**PEMROGRAMAN MOBILE**

**Aplikasi *”Ideal Weight”***

Dosen Pengampu :

I Putu Agus Eka Pratama, ST. MT.



Oleh:

Pande Bagus Narendra Mahaputra 1404505075

I Kadek Dwi Rendika 1404505077

**JURUSAN TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS UDAYANA**

**2016**

# BAB I

# PENDAHULUAN

* 1. **Latar Belakang**

Dewasa ini, produk manakan cepat saji makin banyak diproduksi, makin banyak tempat makan yang menyediakan makanan cepat saji, dan makin banyak individu yang memakan makanan cepat saji tanpa menghiraukan pola hidup sehat dari individu tersebut. Semakin sibuk dan malas individu juga makin membuatnya jarang melakukan aktifitas yang menyehatkan seperti olahraga. Padahal, dengan melakukan olah raga yang rutin, badan kita akan terasa semakin bugar dan berat badan kita akan dapat dikurangi sehingga tubuh ideal dapat diraih dengan mudah.

Oleh karena itu, penulis berencana membuat aplikasi yang dapat menjadi “partner” individu dalam menerapkan pola hidup sehat serta dalam proses penurunan berat badan. Sehingga penulis berinisiatif membuat sebuah aplikasi yang nantinya diharapkan dapat membantu meningkatkan pola hidup sehat semua orang.

* 1. **Rumusan Masalah**

Dari pemaparan latar belakang diatas, dapat dirumuskan permasalahan sebagai berikut :

* + 1. Bagaimanakah UML dari aplikasi yang akan dibuat?
    2. Bagaimana *prototype* (Coding & GUI) dari aplikasi yang akan dibuat?

**BAB II**

**LANDASAN TEORI**

### 

### Object Oriented

Pemrograman berorientasi objek atau Object Oriented Programming disingkat OOP merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, danmengirim pesan ke objek lainnya.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat. Beberapa karakteristik dari pemograman berbasis objek dapat dijabarkan sebagai berikut.

1. Enkapsulasi (Pembungkusan)

Enkapsulasi adalah pelindung program dan data yang sedang diolah. Enkapsulasi mendefinisikan perilaku dan melindungi program dan data agar tidak diakses secara sembarangan oleh program lain.

Dalam pemograman, dasar enkapsulasi adalah class. Anda membuat suatu class yang menyatakan bahwa variable atau method sebuah class tidak dapat diakses oleh class lain dengan menjadikan class tersebut private, atau menjadikan class tersebut protected – yaitu hanya bisa diakses oleh turunannya, atau menjadikan class tersebut public–yaitu bisa diakses oleh sembarang class.

1. Inheritance (Pewarisan)

Inheritance (Pewarisan) dapat diambil perumpamaan dengan benda. Objek-objek yang berada di sekitar kita adalah objek-objek yang saling terhubung secara hirarkis. Misalnya, Lingkaran dan Bujur Sangkar adalah turunan dari bentuk 2D dan bentuk 2D adalah turunan dari objek gambar lingkaran dan bujur sangkar mewarisi (inherit) sifat-sifat dari bentuk 2D, juga mewarisi sifat-sifat dari objek gambar lingkaran dan bujur sangkar dapat dikatakan subclass dari bentuk 2D. Bentuk 3D adalah super class dari bola dan piramida, dan seterusnya.

1. Polimorfisme (Kebanyakrupaan)

Disebut juga kemampuan suatu objek untuk mengungkapkan banyak hal melalui satu cara yang sama. Contohnya adalah : misalkan kita membuat kelas A, kemudian di turunkan lagi menjadi kelas B, C dan D. Dengan kemampuan polimorfisme, kita dapat menjalankan methode(sifat) yang terdapat pada kelas B, C dan D hanya dari objek yang di instansikan dengan kelas A. Polimorfisme bisa disebut juga : dynamic binding, late binding, atau runtime binding.

1. Abstraction (Abstraksi)

Kurang lebih yang di maksud adalah pengabstrakan atau penyembunyian kerumitan suatu proses. Contohnya adalah kita sebagai pengguna sepeda motor tidak perlu berpikir bahwa motor adalah kupulan dari puluhan atau ribuan bagianbagian kecil komponen-komponen. Kita hanya perlu berpikir bahwa motor adalah sebuah objek yang telah memiliki sifat-sifat spesifik, dan motor tersebut dapat kita gunakan sebagai alat transportasi. Sehingga kita hanya perlu tahu cara pakainya saja tanpa perlu tahu kerumitan proses yang ada di dalamnya. Dalam hal ini, insinyur mesin telah menyembunyikan proses-proses yang rumit.

