

BAB 2

TINJAUAN PUSTAKA

2.1 Wearable Device

Wearable device atau perangkat *wearable*, merupakan perangkat yang dapat dikenakan pada tubuh serta diklasifikasikan berdasarkan fungsinya, penampilan, kedekatannya dengan tubuh manusia, dan parameter lainnya. Contohnya seperti yang digunakan untuk mendeteksi kebugaran seseorang. Biasanya teknologi ini disematkan pada *wearable device* seperti *smartwatch* dan *smartband*. Sedangkan dibidang medis, alat yang digunakan untuk memperoleh data fisiologis, data perilaku dan kontekstual untuk diagnosis pengobatan dan pengelolaan penyakit terhadap pasien. Lalu dibidang industri seperti konstruksi yang menggunakan alat untuk pendeteksian keselamatan dan pemantauan fisiologis pekerja nya[10].

Agar *wearable device* dapat mendeteksi aktivitas, diperlukan adanya peranan sensor. Studi menunjukkan sensor yang banyak digunakan untuk mendeteksi aktivitas adalah sensor jenis non visual, yaitu sensor accelerometer (Ranasinghe, Al MacHot, & Mayr, 2016)[30]. *Wearable device* pertama kali ditemukan oleh Ed Thorp dan Claude Shannon pada tahun 1996 yang pada saat itu berupa komputer analog yang memiliki ukuran sebesar kotak rokok. *Wearable device* yang merupakan teknologi *smart* memiliki banyak sensor yang sematkan didalamnya [29] seperti :

A. Accelerometer

Bisa digunakan sebagai alat ukur dalam beberapa parameter seperti Akselerasi, getaran , rotasi, dan gravitasi.

B. Proximity Ambient

Digunakan untuk melacak keberadaan dan jarak *user* dengan memanfaatkan pantulan inframerah.

C. Magnetometer

Digunakan untuk mengukur medan magnet dan juga bisa digunakan sebagai penunjuk arah.

D. Gyroscope

Sebagai alat ukur untuk orientasi yang kemudian penggunaannya dapat dipakai untuk mengukur berat, bentuk serta kecepatan.

E. RGB

Sebagai alat ukur untuk temperatur warna dari suatu lingkungan melalui gelombang cahaya *Red, Green, Blue*.

F. Fingerprint

Biasanya teknologi ini digunakan untuk keamanan suatu sistem atau perangkat. Cara menggunakannya cukup dengan menempelkan sidik jari (biasanya jempol atau telunjuk) yang telah terdaftar di sistem kepada sensor sidik jarinya.

G. Heart Rate

Digunakan untuk mengirim detak jantung, biasanya *device* ditempelkan di dada atau pun di pergelangan tangan yang kemudian saling terhubung dengan *smartphone* melalui konektivitas *bluetooth* agar data detak jantung lebih nyaman untuk dilihat.

H. Pedometer

Memiliki fungsi yang hampir serupa dengan *accelerometer* namun dapat memberikan hasil yang lebih akurat dan efisien.

Ukuran dari *wearable device* relatif kecil, hal ini dimaksudkan agar *user* dapat dengan mudah memakainya atau perangkat tersebut menjadi "*wearable*". Namun karena bentuknya yang kecil, daya yang dapat tersimpan pada *wearable device* tentunya kecil juga[31].

Kesimpulannya *wearable device* merupakan perangkat yang berukuran relatif kecil, dengan tujuan mudah dikenakan oleh *user*. Dalam penggunaannya *wearable device* memakai sensor untuk membaca data dari *user*nya. Saat ini penerapan *wearable device* digunakan sebagai *smart watch*, *smart bracelet*, *smart glasses* dan masih banyak lagi.

2.2 Smartband

Smartband merupakan perangkat *wearable* yang digunakan pada pergelangan tangan layaknya sebuah jam tangan. Smartband memiliki banyak fitur yang dapat digunakan seperti untuk mendeteksi detak jantung, tekanan darah, kandungan oksigen dalam darah dan juga fitur alarm yang dapat di atur dari *smartphone user*. Selain itu smartband juga dapat menerima notifikasi pesan dari handphone seperti pesan whatsapp, *short message service* (SMS) ataupun notifikasi dari Aplikasi lainnya.

