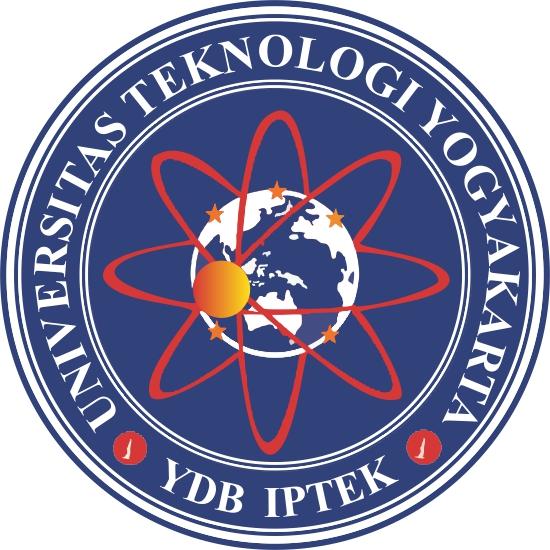
**Metodologi Desain Perangkat Lunak Praktik**

**LAPORAN DevOps AWS**

****

Disusun Oleh :

**Kelompok 6**

5200411045 – Bayu Octabryan Nugraha

5200411365 - Galyleo Ilham Setyaka

5200411275 – Abdul Haris As'ari

5200411502 – Angger anoraga S.M

# **BAB 1**

**LATAR BELAKANG**

Di era sekarang teknologi berkembang sangat pesat, tidak terkecuali di bidang deployment dan updating aplikasi, Banyaknya tools untuk mendeploy dan mengupdate aplikasi, sehingga menyulitkan Developer Operation (DevOps) untuk menentukan tools yang akan digunakan.

DevOps adalah kombinasi dari filosofi budaya, penerapan, dan peralatan yang meningkatkan kemampuan organisasi untuk memberikan aplikasi dan layanan dalam kecepatan yang tinggi: mengembangkan dan meningkatkan produk dalam waktu yang lebih cepat dari organisasi yang menggunakan pengembangan perangkat lunak dan proses pengelolaan infrastruktur tradisional. Kecepatan ini memungkinkan organisasi untuk melayani pelanggan mereka dengan lebih baik dan berkompetisi dengan lebih efektif di pasaran.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.

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 GitHu, GitLab, Bitbucket, atau yang lainnya. Namun SCM saja tak cukup untuk mengkomunikasikan antara tim development dengan tim operational.

Salah satu best practices dalam DevOps yaitu melakukan *update* produk yang kecil dan cepat. Dalam industri IT, ini bisa disebut dengan *minor* / *patch* *update*. Dengan melakukan *update* produk yang kecil, organisasi/perusahaan dapat melakukan inovasi lebih cepat bagi produknya, juga dapat melakukan *rollback* dengan mudah jika ditemukan *bug* yang tidak tercover oleh *automated testing* maupun tim QA. Dengan praktek ini, Organisasi/Perusahaan dapat memonitor produk dan mendapatkan data yang paling berharga yaitu data pengguna.

**Rumusan Masalah**

Rumusan masalah dari latar belakang diatas adalah bagaimana application tools yang dilakukan pada layanan Amazon Web Services (AWS).

**Tujuan**

Tujuan dari Rumusan masalah diatas adalah untuk mengetahui cara kerja dari tools Amazon Web services.

# **BAB 2**

Pembahasan

1. **Pengertian**

Amazon Web Services (AWS) adalah platform cloud paling komprehensif dan digunakan secara luas di dunia, menawarkan lebih dari 200 layanan unggulan yang lengkap dari pusat data secara global. Jutaan pelanggan—termasuk beberapa startup dengan pertumbuhan tercepat, perusahaan terbesar, dan lembaga pemerintah terkemuka—menggunakan AWS untuk memangkas biaya, menjadi lebih sigap, dan berinovasi lebih cepat

1. **Fungsionalitas**

AWS juga memiliki fungsionalitas terdalam dalam semua layananannya. Misalnya, AWS menawarkan ragam database paling luas yang dibangun untuk tujuan beragam jenis aplikasi sehingga Anda dapat memilih alat yang tepat untuk pekerjaan tersebut untuk mendapatkan biaya dan kinerja terbaik

