secara merata sehingga membuat aplikasi menjadi lebih stabil.

### **2) Storage Orchestration**

Kubernetes memungkinkan pengguna melakukan mount pada media penyimpanan (storage) pilihan seperti, storage lokal atau yang berbasis cloud seperti AWS dan lainnya.

### **3) Automated Rollouts and Rollbacks**

Fitur automated rollouts dan rollback sangat penting untuk membantu pengguna terkait deployment. Pada Kubernetes, deployment bisa menggunakan file YAML yang di dalamnya nanti akan berisi ReplicaSet. Apabila terjadi kendala pada deployment, dan pengguna masih memiliki ReplicaSet dari versi sebelumnya, pengguna dapat menggunakannya untuk melakukan rollback. Hal tersebut juga bisa digunakan ketika pengguna akan melakukan deployment lainnya.

### **4) Automatic Bin Packing**

Menggunakan fitur Automatic Bin Packing, pengguna bisa mengatur kapasitas CPU dan sumber daya tiap kontainer secara spesifik. Jadi saat limit kapasitas sudah ditentukan, aplikasi terhindar dari berebut sumber daya. Selain itu, sumber daya jadi lebih hemat.

### **5) Self Healing**

Fitur self-healing sangat penting bagi user yang ingin terus menjalankan aplikasi selama 24 jam. Nah, Kubernetes memiliki kemampuan untuk memeriksa kontainer yang ada, apakah dalam keadaan running (berjalan) atau mengalami error. Jika terjadi kendala pada kontainer, Kubernetes akan menghentikan proses yang berjalan dan memberikan opsi untuk merestart atau mengganti secara otomatis kontainer yang error tersebut.

### **6) Secret and Configuration Management**

Kubernetes memungkinkan pengguna menyimpan data-data sensitif seperti password, auth token hingga SSH keys ke dalam Kubernetes Secret. Hal ini tentu jauh lebih aman dibanding menyimpannya di dalam container image. Secret bisa dibuat oleh sistem atau oleh user, yaitu anda sendiri. Karena secret sendiri secara default disimpan tanpa terenkripsi, pengguna bisa melakukan konfigurasi lebih lanjut sesuai dengan kebutuhan.

## **E. Kekurangan Kubernetes**

1. Penyeimbang beban di Kubernetes tidak otomatis
2. Kubernetes rumit
3. kubernetes tidak memungkinkan untuk digunakan perangkat lunak pihak ketiga untuk logging dan pemantauan
4. Tidak kompatibel dengan alat Docker yang ada.
5. Proses instalasi dan konfigurasi yang kompleks.
6. Migrasi yang lebih kompleks.

## BAB III

### CONTOH PENGGUNAAN TOOLS

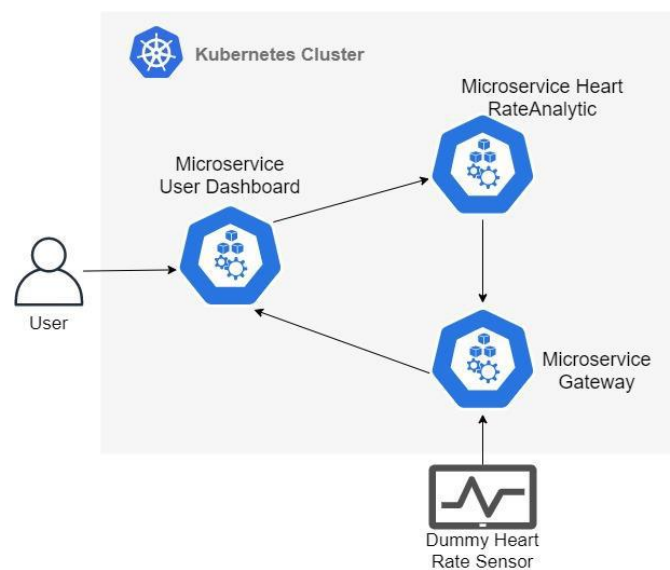
#### A. Pengembangan Infrastruktur Analisis Data Heart Rate berbasis Microservices menggunakan Kubernetes