Terdapat konektivitas *bluetooth* agar *smartband* dapat terhubung dengan *smartphone*. Kemudian agar Aplikasi pada *smartphone* dapat mengambil data dari *smartband* digunakan lah SDK dari smartband tersebut (beda merk maka SDK pun akan berbeda). Didalam SDK tersebut terdapat fungsi-fungsi yang digunakan untuk mengambil data dari *smartband* yang telah terhubung dengan *smartphone* melalui konektivitas *bluetooth*.

2.3 Bluetooth Low Energy (BLE)

Bluetooth low energy merupakan teknologi *bluetooth* yang konsumsi dayanya jauh lebih rendah dibandingkan *bluetooth* biasa. *Bluetooth low energy* bekerja memakai sinyal radio dengan frekuensi 2.4GHz. BLE memiliki jangkauan yang lebih luas, komunikasi yang cepat dan komunikasi yang tidak dipengaruhi benda padat[42]. Beberapa parameter yang terdapat pada BLE diantaranya [43]:

1. Universally Unique Identifier (UUID)

Kumpulan 32 karakter yang berupa *String* yang dibuat secara acak. Kode identifikasi ini dibuat sebagai tanda bagi sebuah perangkat yang memiliki BLE, agar dapat diidentifikasi oleh perangkat lain yang akan dihubungkan.

2. Receive Signal Strength Indicator (RSSI)

Merupakan teknologi yang digunakan untuk mengukur indikator sinyal yang diterima oleh perangkat *wireless* (Sahu dkk, 2013).

3. Major & Minor Value

Standar identifikasi yang berupa tipe Integer yang bernilai antara 0 sampai dengan 65535

BLE banyak dipasangkan pada perangkat *wireless* seperti *Smartwatch*, *Smartband*, *Speaker portable*, dan masih banyak lagi. Hal tersebut dikarenakan penggunaan daya nya yang hemat , yang mana bisa dipakai dalam jangka waktu yang lebih lama dibanding *Bluetooth* yang biasa. Pada penelitian ini, digunakan UUID sebagai *identifier* dari *miband 2* yang akan dihubungkan dengan aplikasi *smartphone*.

2.4 Accelerometer

Accelerometer merupakan suatu alat elektromagnetik yang berfungsi untuk mengubah gerakan fisik menjadi sinyal elektrik yang sebanding dengan kekuatan otot saat melakukan gerakan (Melanson & Freedson, 1996). Pengukuran dilakukan dengan akselerasi yang terjadi, yang mana terbagi menjadi 2 yaitu dinamis dan statis[44]. Accelerometer dinamis diukur melalui gerakan dan getaran yang terdeteksi oleh sensor, dan untuk yang statis diukur melalui sudut, kemiringan, serta gerakan yang diakibatkan gravitasi.

Penggunaan *accelerometer* pada alat-alat elektronik sudah banyak ditemukan saat ini. Dimulai dari yang sering dipakai sehari-hari yaitu

smartphone, sensor ini dipasangkan pada *smartphone* untuk banyak hal semisal menstabilkan tangkapan gambar melalui kamera, mendukung fitur pada *smartphone* seperti melakukan orientasi dari vertikal ke horizontal. Namun respon dari sensor ini tidak begitu cepat serta hanya bisa membaca dalam dua sumbu yaitu atas dan bawah[45].

2.5 Android

Android adalah sistem operasi pada perangkat seluler yang berbasis linux[6]. Sistem operasi tersebut meliputi beberapa layanan yang tersedia pada perangkat seluler seperti menerima pesan, telepon, musik, kamera dan masih banyak lagi.

Hozeng and Syam (2017) Android adalah “sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet”[25]. Menurut Labellapansa et al.,(2017) Android merupakan “sistem operasi berbasis linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet”[26]. Sedangkan menurut Supardi (2014, 2), Android merupakan “sebuah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi”[27].

Dari beberapa pernyataan diatas, dapat disimpulkan bahwa Android merupakan sebuah sistem operasi yang berbasis linux. Dimana sistem operasi tersebut dipakai untuk perangkat seluler yang mencakup sistem operasi, *middleware*, dan aplikasi.

2.5.1 Sejarah Android

Android sebagai sistem operasi dimulai sejak oktober 2003 yang dikembangkan oleh perusahaan Android inc, dimana pendirinya terdiri dari 4 orang bernama Andi Rubin, Rich Minner, Nick Sears dan Chris White[28]. Sebelum pada tahun 2006 google mengakuisisi perusahaan tersebut[7].

