

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Sistem Rekomendasi

Sistem rekomendasi dapat didefinisikan sebagai program yang mencoba untuk merekomendasikan *item* yang paling cocok (produk atau jasa) untuk pengguna tertentu (individu atau bisnis) dengan memprediksi minat pengguna di *item* berdasarkan informasi terkait tentang *item*, pengguna dan interaksi antara *item* dan pengguna [13]. Sistem rekomendasi bertujuan untuk mengurangi informasi yang berlebihan dengan mengambil informasi dan layanan yang paling relevan dari sejumlah besar data.

Tantangan utama sebuah sistem rekomendasi adalah:

1. *Data Sparsity*

Hal ini sangat jarang bahwa dua pengguna menilai item yang sama berkali-kali. Hal ini membuat lebih sulit untuk menghitung kesamaan.

2. *Cold start for user*

Pengguna yang belum melakukan rating terhadap beberapa barang dapat menyebabkan ia mendapatkan hasil yang tidak akurat. Hal ini mungkin terjadi karena pengguna lain yang memiliki kemiripan dengan pengguna ini tidak dapat ditemukan

3. *Cold start for item*

Item yang baru diperkenalkan memungkinkan tidak mendapatkan rating yang cukup, yang menyebabkan tidak direkomendasikan kepada pengguna.

4. *Attacks*

Sistem rekomendasi memiliki ancaman terhadap serangan, seperti menyalin keseluruhan profil pengguna dan membuat sistem berpikir bahwa si penyerang dan pengguna tersebut sangat mirip. Hal ini membuat penyerang dapat membodohi sistem dan membuatnya menyarankan item apapun yang di rating oleh penyerang kepada pengguna [13].

## 2.2 Makanan sehat

Makanan sehat adalah makan yang mengandung zat – zat yang dibutuhkan oleh tubuh dan harus memiliki beberapa syarat, yaitu higienis, bergizi dan berkecukupan, tetapi tidak harus makanan mahal dan enak [14]. Sangat penting untuk memahami hubungan antara makanan yang dikonsumsi dan dampaknya bagi kesehatan, agar masyarakat dapat memilih makanan sehat yang dibutuhkan oleh tubuh. Makanan dibutuhkan oleh tubuh sebagai sumber energi utama bagi tubuh. Makanan yang dikonsumsi sehari-hari memberikan energi bagi tubuh untuk melakukan aktivitas apapun setiap harinya.

Makanan sehat memiliki kandungan gizi yang seimbang antara Karbohidrat, protein, lemak dan vitamin ini berfungsi dalam menjaga badan tetap sehat, tumbuh dan berkembang dengan baik. Fungsi dari masing – masing zat gizi ini, yaitu:

1. Karbohidrat, berfungsi sebagai sumber energi utama untuk beberapa sel seperti sel saraf dan sel darah merah, Melindungi protein agar tidak dipakai sebagai penghasil energi.
2. Protein, berfungsi sebagai alat transportasi untuk molekul kecil dan ion – ion dan juga berperan dalam proses penyimpanan, proteksi tubuh dalam sistem imun, membantu dalam koordinasi gerak, sebagai penunjang tubuh seperti pada kulit dan tulang, membantu dalam proses enzimatik, menigirkan impuls saraf, dan dalam pertumbuhan sel.
3. Lemak, berfungsi sebagai sumber energi yang efektif 2 kali lebih besar dibandingkan dengan protein dan karbohidrat; insulasi panas sehingga menjaga tubuh tetap hangat; sebagai pelindung organ tubuh; membantu melarutkan vitamin A, D, E dan K; pembentukan Prostaglandin dan Sitokin; diferensiasi dan pertumbuhan sel, sebagai struktur membran sel dan myelinasi; pembentukan hormon dan asam empedu.
4. Vitamin, berfungsi untuk proses metabolisme, pertumbuhan yang normal dan membantu menghasilkan energi untuk mendukung kerja dari sel. Sumber dari vitamin ini dapat berasal dari makanan seperti sayuran dan buah buahan dan suplemen makanan.

### 2.3 Kebutuhan Kalori

Kalori adalah suatu unit pengukuran untuk menyatakan jumlah energi dalam makanan. Saat kita makan atau minum, kita memberi energi (kalori) pada tubuh kita. Tubuh kemudian memakai energi tersebut sebagai bahan bakar untuk berbagai aktivitas kita. Semakin banyak aktivitas yang kita lakukan, semakin banyak energi atau kalori yang terpakai [15]. Jumlah kalori dalam suatu makanan biasanya ditulis dalam satuan “kilokalori” atau “kkal”. Sebagai contoh, 500 kalori akan ditulis sebagai 500 kkal. Selain dalam kkal, kalori juga dapat ditulis dalam satuan “kilojoules” atau “kJ”. 1 kJ setara dengan 0,239 kalori.

Menurut Kementerian Kesehatan Republik Indonesia tahun 2014, Tabel kalori makanan pokok perpersi dapat dilihat pada Tabel 2.1:

**Tabel 2.1 Tabel Makanan Pokok Perpersi**

Nama Masakan	Kalori
Nasi Putih	175 kkal
Nasi Uduk	506 kkal
Jagung rebus	90,2 kkal
Kentang Rebus	166 kkal
Ketupat	32 kkal
Roti tawar	128 kkal
Singkong rebus	146 kkal

Tabel Makanan Lauk perpersi dapat dilihat pada Tabel 2.2 :

**Tabel 2.2 Tabel Makanan Lauk Perpersi**

Nama Masakan	Kalori
Ayam Panggang	16,3 kkal
Daging Panggang	150 kkal
Ikan Mas Pepes	143,5 kkal
Tahu Goreng	111 kkal
Telur Dadar	188 kkal

Tabel buah perpersi dapat dilihat pada Tabel 2.3 :

**Tabel 2.3 Tabel Buah Perpersi**

Nama Buah	Kalori
Apel	92 kkal
Belimbing	80 kkal
Jambu Air	35,4 kkal
Mangga	72 kkal
Nanas	104 kkal
Pepaya	46 kkal
Salak	63,6 kkal
Semangka	48 kkal

#### 2.4 Metode Harris Benedict

Komponen utama yang menentukan kebutuhan energi seseorang adalah Angka Metabolisme Basal (AMB) dan Indeks Masa Tubuh (IMT). Angka metabolisme basal adalah kebutuhan energi minimal yang dibutuhkan tubuh untuk menjalankan proses tubuh [16]. Angka metabolisme basal dinyatakan dalam satuan kilokalori (kkal). Angka metabolisme basal pada penelitian ini menggunakan rumus Harris-Benedict yang ditunjukkan pada Persamaan 1 dan 2 berikut :

$$AMB \text{ laki - laki} = 66 + (13,7 \times BB) + (5 \times TB) - (6,8 \times U) \quad (1)$$

atau :

$$AMB \text{ perempuan} = 655 + (9,6 \times BB) + (1,8 \times TB) - (4,7 \times U) \quad (2)$$