Sedangkan contoh abstraksi dalam pemrograman java misalnya, kita tidak perlu tahu secara detail bagaimana cara menggunakan objek tombol yang sudah ada (baik data/properti ataupun method-nya). Dalam kasus ini, pembuat objek telah mengabstraksi semua proses yang berkaitan dengan pembuatan tombol.

### Bahasa Pemrograman Java

Java adalah suatu teknologi di dunia software komputer, yang merupakan suatu bahasa pemrograman, dan sekaligus suatu platform. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi. Java mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++. Java merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan. Sebagai bahasa pemrograman Java dirancang menjadi handal dan aman. Java juga dirancang agar dapat dijalankan di semua platform. Dan juga dirancang untuk menghasilkan aplikasi – aplikasi dengan performansi yang terbaik, seperti aplikasi database Oracle 8i/9i yang core-nya dibangun menggunakan bahasa pemrograman Java. Sedangkan Java bersifat neutral architecture, karena Java Compiler yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras yang disebut sebagai Java Byte code.

Platform Java terdiri atas dua bagian utama, yaitu:

1. Java Virtual Machine (JVM).
2. Java Application Programming Interface (Java API).

Sun membagi arsitektur Java membagi tiga bagian, yaitu:

1. Enterprise Java (J2EE) untuk aplikasi berbasis web, aplikasi sistem tersebar dengan beraneka ragam klien dengan kompleksitas yang tinggi. Merupakansupersetdari Standar Java
2. Standar Java (J2SE), ini adalah yang biasa dikenal sebagai bahasa Java.
3. Micro Java (J2ME) merupakan subset dari J2SE dan salah satu aplikasinya yang banyak dipakai adalah untuk wireless device / mobile device.

### Android

Pengertian Android adalah sistem operasi berbasis Linux yang dipergunakan sebagai pengelola sumber daya perangkat keras, baik untuk ponsel, smartphone dan juga PC tablet. Secara umum Android adalah platform yang terbuka (Open Source) bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai piranti bergerak.

Antarmuka pengguna Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk memanipulasi obyek di layar.

Android memungkinkan penggunanya untuk memasang aplikasi pihak ketiga, baik yang diperoleh dari toko aplikasi seperti Google Play, Amazon Appstore, ataupun dengan mengunduh dan memasang berkas APK dari situs pihak ketiga. Di Google Play, pengguna bisa menjelajah, mengunduh, dan memperbarui aplikasi yang diterbitkan oleh Google dan pengembang pihak ketiga, sesuai dengan persyaratan kompatibilitas Google.

Aplikasi Android dikembangkan dalam bahasa pemrograman Java dengan menggunakan kit pengembangan perangkat lunak Android (SDK). SDK ini terdiri dari seperangkat perkakas pengembangan, termasuk debugger, perpustakaan perangkat lunak, emulator handset yang berbasis QEMU, dokumentasi, kode sampel, dan tutorial. Didukung secara resmi oleh lingkungan pengembangan terpadu (IDE) Eclipse, yang menggunakan plugin Android Development Tools (ADT).

### Android Studio

Android Studio merupakan lingkungan pengembangan Android baru berdasarkan IntelliJ IDEA. Mirip dengan Eclipse dengan ADT Plugin, Android Studio menyediakan alat pengembang terintegrasi untuk pengembangan dan debugging. Fitur yang dimiliki Android Studio:

1. Berbasis Gradle.
2. Android-spesifik refactoring dan perbaikan yang cepat.
3. Alat Lint untuk menangkap kinerja, kegunaan, versi kompatibilitas dan masalah lainnya.
4. ProGuard dan app-signature.
5. Wizard untuk design dan membuat komponen-komponen umum Sebuah layout editor yang memungkinkan untuk drag-and-drop komponen UI, pratinjau layout pada beberapa konfigurasi layar, dan banyak lagi.
6. Built-in dukungan untuk Google Cloud platform , sehingga mudah untuk mengintegrasikan Google Cloud Messaging dan App Engine sebagai komponen server-side.

### UML

UML (Unified Modeling Language) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis Object Oriented. UML sendiri juga memberikan standar penulisan sebuah sistem blueprint, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistemsoftware.

UML sendiri terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 9 diagram, yaitu; use-case, class, object, state, sequence, collaboration, activity, component, dan deployment diagram.

