

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Adapun penelitian terkait yang akan di gunakan sebagai sumber acuan yang relevan.

Menurut Hafiz Fauzi (2018) Fenomena diet saat ini sudah menjadi gaya hidup baginsebagian kalangan, berbagai macam alasan untuk menjalankan diet dari faktor kesehatan sampai menjaga penampilan. Masalah yang timbul adalah ketika ingin memilih cara yang tepat untuk melakukan diet yang sesuai bagi kesehatan. Salah satu cara yang tepat untuk dilakukan adalah diet berdasarakan golongan darah. Diet ini masih tergolong aman digunakan dan telah banyak buku atau panduan untuk melakukan diet berdasarakan golongan darah. Faktor kesibukan dari peserta diet tidak memungkinkan untuk memantau program diet yang dilakukan, maka kebutuhan sebuah aplikasi dapat menjadi solusi dalam memandu program diet.

Menurut Adhitya Irama Putra dan Suryanto (2016) Banyaknya masyarakat yang belum mengetahui pemilihan nutrisi yang baik dan menganggap konsep diet hanya untuk menurunkan berat badan serta kebiasaan pola aktivitas. Jika seseorang memiliki tingkat pengetahuan yang baik mengenai diet, diharapkan dalam praktiknya juga akan baik, akan tetapi jika seseorang memiliki tingkat pengetahuan mengenai diet yang buruk, dapat diasumsikan bahwa dalam praktiknya pun akan buruk, sehingga ada kemungkinan orang tersebut sulit

mendapatkan Indeks Massa Tubuh (IMT) dengan kategori normal. Tujuan penelitian ini adalah untuk mengetahui hubungan tingkat pengetahuan diet dengan IMT member fitness center di GMC. Penelitian ini merupakan penelitian korelatif. Pengambilan data menggunakan angket untuk mengetahui pengetahuan tentang diet dan metode antropometri dengan TB dan BB untuk mengetahui IMT.

II.2. Landasan Teori

II.2.1. Aplikasi

Istilah aplikasi berasal dari bahasa Inggris *application* yang berarti penerapan, lamaran ataupun penggunaan. Sedangkan secara umum, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta jasa pengguna aplikasi lain yang dapat digunakan oleh pengguna yang akan dituju. Menurut kamus komputer eksekutif, pengertian aplikasi merupakan pemecahan masalah yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Aplikasi biasanya berupa perangkat lunak yang berbentuk *Software* yang berisi kesatuan perintah atau program yang dibuat untuk melaksanakan sebuah pekerjaan yang diinginkan.

11.2.2. Java

Bahasa pemrograman Java pada awalnya dibuat oleh James Gosling pada tahun 1995 sebagai bagian dari Sun Microsystems Java Platform”. Sintaks Java banyak diturunkan dari C dan C++ tetapi lebih sederhana, ketat dan mempunyai akses ke OS yang lebih terbatas. Hal ini karena Java ditujukan sebagai bahasa pemrograman yang cukup sederhana untuk dipelajari dan mudah dibaca.

Java ditujukan agar bisa “ditulis sekali, bisa jalan di manapun”. Sekarang ini Java adalah bahasa pemrograman paling populer dan paling banyak digunakan untuk membuat aplikasi baik aplikasi di *embedded system, mobile, desktop hingga web application*. Java mempunyai empat prinsip penting yang dijadikan sebagai tujuannya, keempat prinsip ini adalah :

1. Java harus sederhana, *object oriented* dan mudah dimengerti.
2. Java harus kuat dan aman.
3. Java harus netral terhadap arsitektur system (OS, *processor*) dan bisa jalan di manapun.
4. Java harus bisa dijalankan dengan kinerja yang tinggi.
5. Java harus *interpreted, threaded* dan dinamis.

Dengan kelima prinsip di atas, aplikasi java mempunyai popularitas yang sangat tinggi terutama di dunia *enterprise application*. Dimana semua prinsip di atas sangat cocok untuk jenis aplikasi ini. Industri yang mempunyai *budget* tinggi untuk IT

seperti perbankan dan telekomunikasi menggunakan Java secara ekstensif. Banyak aplikasi dengan skala raksasa dibangun menggunakan platform java.