Keterangan :

AMB = Angka Metabolisme Basal

BB = Berat Badan dalam satuan kilogram (kg)

TB = Tinggi Badan dalam satuan centimeter (cm)

U = Usia

Nilai kebutuhan energi untuk AMB diperhitungkan menurut berat badan normal atau ideal dengan menggunakan Indeks Masa Tubuh (IMT) seperti pada Persamaan berikut:

$$IMT = BB / \left( \frac{TB}{100} \right)^2 \quad (3)$$

Keterangan :

IMT = Indeks Masa Tubuh

BB = Berat Badan dalam satuan kilogram (kg)

TB = Tinggi Badan dalam satuan centimeter (cm)

1m = 100 cm

Nilai IMT ini digunakan sebagai ambang batas kondisi tubuh. Kategori batas ambang IMT dapat dilihat pada Tabel 2.4

**Tabel 2.4 Kategori Batas Ambang IMT**

<b>Kondisi Tubuh</b>	<b>Kategori</b>	<b>Batas Ambang</b>
Kurus	Kekurangan berat badan tingkat berat	< 17,0
	Kekurangan berat badan tingkat ringan	17,0 – 18,5
Normal		18,5 – 25,0
Gemuk	Kelebihan berat badan tingkat ringan	25,0 – 27,0
	Kelebihan berat badan tingkat berat	>= 27,0

Bila nilai IMT kurang dari normal maka berat badan kurang dari ideal untuk itu kebutuhan energinya ditambah sebanyak 500 kilokalori. Jika nilai IMT lebih dari normal maka berat badan lebih dari ideal sehingga kebutuhan energinya dikurangi sebanyak 500 kilokalori.

## 2.5 Metode Perhitungan Kalori Terbakar

Setelah mendapatkan nilai AMB kemudian ditambahkan dengan perhitungan kalori terbakar untuk menentukan total dari kebutuhan energi hariannya. Pembakaran kalori dapat dilakukan dengan cara berjalan. Berjalan merupakan olahraga yang setiap hari kita lakukan, dengan hanya berjalan, kita dapat membakar kalori yang disertai dengan peningkatan aktivitas kerja jantung untuk memompa darah, dan meningkatkan aktivitas paru-paru dalam menyuplai oksigen [4].

Berikut ini merupakan perhitungan kalori terbakar berdasarkan jenis kelamin dan rata-rata lebar setiap langkah kaki.

### 1. Mencari Jarak Langkah Kaki Dalam Kilometer

Studi telah menunjukkan bahwa panjang langkah dibagi dengan tinggi badan dan hasilnya berada pada kisaran 0.41 sampai 0.45.

**Rasio (hasilnya pada kisaran 0.41 – 0.45) = panjang langkah / tinggi badan.**

Rasio rata-rata untuk subjek perempuan adalah 0.413 dan untuk subjek laki – laki adalah 0.415.

$$Jarak = JK * TB * \left( \frac{JLK}{100000} \right) \quad (4)$$

Keterangan:

Jarak = dalam satuan kilometer (km)

JK = Jenis Kelamin

= Jenis Kelamin Laki-laki ( 0,415 )

= Jenis Kelamin Perempuan ( 0,413 )

TB = Tinggi Badan dalam satuan centimeter (cm)

JLK = Jumlah Langkah Kaki

1 km = 100000 cm

### 2. Rumus Menghitung Kalori Terbakar

$$Kalori\ Terbakar = \left( \frac{BB}{0,453592} \right) * 0,30 * \left( \frac{Jarak}{1,60934} \right) \quad (5)$$

Keterangan:

BB = Berat Badan dalam satuan kilogram (kg)

1 lbs = 0.453592 kg

0.30 = nilai konstan

Jarak = hasil perhitungan rumus jarak dalam kilometer (km)

1 mil = 1.60934 km

Setelah mendapatkan nilai kalori terbakar, kebutuhan kalori dapat dihitung dengan menambahkan nilai AMB dan nilai kalori terbakar.

$$\text{Total Kebutuhan Kalori} = \text{AMB} + \text{Kalori Terbakar} \quad (6)$$

## 2.6 *Wearable Device*

*Wearable device* adalah teknologi elektronika atau komputer yang tergabung dalam sebuah benda yang dapat dikenakan atau hiasan yang nyaman digunakan di tubuh. *Wearable device* memiliki beberapa macam kemampuan komunikasi dan memungkinkan penggunanya untuk mengakses informasi secara *real – time* [17]. Contoh dari *wearable device* meliputi gelang, jam tangan, lensa kontak, e-tekstil, kain pintar, topi, perhiasan seperti anting, cincin dan lainnya.

## 2.7 *Smartband*

*Smartband* merupakan alat yang memiliki fungsi sebagai pelacak kesehatan untuk meningkatkan level kebugaran atau bisa disebut dengan *fitness tracker*. Alat ini lebih dititik poinkan pada program kesehatan dan kebugaran. Dengan *smartband* akan lebih memudahkan dalam mencapai kegiatan yang menunjang kesehatan dan juga kebugaran [18]. Disebut dengan *fitness tracker* karena *smartband* memiliki fitur-fitur yang berhubungan dengan kesehatan. Berikut adalah fitur fitur dari *smartband*:

1. Pemantauan tekanan darah
2. Pengukuran detak jantung
3. Penghitungan pembakaran kalori
4. Perekaman langkah kaki

5. Pengelolaan tidur
6. Tampilan data latihan
7. Jam alarm, dll.

### **2.7.2 Xiaomi Mi Band**

Xiaomi Mi Band merupakan gelang pintar keluaran Xiaomi yang memiliki beberapa fitur diantaranya fitur olahraga yaitu *Treadmill*, latihan, lari *outdoor*, bersepeda, jalan, berenang di kolam. Menghitung langkah, jarak, dan kalori yang terbakar. Kemudian Fitur Memantau kesehatan yaitu notifikasi saat mencapai target, setel target, memantau tidur, memantau detak jantung, memantau detak jantung sepanjang hari, detak jantung istirahat, kurva detak jantung, peringatan diam karena diam terlalu lama [19].

## **2.8 Step Detector Sensor**

*Step detector sensor* merupakan sensor yang terdapat pada *platform android* yang digunakan untuk menghitung langkah kaki pengguna. Sensor ini bekerja dengan prinsip *motion sensor* yang mendeteksi dan memicu peristiwa ketika pengguna mengambil langkah [5]. Secara umum *step detector sensor* lebih akurat dan memiliki latensi rendah 1 hingga 2 detik dalam melaporkan langkah – langkah.

## **2.9 Aplikasi**

Aplikasi merupakan suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju [20]. Menurut kamus computer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan.