Untuk mengembangkan sistem operasi Android dibentuklah *Open Handset Alliance* (OHA) dan pada 5 November 2007 Android bersama OHA mendukung pengembangan terbuka pada perangkat seluler[8]. Sampai sekarang sistem operasi Android bersaing dipasar *smartphone* dimulai dari *smartphone* kelas bawah hingga *smartphone* kelas atas. Sistem operasi Android hampir di pakai oleh semua pabrikan *smartphone* mulai dari samsung, xiaomi, oppo, dan masih banyak lagi.

2.5.2 Perkembangan Android

Sejak April 2009, versi Android dikembangkan dengan *codename* yang diberi nama berdasarkan makanan pencuci mulut atau *dessert* [9]. Berikut urutan seri Android :

- a. Android 1.0 Astro
- b. Android 1.1 Blender
- c. Android 1.5 Cupcake
- d. Android 1.6 Donut
- e. Android 2.0-2.1 Éclair
- f. Android 2.2 Froyo
- g. Android 2.3 Gingerbread
- h. Android 3.0/3.1 Honeycomb
- i. Android 4.0 Ice Cream Sandwich
- j. Android 4.1/4.2/4.3 Jelly Bean
- k. Android 4.4 Kitkat
- l. Android 5.0 Lollipop
- m. Android 6.0 Marshmallow
- n. Android 7.0 Nougat
- o. Android 8.0 Oreo
- p. Android 9.0 Pie

2.6 SDK

SDK atau *Software Development Kit* merupakan alat API yang diperlukan untuk pengembangan perangkat lunak pada sistem operasi Android yang menggunakan bahasa pemrograman Java[8]. Menurut Mutia (2018) SDK merupakan alat yang digunakan oleh para *programmer* yang akan mengembangkan sebuah Aplikasi yang berbasis Android. SDK terdiri dari debugger, library, handset emulator, dan dokumentasi[11].

Menurut Kusniyati (12:2016) Android SDK merupakan tools API. (Application Programming Interface).yang dibutuhkan untuk. Mengawali mengembangkan..aplikasi pada platform Android menggunakan..bahasa pemograman java[33].

2.7 API

API adalah antarmuka yang dibangun oleh pengembang sehingga fungsi sistem dapat diakses secara terprogram. Dengan penggunaan API dalam pembuatan *software* dapat memudahkan programmer dalam mengambil data, membuat interface dan yang lainnya. Dikarenakan data – data tersebut sudah tersedia dari penyedia layanan API yang kemudian dapat diakses atau digunakan untuk pembuatan *software* kita dengan cara membuat *request* kepada API tersebut[35].

API ada yang bersifat *public* atau *private*, API *public* memungkinkan untuk diakses oleh banyak orang dan terbuka. Selain menguntungkan bagi *programmer* karena dapat diakses dengan terbuka, API *public* juga membuat si pemilik mendapatkan banyak publisitas, serta kode dari API itu sendiri dapat bertahan lama. Selain yang bersifat *public*, ada juga API yang bersifat *private* yang mana API ini hanya bisa diakses oleh pihak tertentu dan tidak dipublikasi secara luas[36].

2.7.1 Activity Recognition API

API ini digunakan sebagai penentu suatu aktivitas yang dilakukan oleh *user*. Seperti aktivitas berjalan, berkendara, berlari atau diam. Aktivitas tersebut dideteksi ketika perangkat menyala secara berkala dari data yang diambil oleh sensor *smartphone*. Hal tersebut memanfaatkan sensor daya rendah untuk menjaga penggunaan daya seminimal mungkin.

Aktivitas akan diterima beberapa detik setelah terdeteksi oleh sensor. Pada kasus penggunaan yang umum, aplikasi ingin memantau aktivitas di latar belakang dan melakukan tindakan saat aktivitas tertentu terdeteksi. Untuk melakukan ini tanpa memerlukan layanan yang selalu aktif di latar belakang yang memakan sumber daya, aktivitas yang terdeteksi akan dikirim melalui Intent. Aplikasi menetapkan *callback* PendingIntent (atau biasanya IntentService) yang akan dipanggil dengan Intent ketika aktivitas terdeteksi. Ketika aktivitas yang terdeteksi diam atau *still* maka pendeteksian akan menghentikan pelaporan aktivitas secara sementara agar menghemat penggunaan baterai. Dan akan melanjutkan pelaporan ketika aktivitas lain terdeteksi/kembali bergerak.

