

**PENGEMBANGAN SISTEM JARINGAN WATERFALL,
MAINTENANCE, DAN UPDATE BERKALA MEMAKAI
METODE LAIN SYSTEM ARCHITECTURE**

(Database, ERD, DFD, Infrastructur, Software)



USM

DISUSUN OLEH :

- | | |
|---|------------------------|
| 1. Dominica Ardhinia Sekar W. W. P | (G.231.21.0068) |
| 2. Ulya Ghani | (G.231.21.0103) |
| 3. Eliza Cahyaningrum | (G.231.21.0110) |
| 4. Roudhotul Jannah | (G.231.21.0116) |
| 5. Windy Kurniasari | (G.231.21.0128) |

**PROGAM STUDI S1–TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
UNIIVERSITAS SEMARANG**

2023

Pendahuluan

System Architecture adalah model konseptual yang mendefinisikan struktur, perilaku, dan lebih banyak pandangan dari sebuah sistem. Arsitektur sistem perangkat lunak biasanya mencakup elemen-elemen seperti modul-modul, komponen-komponen, antarmuka, dan koneksi antara elemen tersebut. Arsitektur ini memberikan panduan kepada para pengembang dalam merancang, mengimplementasikan, dan mengintegrasikan komponen-komponen perangkat lunak untuk mencapai tujuan yang diinginkan.

system architecture membantu dalam memahami kompleksitas sistem, memfasilitasi komunikasi antara pemangku kepentingan (stakeholders), memandu pengembangan dan implementasi sistem, serta memastikan bahwa sistem dapat mencapai tujuan bisnis atau teknis yang diinginkan.

Software application architecture adalah sebuah proses untuk mendefinisikan struktur dari suatu aplikasi yang dapat memenuhi seluruh kriteria dari sisi teknis dan juga operasional, dengan pertimbangan kualitas seperti performance, security, and manageability.

Untuk menciptakan suatu jembatan antara kebutuhan bisnis dengan kebutuhan teknis melalui use cases yang sudah dibuat, dan menemukan cara bagaimana untuk mengimplementasikan use cases tersebut. Software architecture yang baik akan meminimalisir kemungkinan resiko/kegagalan yang diciptakan oleh para developer. Selain itu, software architecture yang baik juga dapat beradaptasi dengan perubahan yang akan terjadi seiring waktu dalam teknologi perangkat keras dan perangkat lunak, serta dalam skenario dan persyaratan pengguna.

Arsitektur database adalah desain yang melibatkan perangkat lunak, perangkat keras, dan user interface untuk membangun sebuah sistem penyimpanan dan pengelolaan yang terstruktur. Arsitektur database dirancang untuk memastikan efisiensi, kinerja, dan keandalan dalam mengelola dan menyimpan data.

Arsitektur database merujuk pada struktur dan organisasi sistem basis data. Dalam hal ini mencakup berbagai elemen seperti model data (seperti model relasional atau berorientasi dokumen), sistem manajemen basis data (DBMS) yang digunakan untuk mengelola basis data, struktur fisik penyimpanan data di media penyimpanan, bahasa query untuk mengakses data, keamanan data (termasuk hak akses pengguna dan kontrol integritas), replikasi data (membuat salinan data), teknik backup dan pemulihan data, serta strategi skalabilitas (kemampuan sistem untuk mengatasi peningkatan beban kerja).

Infrastruktur arsitektur perangkat lunak (software infrastructure architecture) adalah rencana dan desain yang merinci cara komponen perangkat lunak yang mendasari, seperti kerangka kerja (frameworks), perpustakaan (libraries), layanan (services), dan komponen teknis lainnya, diatur dan berinteraksi dalam suatu sistem atau lingkungan teknologi.

Entity-Relationship Diagrams (ERD) adalah jenis diagram yang digunakan untuk menggambarkan struktur data dan hubungan antara entitas dalam suatu sistem informasi. Simbol yang digunakan dalam ERD adalah: persegi panjang (mewakili entitas), kotak berlian (mewakili hubungan), garis dan notasi standar (mewakili kardinalitas).