Kubernetes merupakan container orchestration atau opensource platform untuk melakukan manajemen terhadap containerized services (Smarvin, 2019). Kubernetes memiliki beberapa object yang digunakan pada penelitian ini yaitu pod, deployment, service, statefulset, persistentvolume, dan persistemvolumeclaim. Pod digunakan untuk menjalankan aplikasi microservices. Namun, pod memiliki siklus hidup yang dapat menyebabkan pod mati sehingga aplikasi yang berjalan di dalam pod tidak dapat diakses serta data yang tersimpan di dalam pod juga akan ikut terhapus.

Semua pod harus dapat diakses dari dalam atau luar cluster kubernetes sehingga digunakan object service dengan tipe ClusterIP dan NodePort untuk mengekspos pod sebagai suatu layanan jaringan. Service tipe ClusterIP memungkinkan pod diakses dari di dalam cluster kubernetes, sedangkan NodePort memungkinkan pod diakses dari luar cluster kubernetes.

##### Perancangan Arsitektur

Secara keseluruhan arsitektur sistem yang dikembangkan dapat dilihat pada gambar:

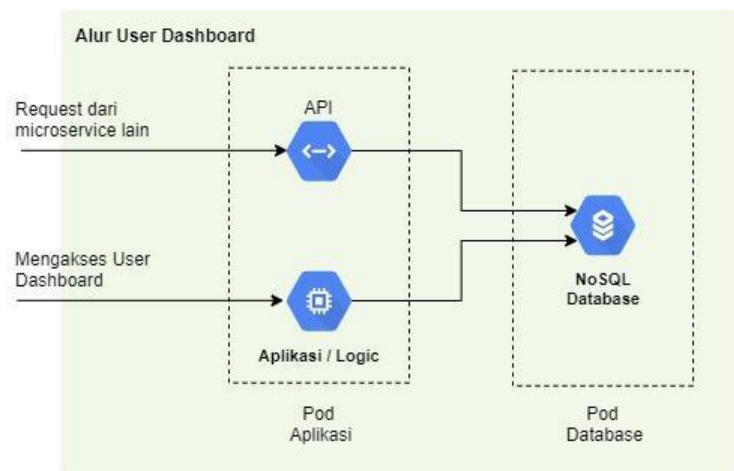


*Gambar 1 - Arsitektur Sistem*

Keterangan dari gambar:

1. Terdapat tiga microservices yaitu user dashboard, heart rate analytic, dan gateway. Setiap microservices dideploy ke pod yang berbeda.
2. Pengguna mengakses microservice user dashboard untuk mengakses keseluruhan sistem melalui web.
3. Sensor mengirimkan data ke microservice gateway.
4. Setiap microservices dapat saling berkomunikasi melalui API, untuk penjelasan lebih lengkap mengenai alur komunikasi dapat dilihat pada bagian 4.2 Perancangan Alur Sistem

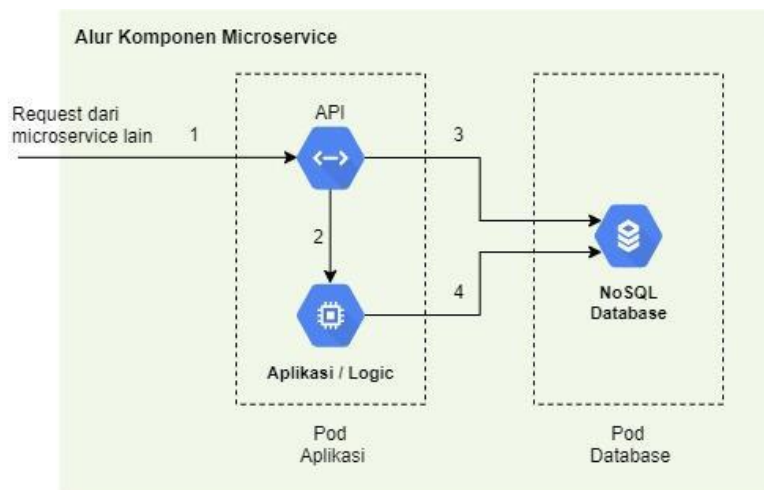
Alur sistem pada komponen microservice heart rate analytic dan gateway adalah sama sehingga dijelaskan ke dalam satu bagian, sedangkan alur sistem pada komponen microservice user dashboard berbeda dengan komponen microservices lainnya.