Parameter

client	GoogleApiClient yang sudah ada harus terhubung pada pemanggilan ini, yang biasanya didapatkan dengan memanggil <i>method</i> GoogleApiClient.connect()
detectionIntervalMillis	Waktu yang diinginkan antara deteksi aktivitas. Nilai yang lebih besar akan menghasilkan lebih sedikit deteksi

	aktivitas sekaligus meningkatkan masa pakai baterai. Nilai 0 akan menghasilkan deteksi aktivitas secepat mungkin
callbackIntent	PendingIntent dipakai pada requestActivityUpdates(GoogleApiClient, long, PendingIntent) atau sama saja dengan mendefinisikan PendingIntent.equals(Object)

Aktivitas yang dapat terdeteksi oleh API ini yaitu :

- a. DetectedActivity.IN_VEHICLE
- b. DetectedActivity.ON_FOOT
- c. DetectedActivity.RUNNING
- d. DetectedActivity.WALKING
- e. DetectedActivity.ON_BICYCLE
- f. DetectedActivity.STILL

Beberapa *method* yang terdapat pada API ini yang berkaitan dengan pendeteksian Aktivitas :

2. extractResult(Intent intent)

Mengekstraksi hasil dari aktivitas yang terdeteksi dari Intent.

3. getActivityConfidence(intActivityType)

Mengembalikan nilai *confidence* atau probabilitas dari aktivitas yang sedang *user* lakukan.

4. getElapsedRealtimeMillis()

Mengembalikan nilai berupa waktu setelah aktivitas terdeteksi dalam milidetik.

5. getMostProbableActivity()

Mengembalikan nilai berupa probabilitas tertinggi dari aktivitas yang sedang dilakukan *user*.

6. getProbableActivities()

Mengembalikan nilai berupa daftar aktivitas dan tingkat *confidence* dari aktivitas tersebut, berdasarkan dengan aktivitas yang dilakukan *user*.

7. getTime()

Mengembalikan nilai berupa waktu dari sebuah pendeteksian

8. hasResult()

Mengembalikan nilai *true* dari Intent yang mengandung hasil dari pendeteksian Aktivitas

2.7.2 FatSecret API

FatSecret merupakan API yang menyediakan berbagai fitur yang berkaitan dengan kesehatan, baik dari makanan ataupun untuk membuat solusi untuk diet, pemilihan nutrisi pada makanan dan manajemen berat badan. Dalam API ini terdapat informasi/variabel yang digunakan untuk mengembalikan nilai yang di *request*. Semisal untuk fitur *get food*, terdapat informasi dasar seperti *food_id*, *food_name*, *food_type*, *brand_name*, *food_url*.

Untuk menggunakan API ini, terlebih dahulu mendaftar menggunakan alamat email aktif, karena kode verifikasi pendaftaran

akan dikirim melalui email tersebut. Setelah pendaftaran selesai maka pendaftar memilih autentikasi yang disediakan. Disini saya memilih OAuth 2.0, setelah itu diberikan *application id* untuk nantinya dipakai ketika melakukan request kepada API ini.

Terdapat beberapa tipe data yang digunakan pada API ini, diantaranya :

- a. Boolean,
- b. Decimal
- c. Int
- d. JSON data
- e. Long
- f. String

Kemudian terdapat *unique identifiers* seperti *food_id*, *serving_id*, *food_entry_id*, dan *exercise_id*. Selain itu terdapat tipe nutrisi yang ditampilkan seperti:

- | | | |
|------------------------|------------------------|--------------|
| a. Calories | g. Monounsaturated_fat | m. Calcium |
| b. Carbohydrate | h. Cholesterol | n. Iron |
| c. Protein | i. Sodium | o. Vitamin_c |
| d. Fat | j. Added_sugar | p. Vitamin_a |
| e. Saturated_fat | k. Vitamin_d | |
| f. Polyunsaturated_fat | l. Trans_fat | |

2.8 Java

Java merupakan bahasa pemrograman yang bersifat umum /non-spesifik (*general purpose*)[12]. Java juga merupakan bahasan pemrograman berorientasi objek (OOP). Bahasa pemrograman Java terdiri dari *Java compiler*, *Java virtual machine*, dan *Java class libraries*[13]. Sintak pada java banyak yang mengadopsi dari bahasa pemrograman C dan C++ namun lebih disederhanakan lagi. Java yang merupakan bahasa pemrograman *multi-*

platform dan *multi-device* menjadikan Java dapat dijalankan hampir disemua komputer dan perangkat lainnya. Oleh karenanya Java memiliki slogann “*Tulis sekali, jalankan dimanapun*”[37].