Data Flow Diagrams (DFD) adalah jenis diagram yang digunakan untuk menggambarkan alur data dalam suatu sistem informasi. DFD menjelaskan aliran dan proses input data, output data, dan penyimpanan data. Simbol yang digunakan dalam DFD adalah: persegi panjang (mewakili entitas data), lingkaran (mewakili proses), panah (mewakili aliran data), oval atau garis sejajar (mewakili penyimpanan data).

Pembahasan

I. Pengembangan Sistem Jaringan Waterfall

Metode pengembangan sistem waterfall adalah pendekatan pengembangan perangkat lunak yang bersifat linier dan sekuen. Pendekatan ini terdiri dari serangkaian fase yang dilakukan secara berurutan, dan setiap fase harus diselesaikan sebelum memasuki fase berikutnya.

Berikut adalah penjelasan singkat tentang setiap fase dalam pengembangan sistem jaringan waterfall :

1. Pemahaman dan Analisis Kebutuhan (Requirements Analysis), identifikasi kebutuhan pengguna dan perangkat lunak yang diinginkan. Ini melibatkan pemahaman mendalam terhadap kebutuhan sistem.
2. Perancangan (Design), merancang arsitektur sistem dan spesifikasi teknis. Ini mencakup perencanaan infrastruktur jaringan, struktur database, dan desain antarmuka pengguna.
3. Pembangunan (Implementation), tahap di mana kode perangkat lunak sebenarnya dibuat berdasarkan spesifikasi desain. Proses pengkodean dan pengujian unit dilakukan dalam fase ini.
4. Pengujian (Testing), mengidentifikasi dan memperbaiki bug serta memastikan bahwa sistem berfungsi sesuai dengan spesifikasi. Pengujian dapat mencakup pengujian integrasi, pengujian sistem, dan pengujian penerimaan pengguna.
5. Pengoperasian dan Pemeliharaan (Deployment and Maintenance), menyusun sistem untuk digunakan secara penuh oleh pengguna akhir dan melakukan pemeliharaan rutin serta perbaikan bug jika ditemukan setelah implementasi.

Keuntungan dari model pengembangan waterfall melibatkan pemisahan tahapan, dengan setiap tahap bergantung pada kelengkapan tahap sebelumnya. Hal ini membuat proyek menjadi lebih terstruktur dan mudah dipahami. Namun, kelemahan utamanya adalah kurangnya fleksibilitas untuk perubahan yang mungkin diperlukan selama pengembangan dan risiko tinggi jika kebutuhan tidak sepenuhnya dipahami di awal proyek. Model ini lebih cocok untuk proyek-proyek dengan kebutuhan yang stabil dan jelas.

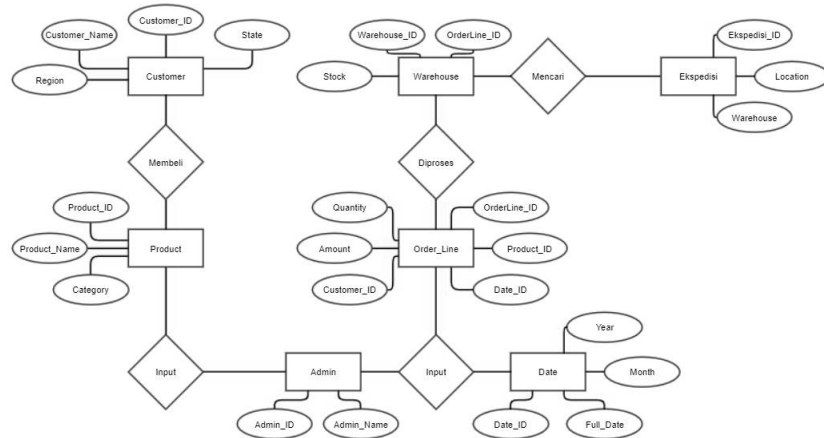
II. Database Architecture

Database Architecture merujuk pada struktur dan organisasi sistem database. Ada beberapa model database architecture, termasuk :