Perkembangan aplikasi saat ini telah tersedia di beberapa *platform*, ada tiga *platform* utama pada pengembangan sebuah aplikasi, yaitu aplikasi berbasis *mobile*, aplikasi berbasis *web*, dan aplikasi berbasis *desktop*.

## 2.10 Android

Android adalah sebuah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi, middleware atau aplikasi [20]. Android merupakan sistem operasi smartphone yang paling banyak digunakan pada saat ini. Android menyediakan platform secara gratis atau terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Pada awalnya, Google Inc. membeli android Inc. yang merupakan pendatang baru dalam pengembangan peranti lunak untuk ponsel atau smartphone. Kemudian dalam pengembangan android ini, Google Inc. membentuk Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia [20].

Dalam perkembangannya, Android saat ini sudah sampai pada generasi ke 9, yaitu Android 9.0 Pie. Perkembangan versi android dari 2008 hingga sekarang dapat dilihat pada Tabel 2.5

**Tabel 2.5 Versi Android**

No	Versi Android	Nama
1	1.0	Astro
2	1.1	Bender
3	1.5	Cupcake
4	1.6	Donut
5	2.0 – 2.1	Éclair
6	2.2 – 2.23	Froyo
7	2.3 – 2.37	Gingerbread
8	3.0 – 3.2.6	Honeycomb
9	4.0 – 4.0.4	Ice Cream Sandwich
10	4.1 – 4.3.1	Jelly Bean

No	Versi Android	Nama
11	4.4 – 4.4.4	Kitkat
12	5.1 – 5.1.1	Lolipop
13	6.0 – 6.0.1	Marshmallow
14	7.1 – 7.1.2	Nougat
15	8.0 – 8.1	Oreo
16	9	Pie

Berikut adalah penjelasan versi android berdasarkan waktu rilisnya:

1. Android 1.0 Astro (Alpha)

Android 1.0, versi komersial pertama Android, dirilis pada 23 September 2008.

Perangkat Android pertama yang tersedia secara komersial adalah HTC Dream.

2. Android 1.1 Bender (Beta)

Pada 9 Februari 2009, pemutakhiran Android 1.1 dirilis, awalnya hanya untuk HTC Dream. Android 1.1 juga dikenal dengan "Petit Four", meskipun nama ini tidak digunakan secara resmi. Versi ini memperbaiki beberapa bug, mengubah API Android, dan menambahkan beberapa fitur.

3. Android 1.5 Cupcake

Pada 27 April 2009, Android 1.5 dirilis, menggunakan kernel Linux 2.6.27. Versi ini adalah rilis pertama yang secara resmi menggunakan nama kode berdasarkan nama-nama makanan pencuci mulut ("Cupcake"), nama yang kemudian digunakan untuk semua versi rilis selanjutnya. Pembaruan pada versi ini termasuk beberapa fitur baru dan perubahan UI.

4. Android 1.6 Donut

Pada 15 September 2009, SDK Android 1.6 – dinamai Donut – dirilis, berdasarkan kernel Linux 2.6.29. versi ini hadir untuk menutupi bug pada versi sebelumnya, sekaligus untuk penambahan beberapa fitur seperti misalnya dukungan untuk perangkat dengan ukuran layar yang lebih besar.

#### 5. Android 2.0 – 2.1 Éclair

Pada 26 Oktober 2009, SDK Android 2.0 – dinamai Eclair – dirilis, berbasis kernel Linux 2.6.29. Versi 2.0.1 rilis tanggal 3 desember 2009 dan versi 2.1 rilis tanggal 12 januari 2010. Beberapa penambahan fitur pada versi ini seperti Bluetooth, flash pada kamera, digital zoom pada kamera, multi touch, live wallpaper dan lainnya.

#### 6. Android 2.2 – 2.2.3 Froyo

Pada 20 Mei 2010, SDK Android 2.2 (Froyo, singkatan untuk frozen yogurt) dirilis, yang berbasis kernel Linux 2.6.32.

#### 7. Android 2.3 – 2.3.7 Gingerbread

Pada tanggal 6 Desember 2010, SDK Android 2.3 (Gingerbread) dirilis, berbasis kernel Linux 2.6.35. Versi 2.3.1 yang rilis desember 2010 dan versi 2.3.2 yang rilis januari 2011 berisi perbaikan bug pada Google Nexus S. Versi 2.3.3 sampai versi 2.3.7 rilis antara bulan februari 2011 sampai bulan september 2011.

#### 8. Android 3.0 – 3.2 Honeycomb

Pada 22 Februari 2011, SDK Android 3.0 (Honeycomb) – pembaruan pertama Android yang ditujukan hanya untuk komputer tablet – dirilis, berdasarkan kernel Linux 2.6.36. Perangkat pertama yang menggunakan versi ini adalah tablet Motorola Xoom, yang dirilis pada 24 Februari 2011. Versi 3.1 rilis tanggal 10 mei 2011. Versi 3.2 rilis tanggal 15 juli 2011. Google TV generasi pertama dan kedua menggunakan Honeycomb.

#### 9. Android 4.0 – 4.0.4 Ice Cream Sandwich

SDK Android 4.0.1 (Ice Cream Sandwich), berdasarkan kernel Linux 3.0.1, dirilis pada 19 Oktober 2011. Petinggi Google, Gabe Cohen, menyatakan bahwa Android 4.0 "secara teoretis kompatibel" dengan perangkat Android 2.3x yang diproduksi pada saat itu. Kode sumber untuk Android 4.0 tersedia pada tanggal 14 November 2011.

#### 10. Android 4.1 – 4.3 Jelly Bean

Google mengumumkan Android 4.1 (Jelly Bean) dalam konferensi Google I/O pada tanggal 27 Juni 2012. Berdasarkan kernel Linux 3.0.31, Jelly Bean adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuka pengguna (UI). Pembaruan ini diwujudkan dalam "Proyek Butter", perbaikan ini termasukantisipasi sentuh, triple buffering, perpanjangan waktu vsync, dan peningkatan frame rate hingga 60 fps untuk menciptakan UI yang lebih halus. Android 4.1 Jelly Bean dirilis untuk Android Open Source Project pada tanggal 9 Juli 2012. Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7, yang dirilis pada 13 Juli 2012.

Google berencana merilis Jelly Bean 4.2 pada sebuah acara di New York City pada 29 Oktober 2012, tapi dibatalkan karena Badai Sandy. Jelly Bean 4.2 didasarkan pada kernel Linux 3.4.0, dan pertama kali digunakan pada Nexus 4 LG dan Nexus 10 Samsung, yang dirilis pada 13 November 2012.

Google merilis Jelly Bean 4.3 pada 24 Juli 2013 di San Francisco. Kebanyakan perangkat Nexus menerima pembaruan dengan segera. Nexus 7 generasi kedua adalah perangkat pertama yang menggunakan sistem operasi ini. Sebuah pembaruan minor dirilis pada tanggal 22 Agustus 2013.