1. Use Case Diagram, menggambarkan sekelompok use cases dan aktor yang disertai dengan hubungan diantaranya diagram use cases ini menjelaskan dan menerangkan kebutuhan yang diinginkan/dikehendaki pengguna, serta sangat berguna dalam menentukan struktur organisasi dan model dari pada sebuah sistem.
2. Class Diagram, yang memperlihatkan struktur statis dari kelas aktual didalam sistem.
3. Object Diagram, yang merupakan varian dari kelas diagram yang memperlihatkan lebih detail banyaknya obyek yang mengintantiasi (instances) kelas.
4. State Diagram, yang memperliatkan semua keadaan (state) yang dapat dimiliki oleh kelas danevent yang dapat merubah keadaan tersebut.
5. Sequence Diagram, yang memperlihatkan kolaborasi dinamik antara objek-objek dengan suatu urutan pesan (a sequence of message) antar objek tersebut.
6. Collaboration Diagram, yang memperlihatkan kolaborasi dinamik antarobjek tanpa memperhatikan aspek waktu.
7. Activity Diagram, yang memperlihatkan aliran urutan aktifitas.
8. Component Diagram, yang memperlihatkan struktur fisik dari source code dalam terminology code components. Komponen berisi informasi tentang logical class dapat berupa komponen source code, komponen biner atau komponen yang dapat dieksekusi.
9. Deployment Diagram, yang memperlihatkan arsitektur fisik dari hardware dan software pada sistem.
10. **Perhitungan Berat Badan Ideal**

Aplikasi ini akan menghasilkan indeks yang nantinya akan menentukan apakah user gemuk, kurus, atau ideal berdasarkan rumus BMI (*Body Mass Index*).

*BMI = Berat Badan / (Tinggi Badan \* Tinggi Badan)*

Berat badan yang dimasukkan adalah dalam satuan kilogram, dan tinggi badan yang dimasukkan dalam satuan meter.

Kategori BMI untuk laki-laki yaitu :

|  |  |
| --- | --- |
| Nilai BMI | Kategori |
| < 17 | Kurus |
| 17-23 | Normal |
| 24-27 | Kegemukan |
| >27 | Obesitas |

Sedangkan, kategori BMI untuk perempuan yaitu :

|  |  |
| --- | --- |
| Nilai BMI | Kategori |
| < 18 | Kurus |
| 18-25 | Normal |
| 26-27 | Kegemukan |
| >27 | Obesitas |