II.2.3. Android

Android adalah suatu sistem operasi yang berbasis *Linux* dengan perangkat *Mobile device*, *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi-aplikasi baru untuk membantu kebutuhan manusia. Awal terdirinya android dimulai dari *Google Inc* yang membeli *Android inc* pendatang bat dari perangkat lunak untuk ponsel/*smartphone*. Semakin banyak *smartphone* dan PC Tablet menggunakan sistem operasi dengan versi yang berbdeda. Semakin tinggi versinya maka fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi *Android* adalah *HTC Dream* yang dirilis pada tanggal 22 Oktober 2008. Perkembangan sistem *Android* berkembang sesuai versi keluarnya adapun versi *Android* dimulai dari (Muhamad Zaelani ; 2017: 5)

II.2.4. Eclipse

Menurut Murya (2016:17) “Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform (platform-independent)*”. Berikut ini adalah sifat dari Eclipse:

1. Multi-platform

Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

2. Multi-language

Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl PHP, dan lain sebagainya.

3. Multi-role

Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

II.2.5. Android SDK

Menurut Murya (2016:15-16) “Android SDK (*Software Development Kit*) adalah tool API (*Application Programming Interface*) yang diperlukan untuk memulai mengembangkan aplikasi pada *platform* Android atau dapat dikatakan perangkat lunak yang digunakan dalam pengembangan aplikasi pada Android”.

II.2.6. Android Development Tools

“Android *Development Tools* atau lebih dikenal dengan sebutan ADT merupakan *plugin* pada Eclipse yang digunakan untuk membuat *project* berbasis Android”. ADT wajib di-*install* sehingga Eclipse yang sudah ter-*install* dikomputer dapat digunakan sebagai tempat atau media untuk melakukan pemrograman Android.

II.2.7. Basis Data (*Database*)

Pangkalan data atau basis data (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system, DBMS*). Sistem basis data dipelajari dalam ilmu informasi. Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Model yang umum digunakan sekarang adalah model relasional, yang mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Model yang lain seperti model

hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel (Neni Purwati dan Hendra Kurniawan, 2015 : 50).

II.2.8. MySQL

MySQL (*My Structure Query Language*) merupakan sebuah program pembuat database yang bersifat *Open Source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik *Windows* maupun *linux*. *MySQL* juga merupakan sebuah perangkat lunak sistem manajemen basis data *SQL* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi multi *user*. *MySQL* juga sering dikenal dengan nama sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam table yang terpisah. Tabel -table tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa table dalam suatu permintaan. Untuk administrasi *database*, seperti pembuatan *database*, pembuatan tabel, dan sebagainya dapat digunakan aplikasi berbasis web seperti *PHP MyAdmin* dengan aplikasi *XAMPP*. (Saipul Anwar, 2016)

II.2.9. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi

apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat *download* langsung dari *web* resminya. (Randi V Palit, 2017 : 2).

II.2.10. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi.

Unified Modeling Language (UML) biasa digunakan untuk :


- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.

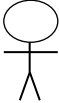


- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *fungsi* dengan *stereotypes*. (Ommi Alfina dan Fitriana Harahap : 2019)

II.2.10.1. Use Case Diagram

Use case diagram merupakan pemodelan untuk sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. Simbol Use Case

Gambar	Keterangan	Deskripsi
	<i>Use case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .

	Aktor	Sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi	Penghubung antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.


----->	<i>Include</i>	Merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
<-----	<i>Extend</i>	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.



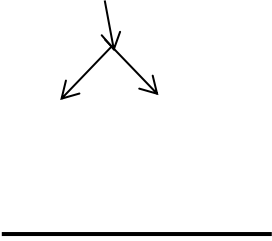
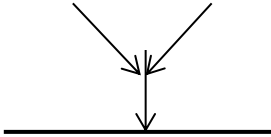
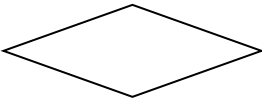

(Sumber : Ade Hendini, 2016 : 109)

II.2.10.2. Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.2. Simbol Activity Diagram

Gambar	Keterangan	Deskripsi
	<i>Start point</i>	Diletakkan pada pojok kiri atas dan merupakan awal aktifitas.

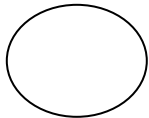
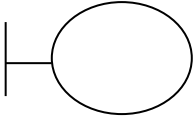
	<i>End point</i>	Akhir aktifitas.
	<i>Activites</i>	Menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan).	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan)	Digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i>	Pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

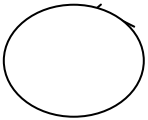

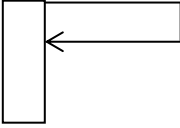


(Sumber : Ade Hendini, 2016 : 109)

II.2.10.3. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan	Deskripsi
	<i>Entity Class</i>	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.

	<i>Control class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i>	Simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Ade Hendini, 2016 : 110)

II.2.10.4. *Class Diagram*

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalitation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality* (Ade Hendini, 2016 : 111).

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Ade Hendini, 2016 : 110)