

BAB II

LANDASAN TEORI

2.1 Definisi Diet

Dalam KBBI, diet diartikan sebagai aturan pola makan khusus yang biasanya ditujukan untuk menjaga kesehatan berdasarkan saran medis. Secara etimologis, kata "diet" berasal dari bahasa Yunani *diaita*, yang berarti "cara hidup". Diet sendiri memiliki makna yang beragam bagi setiap orang, tergantung pada tujuan yang ingin dicapai, seperti menjaga kesehatan, menurunkan berat badan, atau alasan lainnya (Fitria, 2018).

2.2 Definisi Covid-19

Covid-19, atau yang dikenal sebagai Virus Corona, merupakan jenis virus yang sudah dikenal dalam dunia kesehatan hewan. Virus ini termasuk dalam kelompok virus ribonukleat acid (RNA) yang dapat menginfeksi mamalia dan burung. Pada manusia, infeksi virus ini dapat menyebabkan berbagai gejala, mulai dari yang ringan seperti batuk kering dan demam, hingga komplikasi yang lebih serius. Dalam kasus yang parah, infeksi Covid-19 dapat berkembang menjadi gagal napas akibat kerusakan alveolar difus dan sindrom gangguan pernapasan akut (Moderbacher dkk., 2020:996).

2.3 Definisi Imunitas

Kata "imunitas" berasal dari bahasa Latin *immunis*, yang berarti "bebas" atau "dibebaskan". Istilah ini berkembang menjadi *immunity* dalam bahasa Inggris, yang

merujuk pada mekanisme pertahanan tubuh terhadap zat asing seperti mikroorganisme, makanan, bahan kimia, obat-obatan, serbuk sari, dan bulu (Coico & Sunshine, 2015:2).

2.4 Definisi Aplikasi

Dalam buku *Membuat Multiaplikasi Menggunakan Visual Basic 6*, ELEX MEDIA KOMPUTINDO (2004) menyebutkan bahwa multiaplikasi merupakan kumpulan file, seperti form, class, dan report, yang berfungsi untuk menjalankan tugas tertentu secara terorganisir dan saling berkaitan.

2.5 Definisi Android

Edy Winarno ST, M.Eng, Ali, dan Ali Zaki (2014) dalam buku *Tip-Tip Paling Keren BlackBerry dan Android* menjelaskan bahwa Android adalah sistem operasi berbasis Linux yang dikembangkan oleh Google untuk perangkat seluler. Dengan sifatnya yang terbuka, platform ini memungkinkan pengembang membuat dan menjalankan aplikasi di smartphone.

2.5.1 Android Studio

Dalam buku *Belajar Android Studio*, Dayat Suryana Independent (2018) menyebutkan bahwa Android Studio merupakan lingkungan pengembangan terpadu (IDE) berbasis IntelliJ IDEA yang dirancang untuk pengembangan aplikasi Android. IDE ini tidak hanya menyediakan editor kode dan alat pengembang yang canggih, tetapi juga berbagai fitur tambahan yang membantu meningkatkan produktivitas dalam proses pengembangan aplikasi.

2.5.2 Java

Nofriadi, M.Kom (2015) dalam buku Java Fundamental dengan NetBeans 8.0.2 menyebutkan bahwa Java adalah bahasa pemrograman yang dapat berjalan di berbagai sistem operasi, termasuk perangkat seluler. Bahasa ini dikembangkan oleh James Gosling saat masih bergabung dengan Sun Microsystems dan merupakan pengembangan dari C++, dengan sintaks yang mirip dengan C dan C++. Java memiliki sejumlah keunggulan dibandingkan bahasa lain, seperti sifatnya yang multiplatform, dukungan terhadap Pemrograman Berorientasi Objek (PBO), serta pustaka bawaan yang sangat lengkap.

2.5.3 JDK (*Java Development Kit*)

Dalam buku Aplikasi Absensi Dosen dengan Java dan Smartphone sebagai Barcode Reader (2019), Bay Haqi, M.Kom, dan Heri Satria Setiawan, S.E., M.T.I. menyatakan bahwa Java Development Kit (JDK) berfungsi untuk mengompilasi kode Java menjadi bytecode. Bytecode ini kemudian dapat dipahami dan dijalankan oleh Java Runtime Environment (JRE).

2.5.4 SDK (*Software Development Kit*)

Alfa Satyaputra, M.Sc., dan Eva Maulina Aritonang, S.Kom. (2016) menyebutkan dalam bukunya bahwa Software Development Kit (SDK) adalah sekumpulan alat yang mendukung pengembangan perangkat lunak. SDK menyediakan berbagai komponen seperti debugger, pustaka, dan kode sumber yang mempermudah programmer dalam membangun aplikasi.

2.5.5 XML(*eXtensible Markup Language*)

David Roenche (2017) dalam Database Processing Jilid 2 menyatakan bahwa XML (Extensible Markup Language) merupakan bahasa markup yang lebih

unggul dibandingkan HTML karena mampu memisahkan struktur, konten, dan tampilan dokumen secara jelas.