Gambar 2 - Alur Sistem Komponen Microservices User Dashboard

dapat dilihat pengguna dapat mengakses langsung komponen aplikasi melalui user interface berupa aplikasi web, hal ini yang membedakan alur komunikasi komponen microservices user dashboard dan lainnya. Alur komunikasi dibagi menjadi dua. Pertama, pengguna mengakses komponen aplikasi melalui web untuk menjalankan fungsionalitas yang dimiliki oleh microservices user dashboard. Komponen aplikasi mengakses database untuk mendapatkan informasi yang dibutuhkan.

Alur komunikasi kedua, microservices lain mengakses API dari microservice user dashboard kemudian API melakukan query ke database untuk mendapatkan data yang dibutuhkan kemudian dikembalikan oleh API berupa response.



Gambar 3 - Alur Sistem Komponen Microservices HRA & Gateway

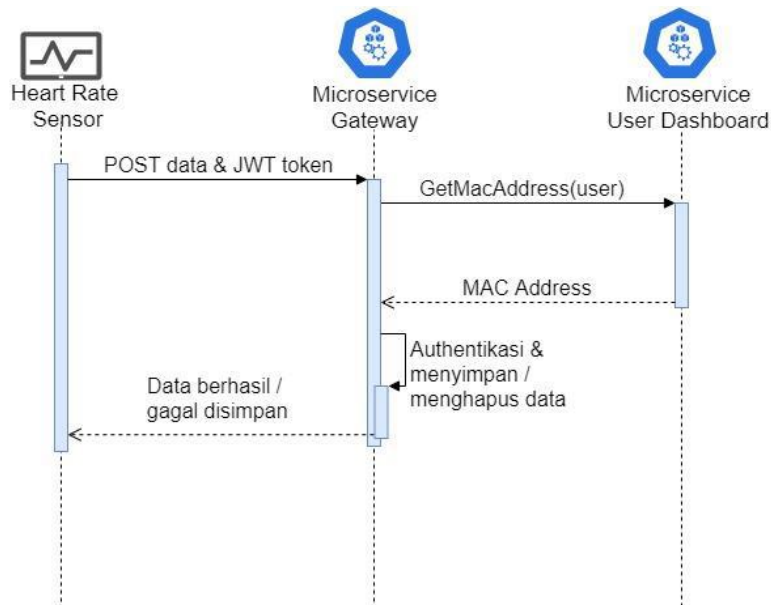
Komponen microservices heart rate analytic dan gateway memiliki alur sistem yang sama. Pada gambar 5 diberikan penomoran untuk mempermudah peneliti menjelaskan alur sistem. Pertama alur sistem dengan urutan 1-2-4, request masuk ke komponen API kemudian menjalankan proses komputasi yang terdapat pada komponen aplikasi, kemudian komponen aplikasi melakukan query ke database untuk mendapatkan data yang diperlukan untuk melakukan



proses komputasi atau untuk menyimpan hasil komputasi. Alur komunikasi yang kedua 1-3, request masuk ke komponen API kemudian melakukan query ke database dan data hasil query dikembalikan dalam bentuk response.

