

BAB II

LANDASAN TEORI

2.1 Landasan Teori

Teori-teori yang mendasar sebagai penunjang proses penelitian yaitu sebagai berikut :

2.1.1 Sistem

Sistem informasi dibuat oleh manusia yang terdiri dari komponen-komponen untuk mencapai suatu tujuan yaitu menyajikan informasi. Komponen-komponen yang bekerja sama untuk mencapai tujuan di dalam sistem[1], adalah sebagai berikut:

1. Tujuan Sistem
Merupakan tujuan akhir dibuatnya sistem.
2. Batasan sistem
Aturan-aturan yang membatasi sistem dalam mencapai tujuan, dapat berupa organisasi, jangka waktu ataupun jumlah biaya
3. Kontrol sistem
Salah satu cara untuk mengawasi jalannya sistem, berupa pengawasan waktu pelaksanaan, biaya dan sebagainya.
4. Masukan (*input*)
Semua data yang digunakan dalam sistem diterima oleh elemen masukan, dapat berupa jenis data dan jumlah nilai
5. Proses
Elemen dari sistem yang bertugas memproses semua inputan data yang kemudian menjadi informasi yang berguna.
6. Keluaran (*output*)
Hasil keluaran dari proses yang merupakan akhir dari sistem, dapat berupa laporan dan grafik.
7. Umpan balik
Pengaruh terhadap organisasi yang menerapkan sistem tersebut. Sistem Informasi adalah kumpulan perangkat keras, perangkat lunak, database, telekomunikasi, manusia dan prosedur yang dirancang untuk mentransformasikan data kedalam bentuk informasi yang berguna. Kualitas dari suatu informasi tergantung tiga hal, yaitu:

- a. Akurat

Informasi dikatakan akurat apabila informasi tersebut tidak menyesatkan, bebas dari kesalahan-kesalahan dan harus jelas mencerminkan maksud dari informasi tersebut.

b. Tepat Waktu

Informasi yang datang pada penerima tidak boleh terlambat, karena informasi merupakan landasan dalam pengambilan keputusan.

c. Relevan

Informasi harus mempunyai manfaat untuk penerima.

2.1.2 Informasi

Informasi adalah data atau objek yang diolah terlebih dahulu sedemikian rupa sehingga dapat disusun dan diperjelas dengan baik sehingga memiliki makna bagi penerima yang kemudian menjadi pengetahuan bagi penerima tentang suatu hal tertentu yang membantu mengambil keputusan dengan tepat[1].

2.1.3 Sistem Informasi

Sistem informasi adalah seperangkat perangkat keras, perangkat lunak, *brainware*, prosedur, dan / atau aturan yang diatur secara integral untuk mengolah data menjadi informasi yang berguna untuk pemecahan masalah dan pengambilan keputusan[1].

2.1.4 Sistem Monitoring

Monitoring didefinisikan sebagai siklus kegiatan yang mencakup pengumpulan, peninjauan ulang, pelaporan, dan tindakan atas informasi suatu proses yang sedang diimplementasikan. Pada umumnya, monitoring digunakan dalam *checking* antara kinerja dan target yang telah ditentukan[2].

2.1.5 Remaja

Remaja merupakan masa terjadinya tumbuh kembang secara pesat baik fisik, *psikologis* atau secara *intelektual*. Remaja berumur sekitar 12 sampai 18 tahun. Masa tumbuh kembang tersebut mengakibatkan para remaja memiliki sifat dan karakter khas yang sama yaitu rasa keingintahuan yang tinggi dan gemar melakukan petualangan serta suka terhadap tantangan. Remaja juga cenderung berani mengambil resiko atas perilaku yang diperbuat tanpa mempertimbangkan secara matang lebih dahulu. Sifat ini dihadapkan pada ketersediaan sarana di sekitarnya yang dapat memenuhi keingintahuan tersebut [3].

2.1.6 Posyandu Remaja

Posyandu Remaja atau Pos Pelayanan Terpadu Remaja adalah sebuah wadah pos kesehatan remaja yang memfasilitasi dalam memahami seluk beluk remaja selama masa puber yang ditujukan kepada siswa dan remaja pada umumnya. Selain itu, pos pelayanan terpadu remaja berfungsi juga sebagai sebuah wadah untuk memberi remaja kesempatan berkembang sesuai dengan kebutuhan dan karakteristiknya melalui pendekatan terpadu dari segi medis [3].