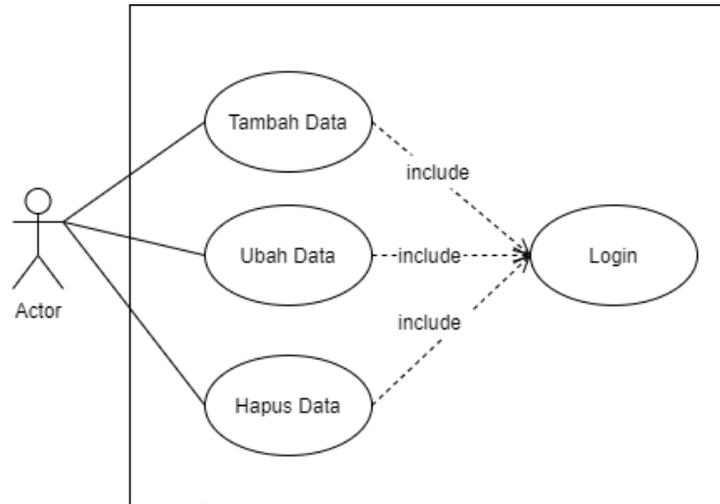
Java terdiri dari sekumpulan teknologi untuk membangun suatu perangkat lunak pada komputer *standalone*. Ketika akan memakai Java untuk membuat perangkat lunak pada komputer kita, diperlukan pemasangan *Java Development Kit* (JDK) sebagai sebuah *Compiler*[32].

2.9 UML

UML merupakan kolaborasi antara beberapa metode seperti Booch, OMT serta OOSE (Nugroho. 2009,4) yang digunakan untuk menganalisa perancangan sistem. Penggunaan dari UML ini sudah banyak sekali diterapkan dalam perancangan sebuah sistem dari perangkat lunak dengan metode OOP atau *Object Oriented Programming* [14]. Dalam membuat perancangan perangkat lunak ada beberapa diagram dari UML yang sering digunakan diantaranya:

a. Use Case Diagram

Use Case Diagram merupakan pemodelan yang mendeskripsikan tipikal interaksi antara user dari sebuah sistem dengan sistemnya sendiri. Melalui sebuah alur yang menggambarkan dan memodelkan perilaku tentang bagaimana sistem tersebut akan digunakan nantinya [15]. Use case dapat digunakan untuk menggambarkan fungsionalitas yang ada pada sistem yang akan dibangun[47]. Untuk contoh *use case diagram* dapat dilihat pada gambar 2.1.

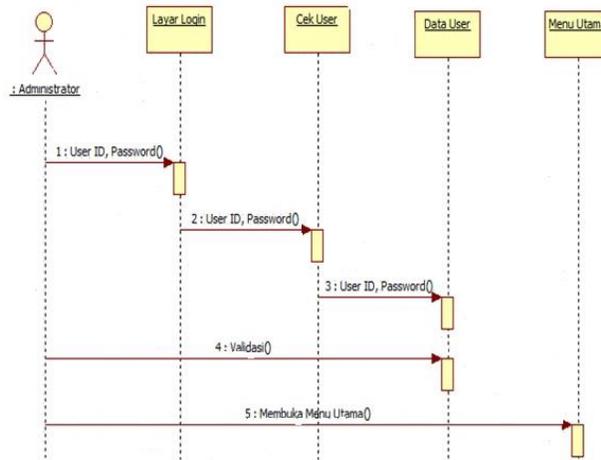


Gambar 0.1 Use Case Diagram

Sumber : <https://sites.google.com/site/rizkyluthpiyanaif15d120/uml>

b. Sequence Diagram

Sequence Diagram menggambarkan perilaku objek pada use case dengan menggambarkan hubungan antar subjek yang kegunaannya untuk menunjukkan pertukaran pesan yang dikirim dari satu objek menuju objek yang lain [16]. Diagram ini akan menunjukkan pesan ketika sebuah objek melakukan method atau tugasnya[47]. Untuk contoh *sequence diagram* dapat dilihat pada gambar 2.2.