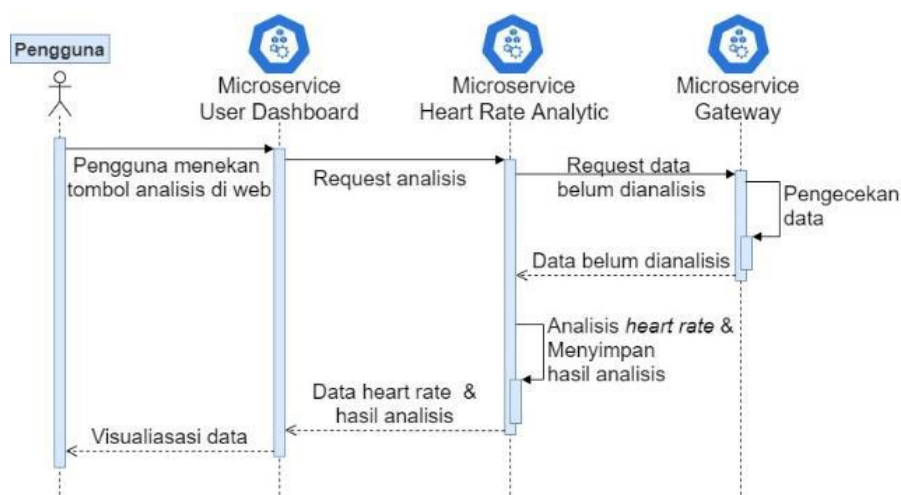
## B. Alur Sistem Microservices

Perancangan alur sistem microservices merupakan rancangan alur komunikasi antara microservices yang dikembangkan pada penelitian ini. Secara umum ada tiga alur komunikasi utama yang dibangun pada penelitian ini yaitu alur pengiriman data ke gateway, alur analisis data dan alur visualisasi data.



Gambar 4 - Sequence Diagram Alur Pengiriman Data Ke Gateway

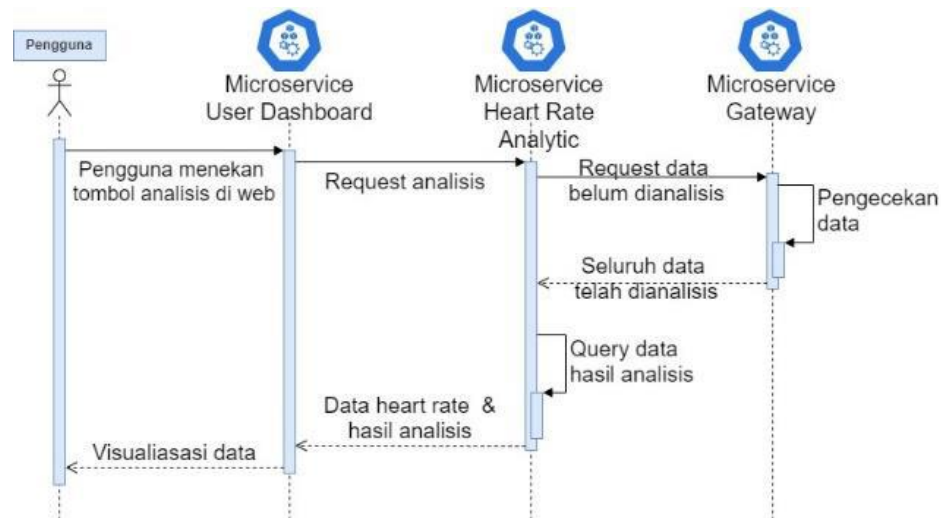
Alur komunikasi pengiriman data ke gateway. Pada alur komunikasi pengiriman data terdapat mekanisme autentikasi untuk memastikan device yang mengirim data mempunyai hak untuk melakukan penyimpanan data.



Gambar 5 - Sequence Diagram Alur Analisis Data

Pada alur komunikasi analisis data dilakukan pengecekan dan query data yang belum dianalisis pada microservices gateway. Setelah itu dilakukan analisis data heart rate pada microservice

heart rate analytic. Setelah analisis selesai dilakukan penyimpanan dan mengembalikan hasil analisis ke microservice user dashboard.



