

BAB 2

LANDASAN TEORI

2.1 Pengertian Jantung

Jantung merupakan salah satu organ terpenting dalam tubuh manusia. Jantung memiliki fungsi sebagai alat pompa darah, sehingga darah dapat mengalir keseluruh tubuh hal ini erat kaitannya dengan peredaran tubuh. Dimana saat itu jantung memompa darah yang kaya akan oksigen yang berasal dari paru-paru ke seluruh tubuh selanjutnya memompa darah hasil metabolisme kembali ke paru-paru untuk mengedarkan kembali oksigen keseluruh tubuh, hal ini terjadi berulang tanpa henti. Dapat dibayangkan begitu vital kerja jantung pada tubuh kita. Informasi mengenai denyut nadi menjadi sangat penting dalam pemantauan kinerja jantung dalam beraktivitas sehari-hari terutama saat aktivitas berat seperti saat berolahraga. Denyut nadi dapat memberikan indikasi seberapa sehat kondisi jantung serta menjadi langkah untuk mendiagnosa kelainankelainan terhadap jantung. Di dunia kedokteran alat bantu yang biasa digunakan dokter untuk mendengar detak jantung adalah stetoskop, dengan bantuan alat ini seorang ahli kesehatan dapat memberikan penilaian dari suara yang dihasilkan dan memberikan diagnosa. Alat lain yang biasa digunakan dibidang kesehatan yaitu Elektrokardiograf (EKG) alat ini dapat merepresentasikan karakteristik sinyal jantung dalam bentuk gelombang sinyal yang dapat digunakan ahli kesehatan untuk menghitung denyut jantung dan mendiagnosis kesehatan jantung. Alat ini dapat merekam kerja jantung dengan akurasi yang sangat baik namun penggunaan alat-alat ini dirasa kurang efisien karena pasien harus mengunjungi rumah sakit atau klinik kesehatan yang tentu cukup memberatkan bagi sebagian besar orang. Serta perkembangan pola pikir manusia yang semakin sadar bahwa olahraga adalah hal yang penting dalam menjaga kesehatan, olahraga juga telah menjadi sebuah gaya hidup bagi sebagian besar orang. Hal ini terbukti dari berkembangnya pusat kebugaran tubuh dan banyaknya event-event olahraga yang diselenggarakan. Perkembangan teknologi mobile sangat pesat diantaranya sistem operasi android. Android merupakan sistem operasi berbasis linux, android

merupakan sistem operasi yang cukup populer dikalangan programmer hal ini tak lepas dari terbukanya platform ini untuk membuat aplikasi. Smartphone android juga sudah menjadi alat komunikasi yang wajib dibawa bagi sebagian besar orang. Ini lah yang membuat banyak developer aplikasi menggunakan android sebagai platform nya. Maka pada tugas akhir ini akan dibangun sebuah aplikasi penghitung denyut nadi berbasis android, Pada aplikasi menghitung denyut nadi ini memanfaatkan kamera dan lampu flash pada perangkat smartphone android. Dengan menggunakan jari tangan untuk mendeteksi jumlah denyut nadi pengguna. aplikasi ini tentu akan mempermudah seseorang untuk menghitung dan mendeteksi denyut nadi sehari-hari dengan mandiri. Hal ini tentu dapat menghemat waktu bagi orang yang memiliki aktivitas tinggi dan tidak dapat mengunjungi rumah sakit. Aplikasi ini juga dapat membantu orang yang memang memiliki rekam medis kelainan jantung untuk memantau kerja jantung nya dan orang yang senang berolahraga dapat mengetahui tingkat kebugaran tubuhnya baik setelah dan sebelum berolahraga [4].

2.2 Teori Dasar

2.2.1 Jantung

Jantung adalah organ tubuh yang terdiri dari otot - otot yang kuat dan memompa darah yang membawa oksigen dan membawa makanan ke seluruh bagian tubuh. Jantung mempunyai dua arteri coroner utama dan memiliki banyak cabang. Jantung juga merupakan salah satu organ tubuh yang vital. Jantung kiri berfungsi memompa darah bersih (kaya oksigen atau zat asam) ke seluruh tubuh, sedangkan jantung kanan menampung darah kotor (rendah oksigen, kaya karbon dioksida atau zat asam arang), yang kemudian di alirkan ke paru-paru untuk dibersihkan. Jantung normal besarnya seenggam tangan kiri pemiliknya. Jantung berdenyut 60-80 kali permenit, denyutan bertambah cepat pada saat aktifitas atau emosi, agar kebutuhan tubuh akan energi dapat terpenuhi. Andaikan denyutan jantung 70 kali permenit, maka dalam 1 jam jantung berdenyut 4200 kali atau 100.800 kali sehari

semalam. Tiap kali berdenyut dipompakan darah sekitar 70 cc, jadi dalam 24 jam jantung memompakan darah sebanyak kira-kira 7000 liter [4].

2.2.2 Denyut Jantung

Denyut jantung (Heart rate) biasanya mengacu pada jumlah waktu yang dibutuhkan oleh detak jantung per satuan waktu, secara umum direspresentasikan sebagai bpm (beats per minute). Denyut jantung yang optimal untuk setiap individu berbeda-beda tergantung pada kapan waktu mengukur detak jantung tersebut (saat istirahat atau saat setelah berolahraga). Variasi dalam detak jantung sesuai dengan jumlah oksigen yang diperlukan oleh tubuh [4].