# **BAB 3**

## Review Jurnal

|  |  |
| --- | --- |
| Judul | IMPLEMENTASI KOMPUTASI AWAN MENGGUNAKAN TEKNOLOGI GOOGLE APP ENGINE (GAE) DAN AMAZON WEB SERVICES (AWS) |
| Jurnal | IMPLEMENTASI AWS |
| Link Jurnal | https://ejournal.unsrat.ac.id/index.php/informatika/article/view/542/435 |
| Volume dan Halaman | Vol I,Hal 1-13 |
| Tahun | 2012 |
| Penulis | 1. Adi Nugroho, ST, MMSI  2. Dr Techn Khabib Mustofa, SSi, MKom |
| Reviewer | Abdul Haris As’ari |
| Tanggal | 10 Januari 2022 |
| Tujuan Penelitian | Penelitian ini mencoba untuk merancang pengembangan aplikasi agar pengguna tidak perlu lagi memikirkan infrastruktur yang biasanya memerlukan investasi dalam bentuk dana yang sangat besar. |
| Metedologi Penelitian | Metodologi penelitian meliputi :  1. Pendahuluan : Komputasi awan (Clou Computing)  2. Implentasi Komputasi Awan Menggunakan Teknologi Google APP Engine (GAE)  3. Implentasi Komputasi Awan Menggunakan Awan Teknologi Amazon Web Service (AWS)  4. Analisis Tentang Implentasi Komputasi Awan Menggunakan Teknologi Google APP Engine (GAE) Dan Teknologi Amazon Web Service (AWS)  6. Kesimpulan |
| Hasil Penelitian | 1. Pada Komputasi awan (Clou Computing) merupakan komputasi masa depan. Dengan difasilitasi oleh jaringan komputer global saat ini (Internet)Komputasi awan membuat investasi di bidang Teknologi Informasi akan menjadi semakin ‘murah’, karena layanan-layanan (services) tertentu bisa diperoleh dari vendor yang menyediakannya (SaaS-Software as a Service)  2. Pada Implentasi Komputasi Awan Menggunakan Teknologi Google APP Engine (GAE) Google adalah pusat data yang ada di luar komputer aplikasi klien. Arsitektur Aplikasi Google App Engine Secara Umum yaitu: Apps Engine Request, DataStore, BigTable, URL Fetch Service.  3. Pada Implentasi Komputasi Awan Menggunakan Awan Teknologi Amazon Web Service (AWS). AWS memiliki komponen-komponen sebagai berikut:   * Amazon S3 (Simple Storage Service), * Amazon Cloud Front, * Amazon SQS (Simple Queue Service), * Amazon SimpleDB, * Amazon RDS (Relational Database Service), * Amazon EC2 (Elastic Compute Cloud)   4. Pada Analisis Tentang Implentasi Komputasi Awan Menggunakan Teknologi Google APP Engine (GAE) Dan Teknologi Amazon Web Service (AWS). Baik Google maupun Amazon memiliki banyak kesamaan seperti menawarkan penggunaan fasilitas komputasi awan kepada pengguna berdasarkan waktu kerja CPU, bandwith, serta berdasarkan ruang penyimpanan yang diperlukan oleh aplikasi pengguna. Dari sudutpandang kemudahan pengembangan aplikasi (termasuk melakukan deploy aplikasi) yang memakan waktu sekitar 30-35% dari keseluruhan waktu yang diperlukan untuk menyelesaikan siklus pengembangan sistem (SDLC-System Development Life Cycle), aplikasi Google App Engine (GAE) relatif lebih mudah dikembangkan daripada pengembangan aplikasi Amazon Web Service (AWS) |