1. Client-Server Architecture, database server memproses permintaan dari klien.
2. Centralized Architecture, semua komponen database terpusat di satu lokasi.
3. Distributed Architecture, database terdistribusi di beberapa lokasi fisik.

a. Entity-Relationship Diagram (ERD)

ERD adalah alat visualisasi yang digunakan untuk merepresentasikan hubungan antara entitas dalam sebuah database. Entitas adalah objek atau konsep yang dapat diidentifikasi, seperti orang, tempat, atau objek. Hubungan antara entitas digambarkan menggunakan garis dan simbol, seperti panah, untuk menunjukkan ketergantungan.

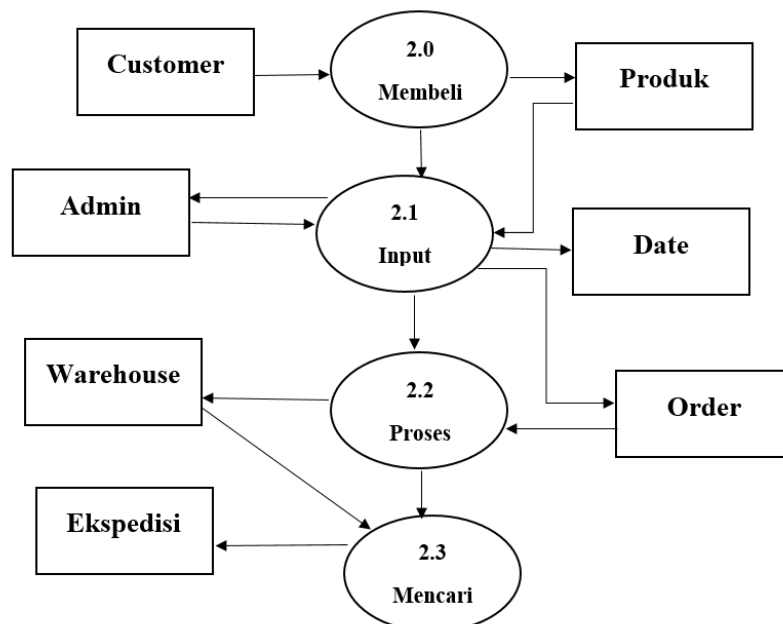


Gambar 1. ERD

b. Data Flow Diagram (DFD)

DFD digunakan untuk menggambarkan aliran data melalui suatu sistem atau proses. Diagram ini terdiri dari proses (kegiatan), data, dan arus data yang menghubungkannya. DFD membantu dalam memodelkan bagaimana data diproses dalam suatu sistem atau antara sistem. DFD memiliki empat komponen utama yaitu :

1. Proses, representasi dari aktivitas atau operasi yang memproses data.
2. Data Flow, representasi jalur atau aliran data antara proses, data store, dan entitas eksternal.
3. Data Store, tempat penyimpanan data yang dapat berupa basis data, file, atau penyimpanan lainnya.
4. External Entities, sumber atau tujuan data yang berada di luar sistem yang sedang dimodelkan.