#### 11. Android 4.4 Kitkat

Google mengumumkan Android 4.4 KitKat (dinamai dengan izin dari Nestlé dan Hershey) pada 3 September 2013, dengan tanggal rilis 31 Oktober 2013. Sebelumnya, rilis berikutnya setelah Jelly Bean diperkirakan akan diberi nomor 5.0 dan dinamai 'Key Lime Pie'.

#### 12. Android 5.0 – 5.1 Lollipop

Android 5.0 pertama kali diperkenalkan di bawah codename "Android L" pada 25 Juni 2014 selama presentasi keynote pada konferensi pengembang Google I/O. Di samping Lollipop, presentasi difokuskan pada sejumlah *platform* Android yang berorientasi dan teknologi baru, termasuk Android TV, pada

*platform* Android Auto, dapat dipakai pada *platform* komputasi Android Wear, dan *platform* pelacakan kesehatan Google Fit.

Bagian dari presentasi didedikasikan untuk bahasa desain *cross-platform* baru yang disebut sebagai "material design". Memperluas pada "kartu" motif pertama kali terlihat di Google Now, adalah desain dengan peningkatan penggunaan layout berbasis grid, animasi dan transisi responsif, padding, dan efek kedalaman seperti pencahayaan dan bayangan.

1. Desain antarmuka (tampilan) yang dinamakan "Material Design".
2. 64-bit ART compiler
3. Project volta, yang berguna untuk meningkatkan daya hidup baterai 30% lebih tahan lama.
4. 'factory reset protection'. Fitur ini berguna ketika smartphone hilang, ia tidak bisa direset ulang tanpa memasukkan id google dan kata sandi (password).

### 13. Android 6.0 Marshmallow

Android Marshmallow memperkenalkan model izin yang didesain ulang: sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi. Sebuah sistem opt-in sekarang digunakan, di mana pengguna akan diminta untuk memberikan atau menolak izin individu (seperti kemampuan untuk mengakses kamera atau mikrofon) untuk aplikasi ketika mereka dibutuhkan. Aplikasi mengingat hibah izin mereka, dan mereka dapat disesuaikan oleh pengguna setiap saat. Model izin baru akan digunakan hanya oleh aplikasi yang dikompilasi untuk Marshmallow menggunakan kit pengembangan perangkat lunak (SDK) tersebut, sementara semua aplikasi lainnya akan terus menggunakan model izin sebelumnya.

Marshmallow juga memiliki skema manajemen daya baru bernama Doze yang mengurangi tingkat aktivitas aplikasi latar belakang saat perangkat menentukan bahwa itu tidak sedang aktif ditangani oleh pengguna, yang, menurut Google, menggandakan pemakaian baterai perangkat. Hal ini juga

memperkenalkan pilihan untuk mengatur ulang semua pengaturan jaringan, tersedia untuk pertama kalinya pada Android, yang membersihkan pengaturan terkait jaringan untuk WI-FI, *Bluetooth* dan koneksi seluler.

Android Marshmallow memberikan dukungan asli untuk pengenalan sidik jari, memungkinkan penggunaan sidik jari untuk membuka perangkat dan otentikasi Play Store dan pembelian Android Pay; API standar juga tersedia untuk melaksanakan otentikasi berbasis sidik jari dalam aplikasi lain. Android Marshmallow mendukung USB Type-C, termasuk kemampuan untuk menginstruksikan perangkat untuk mengisi daya perangkat lain melalui USB. Marshmallow juga memperkenalkan "pranala yang diverifikasi" yang dapat dikonfigurasi untuk membuka langsung dalam aplikasi tertentu mereka tanpa petunjuk pengguna lanjut.

Versi API Android yang disediakan oleh Marshmallow adalah 23. Alat pengembang Android Marshmallow tersedia di Pengelola SDK di bawah tingkat API "MNC".

#### 14. Android 7.0 – 7.1 Nougat

Android "Nougat" (kode nama N dalam pengembangan) adalah rilis 7.0 besar dari sistem operasi Android. Ini pertama kali dirilis sebagai pratinjau pengembang pada tanggal 9 Maret 2016, dengan gambar pabrik untuk perangkat Nexus saat ini, serta dengan "Program Beta Beta" baru yang memungkinkan perangkat yang didukung ditingkatkan versinya ke versi Android Nougat melalui over-the-air update. Rilis terakhir adalah pada tanggal 22 Agustus 2016. Pratinjau akhir pembuatannya dirilis pada tanggal 18 Juli 2016, dengan nomor bangunan NPD90G.

Pada tanggal 19 Oktober 2016, Google merilis Android 7.1.1 sebagai pratinjau pengembang untuk Nexus 5X, Nexus 6P dan Pixel C. Pratinjau kedua mulai tersedia pada 22 November 2016, sebelum versi final diluncurkan ke publik pada bulan Desember. 5, 2016

#### 15. Android 8.0 – 8.1 Oreo

Android Oreo adalah rilis utama ke 8 dari sistem operasi Android. Ini pertama kali dirilis sebagai preview pengembang pada tanggal 21 Maret 2017, dengan gambar pabrik untuk perangkat Nexus dan Pixel saat ini. Pratinjau pengembang terakhir dirilis pada tanggal 24 Juli 2017, dengan rilis stabil yang diharapkan pada bulan Agustus atau September 2017.