Gambar 6 - Sequence Diagram Alur Visualisasi Data

Pada alur komunikasi visualisasi data dilakukan pengecekan data pada microservice gateway. Jika seluruh data telah dianalisis maka microservices heart rate analytic akan melakukan query data hasil analisis dan mengembalikan ke microservice user dashboard.

Jurnal :

Jahiduddin, A., Pramukantoro, E., & Bachtiar, F. Pengembangan Infrastruktur Analisis Data Heart Rate berbasis Microservices menggunakan Kubernetes. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 4, no. 1, p. 100-108, feb. 2020. ISSN 2548-964X. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6870>

## **BAB IV**

### **PENUTUP**

1. *Waterfall* merupakan metode pengembangan perangkat lunak tradisional yang sistematis. Metode ini memiliki lima tahapan proses: di antaranya *Communication*, *Planning*, *Modeling*, *Construction*, dan *Deployment*. *Communication* merupakan fase di mana pelanggan atau pemilik proyek menyampaikan kebutuhan dan permasalahannya kepada pengembang. Lalu, bersama-sama mengumpulkan data-data yang diperlukan dan merumuskan fitur-fitur perangkat lunak. Selanjutnya, menginjak pada proses perancangan. Dimulai dengan merumuskan estimasi kerja, kebutuhan sumber daya, serta perencanaan alur kerja. Berlanjut dengan tahap perancangan struktur data, arsitektur, tampilan, dan algoritma perangkat lunak. Rancangan kemudian coba diaplikasikan pada perangkat keras komputer dalam bentuk bahasa pemrograman. *Construction* juga mencakup tahapan uji coba pengoperasian perangkat lunak untuk mengetahui kelemahannya. Setelah berhasil dibuat, perangkat lunak disebarluaskan untuk diimplementasikan pada perangkat pengguna secara umum. Temuan-temuan dari pengguna, akan menjadi bahan bagi pengembang untuk mengevaluasi dan memperbaiki perangkat lunak lebih jauh lagi.

#### **A. Kekurangan dan Kelebihan Metode Waterfall**

Metode *waterfall* melibatkan berbagai proses yang sistematis dan komprehensif. Sumber daya dan tahapan pengerjaannya dikumpulkan secara lengkap sehingga dapat mencapai hasil maksimal. Sayangnya, proses tersebut memakan waktu lebih lama.

Sistem dalam metode *waterfall* merupakan proses yang baku, sehingga pengembang sulit melakukan improvisasi. Itulah mengapa metode ini dianggap kurang efektif dan seringkali hanya dipakai dalam pengembangan perangkat lunak atau sistem berskala besar.

2. *Prototype* dalam bahasa Indonesia diartikan dengan istilah purwarupa. Istilah tersebut berarti model awal atau rancangan sementara yang masih membutuhkan berbagai penyesuaian sebelum dinyatakan telah memenuhi hasil yang diinginkan. Terdapat lima tahapan dalam pengembangan perangkat lunak menggunakan metode *prototype*. Dimulai dengan pengumpulan informasi dari pelanggan ke tim pengembang. Lalu, tim akan merencanakan sistem dan mengerjakan purwarupa perangkat lunak. Hasilnya kemudian diserahkan kepada pelanggan untuk dievaluasi. Jika terdapat permasalahan, tim akan merevisi sistem tersebut hingga benar-benar sesuai dengan kebutuhan pelanggan. Ketika revisi selesai dan telah diterima, berarti perangkat lunak telah siap diterjemahkan ke dalam perangkat keras. Kemudian, dilanjutkan dengan proses uji coba dan berbagai revisi sebelum mulai dapat digunakan.