Gambar 2. DFD

Tabel Customer

Customer_ID	Customer_Name	Region	State

Gambar 3. Tabel Customer

Tabel Product

Product_ID	Product_Name	Category

Gambar 4. Tabel Product

Tabel Admin

Admin_ID	Admin_Name

Gambar 5. Tabel Admin

Tabel Date

Date_ID	Month	Year	Full_Date

Gambar 6. Tabel Date

Tabel Order_Line

OrderLine_ID	Product_ID	Customer_ID	Date_ID	Amount	Quantity

Gambar 7. Tabel Order Line

Tabel Warehouse

Warehouse_ID	OrderLine_ID	Stock

Gambar 8. Tabel Warehouse

Tabel Ekspedisi

Ekspedisi_ID	Location	Warehouse

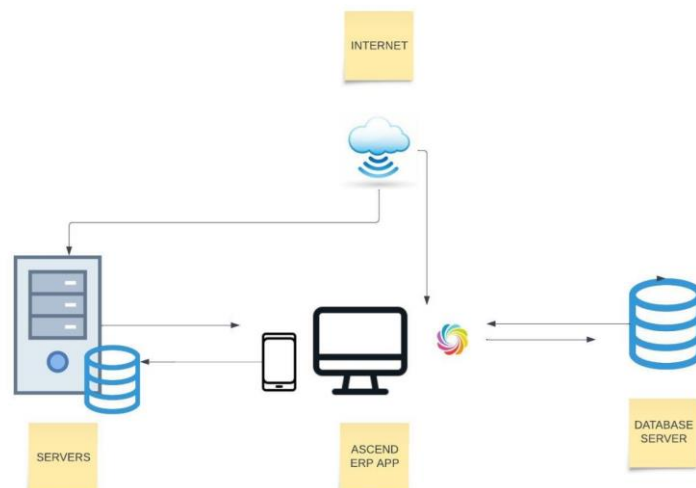
Gambar 9. Tabel Ekspedisi

III. Infrastructure Architecture (Arsitektur Infrastruktur)

Arsitektur Infrastruktur merujuk pada desain dan konfigurasi elemen-elemen inti yang membentuk landasan teknologi suatu organisasi. Ini mencakup perangkat keras, perangkat lunak, jaringan, dan infrastruktur lainnya yang mendukung operasi dan kebutuhan teknologi informasi.

Infrastruktur Architecture berperan penting dalam mendukung operasional, keamanan, dan pertumbuhan suatu organisasi dengan memastikan bahwa elemen-elemen teknologi bekerja bersama secara efisien sesuai dengan kebutuhan bisnis. Berikut adalah beberapa poin kunci terkait Arsitektur Infrastruktur :

1. Perangkat Keras (Hardware), mencakup semua perangkat keras fisik seperti server, komputer, perangkat penyimpanan, dan perangkat jaringan. Desain infrastruktur perangkat keras harus mempertimbangkan kebutuhan kinerja, skalabilitas, dan redundansi.
2. Perangkat Lunak (Software), infrastruktur melibatkan konfigurasi perangkat lunak termasuk sistem operasi, middleware, basis data, dan aplikasi bisnis. Arsitektur perangkat lunak harus mendukung kebutuhan aplikasi dan memastikan kompatibilitas yang baik di seluruh sistem.
3. Jaringan (Network), desain jaringan sangat penting dalam Arsitektur Infrastruktur. Ini mencakup topologi jaringan, kecepatan koneksi, keamanan jaringan, dan teknologi jaringan lainnya. Infrastruktur harus mendukung aliran data yang efisien dan aman.
4. Keamanan (Security), aspek keamanan melibatkan perlindungan terhadap akses yang tidak sah, serangan siber, dan perlindungan data. Ini mencakup penggunaan firewall, enkripsi, kontrol akses, dan langkah-langkah keamanan lainnya.
5. Skalabilitas (Scalability), infrastruktur harus dirancang agar dapat dengan mudah berkembang atau menyusut sesuai dengan kebutuhan bisnis. Ini termasuk pemilihan perangkat keras dan perangkat lunak yang mendukung pertumbuhan yang bersifat linier.
6. Manajemen Sumber Daya (Resource Management), manajemen sumber daya melibatkan pemantauan, manajemen kapasitas, dan optimisasi sumber daya. Tujuannya adalah untuk memastikan penggunaan sumber daya yang efisien dan mengidentifikasi area-area di mana peningkatan dapat dilakukan.
7. Ketersediaan dan Redundansi, infrastruktur harus mempertimbangkan ketersediaan layanan. Ini melibatkan penggunaan redundansi dan pemulihan bencana untuk memastikan kelangsungan operasi dalam situasi darurat.
8. Kompatibilitas dan Integrasi, infrastruktur harus kompatibel dengan teknologi yang ada dan dapat diintegrasikan dengan sistem eksternal. Ini memfasilitasi pertukaran data dan integrasi yang lancar dengan aplikasi lain.