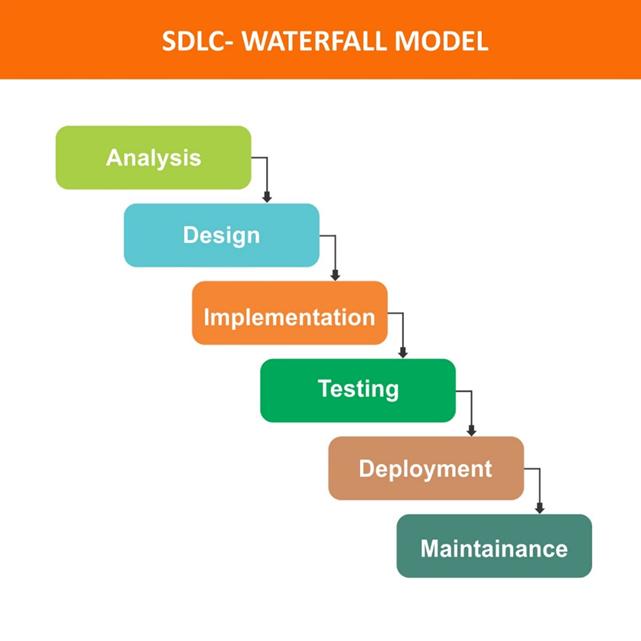
# BAB 4

PENUTUP

(PERBANDINGAN ANTARA WATERFALL, PROTOTYPE, RAD, AGIL, DEVOPS)

## **WATERFALL**

Model waterfall adalah model pengembangan perangkat lunak yang paling sering digunakan. Model pengembangan ini bersifat linear dari tahap awal pengembangan system yaitu tahap perencanaan sampai tahap akhir pengembangan system yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya. Menurut [2] “model SDLC air terjun (waterfall) sering disebut juga model sekuensial linier (sequential linear) atau alur hidup klasik (classic life cycle). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukuang (support)”. Berikut adalah gambar model air terjun:

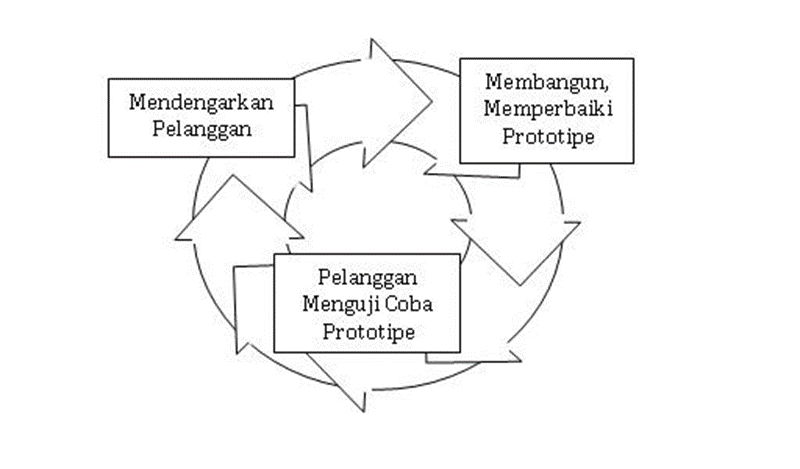


## 

## **PROTOTYPE**

Model prototyping merupakan suatu teknik untuk mengumpulkan informasi tertentu mengenai kebutuhankebutuhan informasi pengguna secara cepat. Berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai. Prototipe tersebut akan dievaluasi oleh pelanggan/pemakai dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. Prototype didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya, dan proses untuk menghasilkan sebuah prototype disebut prototyping[6] Bentuk dari prototype pada gambar 2 merupakan versi awal dari tahapan sebuah sistem software yang digunakan dalam mempresentasikan gambaran dari ide, eksperimen dari sebuah rancangan, mencari sebanyak mungkin masalah yang ada serta penyelesaian terhadap masalah tersebut [3]. Sistem dengan model prototype mengijinkan pengguna agar mengetahui seperti apa tahapan sistem dibuat sehingga sistem mampu beroperasi dengan baik.

Metode prototype digunakan dimaksudkan untuk mendapatkan representasi dari pemodelan aplikasi yang akan dibuat. Rancangan aplikasi awal mulanya berbentuk mockup selanjutnya akan dievaluasi oleh pengguna. Setelah mockup dievaluasi pengguna tahap selanjutnya mockup menjadi bahan rujukan bagi pengembang software untuk merancang aplikasi.



**RAD**

RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkatperangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi.