Denyut nadi (pulse rate) menggambarkan frekuensi kontraksi jantung seseorang. Pemeriksaan denyut nadi sederhana, biasanya dilakukan secara palpasi. Palpasi adalah cara pemeriksaan dengan meraba, menyentuh, atau merasakan struktur dengan ujung-ujung jari; sedangkan pemeriksaan dikatakan auskultasi, apabila pemeriksaan dilakukan dengan mendengarkan suara-suara alami yang diproduksi dalam tubuh [4].

Pengukuran denyut nadi selama bekerja merupakan suatu metode untuk menilai *cardiovascular strain*. Salah satu peralatan yang dapat digunakan untuk menghitung denyut nadi adalah telemetri dengan menggunakan ranggsangan *ElectroCardio Graph (ECG)*. Apabila peralatan tersebut tidak tersedia, maka dapat dicatat secara manual memakai *stopwatch* dengan metode 10 denyut[4].

2.2.3 Denyut Nadi Maksimal

Menurut DR Suhantoro cara yang aman mengukur denyut nadi maksimal (DNM). DNM adalah denyut nadi maksimal yang dihitung berdasarkan rumusan $DNM=220-Umur$, dalam olahraga, diberikan 3 (tiga) tingkatan kebutuhan yaitu:

- a. Untuk sehat (*Health*) : 50-70% denyut nadi maksimum
- b. Untuk kebugaran (*Fitness*) : 70-80% denyut nadi maksimum
- c. Untuk atlit (*Performance*) : 80-100% denyut nadi maksimum

Kemudian dikalikan dengan intensitas latihan yang sedang dilakukan 50-70 persen DNM. DR Suhantoro mencontohkan orang yang berusia 40 tahun maka DNM saat ia berolahraga adalah $220-40 = 180$. Kemudian angka 180 dikalikan

dengan 50 persen untuk batas ringan dan 70 persen untuk batas atas yang hasilnya 90- 126. Dengan mengetahui denyut nadi tersebut, maka orang yang berusia 40 tahun harus berhenti sejenak dari olahraganya ketika denyut nadinya sudah melampaui 126. Jika masih dipaksakan yang terjadi adalah kram jantung yang membuat serangan jantung. "Sekali lagi perlu diperhatikan kondisi denyut jantung saat berolahraga jangan sampai melebihi batas maksimal yang bisa membahayakan jantung,"Jika sudah merasa melampaui dosis saat lari di futsal berikan saja bolabola itu ke orang lain yang masih kuat. Satu lagi saat istirahat minumlah air dengan suhu 15-16 derajat atau minuman manis dengan kadar gula 2,5-5 persen. "Minuman yang terlalu dingin akan sulit diserap tubuh karena suhu tubuh setelah olahraga sedang dalam kondisi panas [4].

2.3 Suhu Tubuh

Suhu tubuh adalah perbedaan antara jumlah panas yang diproduksi oleh proses tubuh dan jumlah panas yang hilang ke lingkungan luar. Suhu tubuh mudah sekali berubah dan dipengaruhi oleh banyak faktor, baik faktor eksternal maupun faktor internal. Perubahan suhu tubuh sangat erat kaitannya dengan produksi panas maksimal maupun pengeluaran panas yang berlebihan. Sifat perubahan panas tersebut sangat mempengaruhi masalah klinis yang dialami setiap orang, menurut WHO suhu tubuh normal manusia berkisar $36,5 - 37,5^{\circ}\text{C}$ [3].

Hubungan denyut jantung dan suhu tubuh pada penelitian ini karena pada penelitian ini bertujuan membuat suatu aplikasi marathon trainer dengan indikator kebugaran berdasarkan denyut jantung dan suhu tubuh yang bisa membantu para pelari dalam menentukan indikator level kebugaran mereka.

2.4 Android

Android adalah sistem operasi yang dimodifikasi untuk *mobile device* berbasis linux yang terdiri dari sistem operasi, software penghubung antar sebuah aplikasi dengan aplikasi lain, dan aplikasi-aplikasi utama. Android dikembangkan oleh Android Inc. perusahaan ini kemudian dibeli oleh Google pada tahun 2005 dan dirilis secara resmi pada tahun 2007 [5].

Android telah mengalami beberapa pembaruan sejak pertama kali rilis, berikut ini jenis-jenis versi dan *Codename* android bisa dilihat pada table 2.1 [6].

Tabel 2.1 Daftar Versi Android

Versi	Peluncuran	Kode
1.5	30 April 2009	Cupcake
1.6	15 September 2009	Donut
2.0/2.1	26 Oktober 2009	Éclair
2.2	20 Mei 2010	Frozen Yoghurt (Froyo)
2.3	6 Desember 2010	Gingerbread
3.0	22 Februari 2011	Honeycomb
4.0	19 Oktober 2011	Ice Cream Sandwich
4.1/4.2/4.3	27 Juni 2012	Jelly Bean
4.4	3 September 2013	Kitkat
5.0 – 5.1	12 November 2014	Lollipop
6.0	5 Oktober 2015	Marshmallow
7.0/7.1	Agustus / September 2016	Nougat
8.0/8.1	Agustus 2017	Oreo

Secara umum, arsitektur Android terdiri dari beberapa lapisan software, yaitu *Applications and Widgets*, *Applications Framework*, *Libraries*, *Android Run-time* sebagai lapisan perantara (*middleware*), serta *Linux Kernel* yang membungkus di bawahnya[7]. Untuk penjelasan setiap lapisan tersebut adalah sebagai berikut :

1. *Applications and Widget*, Lapisan dimana pengguna hanya berhubungan dengan aplikasi saja.
2. *Applications Framework*, Lapisan dimana para pengembang melakukan pembuatan aplikasi yang akan dijalankan di sistem operasi Android dengan komponen-komponennya meliputi view, sumberdaya, pemberitahuan (notifikasi), dan lain-lain.