2.1.7 Indeks Massa Tubuh

Indeks Massa Tubuh (IMT) adalah suatu ukuran yang digunakan untuk menentukan apakah berat badan seseorang termasuk kategori ideal, kurang, atau berlebih. IMT diperoleh dengan membagi berat badan dengan kuadrat tinggi badan (kg/m^2). Rumusnya adalah sebagai berikut :

$$\text{Indeks Massa Tubuh} = \frac{\text{Berat Badan (kg)}}{\text{Tinggi Badan}^2 (\text{m}^2)}$$

IMT mempunyai keunggulan utama yakni menggambarkan lemak tubuh yang berlebihan, sederhana dan bisa digunakan dalam penelitian populasi berskala besar. Pengukurannya hanya membutuhkan 2 hal yakni berat badan dan tinggi badan [4]. Berat badan pada masa remaja merupakan prediktor yang sangat baik untuk mengetahui kemungkinan seseorang dapat terkena obesitas atau penyakit merugikan lainnya. Indeks masa tubuh (IMT) merupakan metode yang dianjurkan dan banyak digunakan untuk mengetahui tingkat obesitas [5]. Berikut adalah ambang batas status gizi berdasarkan indeks massa tubuh mengacu pada Kemenkes RI Tahun 2020 dapat dilihat pada Tabel 2.1.

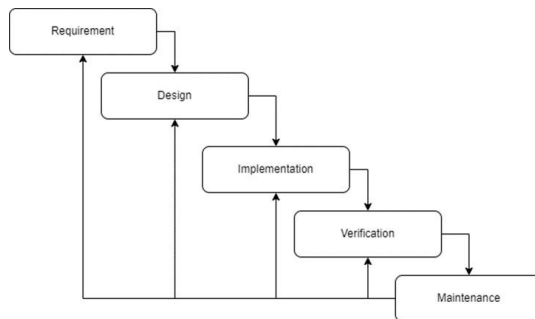
Tabel 2. 1 Ambang Batas Indeks Massa Tubuh

Kategori	IMT
Sangat Kurus	< 17,0
Kurus	17,0 – 18,4
Normal	18,5 – 25,0
Gemuk	25,1 – 27,0
Obesitas	>27,0

2.1.8 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan perihal kegiatan yang kreatif dan sistematis berdasar suatu disiplin ilmu yang membangun suatu perangkat lunak berdasar suatu aspek masalah tertentu. Proses perangkat lunak sebagai sebuah kerangka kerja untuk tugas-tugas yang dibutuhkan dalam membangun perangkat lunak dengan kualitas yang baik. Proses perangkat lunak menentukan pendekatan yang digunakan pada perangkat lunak yang dikembangkan. Pengembangan perangkat lunak juga meliputi teknologi yang mempopulasikan metode-metode, teknis, alat-alat bantu otomatis, dan prosedur-prosedur atau sering disebut dengan proses [6].

System Development Life Cycle (SDLC) merupakan proses mengembangkan atau mengubah sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya. Untuk mempermudah dalam proses pengembangan perangkat lunak dibutuhkan model atau metodologi pengembangan perangkat lunak, salah satunya model *waterfall* menurut Pressman. *Flowchart* metode *waterfall* dapat lihat pada Gambar 2.1.



Gambar 2. 1 *Flowchart* metode *waterfall*

Tahapan-tahapan dari metode ini sebagai berikut :

a. *Requirement*

Requirement merupakan proses dari analisa atau pengumpulan data - data yang berkaitan dengan sistem yang akan dibuat. Pada tahap ini, pengembang harus mengetahui dan memahami perangkat lunak yang diharapkan oleh pengguna. Metode pengumpulan data dapat diperoleh melalui diskusi, observasi, survei, dan wawancara.

Kemudian data yang didapat diolah dan dianalisis sehingga memperoleh informasi yang lengkap mengenai spesifikasi kebutuhan pengguna terhadap perangkat lunak yang akan dikembangkan.

b. *Design*

Pada tahap ini, pengembang sistem akan membuat desain terkait sistem yang akan dibuat. Design ini dapat digunakan untuk menentukan perangkat keras (*hardware*) dan membantu mendefinisikan arsitektur sistem secara keseluruhan.

c. *Implementation*

Pada tahap ini, mengimplementasi hasil desain ke dalam kode atau bahasa yang dimengerti oleh mesin komputer dengan menggunakan bahasa pemrograman tertentu.