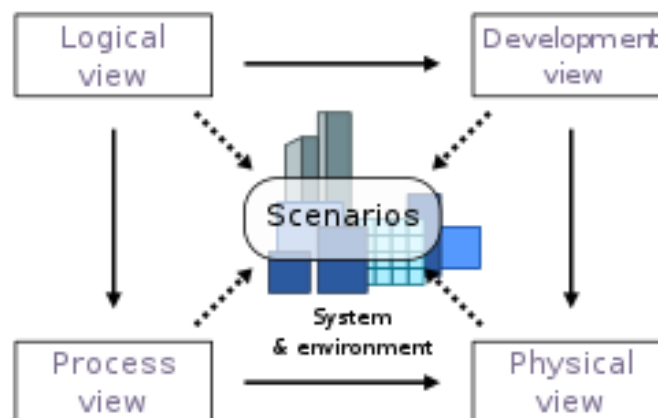


Gambar 10. Infrastuctur Architecture

IV. Software Architecture

Arsitektur perangkat lunak adalah struktur atau kerangka kerja yang digunakan untuk merancang, mengorganisir, dan membangun sistem perangkat lunak. Ini melibatkan pengambilan keputusan strategis terkait dengan elemen-elemen perangkat lunak, seperti komponen, modul, dan interaksi antar mereka. Berikut adalah beberapa poin kunci terkait Arsitektur Perangkat Lunak :

1. Struktur Sistem, arsitektur perangkat lunak mendefinisikan struktur keseluruhan sistem, termasuk bagaimana komponen-komponen perangkat lunak berinteraksi satu sama lain. Ini mencakup pembagian tugas, hubungan, dan aliran data.
2. Komponen Perangkat Lunak, arsitektur mengidentifikasi dan mendefinisikan komponen-komponen perangkat lunak yang membentuk aplikasi. Komponen ini dapat berupa modul, kelas, atau layanan yang bekerja bersama untuk mencapai tujuan sistem.
3. Hubungan Antar Komponen, menentukan bagaimana komponen-komponen tersebut berinteraksi satu sama lain. Ini mencakup aliran data, panggilan fungsi, dan komunikasi antar modul atau layanan.
4. Polanya Desain (Design Patterns), penggunaan pola desain yang telah terbukti untuk menyelesaikan masalah desain umum. Pola desain membantu dalam memecahkan masalah secara efisien dan menyediakan solusi yang dapat digunakan kembali.
5. Keamanan, mempertimbangkan aspek keamanan dalam perancangan perangkat lunak, seperti kontrol akses, enkripsi data, dan mekanisme perlindungan dari serangan siber.
6. Kinerja (Performance), memastikan bahwa arsitektur mendukung kinerja yang memadai sesuai dengan kebutuhan aplikasi. Ini melibatkan manajemen sumber daya dan optimisasi kinerja.
7. Skalabilitas (Scalability), desain sistem agar dapat dengan mudah dikembangkan atau dikurangi sesuai dengan kebutuhan pertumbuhan. Ini termasuk kemampuan untuk menangani beban kerja yang meningkat.
8. Keterbacaan dan Pemeliharaan, menyusun kode sumber agar mudah dibaca, dimengerti, dan dipelihara oleh tim pengembang. Ini melibatkan pemilihan struktur yang jelas dan dokumentasi yang baik.
9. Interoperabilitas, menentukan kemampuan sistem untuk berinteraksi dengan sistem lainnya. Ini melibatkan pemilihan standar komunikasi dan format data yang umum.
10. Keselarasan dengan Kebutuhan Bisnis, memastikan bahwa arsitektur perangkat lunak mendukung dan mencerminkan tujuan bisnis dan kebutuhan pengguna.



Gambar 11. Software Architecture