3. *Libraries*, Android menyediakan sejumlah *Libraries* seperti grafis 2D dan 3D, *multimedia playback*, *browser engine*, pencetakan font, database, dan lain-lain.
4. *Android Run-time*, Lapisan yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux yang terbagi menjadi dua bagian yaitu Core Libraries dan Dalvik virtual Machine.
5. *Linux Kernel*, Menyediakan layanan dasar seperti keamanan, pengelolaan proses, pengelolaan file, pengelolaan sumber daya memori, dan hard ware.

Alasan digunakannya Android pada penelitian ini karena Android menjadi sebuah sistem operasi perangkat bergerak (*mobile device*) yang paling banyak digunakan di dunia. Dari sample 50 orang yang sudah dilakukan didapat data 100% orang memiliki perangkat android. Teknologi yang terdapat pada perangkat android bisa dimanfaatkan untuk membantu permasalahan yang sedang diteliti saat ini.

2.5 Java

Java diciptakan oleh Patrick Naughton dan James Gosling bersama timnya dalam suatu proyek dari Sun Microsystems dengan tujuan untuk menghasilkan bahasa pemrograman yang sederhana. Awalnya nama bahasa pemrograman ini bukanlah Java, tetapi OAK terinspirasi dari sebuah pohon yang berada pada sebrang kantornya. Namun Sun menyadari OAK merupakan bahasa pemrograman komputer yang sudah ada. Maka Sun mengubahnya menjadi Java yang terinspirasi saat mereka sedang menikmati secangkir kopi di sebuah kedai kopi. Java termasuk bahasa pemrograman tingkat tinggi dan bahasa pemrograman berorientasi objek untuk membuat dan menjalankan perangkat lunak pada komputer [8].

Berikut ini beberapa fitur java yang ditawarkan java API antara lain [9] :

1. Applet

Program Java yang dapat berjalan di atas browser, yang dapat membuat halaman HTML lebih dinamis dan menarik.

2. Java Networking

Sekumpulan API (*Application Programming Interface*) yang menyediakan fungsi-fungsi untuk aplikasi-aplikasi jaringan, seperti penyediaan akses untuk TCP, UDP, IP address dan URL. Namun pada Java Networking tidak menyediakan ICMP karena *security* dan hanya administrator (root) yang bias memanfaatkan protocol ICMP.

3. Java Database Connectivity (JDBC)

JDBC menyediakan sekumpulan API yang dapat digunakan untuk mengakses database seperti Oracle, MySQL, PostgreSQL, Microsoft SQL Server

4. Java Security

Java Security menyediakan sekumpulan API untuk mengatur security dari aplikasi Java baik secara *high level* atau *low level*, seperti *public/private key management* dan *certificates*.

5. Java Swing

Java Swing menyediakan sekumpulan API untuk membangun aplikasi – aplikasi GUI (*Graphical User Interface*) dan model GUI yang diinginkan bisa bermacam-macam, bisa model Java, model Motif/CDE atau model yang dependent terhadap platform yang digunakan

6. Java RMI

Java RMI menyediakan sekumpulan API untuk membangun aplikasi – aplikasi Java yang mirip dengan model RPC (Remote Procedure Call) jadi objek-objek Java bisa di panggil secara remote pada jaringan komputer.

7. Java 2D/3D

Java 2D/3D menyediakan sekumpulan API untuk membangun grafik-grafik 2D/3D yang menarik dan juga akses ke printer.

8. Java Server Pages

Berkembang dari Java Servlet yang digunakan untuk menggantikan aplikasi-aplikasi CGI, JSP (Java Server Pages) yang mirip ASP dan PHP merupakan alternatif terbaik untuk solusi aplikasi Internet.

9. Java Native Interface (JNI)

JNI menyediakan sekumpulan API yang digunakan untuk mengakses fungsi-fungsi pada library (*.dll atau *.so) yang dibuat dengan bahasa pemrograman yang lain seperti C,C++, dan Basic.

10. Java Sound

Java Sound menyediakan sekumpulan API untuk manipulasi suara.

11. Java IDL + CORBA

Java IDL (*Interface Definition Language*) menyediakan dukungan Java untuk implementasi CORBA (*Common Object Request Broker*) yang merupakan model distributed-Object untuk solusi aplikasi besar di dunia *networking*

12. Java Card

Java Card utamanya digunakan untuk aplikasi-aplikasi pada smart card, yang sederhana wujudnya seperti SIM Card pada handphone.

13. Java Telephony API (JTAPI)

Java Telephony API menyediakan sekumpulan API untuk memanfaatkan devices-devices telephony, sehingga akan cocok untuk aplikasi-aplikasi CTI (*Computer Telephony Integration*) yang dibutuhkan seperti ACD (*Automatic Call Distribution*), PC-PBX dan lainnya.

Adapun hubungan Java pada penelitian ini karena aplikasi yang dibuat menggunakan tools Android studio dengan bahasa pemrograman yang digunakan yaitu bahasa pemrograman Java. Android Studio dipilih karena memiliki banyak fitur yang memudahkan saat pembangunan aplikasi android.