2.5.6 Apk(Android application package)

Dalam buku *Beginning Android Programming with ADT Bundle*, Elex Media Komputindo (2014) menyatakan bahwa APK (Android Application Package) adalah format file yang digunakan untuk menyimpan aplikasi Android dalam ekstensi .apk. File ini memungkinkan aplikasi dijalankan pada perangkat Android atau emulator.

2.5.7 Firebase

Dalam buku *Google Android Firebase: Learning the Basics*, Bill Stonehem (2016) menjelaskan bahwa Firebase adalah layanan cloud yang menyediakan infrastruktur backend bagi aplikasi mobile, memungkinkan pengelolaan data yang lebih sistematis.

2.6 Forward Chaining

Forward chaining adalah metode inferensi yang memulai proses penelusuran dari sekumpulan fakta dengan mencari aturan yang sesuai untuk menguji hipotesa, sehingga akhirnya menghasilkan suatu kesimpulan (Arhami M., 2015:111). Metode ini bekerja dengan cara mengaitkan masalah yang ada dengan solusinya; apabila klausa premis yang diuji bernilai TRUE, maka sistem akan mengeluarkan konklusi yang sesuai.

2.7 Fuzzy

Mochammad Rusli (2017) menjelaskan bahwa sistem fuzzy merupakan pendekatan berbasis aturan (rule-based system) yang memanfaatkan pernyataan

Jika-Maka (IF-THEN) untuk menentukan hasil keluaran berdasarkan input tertentu. Sebagai ilustrasi, sistem kontrol kecerahan layar televisi dapat menggunakan aturan seperti: "Jika jarak pemirsa ke layar dekat dan ruangan sangat gelap, maka kecerahan layar diturunkan."

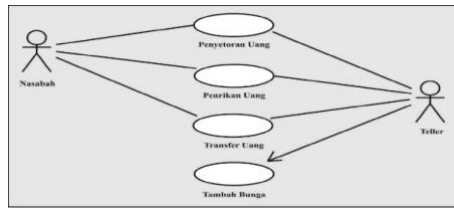
Dalam implementasi fuzzy, variabel linguistik seperti "dekat", "sangat gelap", dan "kecil" direpresentasikan menggunakan fungsi keanggotaan, yang memungkinkan sistem melakukan penyesuaian berbasis derajat keanggotaan (membership function). Dengan kombinasi aturan yang terstruktur, sistem fuzzy dapat berfungsi sebagai pengontrol otomatis yang banyak diterapkan dalam berbagai bidang teknologi.

2.8 UML

Adi Nugroho (2004) menjelaskan dalam bukunya bahwa UML (Unified Modeling Language) adalah sebuah bahasa yang digunakan untuk mendeskripsikan, memvisualisasikan, dan merancang sistem perangkat lunak. UML juga memungkinkan pengembang untuk memasukkan aturan bisnis ke dalam model yang dibuat.

2.9.1 Use Case Diagram

Muhamad Muslihudin Oktafianto (2016) dalam bukunya menjelaskan bahwa Use Case berfungsi sebagai alat untuk mendeskripsikan bagaimana suatu sistem dipandang dari sudut pandang eksternal, sebelum diimplementasikan dalam pemodelan sistem yang lebih detail.



Gambar 2.1 Contoh *Use Case Diagram* (Muslihudin, 2016)

Komponen pembentukan *use case diagram* adalah :

A. Aktor

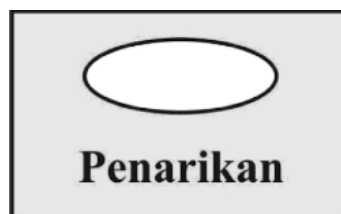
Dalam buku Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML, Muhamad Muslihudin Oktafianto (2016) menjelaskan bahwa Aktor (Actor) adalah individu atau pihak yang memiliki peran dalam sistem dan berinteraksi dengannya.



Gambar 2.2 Contoh Aktor (Muslihudin, 2016)

B. Use Case

Muhamad Muslihudin Oktafianto (2016) menyebutkan bahwa Use case adalah representasi dari berbagai aktivitas atau fungsi yang disediakan oleh sistem atau bisnis untuk penggunaannya. Dalam pemodelan, use case divisualisasikan dalam bentuk elips atau oval untuk menunjukkan interaksi antara sistem dan aktor.



Gambar 2.3 Contoh *Use Case* (Muslihudin, 2016)

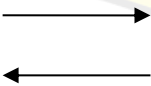
C. Hubungan (*Link*)

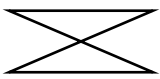
Muhamad Muslihudin Oktafianto (2016) menyatakan bahwa link merupakan komponen yang berfungsi menghubungkan aktor dengan use case yang berinteraksi dengannya dalam sebuah sistem.

2.9.2 Activity Diagram

Munawar (2005) dalam bukunya *Pemodelan Visual dengan UML* menjelaskan bahwa activity diagram digunakan untuk merepresentasikan logika prosedural, alur kerja, dan proses bisnis dalam berbagai skenario.