## **B. Kekurangan dan Kelebihan Metode *Prototype***

Metode *prototype* digunakan apabila pemilik proyek tahu benar apa yang diinginkannya, tetapi tidak mengetahui bagaimana cara mengaplikasikannya. Kuncinya terletak pada komunikasi yang baik antara pelanggan dan pengembang. Pelanggan harus menyampaikan kebutuhannya secara jelas. Sementara pengembang juga diharapkan mampu menerjemahkan informasi tersebut agar dapat menghasilkan perangkat lunak yang sesuai. Jika tidak, *prototype* ini tidak akan menjadi metode yang efektif.

Pengembang memperoleh tantangan besar dari pelanggan. Sebaliknya, pelanggan dapat terpuaskan jika pengembang berhasil memenuhi kebutuhannya. Kerja sama kedua pihak akan saling menguntungkan.

3. RAD merupakan singkatan dari *Rapid Application Development*. Metode ini juga menggunakan pendekatan iteratif dan inkremental, tetapi lebih menekankan pada tenggat waktu dan efisiensi biaya yang sesuai dengan kebutuhan. Proses pengembangan dengan Metode RAD dianggap lebih singkat. Pasalnya, semua pihak, baik pelanggan maupun pengembang, terus terlibat secara aktif dalam setiap proses hingga hasil dapat tercapai. Di samping itu, tahapan kerja pada metode ini juga lebih sedikit.

Alur kerja hanya dibagi menjadi tiga tahap yang semuanya padat. Identifikasi tujuan yang langsung diiringi dengan komunikasi dan perancangan, di mana seluruh pihak terlibat aktif dalam setiap perumusannya. Proses ini menjadi tahap awal dari Metode RAD.

Tahap kedua masih melibatkan semua pihak, yaitu proses mendesain sistem atau perangkat lunak sesuai kebutuhan. Pelanggan atau pengguna ikut terjun dalam menguji coba perangkat lunak. Perbaikan pun langsung diterapkan jika pengguna menemukan kesalahan.

Ketika pengguna terpuaskan dengan desain perangkat lunak, setelah melalui berbagai perbaikan, barulah proses kerja menginjak pada tahap terakhir, yaitu implementasi. Desain perangkat lunak mulai diterjemahkan dalam bahasa mesin dan bisa digunakan.

## **C. Kekurangan dan Kelebihan Metode RAD**

Beberapa kekurangan dari metode RAD, antara lain dilihat dari segi konsistensi dan kemampuan personel. Metode ini membutuhkan pengembang ahli, sekaligus kerjasama yang aktif dan konsisten antara pemilik proyek beserta semua tim. Tanpa kedua hal tersebut, mustahil menerapkan metode RAD dalam pengembangan perangkat lunak, apalagi yang berskala besar. Namun jika kedua hal itu terakomodasi dengan baik, metode RAD adalah cara paling efektif untuk menghemat waktu dan biaya.

4. Pembahasan tentang *scrum* telah sedikit menyinggung metode *agile*. Metode *agile* merupakan induk dari *scrum*. Jika *scrum* adalah kerangka kerja, *agile* adalah pelaksanaan proyek secara keseluruhan yang berskala besar. Metode ini tergolong modern, karena menekankan pada improvisasi dan adaptasi. Meskipun begitu, alur kerjanya masih menerapkan pola tradisional yang sistematis. Dimulai dari perencanaan, analisis kebutuhan, perancangan, uji coba, implementasi, dan pemeliharaan. Salah satu kunci dari *agile* adalah dokumentasi pekerjaan. Dokumentasi harus tersusun rapi dan terstruktur. Hal ini berkaitan erat dengan metode *agile* yang bersifat adaptif terhadap perubahan fenomena pengguna perangkat lunak. Pengembang membutuhkan data -data penting dari pekerjaan yang telah lalu untuk melakukan perbaikan secara terus menerus. Dokumentasi yang baik akan mempersingkat waktu yang dibutuhkan pengembang dalam memperbaiki sistem atau perangkat lunak.