**BAB III**

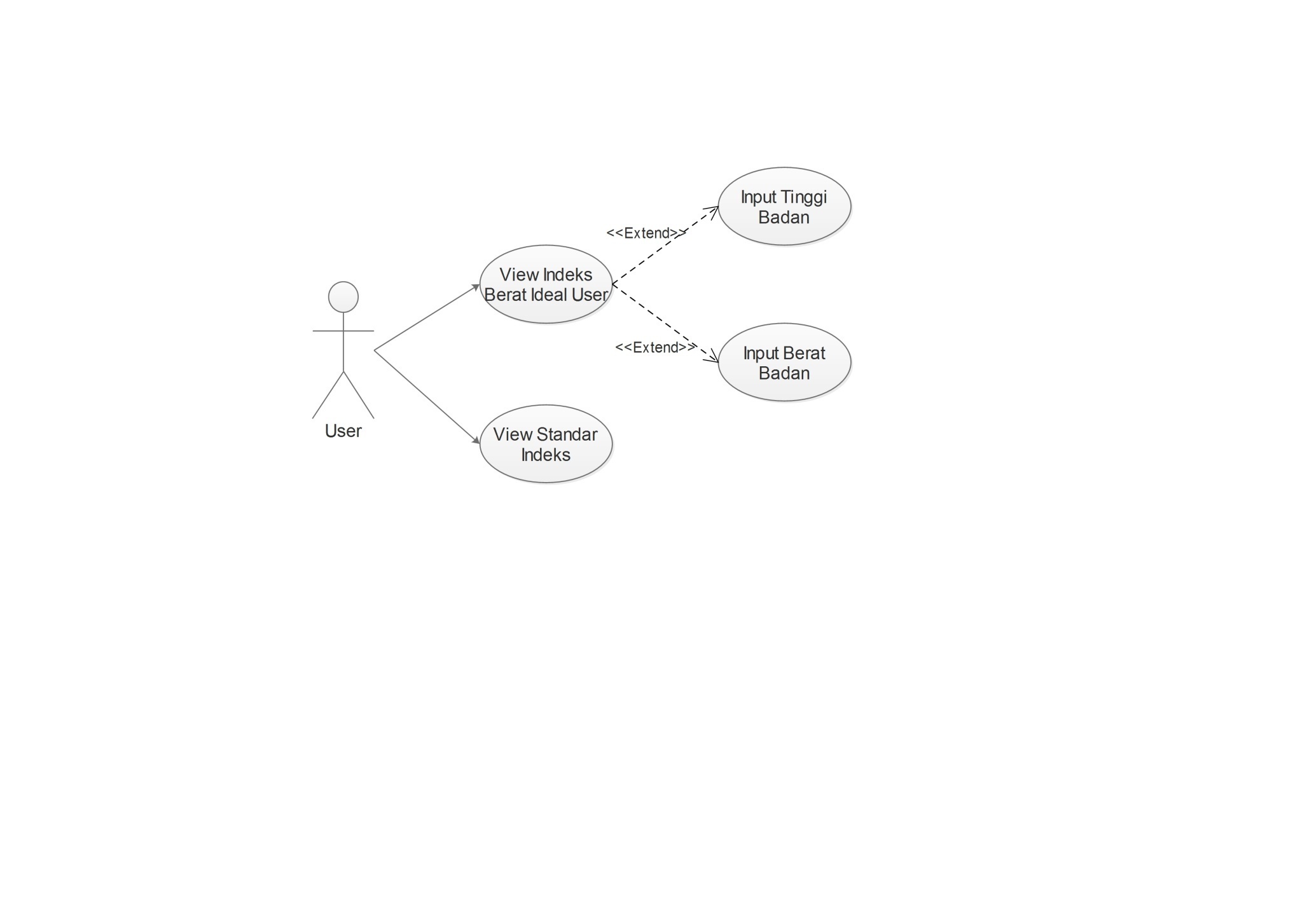
**PEMBAHASAN**

**3.1 Gambaran Umum Aplikasi**

Aplikasi yang akan dibuat yaitu aplikasi yang dapat digunakan untuk menghitung indeks berat badan dari user yang menggunakan. Nama aplikasi yang dibuat adalah *Ideal Weight*. Aplikasi ini memungkinkan user untuk mengetahui indeks berat badan dirinya berdasarkan berat badan dan tinggi badan.

**3.2 Use Case**

Use case yang telah dibuat untuk merancang aplikasi *Ideal Weight* dapat dilihat sebagai berikut.