2.6 OOP (*Object Oriented Programming*)

Object oriented programming atau pemrograman berorientasi objek adalah suatu pendekatan pemrograman untuk membuat kode program yang lebih terkelompok berdasarkan objek-objek yang terlibat. Untuk membuat aplikasi yang besar kita harus membagi aplikasi tersebut menjadi bagian-bagian kecil, pada kasus seperti ini programmer menerapkan konsep pemrograman berorientasi

objek dengan membagi-bagi kode program aplikasi menjadi kumpulan bungkus objek [8].

Sebuah bahasa pemrograman dinyatakan bisa menggunakan OOP jika sudah memenuhi syarat-syarat berikut, yaitu :

1. *Encapsulation* atau pembungkusan yaitu memaketkan atribut bersama dengan metode-metodenya dalam sebuah *Class* dan memberi akses langsung terhadap atribut dan metode-metode yang ingin dilindungi di dalam *Class* tersebut..
2. *Inheritance* atau pewarisan yaitu mewariskan sesuatu dari suatu *Class* untuk dipakai *Class-class* yang lain yang memiliki beberapa sifat atau perilaku dari *Class* induknya. Misalnya seorang anak yang mendapatkan beberapa sifat atau perilaku dari ibu/bapaknya. Pada pemrograman berbasis objek ini inherent merupakan suatu hubungan dari dua buah class atau lebih dimana dari class yang berhubungan tersebut memiliki atribut dan metode yang sama.
3. *Polymorphism* berarti banyak bentuk yaitu pembuatan metode-metode dengan nama yang sama tetapi berbeda *Class* atau parameter masukkan.

Hubungan OOP (*Object Oriented Programming*) pada penelitian ini adalah pada penelitian ini menggunakan konsep OOP yaitu membagi-bagi kode program menjadi kelas-kelas yang di buat berdasarkan objek-objek yang terlibat saat pembangunan program.

2.7 UML (*Unified Modeling Language*)

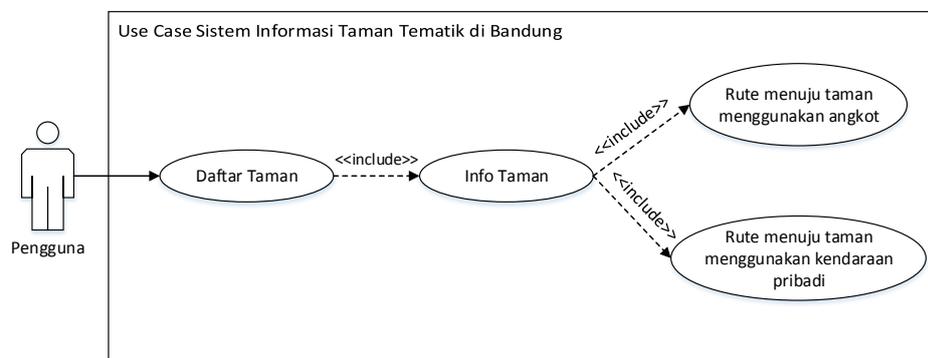
UML (*Unified Modeling Language*) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM^[11].

Berikut ini diagram-diagram UML yang paling sering digunakan dalam memodelkan sebuah sistem berorientasi objek diantaranya :

1. Use Case Diagram

Use case diagram adalah form tertinggi tentang suatu sistem yang merupakan gambaran pada pihak manajemen, customer, atau orang lain yang terlibat langsung pada sistem tersebut tetapi tidak secara mendetail^[11].

Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Selain Include, Sebuah use case juga dapat meng-extend use case lain dengan behaviour-nya sendiri. Berikut ini adalah contoh dari Use Case Diagram



Gambar 2.1 Contoh Use Case Diagram

2. Class Diagram

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain [10].

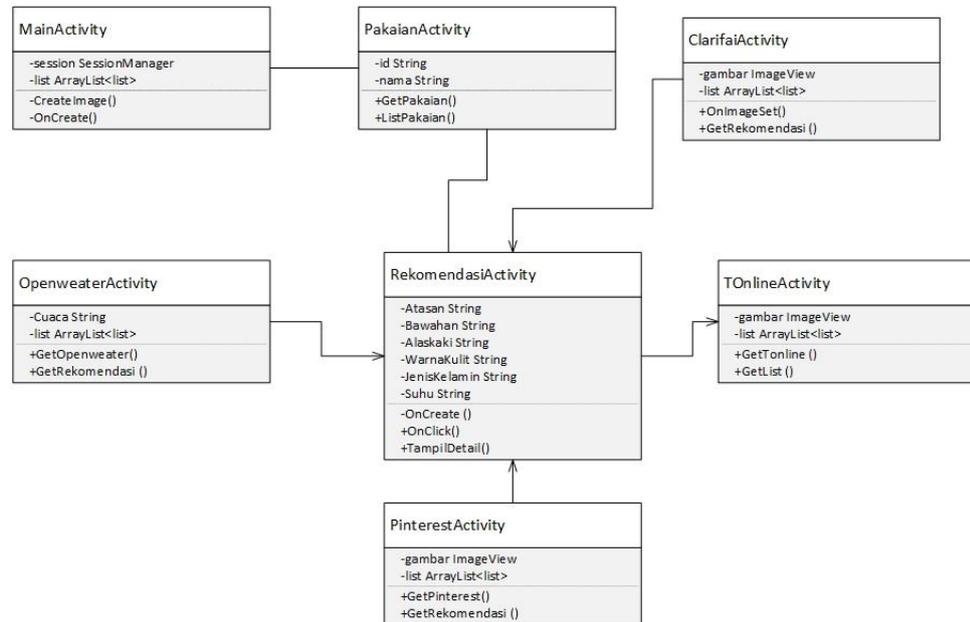
Class memiliki tiga area pokok:

1. Nama memiliki fungsi merepresentasikan identitas dari sebuah class.
2. Atribut memiliki fungsi merepresentasikan atribut – atribut yang ada di dalam suatu class.
3. Metode memiliki fungsi merepresentasikan metode-metode yang ada pada suatu class.