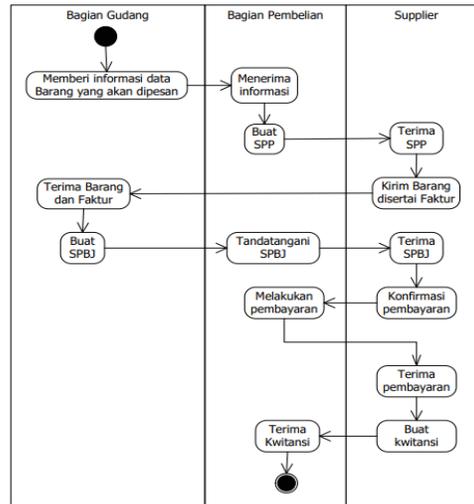
Gbr 8. Contoh Diagram Sequence

Gambar 0.2 Sequence Diagram

Sumber : <https://freezcha.files.wordpress.com/2010/04/diagram-sequence.jpg>

c. Activity Diagram

Activity Diagram menggambarkan *workflow* atau alur kerja dari sebuah sistem. Jadi pada diagram ini, aktivitas lain seperti aktivitas *user* tidak termasuk, karena hanya aktivitas dan alur kerja sistem yang di gambarkan[17]. Secara keseluruhan activity diagram menggambarkan bagaimana perilaku antara *user* dengan fungsionalitas dari sistem[47]. Untuk contoh *activity diagram* dapat dilihat pada gambar 2.3.

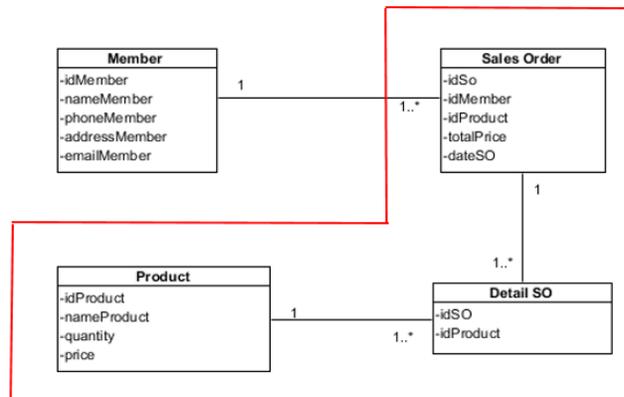


Gambar 0.3 Activity Diagram

Sumber : <https://www.dicoding.com/blog/wp-content/uploads/2020/04/intern-rendi-activity.png>

d. Class Diagram

Class Diagram merupakan pengembangan dari desain yang berorientasi objek. Dimana pada diagram ini menggambarkan struktur serta deskripsi dari kelas-kelas yang ada pada suatu sistem yang berelasi satu sama lain, baik itu relasi antar kelas seperti pewarisan, asosisasi, dan lain-lain[18]. Untuk contoh *class diagram* dapat dilihat pada gambar 2.4.



Gambar 0.4 Class Diagram

Sumber : <https://sis.binus.ac.id/2016/06/20/domain-class-diagram/>

2.10 Fatigue

Fatigue atau kelelahan adalah penurunan kondisi tubuh seseorang setelah melakukan aktivitas. Aktivitas tersebut bervariasi mulai dari aktivitas ringan seperti mengerjakan pekerjaan rumah, sampai aktivitas yang berat seperti olahraga ataupun bekerja [19]. Banyak faktor yang menyebabkan kelelahan mudah terjadi, salah satunya adalah faktor usia dimana semakin tua tubuh kita akan melemah dan kelelahan lebih mudah terjadi, selain itu tingkat kebugaran tubuh juga dapat menjadi salah satu faktor terjadinya kelelahan [20].

Selain itu faktor psikis, kecukupan gizi dan kondisi lingkungan dapat berpengaruh terhadap kelelahan ketika sedang beraktivitas. Kelelahan juga dapat terjadi karena bentuk perlindungan terhadap tubuh, agar tubuh tersebut tidak bekerja melebihi kapasitas atau kekuatannya. Tentunya dengan kelelahan yang muncul, seseorang tidak dapat melanjutkan aktivitasnya dengan normal. Karena konsentrasi pun kian menurun dikarenakan tubuh sudah tidak sanggup untuk tetap melanjutkan aktivitas tersebut [21].