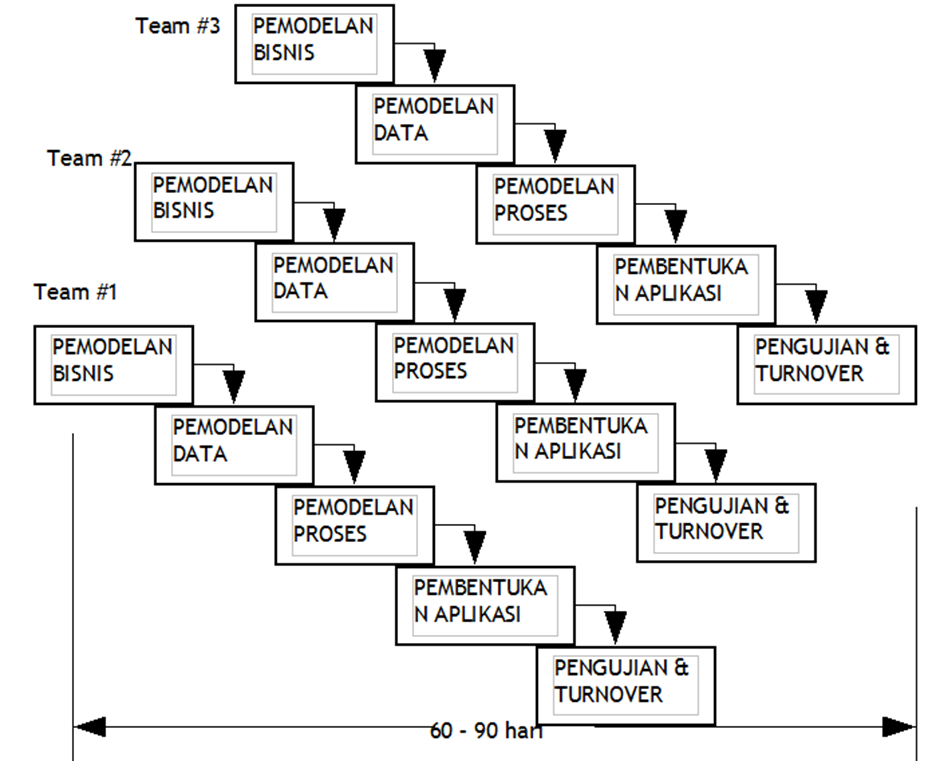
RAD menggunakan metode iteratif (berulang) dalam mengembangkan sistem dimana working model (model bekerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (requirement) pengguna dan selanjutnya disingkirkan.

Model RAD memiliki 3 tahapan sebagai berikut:

1) Rencana Kebutuhan (Requirement Planning): User dan analyst melakukan pertemuan untuk mengidentifikasi tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan.

2) Proses Desain Sistem (Design System): proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara user dan analyst. Keluaran dari tahapan ini adalah spesifikasi software yang meliputi organisasi sistem secara umum, struktur data dan yang lain.

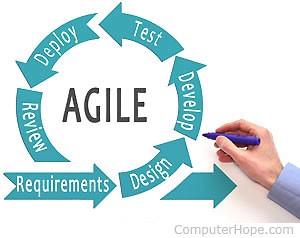
3) Implementasi (Implementation): Tahapan ini adalahan tahapan programmer yang mengembangkan desain suatu program yang telah disetujui oleh user dan analyst. Sebelum diaplikasikan pada suatu organisasi terlebih dahulu dilakukan proses pengujian terhadap program tersebut.



## AGILE

Metode Agile mengutamakan interaksi antar tim pengembang dan review/tanggapan dari client walaupun itu dapat menyebabkan keterlambatan dalam proses pengembangan, tapi dari tim development tetap mengutamakan kecepatan dalam bekerja, itulah mengapa Agile lebih sering saya sebut sebagai semangat bekerja.

Saat belajar tentang Agile, kita akan menemukan proses tools pembantu seperti Scrum, Extreme Programming, dan Adaptive Software Development mereka biasanya dibundle menjadi 1 paket aplikasi, sebagai contoh yaitu Jira Software. Itu adalah contoh proses yang menerjemahkan nilai dan prinsip Agile dalam praktik. Selain itu, ada metode yang populer di dunia Agile seperti user story, story point, burndo



## DEVOPS

Sedangkan untuk DevOps yaitu metodologi pengembangan perangkat lunak yang bertujuan untuk menyatukan tim pengembangan perangkat lunak dan tim teknologi informasi. Ini adalah konsep yang menumbuhkan budaya kolaborasi antara kedua tim yang sebelumnya bekerja secara terpisah, dari tahap desain awal hingga rilis sebuah produk.

DevOps adalah metodologi yang menggabungkan pengembangan perangkat lunak lebih cepat dan dengan efisiensi dan kecepatan yang lebih besar.