Pada atribut dan metode bisa memiliki sifat sifat berikut :

1. *Private*, hanya bisa dipanggil oleh *Class* yang bersangkutan, tidak bisa dipanggil oleh *Class* lain yang tidak bersangkutan.
2. *Public*, bisa di panggil oleh seluruh *Class*.
3. *Protected*, hampir sama sifatnya dengan *Private* dan *Class* turunannya.

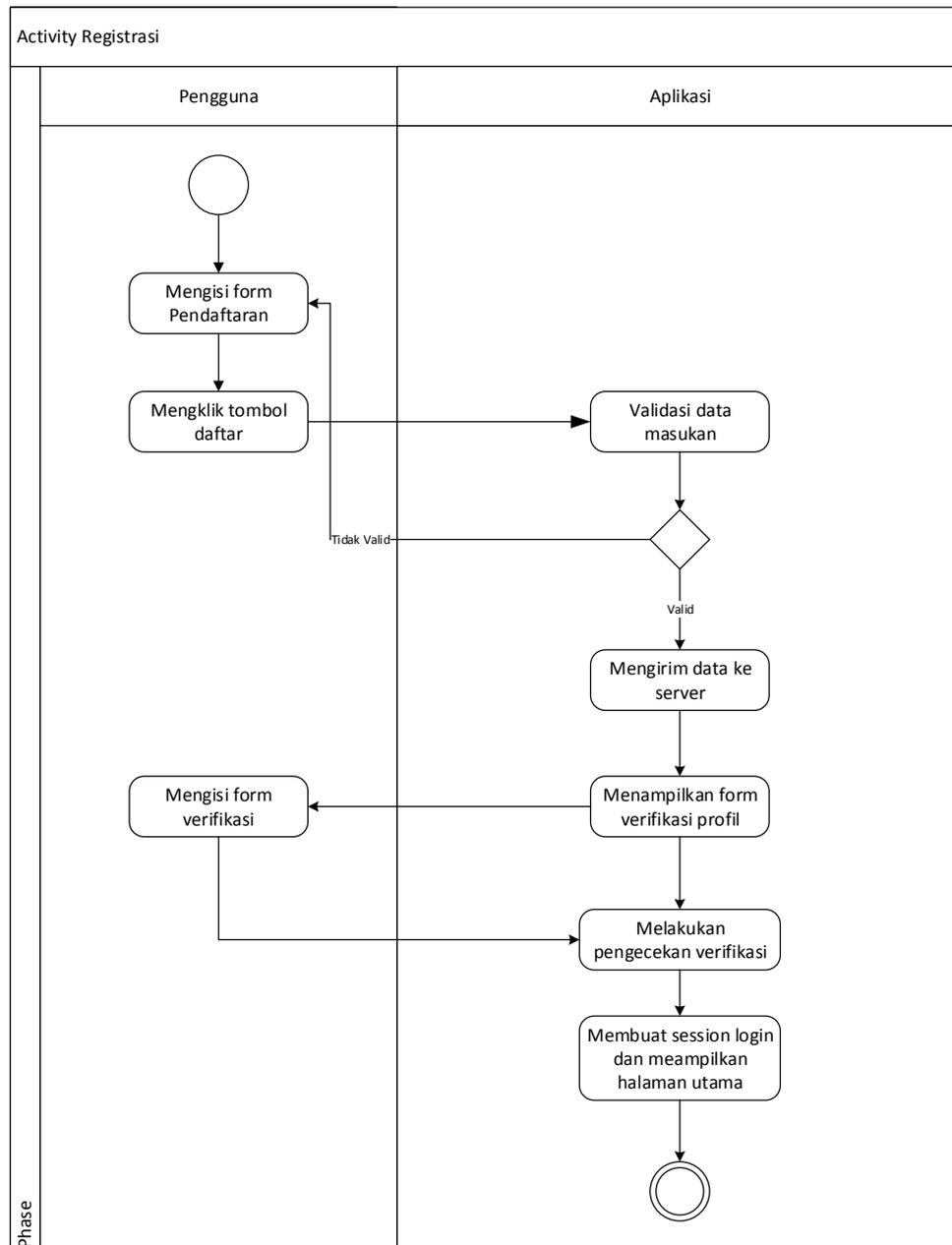
Pada gambar 2.2 adalah contoh dari *Class* diagram:



Gambar 2.2 Contoh Class Diagram

3. *Activity* Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin. Pada Gambar 2.3 adalah contoh dari *Activity* Diagram