#### D. Kekurangan dan Kelebihan Metode Agile

Jika berorientasi pada kepuasan pengguna, metode *agile* menyediakan prosedur yang tepat. Demikian halnya pada proyek besar yang membutuhkan pemeliharaan jangka panjang. Metode *agile* menguntungkan dari segi efisiensi waktu dan tenaga. Keunggulan metode *agile* juga terletak pada adaptasi dan kebebasan bagi pengembang untuk melakukan improvisasi. Tidak ada batasan baku bagi pengembang ketika ingin menerapkan ide-ide barunya dalam menjawab tantangan zaman. Namun, bukan berarti metode ini tidak memiliki kelemahan. Tantangan terbesar akan dihadapi pengembang yang terus dipacu untuk beradaptasi dengan perubahan zaman. Tim pengembangan pun tidak boleh asal pilih, harus solid dan sama-sama berkomitmen kuat.

5. DevOps merupakan singkatan dari dua kata yaitu *Development* dan *Operation*. Di mana kedua kata tersebut bermakna “operasional pengembang”. Seperti yang disebutkan sebelumnya, DevOps adalah sebuah prinsip developer untuk mengkoordinasikan antar tim yaitu tim *development* dengan tim *operations* dengan efektif dan efisien.

DevOps merupakan singkatan dari dua kata yaitu *Development* dan *Operation*. Di mana kedua kata tersebut bermakna menggabungkan proses *development*/pengembangan dari sebuah sistem/aplikasi dengan *operation*/operasional. Seperti yang disebutkan sebelumnya, DevOps adalah sebuah prinsip developer untuk mengkoordinasikan antar tim yaitu tim *development* dengan tim *operations* dengan efektif dan efisien.

Pola pikir yang dibentuk oleh DevOps adalah koordinasi antar tim yang dapat dilakukan dengan cara singkat sehingga tidak membutuhkan banyak pertanyaan.

Tim *operation* atau *development* cukup mengonfigurasi beberapa komponen yang dibutuhkan melalui prosedur yang dibuat.

Tentunya koordinasi yang diterapkan pada DevOps membutuhkan sebuah tools.

Banyak tools yang bisa kamu gunakan, salah satunya adalah *Source Code Management* (SCM) yang biasa digunakan secara umum oleh tim *development*.

Produk SCM yang paling terkenal adalah Git, ditemani oleh *Source Code Repository* (SCR) seperti GitHub, GitLab, Bitbucket, atau yang lainnya. Namun SCM saja tak cukup untuk mengomunikasikan antara tim *development* dengan tim *operational*.

Agar tim *operational* dapat mengetahui permasalahan yang terjadi, biasanya akan dihubungkan ke Product Management Software, seperti Jira. Melalui Product Management Software, tim *operation* dapat mengetahui berbagai permasalahan yang terjadi pada sistem/aplikasi. Sehingga antara pihak *development* dengan *operational* akan saling terhubung satu sama lain.

## **E. Kelebihan dan Kekurangan DevOps**

Penerapan metode DevOps memiliki banyak sekali kelebihan, akan tetapi terdapat juga kekurangan juga dari penerapan metode ini. Berikut adalah kelebihan dan kekurangan dari penerapan metode DevOps.

Kelebihan:

1. Siklus pengembangan yang lebih pendek.
2. Meningkatkan kualitas dan fleksibilitas.
3. Biaya yang lebih efisien.
4. Kontrol Resiko dan Pemulihan yang lebih baik.
5. Dapat meningkatkan Praktik Keamanan.

Kekurangan:

1. Penerapan Metode DevOps harus merubah kebiasaan atau kebudayaan dalam pengembangan proyek.
2. Membutuhkan Software Engineering yang ahli.
3. Memerlukan kolaborasi yang kuat

## **DAFTAR PUSTAKA**

<https://www.dicoding.com/blog/apa-itu-devops/>

<https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6870/3323>