Dalam memulihkan kembali tenaga setelah mengalami kelelahan, Rose et al. (2014) mengatakan bahwa pemulihan tidak terjadi begitu saja. Dikarenakan diperlukan waktu dan juga dapat dipengaruhi oleh variabel yang berbeda-beda.

Pemulihan dapat dipercepat dengan beberapa variabel seperti konsumsi gizi yang baik, kondisi lingkungan yang mendukung untuk kita melakukan istirahat dan juga kondisi kebugaran dari tubuh orang tersebut[22].

2.11 Metode Pembangunan Perangkat Lunak *Waterfall*

Menurut Rosa dan M Shalahuddin (2013:28) Metode *Waterfall* sering disebut model sekuensial linier. Metode ini juga menyediakan penelekat alur pembangunan perangkat lunak yang terurut dimulai dari analisis hingga ke tahap *maintenance*[23]. Penggunaan metode ini lebih praktis dalam implementasi nya terhadap pembuatan sistem karena lebih terkontrol dan pemeliharaan menjadi lebih mudah[24]. Berikut merupakan tahapan metode *Waterfall* [34] :

a. *Analisa Kebutuhan Software*

Tahapan ini dilakukan untuk menganalisa kebutuhan yang diperlukan dalam membuat sebuah perangkat lunak agar ditemukan sebuah solusi yang akan digunakan sebagai proses komputerisasi pada sistem Aplikasi yang dibuat

b. *Desain*

Dalam tahapan ini semua perancangan baik dari segi frontend dan backend dilakukan. Dimaksudkan untuk menjelaskan secara lebih rinci mengenai rancangan terhadap perangkat lunak yang akan dibangun.

c. *Implementasi*

Rancangan yang telah dibuat pada tahapan sebelumnya, ditranslasikan menjadi bentuk program pada tahapan ini. Untuk penggunaan IDE ataupun bahasa pemrograman menyesuaikan terhadap kebutuhan dan rencana awal yang telah disetujui oleh tim *developer*.

d. Pengujian

Pada tahapan ini perangkat lunak akan diuji agar terhindar dari error yang nantinya hasil akhir dari pembuatan perangkat lunak sesuai dengan rencana awal.

e. *Support*

Bisa disebut juga sebagai tahapan pemeliharaan, dimana ketika sistem diperlukan pembaharuan untuk menyesuaikan dengan kebutuhan *user*.

2.12 Tidur

Tidur merupakan suatu keadaan dimana kita tidak sadar. Meski dalam kondisi tidak sadar, ketika tertidur manusia masih dapat merasakan rangsangan yang terjadi disekitarnya. Seperti ketika ada seseorang yang mencubit ketika kita sedang tidur, tentu saja tubuh akan bereaksi terhadap hal tersebut atau ketika kita merasa tidak nyaman akibat suhu ruangan yang panas. Bahkan ketika kita mengalami gangguan dari dalam tubuh itu sendiri, seperti misalnya rasa ingin buang air kecil atau sakit perut. Tentu saja reaksi akan rangsangan tersebut lebih lemah dibandingkan ketika sedang tidak tidur. Tidur memiliki 2 Kategori [38] yaitu :

1. Tidur NREM

Kategori ini bisa disebut juga sebagai *deep sleep*, dimana seseorang akan merasa nyaman karena pada tahap ini mereka cenderung rileks dikarenakan kinerja tubuh seperti metabolisme dan kecepatan pernapasan menurun. Ketika metabolisme dan kecepatan pernapasan menurun, tentu detak jantung akan melemah/menurun.

2. Tidur REM

Ketika kita tertidur seringkali kita merasa tidak nyaman, mulai dari detak jantung yang semakin meningkat, kondisi tubuh menjadi tegang dan beberapa kondisi lain yang tidak nyaman. Gejala tersebut

menjadi tanda bahwa kita sedang tidur, kita berada pada kategori tidur REM. Dimana tidur menjadi tidak nyaman, mudah terganggu dengan rangsangan dari lingkungan, yang menyebabkan tidur menjadi tidak lelap.