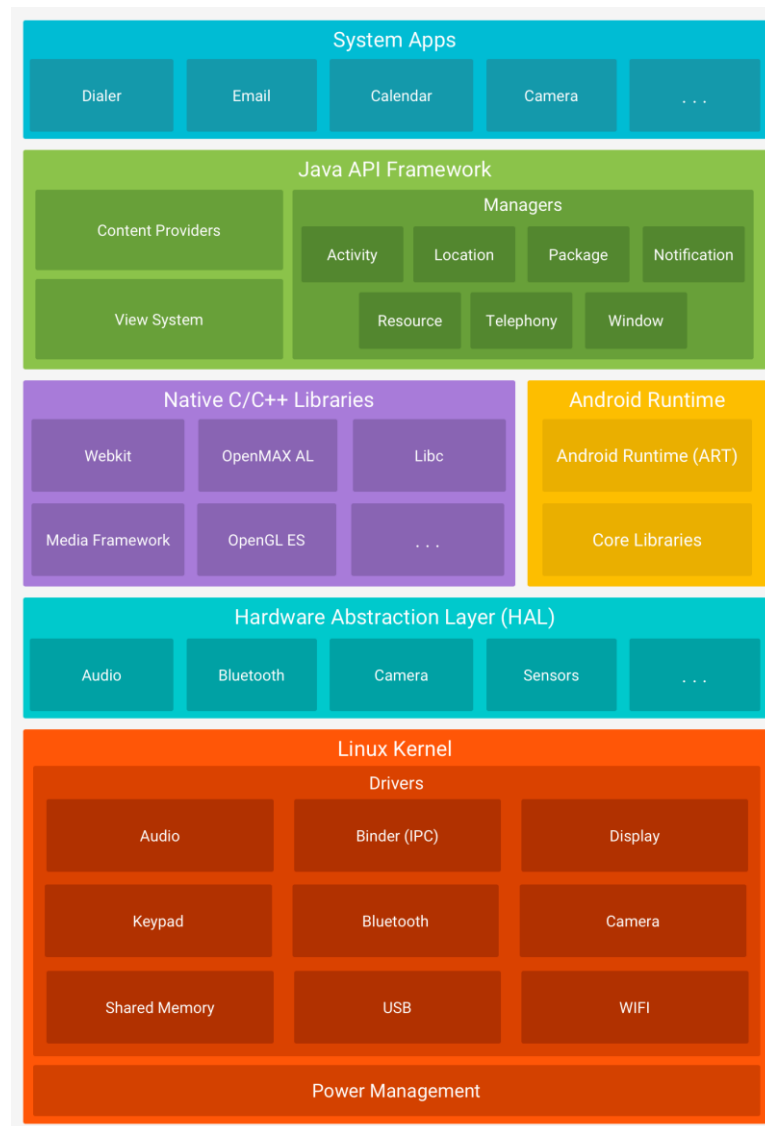
#### 16. Android 9.0 Pie

Terakhir ada Android 9.0 Pie yang secara resmi diperkenalkan pada Agustus 2018. Sistem operasi Android ini memberi banyak ubahan, terutama untuk HP dengan desain baru.

Misal Android 9.0 Pie memberikan navigasi berupa gesture yang menggantikan tombol fisik Home, Back dan Recent Apps. Fitur lainnya yang cukup berguna adalah sistem notifikasi, pengatur kecerahan hingga sistem screenshot terbaru yang lebih memudahkan.

#### **2.10.1 Arsitektur Android**

Android adalah tumpukan perangkat lunak berbasis *linux* sumber terbuka yang dibuat untuk berbagai perangkat dan faktor bentuk [21]. Android memiliki beberapa komponen utama yang disebut dengan Arsitektur Platform. Komponen utama pada android dapat dilihat pada Gambar 2.1.



Sumber gambar : <https://developer.android.com/guide/platform/index.html?hl=id>

[21]

**Gambar 2.1** Arsitektur Platform Android

Berikut merupakan penjelasan dari Gambar 2.1 :

a. Linux Kernel

Fondasi *platform* Android adalah kernel Linux. Sebagai contoh, *Android Runtime* (ART) bergantung pada kernel Linux untuk fungsionalitas dasar seperti threading dan manajemen memori tingkat rendah.



Menggunakan kernel Linux memungkinkan Android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen perangkat untuk mengembangkan driver perangkat keras untuk kernel yang cukup dikenal.

b. Hardware Abstraction Layer (HAL)

Hardware Abstraction Layer (HAL) menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras di perangkat ke kerangka kerja Java API yang lebih tinggi. HAL terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau *bluetooth*. Bila API kerangka kerja melakukan panggilan untuk mengakses perangkat keras, sistem Android memuat modul pustaka untuk komponen perangkat keras tersebut.

c. Android Runtime

Untuk perangkat yang menjalankan Android versi 5.0 (API level 21) atau yang lebih tinggi, setiap aplikasi menjalankan proses masing-masing dengan tahap Android Runtime (ART). ART ditulis guna menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi file DEX, *format* bytecode yang didesain khusus untuk Android yang dioptimalkan untuk footprint memori minimal. Buat rantai aplikasi, misalnya Jack, mengumpulkan sumber Java ke bytecode DEX, yang dapat berjalan pada *platform* Android.

Beberapa fitur utama ART mencakup:

1. Kompilasi mendahului waktu (AOT) dan tepat waktu (JIT).
2. Pengumpulan sampah (GC) yang dioptimalkan.
3. Dukungan debug yang lebih baik, mencakup *profiler* sampling terpisah, pengecualian diagnostik *mentail* dan laporan kerusakan dan kemampuan untuk mengatur titik pantau guna memantau bidang tertentu.

Sebelum ke Android versi 5.0 (API level 21), Dalvik adalah waktu proses Android. Jika aplikasi Anda berjalan baik pada ART, semestinya berfungsi baik juga pada Dalvik, tetapi mungkin tidak sebaliknya.

Android juga menyertakan serangkaian pustaka waktu proses inti yang menyediakan sebagian besar fungsionalitas bahasa pemrograman Java, termasuk beberapa fitur bahasa Java 8, yang digunakan kerangka kerja Java API.

#### d. Pustaka C/C++ Asli

Banyak komponen dan layanan sistem Android inti seperti ART dan HAL dibuat dari kode asli yang memerlukan pustaka asli yang tertulis dalam C dan C++. Platform Android memungkinkan kerangka kerja Java API mengekspos fungsionalitas beberapa pustaka asli pada aplikasi. Misalnya, Anda bisa mengakses OpenGL ES melalui kerangka kerja Java OpenGL API Android guna menambahkan dukungan untuk menggambar dan memanipulasi grafik 2D dan 3D pada aplikasi Anda.

Jika Anda mengembangkan aplikasi yang memerlukan kode C atau C++, Anda bisa menggunakan Android NDK untuk mengakses beberapa pustaka *platform* asli langsung dari kode asli.

#### e. Kerangka Kerja Java API

Keseluruhan rangkaian fitur pada Android OS tersedia untuk Anda melalui API yang ditulis dalam bahasa Java. API ini membentuk elemen dasar yang Anda perlukan untuk membuat aplikasi Android dengan menyederhanakan penggunaan kembali inti, komponen dan layanan sistem modular, yang menyertakan berikut ini:

1. Tampilan Sistem yang kaya dan luas bisa Anda gunakan untuk membuat UI aplikasi, termasuk daftar, kisi, kotak teks, tombol, dan bahkan browser *web* yang dapat disematkan.
2. Pengelola Sumber Daya, memberikan akses ke sumber daya bukan kode seperti string yang dilokalkan, grafik, dan file layout.

3. Pengelola Notifikasi yang mengaktifkan semua aplikasi guna menampilkan lansiran khusus pada bilah status.
4. Pengelola Aktivitas yang mengelola daur hidup aplikasi dan memberikan back-stack navigasi yang umum.
5. Penyedia Materi yang memungkinkan aplikasi mengakses data dari aplikasi lainnya, seperti aplikasi Kontak, atau untuk berbagi data milik sendiri.

Developer memiliki akses penuh ke API kerangka kerja yang sama dengan yang digunakan oleh aplikasi sistem Android.