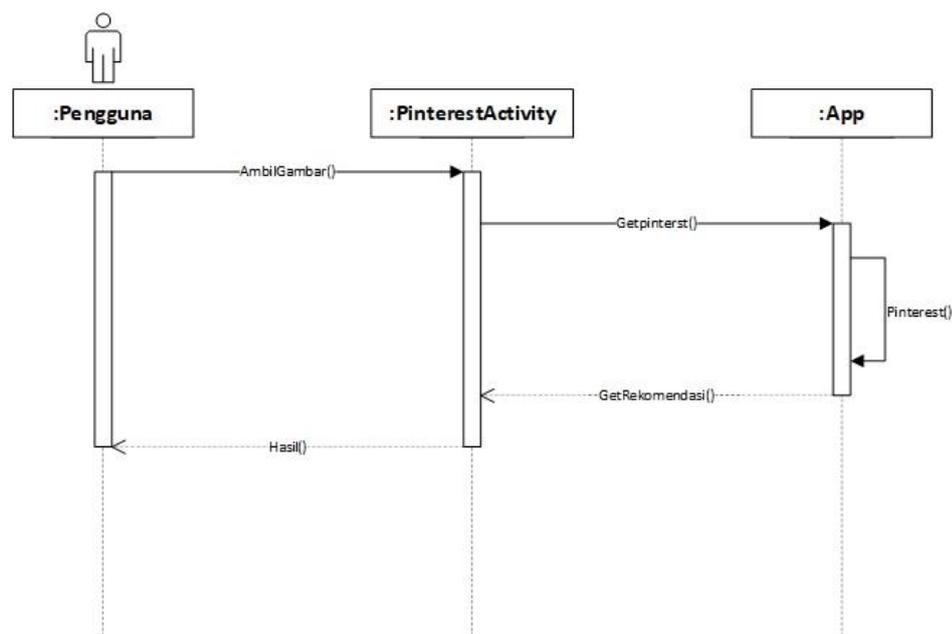
Gambar 2.3 Contoh Activity Diagram

4. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence* diagram terdiri

antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan [11]. Pada Gambar 2.4 adalah contoh dari *Sequence* Diagram



Gambar 2.4 Contoh Squence Diagram

Adapun hubungan UML pada penelitian ini adalah pada penelitian ini untuk memodelkan suatu aplikasi yang akan dibuat menggunakan konsep berorientasi objek. Sehingga bahasa pemodelan yang digunakan dalam perancangan aplikasi adalah UML.

2.8 Facebook API

Facebook memberikan kemudahan untuk mengintegrasikan sebuah aplikasi dengan *facebook*. *Facebook* telah menyediakan API untuk *developer* yang dapat digunakan secara gratis. Aplikasi yang dikembangkan dapat diintegrasikan dari banyak aspek, seperti *news feed*, *notification*, informasi

tentang *user*, *registrasi* atau *login* dengan *facebook* dan masih terdapat beberapa fasilitas yang ditawarkan. Untuk mempermudah pengembangannya, aplikasi yang dikembangkan tidak perlu dihosting di *server facebook*, melainkan cukup di *server* kita sendiri [12].

2.9 GPS (*Global Positioning System*)

Fitur yang semakin terjangkau dan populer pada piranti bergerak adalah kemampuan GPS yang memungkinkan piranti memberitahu lokasi setiap saat. Fungsi utama GPS adalah pemetaan, pencarian lokasi, dan arah. Teknologi GPS adalah teknologi sistem navigasi yang awalnya dikembangkan untuk menunjang kegiatan militer, namun kemudian dibuka untuk umum.

Teknologi GPS terdiri dari konsentelasi 27 satelit yang mengelilingi bumi (24 satelit aktif ditambah 3 satelit cadangan). Tiap satelit mengelilingi bumi dua kali dalam satu hari. Orbit tiap satelit sudah di atur sehingga pada tiap titik di seluruh permukaan bumi, orang akan menemukan paling tidak 4 setelit di langit tiap saat. Bicara piranti GPS artinya mengacu pada piranti penerima yang tugas utamanya adalah menemukan lokasi empat satelit atau lebih, mendapat jarak piranti penerima kemasing-masing satelit, dan menentukan lokasi piranti penerima berdasarkan jarak ke masing-masing satelit [6].

Selain memberikan keuntungan, tentu ada beberapa tantangan yang harus dihadapi ketika menggunakan layanan berbasis lokasi, yaitu :

1. Konsumsi Energi

Tingkat konsumsi energi penyedia layanan berbeda-beda. Ketika aktif, piranti GPS paling banyak menguras energi disbanding Cell-ID atau Wi-Fi. Oleh karena itu, penting untuk mematikan piranti ini bila tidak diperlukan.

2. Akurasi

Tingkat akurasi penyedia layanan lokasi juga berbeda-beda, `LocationManager.NETWORK_PROVIDER` memiliki akurasi terendah. Sedangkan `LocationManager.GPS_PROVIDER` memiliki akurasi lebih baik.

3. Ketersediaan Layanan

Layanan berbasis lokasi dengan GPS, Cell-ID, atau Wi-Fi hanya dapat dinikmati bila piranti Anda mendukungnya. Namun, meskipun sudah mendukung, tetap tidak ada jaminan layanan akan selalu tersedia. Sinyal GPS akan bagus apabila Anda berada diruang terbuka dan posisi pancaran sinyal dari satelit tidak terhalang. Bila berada dibawah tanah atau ruang berdinding tebal, penerimaan sinyal GPS mungkin buruk karena tidak cukup kuat untuk menembusnya.

4. Waktu Untuk Memperoleh Data Lokasi

Penentuan lokasi piranti penerimaan GPS membutuhkan waktu yang lama di bandingkan Cell-ID. Lama waktu tergantung piranti penerima GPS [6].

Hubungan GPS pada penelitian ini adalah pada penelitian ini membutuhkan GPS untuk menemukan lokasi pengguna untuk pengambilan data cuaca lokasi pengguna saat itu.

2.10 Android Studio

Android Studio adalah IDE (*Integrated Developmenat Environment*) resmi Android yang dirancang untuk membantu Anda membangun aplikasi berkualitas tinggi untuk setiap perangkat Android. Android Studio merupakan pembaharuan IDE dari Eclipse yaitu versi IDE yang sebelumnya. Saat ini, Android Studio sudah dilengkapi dengan berbagai kemudahan dan kenyamanan [13].