d. *Verification*

Pada tahap ini, seluruh program kecil (*unit*) yang dikembangkan dilakukan verifikasi dalam sistem secara keseluruhan. Selanjutnya dilakukan verifikasi dan pengujian sistem apakah perangkat lunak telah sesuai dengan spesifikasi kebutuhan pengguna atau terdapat *error* dalam sistem sebelum kemudian diperbaiki ulang.

e. *Maintenance*






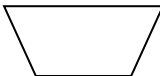

Tahap ini merupakan tahap akhir dari metode *waterfall*. Perangkat lunak yang telah dibuat akan dioperasikan pengguna dan dilakukan pemeliharaan. Pemeliharaan adalah proses memperbaiki aplikasi dari setiap *error* atau *bug*, peningkatan kinerja aplikasi, penambahan program kecil (*unit*) baru untuk pengembangan aplikasi, dan penyesuaian sistem sesuai dengan kebutuhan pengguna.

Adapun struktur data merupakan cara menyimpan atau mempresentasikan data didalam komputer agar bisa dipakai secara efisien. Berikut bagian-bagian yang ada pada struktur data:

A. *Flowchart*

Flowchart merupakan suatu diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program [7]. Pada Tabel 2.2 merupakan simbol-simbol *flowchart*:

Tabel 2. 2 Simbol-simbol *flowchart*





No	Simbol	Nama	Keterangan
1		<i>Terminator Symbol</i>	Simbol yang menyatakan awal atau akhir suatu program.
2		<i>Input-Output</i>	Memasukkan data maupun menunjukkan hasil dari suatu proses tanpa tergantung dengan jenis pendataannya.
3		<i>Processing Symbol</i>	Menyatakan suatu proses yang dilakukan oleh computer
4		<i>Decision symbol</i>	Menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya atau tidak.
5		<i>Flow symbol</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. Simbol arus ini sering disebut juga dengan <i>connecting line</i> .
6		<i>Manual symbol</i>	Menyatakan suatu proses yang tidak dilakukan computer
7		<i>Document Symbol</i>	Merupakan simbol untuk data yang terbentuk informasi.




B. Unified Modeling Language (UML)

Salah satu standar bahasa yang banyak digunakan didunia industry untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur disebut dengan *Unified Modeling Language* (UML). UML menyederhanakan permasalahan-permasalahan yang kompleks sehingga dapat dipelajari dan dipahami dengan mudah. UML dapat menggambarkan hubungan antar kelas dengan garis lurus[7]. Model UML yang dipakai dalam pengembangan sistem ini adalah *use case diagram*.

Use case diagram merupakan permodelan untuk melakukan sistem informasi yang akan dibuat. *Use case diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Tabel 2.3 menunjukkan simbol-simbol yang ada pada *use case diagram*.

Tabel 2. 3 Simbol-simbol *use case diagram*

No	Simbol	Nama	Keterangan
1		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
2		<i>Actor</i>	Mewakili peran orang, sistem yang lain, atau alat Ketika berkomunikasi dengan <i>use case</i> .
3		<i>Assosiation</i>	Menghubungkan antara objek satu dengan objek lainnya.
4		<i>Include</i>	Menspesifikan bahwa <i>use case</i> sumber segala eksplisit.

5		<i>Extend</i>	Menspesifikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber suatu titik yang diberikan.
6		<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.
7		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.

2.1.9 Pemrograman Berorientasi Objek (PBO)

Pemrograman berorientasi objek merupakan pemrograman berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam *class-class* atau *object-object*. Syarat sebuah bahasa pemrograman bisa digolongkan sebagai berorientasi objek adalah bila Bahasa pemrograman tersebut memiliki fitur untuk mengimplementasikan ke 4 (empat) konsep berorientasi objek yaitu [8]:

- a). *Abstraksi* adalah sebuah prinsip yang digunakan untuk merepresentasikan dunia nyata yang kompleks ini menjadi sebuah model yang sederhana dengan menghiraukan aspek - aspek lainnya yang tidak sesuai dengan permasalahan.
- b). *Encapsulation* adalah bagaimana membungkus data dan method yang menyusun class hingga class dipandang sebagai suatu modul.
- c). *Polymorphison* adalah sesuatu yang memiliki banyak bentuk, diartikan sebagai modul yang memiliki kesamaan nama, namun behaviour (tingkah laku yang berbeda) sehingga listing kode implementasinya berbeda.
- d). *Inheritance* (Penurunan sifat) adalah proses pewarisan data dan method dari suatu class kepada class yang lain.

Metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi kelompok data dan fungsi. Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek. Metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi data dan fungsi. Sehingga, objek dapat memiliki kelakuan yang berbeda, apabila diperbarui implementasi dari objek tersebut sesuai dengan karakteristiknya[9].

2.1.10 Framework Laravel

Laravel adalah salah satu php framework paling populer dan paling banyak digunakan untuk membuat website saat ini. Dengan menggunakan laravel proses pengembangan aplikasi menjadi lebih cepat dan powerfull. Dalam pengembangan website menggunakan laravel, ada 2 tools yang akan sering kita pakai. yaitu composer dan artisan. *Framework* adalah kumpulan intruksi-intruksi yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masingmasing untuk memudahkan *developer* dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama berulang-ulang serta dapat menghemat waktu. Kelebihan dari *framework laravel* jika dibandingkan dengan *framework* lain adalah sebagai berikut :

1. Gratis (*Open-Source*)

Laravel merupakan framework yang bersifat open source, bisa digunakan dengan gratis. Framework ini juga banyak digunakan karena developer dapat menggunakannya untuk dikembangkan lagi.

2. Lebih aman

Laravel adalah framework PHP yang memberikan beberapa pilihan yang dapat digunakan agar aplikasi tetap aman. Salah satu sistem keamanan yang dimiliki Laravel adalah PDO yang bisa mencegah SQL Injection.

3. Menggunakan konsep M-V-C

Laravel merupakan konsep M-V-C (*Hierarchical ModelView-Controller*) yang memungkinkan pemisahan antara *layer application-logic* dan *presentation*.

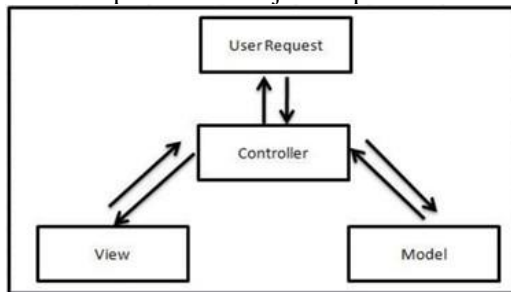
Menggunakan konsep ini kode PHP, query Mysql, Javascript dan CSS dapat saling dipisah-pisahkan sehingga ukuran file menjadi lebih kecil dan lebih mudah dalam perbaikan kedepannya atau *maintenance*.

- a. Model Kode merupakan program (berupa OOP *class*) yang

digunakan untuk berhubungan dengan database MySQL sekaligus untuk memanipulasinya (*input-edit-delete*).

- b. *View* Merupakan kode program berupa *template* atau PHP untuk menampilkan data pada browser.
- c. *Controller* merupakan Kode program (berupa OOP *class*) yang digunakan untuk mengontrol aliran atau dengan kata lain sebagai pengontrol model dan *view*.

Adapun alur dari program aplikasi berbasis *laravel* yang menggunakan konsep M-V-C ditunjukkan pada Gambar 2.2 berikut :



Gambar 2. 1 Konsep Aliran

2.1.11 *Design* Antarmuka Pengguna

Antarmuka pengguna adalah suatu istilah yang digunakan untuk menggambarkan tampilan dari mesin atau komputer yang berinteraksi langsung dengan pengguna. *User Interface* atau tampilan antarmuka pengguna merupakan bagian yang penting dalam sebuah sistem atau aplikasi. UI adalah bagian dari sistem yang digunakan untuk berinteraksi langsung dari pengguna. Desain dan penyusunan tampilan antarmuka perlu diperhatikan untuk menghasilkan tampilan yang bagus. Dengan mengetahui desain tampilan antarmuka yang mudah digunakan maka pengembang aplikasi akan dapat membuat sistem yang menarik pengguna untuk menggunakan aplikasinya. Dari sisi pengguna diharapkan semua aplikasi untuk ke depannya menggunakan desain tampilan antarmuka yang *user-friendly* sehingga pengguna tidak perlu merasa terganggu ketika menggunakan sebuah sistem informasi.