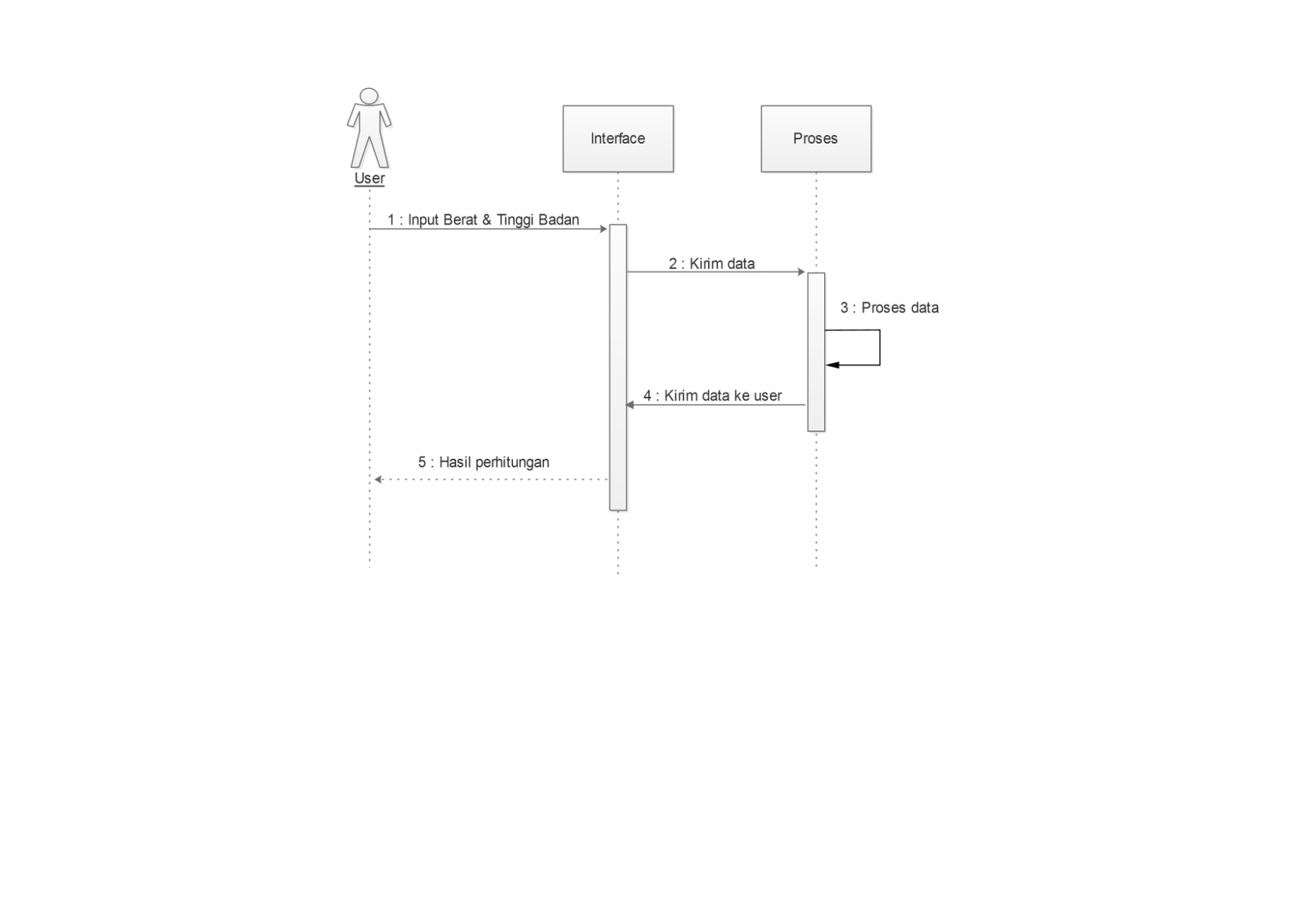


**Gambar 3.1** *Use Case Ideal Weight*

Dari use case diatas, dapat diihat bahwa user dapat mengetahui indeks berat badan idealnya jika user sudah memasukkan tinggi badan serta berat badan user tersebut. User juga dapat melihat standar indeks untuk menentukan apakah ia kurus, atau gemuk.

**3.3 Sequence Diagram**

Sequence diagram yang dibuat berdasarkan pemaparan use case diatas yaitu :



**Gambar 3.2** *Sequence Diagram Ideal Weight*

Pada Sequence Diagram diatas, dapat dijelaskan bahwa user harus memasukkan data berat badan dan tinggi badan user, selanjutnya data tersebut akan diproses sehingga menghasilkan indeks berat ideal user, selanjutnya data indeks akan ditampilkan di interface sehingga user dapat melihat.

**3.4 Class Diagram**

Class Diagram yang dapat dibuat untuk aplikasi *Ideal Weight* yaitu sebagai berikut.



**Gambar 3.3** *Class Diagram Ideal Weight*

Berdasarkan gambar diatas, dapat dijelaskan bahwa data yang dimasukkan adalah tinggi badan dan berat badan yang keduanya memiliki tipe data integer. Selanjutnya data akan dibawa ke entitas Berat\_Ideal untuk menghasilkan indeks berat ideal berdasarkan data yang dimasukkan user.

**3.5 Coding**

Coding yang dimiliki aplikasi Ideal Weight dalam aplikasi Android Studio yaitu sebagai berikut.

* + 1. Coding dalam MainActivity.java

|  |
| --- |
| package com.rzproject.idealweight;  import android.content.Intent;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  public class MainActivity extends AppCompatActivity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  Button start=(Button)findViewById(R.id.start);  start.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent i =new Intent(getApplicationContext(),Start.class);  startActivity(i);  }  });  Button about=(Button)findViewById(R.id.about);  about.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent j =new Intent(getApplicationContext(),About.class);  startActivity(j);  }  });  Button exit=(Button)findViewById(R.id.exit);  exit.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  moveTaskToBack(true);  }  });  }  } |

**Code 3.1** Code dalam MainActivity.java

* + 1. Coding dalam Activity\_main.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  tools:context="com.rzproject.idealweight.MainActivity">  <ImageView  android:layout\_width="fill\_parent"  android:layout\_height="150dp"  android:src="@drawable/logoname"  android:id="@+id/layout" />  <TextView  android:id="@+id/desc"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center"  android:text="Android Software to Count Your Ideal Weight"  android:textSize="20sp"  android:layout\_marginTop="30dp"  android:layout\_below="@id/layout" />  <Button  android:id="@+id/start"  android:layout\_width="fill\_parent"  android:layout\_height="70dp"  android:text="Mulai Menghitung"  android:textSize="20sp"  android:layout\_marginBottom="20dp"  android:layout\_above="@+id/about" />  <Button  android:id="@+id/about"  android:layout\_width="160dp"  android:layout\_height="wrap\_content"  android:text="Tentang Aplikasi"  android:layout\_alignParentBottom="true" />  <Button  android:id="@+id/exit"  android:layout\_width="160dp"  android:layout\_height="wrap\_content"  android:text="Keluar"  android:layout\_alignParentRight="true"  android:layout\_alignParentBottom="true"/>  </RelativeLayout> |