Kemampuan yang ditawarkan dari Android Studio antara lain :

1. Mendukung pengembangan berbasis Gradle.
2. Dapat menspesifikasikan pemfaktoran android dan melakukan perbaikan dengan cepat.
3. memiliki lint tools untuk menangkap kinerja, kegunaan kompatibilitas versi dan masalah lainnya.
4. memiliki kemampuan penandaan aplikasi dan ProGUard.
5. memiliki kemampuan wizard berdasarkan template untuk membuat desain dan komponen android yang umum.

6. sebuah layout editor yang kaya dan memungkinkan anda untuk melakukan drag-drop UI, pratinjau layout pada beberapa konfigurasi layar, dan banyak lagi.
7. di bangun untuk mendukung Google Cloud Platform, sehingga mudah untuk mengintegrasikan Google Cloud Messaging dan App Engine sebagai komponen server-side [14].

Alasan digunakan Android Studio pada penelitian karena tools yang digunakan dalam pembangunan aplikasi android yaitu menggunakan IDE Android Studio. Android studio menyediakan alat pengembang android yang terintegrasi dalam pengembangan dan debugging program.

2.11 JSON

JSON adalah sebuah format data-interchange yang ringan. JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga-C, termasuk C, C++, C#, Java, JavaScript, Perl, dan Python [15].

JSON dibangun dalam dua struktur, yaitu :

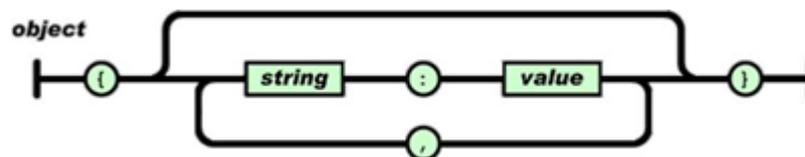
1. Beberapa pasangan dari nama / nilai. Dalam beberapa bahasa pemrograman biasa disebut dengan istilah object, record, struct, tabel hash, key list atau associative array.
2. Nilai – nilai tersusun secara ordered list. Biasa disebut dengan array, vector, list atau daftar dalam bahasa pemrograman.

JSON adalah struktur data yang universal, bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON.

JSON menggunakan bentuk sebagai berikut:

A. Object

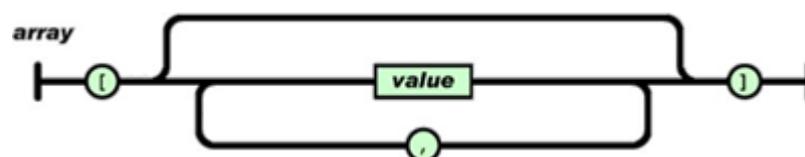
Objek adalah sepasang nama / nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma). Objek biasanya digunakan untuk menyimpan data tunggal dalam bentuk JSON.



Gambar 2.5 Object JSON

B. Array

Array adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma). Larik dalam JSON dapat digunakan sebagai value dari JSON object hal ini dapat berguna jika JSON menyimpan data bertingkat.

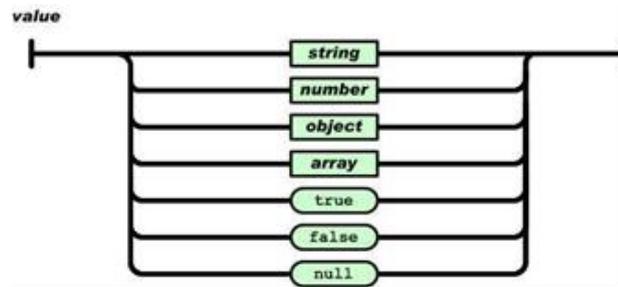


Gambar 2.6 Array JSON

Bentuk data JSON objek dan larik dapat saling dikombinasikan untuk mendukung struktur data yang lebih kompleks. JSON mendukung beberapa tipe data untuk menjadi *value* seperti Angka, *String*, *Boolean* dan nilai *NULL*.

C. Nilai (*Value*)

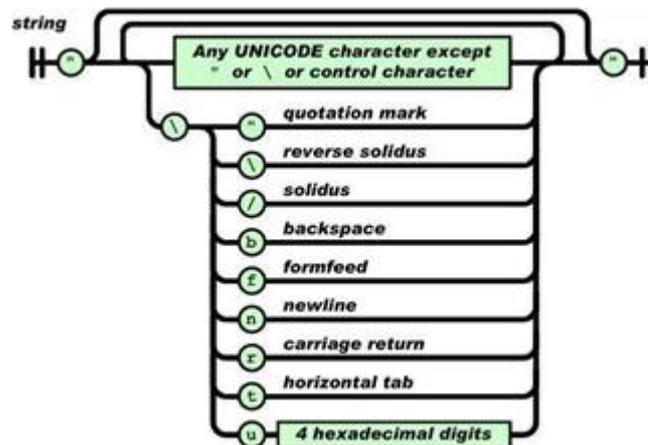
Nilai (*value*) dapat berupa sebuah *string* dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null* atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



Gambar 2.7 Nilai (Value) JSON

D. String

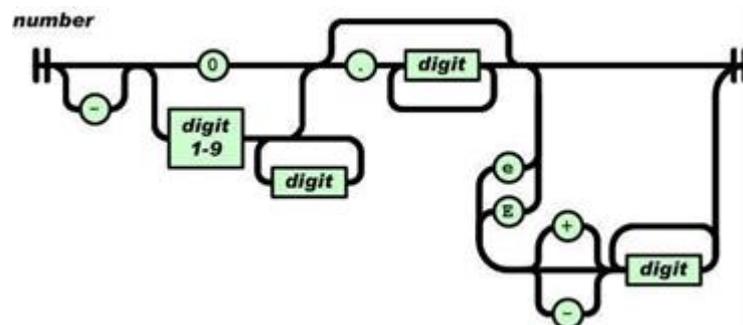
String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Pada *string* dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string*. *String* sangat mirip dengan *string* C atau Java .