2.1.12 *Basis* Data

Basis data merupakan koleksi dari data yang terorganisasi dengan cara sedemikian rupa sehingga data tersebut mudah disimpan

dan dimanipulasi. Disamping berisi atau menyimpan data, setiap basis data juga mengandung/menyimpan definisi struktur[10].

a) **Database Management System (DBMS)**

Database Management System (DBMS) atau dalam Bahasa Indonesia sering disebut sebagai sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. Dalam penggunaan DBMS dibutuhkan komponen-komponen antara lain :



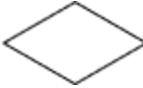

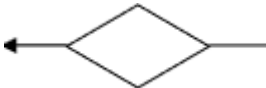
- (1) *Query processor*, komponen yang mengubah bentuk *query* dalam bentuk instruksi ke dalam *database manager*.
- (2) *Database manager*, menerima *query*, menguji eksternal dan konseptual untuk menentukan apakah *record-record* tersebut dibutuhkan untuk memenuhi permintaan kemudian hari dari *database manager* dengan memanggil *file manager* untuk menyelesaikan permintaan.
- (3) *File manager*, memanipulasi penyimpanan file dan mengatur alokasi ruang penyimpanan disk.
- (4) *Data manipulation language processor*, modul yang mengubah perintah DML yang ditempelkan ke dalam program aplikasi dalam bentuk fungsi-fungsi.
- (5) *Data definition language compiler*, mengubah statement DDL menjadi kumpulan tabel atau *file* yang berisi data *dictionary* atau meta data.
- (6) *Dictionary manager*, mengatur akses dan memelihara data *dictionary*.

Salah satu *software* yang tergolong ke dalam DBMS adalah *MySQL*. *MySQL* merupakan salah satu *Relational Database Management System* (RDBMS) yang saat ini sedang banyak diminati.

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). *Entity Relationship Diagram* (ERD) adalah sekumpulan cara atau peralatan untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut entitas (*entity*) serta hubungan (*relationship*) antar entitas-entitas tersebut dengan menggunakan beberapa notasi[10]. Komponen-komponen pembentuk ERD dapat di lihat pada Tabel 2.4 di bawah ini :

Tabel 2. 4 Komponen-Komponen ERD

Notas	Komponen	Keterangan
-------	----------	------------

i		
	Entitas	Individu yang mewakili suatu objek dan dapat dibedakan dengan objek yang lain.
	Atribut	Properti yang dimiliki oleh suatu entitas, dimana dapat mendeskripsikan karakteristik dari entitas tersebut.
	Relasi	Menunjukkan hubungan antara sejumlah entitas yang berbeda.
	Relasi 1:1	Relasi yang menunjukkan bahwa setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak satu entitas pada himpunan entitas kedua.
	Relasi 1:N	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak atau sebaliknya. Setiap entitas dapat berelasi dengan banyak entitas pada himpunan entitas yang lain.

Metode pemodelan data dengan menggunakan *Entity-Relationship Diagram* (ERD) digunakan untuk mengidentifikasi objek data dan hubungannya dengan menggunakan notasi grafis dalam perancangan perangkat lunak. Atribut dari masing-masing objek data digambarkan dengan menggunakan deskripsi objek data.

ERD hanya berfokus pada data dengan menunjukkan jaringan data yang ada untuk suatu sistem yang diberikan.

b) *Structural Query Language (SQL)*

Structural Query Language (SQL) adalah Bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. Secara umum SQL terdiri dari 2 (dua) bahasa yaitu *Data Definition Language (DDL)* dan *Data Manipulation Language (DML)*.

(1) *Data Definition Language (DDL)*

Data Definition Language (DDL) merupakan suatu perintah yang berfungsi untuk mendefinisikan atribut atribut basis data, tabel, atribut serta hubungan antar tabel. DDL berfungsi lebih ke dalam memanipulasi struktur *database*. DDL digunakan untuk membuat tabel atau menghapus tabel, membuat *key* atau *indeks*, membuat relasi antar tabel. Berikut sintaks yang ada didalam DDL:

(a) *Create*

Perintah *create* digunakan untuk membuat objek baru, baik berupa *database*, tabel, indeks atau prosedur yang tersimpan.

(b) *Alter*

Perintah *alter* digunakan untuk memodifikasi onjek pada *database*, seperti *indeks*, dan lokasi.

(c) *Drop*

Perintah *drop* digunakan untuk menghilangkan atau menghapus objek pada *database*.

(2) *Data Manipulation Language (DML)*

Data Manipulation Language (DML) merupakan kelompok perintah yang berfungsi untuk melakukan proses insert, update atau delete ke dalam suatu *database*. Berikut sintaks yang ada didalam DML:

(a) *Select*

Perintah *select* digunakan untuk menampilkan data/isi tabel dari *database*.

(b) *Insert*

Perintah *insert* untuk menambahkan data baru dalam *database*.

(c) *Update*

Perintah *update* digunakan untuk merubah data didalam *database*.

(d) *Delete*

Perintah *delete* digunakan untuk menghapus data dari *database*.