**Code 3.2** Code dalam Activity\_main.xml

* + 1. Coding dalam Start.java

|  |
| --- |
| package com.rzproject.idealweight;  import android.content.Intent;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.TextView;  import android.widget.Toast;  public class Start extends AppCompatActivity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_start);  Button ulang = (Button)findViewById(R.id.ulang);  Button back = (Button)findViewById(R.id.back);  final EditText tinggi = (EditText) findViewById(R.id.tinggi);  final EditText berat = (EditText) findViewById(R.id.berat);  final Button hitung = (Button) findViewById(R.id.Hitung);  final TextView hasil = (TextView) findViewById(R.id.Hasil);  back.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  startActivity(new Intent(Start.this,MainActivity.class));  finish();  }  });  ulang.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  tinggi.setText("");  berat.setText("");  hasil.setText("");  }  });  hitung.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if (tinggi.length() == 0 && berat.length() == 0) {  Toast.makeText(getApplication(), "Tinggi dan Berat Tidak Boleh Kosong ", Toast.LENGTH\_SHORT).show();  } else if (tinggi.length() == 0) {  Toast.makeText(getApplication(), "Tinggi Tidak Boleh Kosong ", Toast.LENGTH\_SHORT).show();  } else if (berat.length() == 0) {  Toast.makeText(getApplication(), "Berat Tidak Boleh Kosong ", Toast.LENGTH\_SHORT).show();  } else {  String isiTinggi = tinggi.getText().toString();  String isiBerat = berat.getText().toString();  float ntinggi = Float.parseFloat(isiTinggi);  float nberat = Float.parseFloat(isiBerat);  float iw = IdealWeight(ntinggi, nberat);  String output = String.valueOf(iw);  hasil.setText(output.toString());  }  }  });  }  public float IdealWeight(float ti, float be){  return (float) be / ((ti/100)\*(ti/100));  }  } |

**Code 3.3** Code dalam Start.java

* + 1. Coding dalam Activity\_start.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  tools:context="com.rzproject.idealweight.Start">  <ImageView  android:layout\_width="match\_parent"  android:layout\_height="100dp"  android:src="@drawable/logoname"  android:layout\_alignParentTop="true"  android:id="@+id/logoLayout"/>  <RelativeLayout  android:id="@+id/relative"  android:layout\_marginTop="15dp"  android:orientation="horizontal"  android:layout\_width="fill\_parent"  android:layout\_height="60dp"  android:background="@drawable/kolom"  android:layout\_below="@id/logoLayout">  <LinearLayout  android:id="@+id/linear1"  android:orientation="vertical"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="30dp">  <TextView  android:id="@+id/txtTinggi"  android:layout\_width="150dp"  android:layout\_height="wrap\_content"  android:text="Tinggi (CM)"/>  <EditText  android:layout\_width="80dp"  android:layout\_height="wrap\_content"  android:inputType="number"  android:ems="10"  android:id="@+id/tinggi" />  </LinearLayout>  <LinearLayout  android:id="@+id/linear2"  android:orientation="vertical"  android:layout\_width="150dp"  android:layout\_height="wrap\_content"  android:layout\_toRightOf="@id/linear1"  android:paddingLeft="30dp">  <TextView  android:id="@+id/txtBerat"  android:layout\_width="150dp"  android:layout\_height="wrap\_content"  android:text="Berat (KG)"/>  <EditText  android:layout\_width="80dp"  android:layout\_height="wrap\_content"  android:inputType="number"  android:ems="10"  android:id="@+id/berat" />  </LinearLayout>  </RelativeLayout>  <LinearLayout  android:id="@+id/lineardata"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical"  android:layout\_below="@id/relative">  <TextView  android:id="@+id/standar"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Standar BMI"  android:gravity="center\_horizontal"  android:textSize="30sp"  android:layout\_marginTop="15sp" />  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <TextView  android:layout\_width="140dp"  android:layout\_height="wrap\_content"  android:text="Pria" />  <TextView  android:id="@+id/ktgr"  android:layout\_width="140dp"  android:layout\_height="wrap\_content"  android:text="Kategori" />  <TextView  android:id="@+id/wnt"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Wanita" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <TextView  android:layout\_width="140dp"  android:layout\_height="wrap\_content"  android:text="&lt; 17" />  <TextView  android:layout\_width="140dp"  android:layout\_height="wrap\_content"  android:text="Kurus" />  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="&lt; 18"/>  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <TextView  android:layout\_width="140dp"  android:layout\_height="wrap\_content"  android:text="17 - 23" />  <TextView  android:layout\_width="140dp"  android:layout\_height="wrap\_content"  android:text="Ideal" />  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="18 - 25"/>  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <TextView  android:layout\_width="140dp"  android:layout\_height="wrap\_content"  android:text="23 - 27" />  <TextView  android:layout\_width="140dp"  android:layout\_height="wrap\_content"  android:text="Kegemukan" />  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="25 - 27" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <TextView  android:layout\_width="140dp"  android:layout\_height="wrap\_content"  android:text="> 27" />  <TextView  android:layout\_width="140dp"  android:layout\_height="wrap\_content"  android:text="Obesitas" />  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="> 27" />  </LinearLayout>  </LinearLayout>  <Button  android:id="@+id/Hitung"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hitung"  android:layout\_below="@id/lineardata"  android:layout\_marginTop="15dp"  android:layout\_centerHorizontal="true"  android:textSize="20sp" />  <TextView  android:id="@+id/txtHasil"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hasil : "  android:layout\_below="@id/Hitung"  android:layout\_marginTop="15dp"  android:paddingRight="15dp"  android:textSize="30dp"/>  <TextView  android:id="@+id/Hasil"  android:layout\_width="60dp"  android:layout\_height="40dp"  android:layout\_toRightOf="@id/txtHasil"  android:layout\_marginTop="15dp"  android:layout\_below="@id/Hitung"  android:textSize="30dp"/>  <Button  android:id="@+id/ulang"  android:layout\_width="160dp"  android:layout\_height="wrap\_content"  android:text="Ulang"  android:layout\_alignParentBottom="true" />  <Button  android:id="@+id/back"  android:layout\_width="160dp"  android:layout\_height="wrap\_content"  android:text="Kembali"  android:layout\_alignParentRight="true"  android:layout\_alignParentBottom="true"/>  </RelativeLayout> |