#### f. Aplikasi Sistem

Android dilengkapi dengan serangkaian aplikasi inti untuk email, perpesanan SMS, kalender, menjelajahi internet, kontak, dll. Aplikasi yang disertakan bersama *platform* tidak memiliki status khusus pada aplikasi yang ingin dipasang pengguna. Jadi, aplikasi pihak ketiga dapat menjadi browser *web* utama, pengolah pesan SMS atau bahkan keyboard utama (beberapa pengecualian berlaku, seperti aplikasi Settings sistem).

Aplikasi sistem berfungsi sebagai aplikasi untuk pengguna dan memberikan kemampuan kunci yang dapat diakses oleh developer dari aplikasi mereka sendiri. Misalnya, jika aplikasi Anda ingin mengirimkan pesan SMS, Anda tidak perlu membangun fungsionalitas tersebut sendiri—sebagai gantinya Anda bisa menjalankan aplikasi SMS mana saja yang telah dipasang guna mengirimkan pesan kepada penerima yang anda cantumkan.

## 2.11 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (*Integrated Development Environment/IDE*) resmi untuk pengembangan aplikasi android, yang didasarkan pada IntelliJ IDEA [22]. Selain sebagai editor kode dan fitur developer IntelliJ yang handal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat aplikasi android, seperti:

1. Sistem build berbasis *Gradle* yang fleksibel

2. *Emulator* yang cepat dan kaya fitur
3. Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android
4. Terapkan Perubahan untuk melakukan *push* pada perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
5. *Template* kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
6. *Framework* dan fitur pengujian yang lengkap
7. Fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine*.

## 2.12 Java

Java menurut definisi dari Sun adalah mana untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *stand alone* ataupun pada lingkungan jaringan.

Bahasa java bersifat *case sensitive*, sehingga Anda harus memperhatikan penggunaan huruf besar dan kecil. Selain itu penulisan *source code* program tidak harus memperhatikan bentuk tertentu, sehingga Anda bisa saja menuliskan semua baris *source code* tersebut dalam satu baris asal Anda tidak lupa membubuhkan tanda titik koma (;), atau menuliskan tiap kata dalam satu baris tersendiri.

Bahasa java memiliki kelemahan dan keunggulan. Kelemahan bahasa java adalah sebagai berikut :

1. Mudah didekompilasi

Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan *bytecode* yang menyimpan

banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data.

## 2. Penggunaan memori tidak banyak

Penggunaan memori untuk program berbasis java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal).

## 3. Implementasi J2ME tidak global

Misalnya, J2ME untuk Motorola dengan J2ME untuk Sony Ericson tidak sama. Berbeda lagi J2ME untuk Nokia. Setiap produk selalu mempunyai modul tersendiri yang dinilai aneh penerapannya dan harus di-compile dengan modul yang berbeda-beda.

Sedangkan keunggulan bahasa java adalah sebagai berikut :

### 1. Platform independent

Salah satu keunggulan java adalah sifatnya yang '*platform independence*', artinya Java – baik *source* program maupun hasil kompilasinya – sama sekali tidak bergantung kepada sistem operasi dan *platform* yang digunakan.

### 2. Sederhana dan berorientasi objek

Java lahir dari suatu pemikiran mendalam akan bahasa pemrograman yang ada pada saat itu, seperti C dan C++. Hal ini akan memudahkan programmer profesional untuk dapat mengerti lebih jelas tentang Java, fungsionalitas, dan lain sebagainya apabila ia memiliki pengetahuan dasar tentang C++ dan konsep pemrograman berorientasi objek.

### 3. *Automatic garbage collection*

*Automatic garbage collection* atau pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++) yang dipakai secara luas.

### 4. Menghilangkan pewarisan berganda yang terdapat pada C++

Walaupun kelihatannya lebih sebagai suatu kekurangan, namun banyak para ahli yang mengakui bahasa konsep pewarisan berganda lebih banyak mengakibatkan kerugian dari pada keuntungan. java telah didesain sedemikian rupa sehingga Anda tidak akan memerlukan tehnik ini dalam pembuatan program apa pun.

#### 5. Mengurangi *pointer* aritmetik

Pengaksesan lokasi memori secara langsung dengan menggunakan *pointer* memungkinkan program untuk melakukan suatu tindakan yang tidak seharusnya atau tidak boleh dilakukan.

#### 6. *Library* yang lengkap

Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya

#### 7. Karena OOP jadi sangat cocok bila digunakan untuk membangun program yang besar.

OOP (*Object Oriented Programming* – Pemrogram Berorientasi Objek) yang artinya semua aspek yang terdapat di java adalah Objek. java merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut *Object*.

#### 8. Bergaya C++

Memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke java. Saat ini pengguna java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke java.

### 2.13 Java Development Kit (JDK)

*Java Development Kit* (JDK) adalah sekumpulan perangkat lunak yang dapat digunakan untuk mengembangkan perangkat lunak berbasis Java. Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber compiler java, bulding, debuggers, development libraries dan lain sebagainya. JRE

adalah sebuah implementasi dari *Java Virtual Machine* yang benar-benar digunakan untuk menjalankan program Java [20].

#### 2.14 *Web Service*

*Web Service* adalah sekumpulan aplikasi logik beserta objek-objek dan metode-metode yang dimilikinya, yang terletak disuatu server yang terhubung ke internet sehingga dapat diakses menggunakan protokol HTTP (*Hypertext Transfer Protocol*) maupun SOAP (*Simple Object Access Protocol*) [20].

Dalam penggunaannya, *web service* dapat digunakan untuk memeriksa data user yang *login* ke sebuah *website* ataupun untuk digunakan pada transaksi perbankan online yang rumit. Tujuan dari *web service* adalah untuk memudahkan beberapa aplikasi kompenennya untuk saling terhubung dengan aplikasi lain dalam sebuah organisasi maupun diluar organisasi menggunakan standar yang tidak terikat dengan *platform* dan tidak terikat dengan bahasa pemrograman yang digunakan [20].

#### 2.15 *Node.js*

Node.js adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi berbasis *web*. Node.js ditulis dalam bahasa *Javascript* dengan menggunakan basis *event* dan *asynchronous I/O*. Node.js tidak seperti *Javascript* pada umumnya yang dijalankan pada peramban melainkan berjalan pada sisi server. Node.js terdiri dari *V8 Javascript Engine* buatan Google dan beberapa modul bawaan yang terintegrasi [23].

Modul-modul yang digunakan dalam implementasi klien SIP ini antara lain *Sip.js* sebagai implementasi protokol SIP pada node.js, *Websocket-Node* yang merupakan implementasi *Websocket* pada Node.js dan *Express* yang merupakan kerangka kerja HTTP pada Node.js [23].