Lama seseorang tidur dipengaruhi oleh usia serta jenis kelamin, untuk usia 1 – 18 Bulan (kategori Bayi) memiliki jam tidur normal berkisar 12-14 jam/hari dan untuk bayi dengan usia 1 – 3 tahun memiliki jam tidur normal 10 – 12 jam/hari. Rentang usia 3 – 6 tahun (kategori Anak-anak) memiliki jam tidur normal berkisar \pm 11 jam/hari. Rentang usia 6 – 12 tahun (kategori usia sekolah) memiliki jam tidur normal \pm 10 jam/hari. Rentang usia 12 – 18 Tahun (kategori remaja) memiliki jam tidur normal berkisar 7-8,5 jam/hari. Rentang usia 18 – 40 tahun (kategori dewasa muda) memiliki jam tidur normal berkisar 7-8 jam/ hari. Rentang usia 40 – 60 tahun (kategori dewasa menengah) memiliki jam tidur yang sama dengan dewasa muda yaitu 7-8 jam/ hari. Dan untuk rentang usia >60 tahun (kategori dewasa tua) memiliki jam tidur normal berkisar 6 jam/hari[39]. Untuk pembagian rentang durasi tidur berdasarkan usia dapat dilihat pada tabel 2.1.

Tabel 0.1 Tabel Jam Tidur

Rentang Usia	Kategori	Jam Tidur/Hari
1 – 4 Tahun	Bayi	\pm 12 Jam
5 – 11 Tahun	Anak-anak	\pm 11 Jam
12 – 18 Tahun	Remaja	7 – 8.5 Jam
19 – 60 Tahun	Dewasa	7 – 8 Jam
>60 Tahun	Lansia	\pm 6 Jam

2.13 Detak Jantung

Detak jantung merupakan tekanan yang dialirkan darah dalam bentuk gelombang yang dipompa dari jantung menuju semua bagian-bagian tubuh. Intensitas gelombang yang dialirkan akan berbeda tergantung dari usia dan aktivitas yang dilakukan oleh manusia itu sendiri. Selain itu ada faktor luar yang mempengaruhi seperti konsumsi obat-obatan, dan lainnya. Kategori usia dalam mempengaruhi detak jantung terbagi menjadi 3 yaitu bayi (100-140), anak-anak (80-100) dan dewasa 60-80[40]. Perubahan detak jantung ketika berolahraga bisa sampai 60-80 persen dari batas maksimum detak jantung. Untuk jelasnya dapat dilihat pada tabel 2.2

Tabel 0.2 Tabel Detak Jantung

Kategori	BPM		
	Normal (Diam, duduk, dll)	Tinggi	Rendah
Bayi	100 – 140	>140	<100
Anak – anak	80 – 100	>120	<80
Dewasa	60 - 80	>110	<60

2.14 Kebutuhan Kalori

Kebutuhan kalori yang dibutuhkan manusia dewasa berkisar 2000 –2500. Sekitar 20% dari energi kita dipakai untuk metabolisme otak, sisanya terpakai untuk aktivitas yang dilakukan, sirkulasi darah, pencernaan dan pernafasan [41]. Jika asupan kalori kurang, tentu saja badan akan terasa lebih lemah dari biasanya, itu karena energi yang terdapat dalam tubuh juga tidak banyak. Untuk menyesuaikan asupan kalori pada tubuh, selain melakukan olahraga

untuk kebugaran, asupan makanan juga penting agar energi dapat terisi kembali setelah dikuras oleh aktivitas sehari-hari. Salah satunya dengan konsumsi karbohidrat yang mana dapat memulihkan tenaga lebih cepat karena kadar insulin akan meningkat secara signifikan. Hal itu disebabkan karena karbohidrat terserap kedalam tubuh dalam bentuk gula atau glukosa yang cepat masuk ke aliran darah.

2.15 Indeks Massa Tubuh

Indeks massa tubuh biasanya digunakan untuk menentukan apakah seseorang kekurangan berat badan atau berlebihan berat badan. Indeks Massa Tubuh diukur dengan cara membagi berat badan dengan tinggi badan (m)². Selain itu IMT juga dapat mempengaruhi detak jantung manusia ketika kondisi istirahat. Jika berat badan semakin tinggi, maka detak jantung ketika istirahat juga akan semakin tinggi dan jika berat badan semakin rendah maka detak jantung ketika istirahat juga semakin rendah[46].

Tabel 0.3 Tabel Indeks Massa Tubuh

IMT	Status
<18,5	Berat badan kurang
18,5 – 25	Berat badan normal
25,1 – 27	Kelebihan berat badan
>27	Obesitas