**Code 3.4** Code dalam Activity\_start.xml

* + 1. Coding dalam About.java

|  |
| --- |
| package com.rzproject.idealweight;  import android.content.Intent;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.view.View;  import android.widget.Button;  public class About extends AppCompatActivity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_about);  Button aboutback=(Button)findViewById(R.id.aboutback);  aboutback.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  startActivity(new Intent(About.this,MainActivity.class));  finish();  }  });  }  } |

**Code 3.5** Code dalam About.java

* + 1. Coding dalam Activity\_about.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  tools:context="com.rzproject.idealweight.About">  <TextView  android:id="@+id/judul"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Judul Aplikasi : " />  <TextView  android:id="@+id/judul1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Ideal Weight"  android:layout\_toRightOf="@id/judul" />  <TextView  android:id="@+id/deskripsi"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Deskripsi : "  android:layout\_below="@id/judul" />  <TextView  android:id="@+id/deskripsi1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Aplikasi untuk menghitung berat badan ideal"  android:layout\_toRightOf="@id/deskripsi"  android:layout\_below="@id/judul"  android:layout\_toEndOf="@+id/judul" />  <TextView  android:id="@+id/namanim"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Nama Nim : "  android:layout\_below="@id/deskripsi1" />  <TextView  android:id="@+id/namanim1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Pande Bagus Narendra Mahaputra (1404505075)"  android:layout\_below="@id/deskripsi1"  android:layout\_toRightOf="@id/namanim"  android:layout\_toEndOf="@+id/judul" />  <TextView  android:id="@+id/namanim2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="I Kadek Dwi Rendika (1404505077)"  android:layout\_below="@id/namanim1"  android:layout\_toRightOf="@id/namanim"  android:layout\_toEndOf="@+id/judul" />  <TextView  android:id="@+id/mk"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="MK : "  android:layout\_below="@id/namanim2" />  <TextView  android:id="@+id/mk1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Pemrograman Mobile (C)"  android:layout\_alignBaseline="@+id/mk"  android:layout\_alignBottom="@+id/mk"  android:layout\_toRightOf="@+id/judul"  android:layout\_toEndOf="@+id/judul" />  <TextView  android:id="@+id/kodemk"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Kode MK : "  android:layout\_below="@id/mk1" />  <TextView  android:id="@+id/kodemk1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="TI024330"  android:layout\_below="@+id/mk1"  android:layout\_alignLeft="@+id/mk1"  android:layout\_alignStart="@+id/mk1" />  <TextView  android:id="@+id/dosen"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Dosen : "  android:layout\_below="@id/kodemk1" />  <TextView  android:id="@+id/dosen1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="I Putu Agus Eka Pratama, ST, MT"  android:layout\_below="@+id/kodemk1"  android:layout\_alignLeft="@+id/kodemk1"  android:layout\_alignStart="@+id/kodemk1" />  <TextView  android:id="@+id/jurusan"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Jurusan : "  android:layout\_below="@id/dosen1" />  <TextView  android:id="@+id/jurusan1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Teknologi Informasi"  android:layout\_below="@+id/dosen1"  android:layout\_alignLeft="@+id/dosen1"  android:layout\_alignStart="@+id/dosen1" />  <TextView  android:id="@+id/fakultas"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Fakultas : "  android:layout\_below="@id/jurusan1" />  <TextView  android:id="@+id/fakultas1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Teknik"  android:layout\_below="@+id/jurusan1"  android:layout\_alignLeft="@+id/jurusan1"  android:layout\_alignStart="@+id/jurusan1" />  <TextView  android:id="@+id/universitas"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Universitas : "  android:layout\_below="@id/fakultas1" />  <TextView  android:id="@+id/universitas1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Universitas Udayana"  android:layout\_below="@+id/fakultas1"  android:layout\_alignLeft="@+id/fakultas1"  android:layout\_alignStart="@+id/fakultas1" />  <TextView  android:id="@+id/tahun"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Tahun : "  android:layout\_below="@id/universitas1" />  <TextView  android:id="@+id/tahun1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="2016"  android:layout\_below="@id/universitas1"  android:layout\_toRightOf="@id/tahun"  android:layout\_toEndOf="@+id/judul" />  <Button  android:id="@+id/aboutback"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Kembali"  android:layout\_centerHorizontal="true"  android:layout\_alignParentBottom="true"/>  </RelativeLayout> |