Berikut adalah contoh *source code* pada Node.js [24] :

```
var http = require('http');  
var server = http.createServer(function (req, res) {
```

```

    res.end("Hi, selamat datang di nodejs");
  });
  server.listen(8000);
  console.log("server          running          on
  http://localhost:8000");
node server.js

```

## 2.16 MySQL

MySQL adalah suatu perangkat lunak *database* relasi atau *Relational Database Management System* (RDBMS) yang didistribusikan gratis di bawah lisensi GPL (*General Public License*) [25]. MySQL merupakan *database* yang bersifat gratis digunakan oleh siapa saja, akan tetapi tidak diperbolehkan untuk dijadikan produk turunan yang dijadikan closed source atau komersial.

Berikut adalah contoh *source code* pada MySQL :

```

/* Membuat Database */
CREATE DATABASE peminjaman_buku;

```

```

/* Melihat Database */
SHOW DATABASES;

```

```

/* Membuat Tabel */
CREATE TABLE operator(
  id VARCHAR (20) NOT NULL,
  nama VARCHAR (50) NOT NULL,
  password VARCHAR(100) NOT NULL,
  created_at DATETIME NOT NULL,

```



```
    updated_at TIMESTAMP,  
    PRIMARY KEY (id)  
);
```

```
/* Melihat Tabel */  
SHOW TABLE;
```

```
/* Menghapus Tabel */  
DROP TABLE operator;
```

```
/* Menghapus Database */  
DROP DATABASE peminjaman_buku;
```

## 2.17 JSON

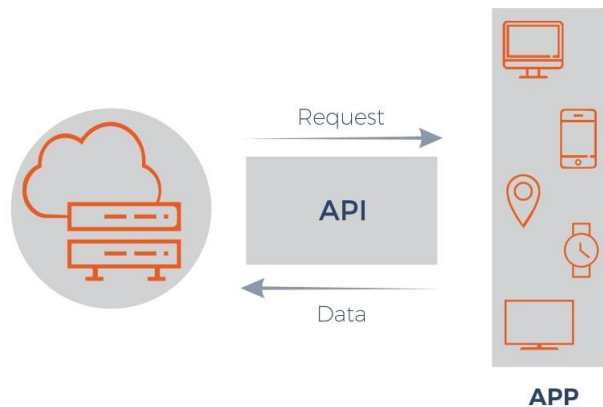
JSON (*Javascript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. [26].

JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data. JSON terbuat dari dua struktur [26] :

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (object), rekaman (record), struktur (struct), kamus (dictionary), tabel hash (hash table), daftar berkunci (keyed list), atau associative array.
2. Daftar nilai terurutkan (an ordered list of values). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (array), vektor (vector), daftar (list), atau urutan (sequence).

### 2.18 *Application Programming Interface (API)*

*Application Programming Interface (API)* adalah seperangkat definisi subrutin, protokol, dan alat untuk membangun perangkat lunak aplikasi. API yang bagus memudahkan pengembangan program komputer dengan menyediakan semua blok bangunan, yang kemudian disatukan oleh pemrogram. API mungkin untuk sistem berbasis *web*, sistem operasi, sistem basis data, perangkat keras komputer atau perpustakaan perangkat lunak.



Sumber gambar : <https://qatestlab.com/resources/knowledge-center/application-programming-interface/> [27]

**Gambar 2.2 *Application Programming Interface (API)***

## 2.19 Google Maps API

Google Maps adalah layanan pemetaan berbasis *web service* yang disediakan oleh Google dan bersifat gratis, yang memiliki kemampuan terhadap banyak layanan pemetaan berbasis *web*. Google Maps juga memiliki sifat *server side*, yaitu peta yang tersimpan pada *server* Google dapat dimanfaatkan oleh pengguna. Google Maps API adalah suatu library yang berbentuk *Javascript* yang berguna untuk memodifikasi peta yang ada di Google Maps sesuai kebutuhan. Untuk membangun aplikasi yang memanfaatkan Google Maps di desktop dan mobile device maka akan digunakan Google Maps *Javascript* API v3 yang memiliki keunggulan lebih cepat dari versi sebelumnya [28].

## 2.20 Google Direction API

Google Direction API merupakan layanan untuk menentukan arah antar lokasi dengan memanfaatkan HTTP. Arah antar lokasi antara lain tempat dimana pengguna berada, tempat yang akan dituju, atau koordinat penentu *latitude* (garis lintang) dan *longitude* (garis bujur). Google Directions API dapat mengembalikan beberapa directions dengan menggunakan satu garis titik lokasi (waypoints). Pengguna directions API dibatasi untuk 2.500 titik permintaan dengan 8 titik directions per hari dan juga directions URL dibatasi 2.048 karakter [29].

## 2.21 Firebase

Firebase adalah API yang disediakan google untuk penyimpanan dan penyelarasan data ke dalam aplikasi Android, iOS, atau web. Realtimedatabase adalah salah satu fasilitas yang menyimpan data ke database dan mengambil data darinya dengan sangat cepat tetapi firebase bukan hanya realtimedatabase, jauh lebih dari itu. Firebase memiliki banyak fitur seperti authentication, database, storage, hosting, pemberitahuan dan lain-lain [30].

Firebase memiliki beberapa fitur yang menarik yaitu

1. *Analytics*

Fitur ini digunakan untuk keperluan koleksi data dan reporting untuk aplikasi android / IOS. Dengan fitur ini dapat melihat bagian mana dari sebuah aplikasi yang sering digunakan oleh user.

#### 2. *Cloud Messaging and Notifications*

Fitur ini memberikan *push notification* dan membuat komunikasi dua arah antara *device*.

#### 3. *Firebase Remote Config*

Fitur ini memberikan akses untuk melakukan perubahan konfigurasi di dalam aplikasi Android / iOS, tanpa harus melakukan update aplikasi di Play Store / App Store.

#### 4. *Crash Reporting*

Fitur ini digunakan untuk keperluan merekam setiap exception yang terjadi pada aplikasi itulah *Crash Reporting*.

#### 5. *Realtime Database*

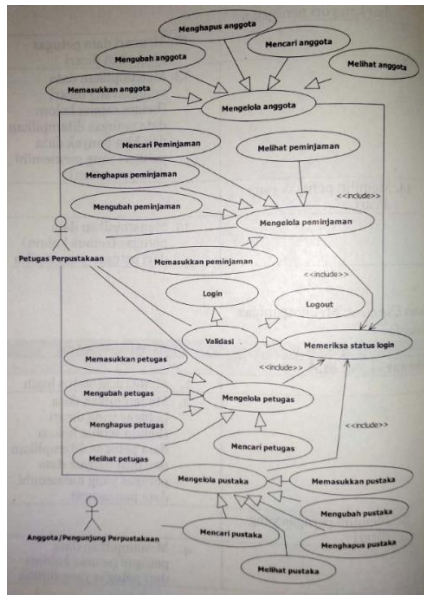
Fitur ini memungkinkan aplikasi yang kita kembangkan bisa diakses secara langsung oleh pengguna. Aplikasi juga bisa menyimpan data secara lokal ketika tidak ada akses internet, kemudian melakukan *syncdata* segera setelah mendapatkan akses internet.