Gambar 2.8 String JSON

E. Angka

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Gambar 2.9 Angka JSON

2.12 Pengujian Kuisisioner

Kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi beberapa pertanyaan atau pernyataan tertulis kepada responden untuk mendapatkan jawaban dari pertanyaan atau pernyataan tersebut. Metode yang digunakan dalam kuisisioner ini menggunakan skala Likert. Skala likert diciptakan oleh Rensis Likert untuk pengukuran sikap, pendapat, dan persepsi seseorang atau kelompok orang terhadap sesuatu, misalnya tentang fenomena sosial. Untuk setiap satu pernyataan responden memilih salah satu tanggapan berupa SS = Sangat Setuju, S = Setuju, R = Ragu, TS = Tidak Setuju, STS = Sangat Tidak Setuju. Tanggapan responden dikodekan kedalam bilangan 1 sampai dengan 5. Pemberian bilangan ini bergantung kepada bentuk pernyataan yang dijawab oleh responden [16].

Ada dua macam bentuk pernyataan dalam skala likert, pertama adalah pernyataan yang diharapkan untuk disetujui oleh responden, disebut bentuk positif. Yang kedua adalah pernyataan yang diharapkan untuk tidak disetujui oleh responden, bentuk negatif. Biasanya kuisisioner berisikan pernyataan positif dan negatif agar responden berpikir dulu sebelum memberi jawaban.

Tabel 2.2 Skala Likert

Jenis Pernyataan	Pernyataan	
	Positif	Negatif
Sangat Setuju	5	1

Setuju	4	2
Ragu	3	3
Tidak Setuju	2	4
Sangat Tidak Setuju	1	5

Hubungan pengujian kuisisionar dengan penelitian ini adalah pada panelitian ini menggunakan kuisisionare untuk mendapatkan jawaban dari responden berupa pernyataan positif dan negatif terhadap penelitian yang dilakukan. Dengan mengukur sikap, pendapat, dan persepsi seseorang orang mengenai suatu fenomena saat memilih pakaian.

2.13 Pengujian *Black Box*

Pengujian *Black Box* berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetestan pada spesifikasi fungsional program. *Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (interface errors).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (performance errors).
5. Kesalahan inisialisasi dan terminasi [17].

Saat ini terdapat banyak metoda atau teknik untuk melaksanakan *Black Box Testing*, antara lain:

1. Equivalence Partitioning
2. Boundary Value Analysis/Limit Testing
3. Comparison Testing
4. Sample Testing
5. Robustness Testing
6. Behavior Testing
7. Requirement Testing
8. Performance Testing

9. Uji Ketahanan (Endurance Testing)

10. Uji Sebab-Akibat (Cause-Effect Relationship Testing)

Alasan menggunakan Pengujian *Black Box* pada penelitian ini karena pengujian pada penelitian ini berfokus pada spesifikasi fungsional dari perangkat lunak. Pada pengujian Black Box dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak yang dibangun.

2.14 Sensor Proximity

Sensor proximity adalah sebuah sensor yang bisa mendeteksi keberadaan benda tanpa kontak fisik. Sensor proximity memancarkan medan elektromagnetik atau sinar radiasi elektromagnetik (misalnya inframerah) dan mendeteksi perubahan bidang dengan mengembalikan sinyal. Ada empat jenis teknologi sensor proximity, diantaranya Electrical (Inductive dan Capacitive), Optical (IR dan Laser), Magnetic, Sonar .

Dari beberapa jenis sensor proximity, jenis sensor proximity dengan tipe optical paling banyak digunakan, karena jenis sensor proximity dengan tipe optical lebih murah dan lebih sederhana dibandingkan dengan jenis sensor yang lainnya, dan bisa mendeteksi benda hingga jarak 5CM. Sensor proximity pada ponsel android biasanya terdapat di depan ponsel, tepatnya diatas layar ponsel.

Pada ponsel android sensor proximity biasa digunakan untuk mematikan layar saat terjadi panggilan masuk. Hal ini bertujuan untuk menghemat konsumsi baterai saat pengguna sedang melakukan panggilan telepon. Sensor proximity akan aktif otomatis dan mendeteksi ketika benda dalam jangkauan sensor, kemudian sensor akan mengirimkan informasi ke sistem ketika benda yang mendekati sensor, dan sistem akan mematikan layar selama benda dekat dengan sensor selama panggilan terjadi. Proximity Switch atau Sensor Proximity adalah alat pendeteksi yang bekerja berdasarkan jarak obyek terhadap sensor. Karakteristik dari sensor ini adalah mendeteksi obyek benda dengan jarak yang cukup dekat, berkisar antara 1 mm sampai beberapa centi meter saja sesuai type sensor yang digunakan. Proximity Switch ini mempunyai tegangan kerja antara 10-30 Vdc dan ada juga yang menggunakan tegangan 100-200VAC [18].