**Code 3.6** Code dalam Activity\_about.xml

**3.6 GUI *Ideal Weight***

GUI dari aplikasi *Ideal Weight* yaitu dapat dilihat sebagai berikut.

|  |
| --- |
| D:\PROGMOB\main logo.jpg |

**Gambar 3.4** *Icon Laucher Ideal Weight*

Gambar diatas adalah icon launcher untuk aplikasi *Ideal Weight*. Warna dasar yang digunakan adalah hijau, dengan gambar timbangan sebagai lambang keseimbangan.

|  |
| --- |
| D:\PROGMOB\logo+name.jpg |

**Gambar 3.5** *Logo Ideal Weight*

Gambar diatas adalah logo dari *Ideal Weight*. Timbangan berisi buah apel dan kue mengartikan bahwa harus ada keseimbangan antara junk food dan healty food sehingga berat ideal akan sesuai keinginan. Warna hijau digunakan agar tampilan menjadi lebih cerah.

|  |
| --- |
| D:\PROGMOB\main.png |

**Gambar 3.6** GUI Halaman Awal Aplikasi

Gambar ditas adalah interface dari halaman awal aplikasi *Ideal Weight*. Dalam GUI tersebut terdapat logo aplikasi, deskripsi singkat aplikasi, tombol Mulai Menghitung untuk menuju menu penghitungan indeks berat ideal, tombol Tentang Aplikasi untuk menampilkan informasi rinci dari aplikasi, dan tombol Keluar untuk keluar dari aplikasi.

|  |
| --- |
| D:\PROGMOB\start.png |

**Gambar 3.7** GuI Halaman Utama Aplikasi

Gambar diatas adalah interface setelah menekan tombol Mulai Menghitung pada halaman awal aplikasi. Interface ini akan melakukan perhitungan berat ideal setelah user memasukkan tinggi badan pada form Tinggi (CM), dan berat badan pada form Berat (KG). Terdapat juga standar BMI (Body Mass Index) untuk pria dan wanita sehingga user dapat menentukan apakah dirinya kurus, ideal, gemuk, atau obesitas. Tombol Hitung berfungsi untuk melakukan perhitungan indeks berat ideal setelah user memasukkan tinggi dan berat badan. Hasil dari perhitungan akan tampil pada form Hasil dibawah tombol Hitung. Terdapat juga tombol Ulang untuk mereset data di aplikasi, dan tombol Kembali untuk menuju halaman awal aplikasi.

|  |
| --- |
|  |

**Gambar 3.8** GUI Halaman Tentang Aplikasi

Gambar diatas adalah GUI saat user menekan tombol Tentang Aplikasi pada halaman awal, maka akan muncul data secara rinci tentang aplikasi seperti judul aplikasi, deskripsi, serta nama pembuat dari aplikasi tersebut.