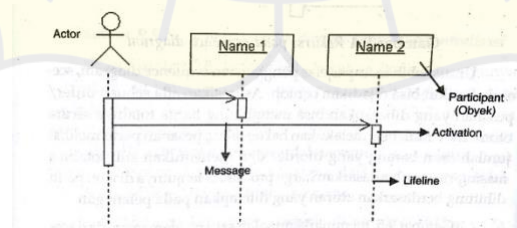
Tabel 2.1 Simbol-simbol yang sering digunakan pada *Activity* diagram (Munawar, 2005)

NO	SIMBOL	KETERANGAN
1		Titik awal
2		Menggambarkan sebuah kegiatan atau tugas yang perlu dilakukan (activity)
3		Menggambarkan sasaran yang mengawali kegiatan
4		Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu

5		Tanda waktu
6		Pilihan untuk pengambilan keputusan
7		Titik akhir

2.9.3 Sequence Diagram

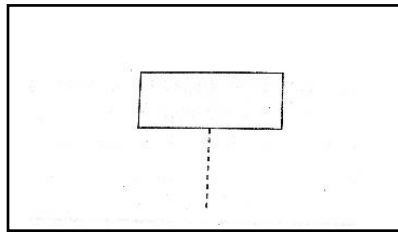
Munawar (2005) dalam bukunya *Pemodelan Visual dengan UML* menjelaskan bahwa sequence diagram digunakan untuk merepresentasikan perilaku dalam suatu skenario. Diagram ini menggambarkan interaksi antara beberapa objek dalam sebuah use case dengan menampilkan pesan-pesan yang dikirimkan di antara objek-objek tersebut. Komponen utama dalam sequence diagram terdiri dari objek yang direpresentasikan dalam bentuk persegi panjang dengan nama tertentu. Pesan dalam diagram ini ditunjukkan menggunakan garis panah, sementara urutan waktu diilustrasikan dalam progresi vertikal.



Gambar 2.4 Contoh *sequence diagram* (Munawar, 2005)

A. Obyek/Participant

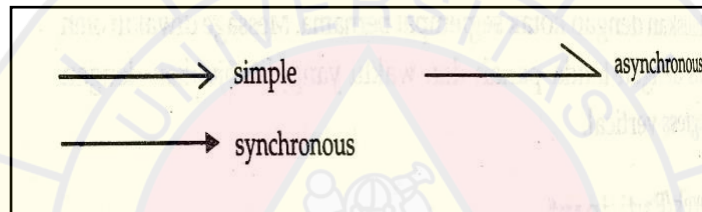
Obyek dalam diagram ditempatkan di bagian atas dengan susunan dari kiri ke kanan untuk mempermudah pemahaman.



Gambar 2.5 Contoh *participant* atau obyek di *sequence* diagram (Muhawar, 2005)

B. *Message*

Message menggambarkan interaksi antar partisipan dan berpindah dari satu lifeline ke lifeline lainnya dalam diagram.



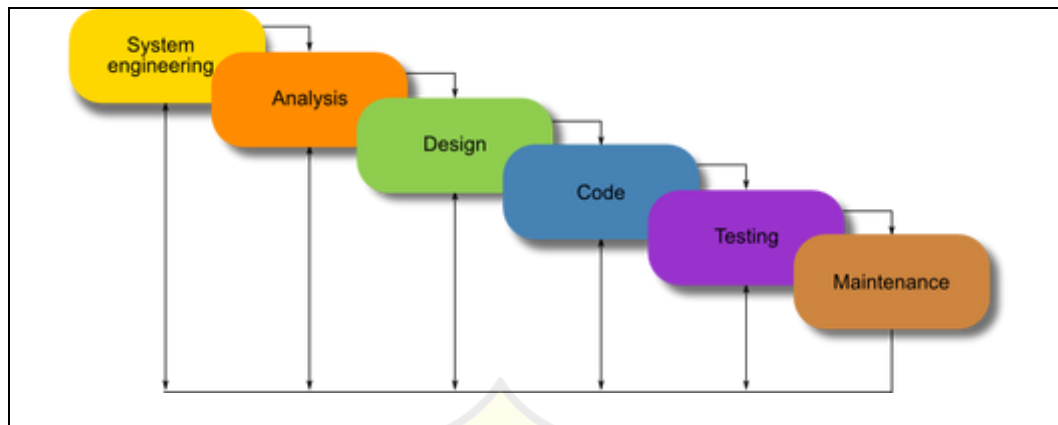
Gambar 2.6 Simbol – simbol *message* pada *sequence* diagram (Munawar, 2005)

C. *Time*

Diagram ini merepresentasikan dimensi waktu dalam arah vertikal, di mana waktu dimulai dari bagian atas dan bergerak ke bawah. Pesan yang lebih dekat dengan bagian atas akan dieksekusi lebih dulu dibandingkan dengan pesan yang lebih bawah.

2.9 *Waterfall*

Model Waterfall dalam SDLC memiliki prinsip bahwa setiap tahap pengembangan harus selesai terlebih dahulu sebelum masuk ke tahap selanjutnya. Hal ini memastikan bahwa setiap fase mendapatkan perhatian penuh tanpa ada pekerjaan yang dilakukan secara bersamaan. Berikut adalah tahap-tahap dalam model Waterfall.



Gambar 2.7 Fase Metode Waterfall