## 2.22 *Unified Modelling Language (UML)*

*Unified Modelling Language (UML)* adalah standar bahasa yang banyak digunakan di dunia industry untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [31].

### 2.22.1 *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk kelakuan sistem informasi (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu [31]

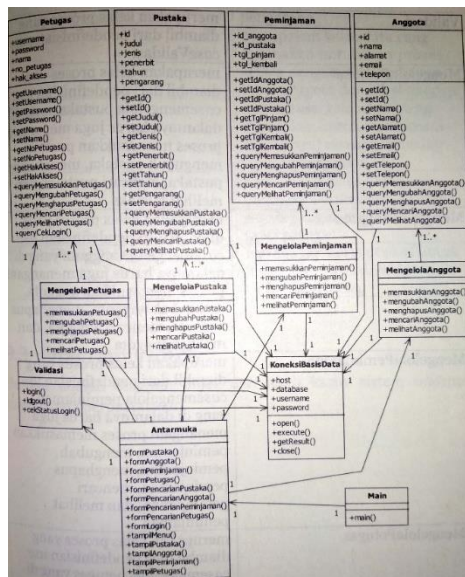


Sumber gambar : Book Rekayasa Perangkat Lunak (2018) [31]

Gambar 2.3 Use Case Diagram

2.22.2 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem.

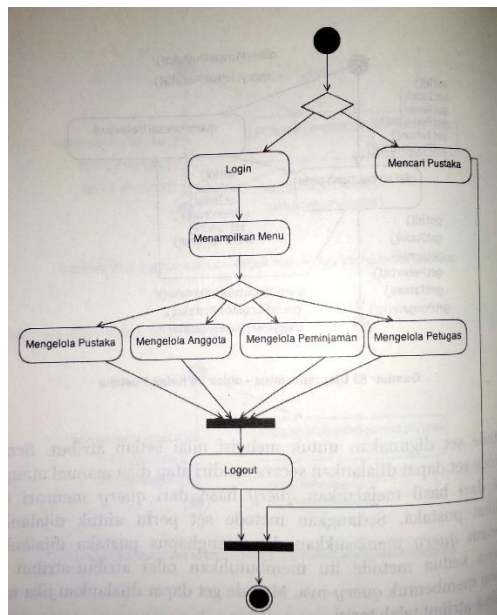


Sumber gambar : Book Rekayasa Perangkat Lunak (2018) [31]

Gambar 2.4 Class Diagram

### 2.22.3 Activity Diagram

*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah bahwa *activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem [31].

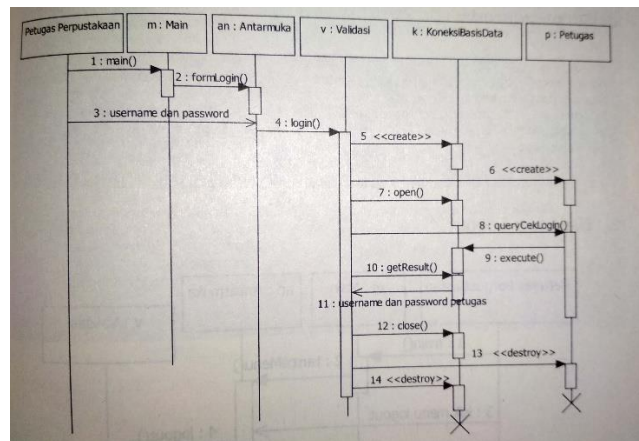


Sumber gambar : *Book Rekayasa Perangkat Lunak (2018)* [31]

**Gambar 2.5 Activity Diagram**

### 2.22.4 Sequence Diagram

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Untuk menggambarkan *sequence diagram* maka harus diketahui objek – objek yang terlibat dalam sebuah *use case* beserta metode – metode yang dimiliki kelas yang diinstansiasi menjadi objek itu [31].



Sumber gambar : *Book Rekayasa Perangkat Lunak (2018)* [31]

**Gambar 2.6 Sequence Diagram**

## 2.23 Pengujian

### 2.23.1 Blackbox Testing

*Blackbox testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Blackbox testing* mempunyai ciri-ciri yaitu :

1. *Blackbox testing* berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari software.

Black box testing melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. Juga disebut sebagai behavioral testing, specification-based testing, input/output testing atau functional testing [32].

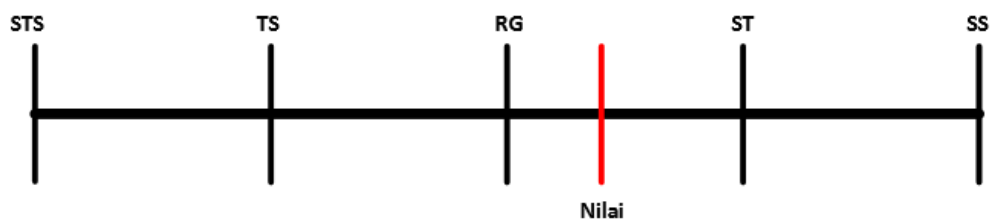
### 2.23.2 Skala Likert

Skala likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang kejadian atau gejala sosial yang telah ditetapkan secara spesifik oleh peneliti (selanjutnya disebut dengan variable penelitian). Dengan menggunakan skala likert, variable yang akan diukur dijabarkan menjadi dimensi. Dimensi lalu dijabarkan menjadi sub variable. Kemudian, subvariable dijabarkan menjadi indicator indikator yang dapat diukur.

Akhirnya, indikator-indikator terukur dapat dijadikan titik tolak untuk membuat item instrument yang berupa pertanyaan atau pernyataan yang perlu dijawab oleh responden. Setiap jawaban dihubungkan dengan bentuk pernyataan atau dukungan sikap yang diungkapkan dengan kata-kata sebagai berikut:

- i. Pernyataan sangat tidak setuju (1)
- ii. Pernyataan tidak setuju (2)
- iii. Pernyataan ragu-ragu (3)
- iv. Pernyataan setuju (4)
- v. Pernyataan sangat setuju (5)

Dengan data yang direkap dari pengumpulan data angket yang didapatkan data dan di skor perhitungkan dengan cara sebagai berikut. Jumlah penjawab dikalikan dengan pernyataan sangat tidak setuju, tidak setuju, ragu-ragu, setuju dan sangat setuju, lalu hasil dari total nilai yang didapat (diberi contoh a) dari pernyataan dibagi dengan skor maksimal (diberi contoh y) dan dikalikan dengan 100% untuk mendapatkan persentase tingkat persetujuan dari suatu pernyataan. Seperti berikut.  $= (a \times y) : 100\% = x\%$ , dan hasil dituangkan pada gambar berikut.



**Gambar 2.7 Skala likert**

Jadi berdasarkan data yang diperoleh dari x responden maka rata-rata (a) yang berwarna merah terletak pada daerah setuju [33].