BAB 2
LANDASAN TEORI

2.1 Makanan

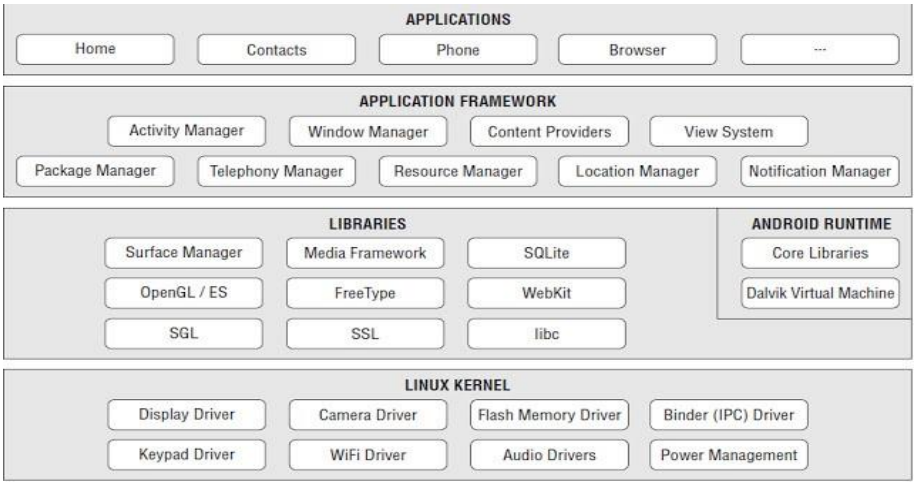
Makanan adalah bahan, biasanya berasal dari hewan atau tumbuhan, yang dimakan oleh makhluk hidup mendapatkan tenaga dan nutrisi. Cairan yang dipakai untuk maksud ini sering disebut minuman. Makanan yang dibutuhkan manusia biasanya diperoleh dari hasil bertani atau berkebun yang meliputi sumber hewan, dan tumbuhan. Beberapa orang menolak untuk memakan makanan dari hewan seperti, daging, telur, dan lain-lain. Mereka yang tidak suka memakan daging, dan sejenisnya disebut *vegetarian* yaitu orang yang hanya memakan sayuran sebagai makanan pokok mereka.

2.2 Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile,dan Nvidia [4].

2.2.1 Arsitektur Android

Arsitektur android dan secara garis besar Arsitektur Android dapat dijelaskan seperti pada gambar 2.1 :



Gambar 2. 1 Arsitektur Android [5]

1. Application dan Widget

Inilah layer pertama pada OS Android, biasa dinamakan layer Applications dan Widget. Layer ini merupakan layer yang berhubungan dengan aplikasi-aplikasi inti yang berjalan pada Android OS. Seperti klien email, program SMS, kalender, browser, peta, kontak, dan lain-lain. Semua aplikasi ini dibuat dengan menggunakan bahasa Java. Apabila kalian membuat aplikasi, maka aplikasi itu ada di layer ini.

2. Application Frameworks

Applications Framework merupakan layer dimana para pembuat aplikasi menggunakan komponen-komponen yang ada di sini untuk membuat aplikasi mereka. Beberapa contoh komponen yang termasuk di dalam Applications Framework adalah sebagai berikut :

2. Views
3. Content Provider
4. Resource Manager
5. Notification Manager
6. Activity Manager

3. Libraries

Libraries ini adalah layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan

diatas Kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc SSL, serta:

1) *Libraries* media untuk pemutaran media audio dan video
2) *Libraries* untuk manajemen tampilan
3) *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D
4) *Libraries* SQLite untuk dukungan database
5) *Libraries* SSL dan WebKit terintegrasi dengan web browser dan security
6) *Libraries* LiveWebcore mencakup modern web browser dengan engine embedded web view
7) *Libraries* 3D yang mencakup implementasi OpenGL ES1.0 API's.

4. Android Run Time

Android RunTime merupakan layer yang membuat aplikasi android bisa dijalankan. Android RunTime dibagi menjadi dua bagian yaitu:

1. Core Libraries

   Aplikasi Android dibangun dalam bahasa Java, sementara Dalvik sebagai virtual mesinnya bukan Virtual Machine Java, sehingga diperlukan sebuah libraries yang berfungsi untuk menerjemahkan bahasa java/C yang ditangani oleh Core Libraries.

2. Dalvik Virtual Machine

   Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat Linux Kernel untuk melakukan threading dan manajemen tingkat rendah

5. Linux Kernel

Linux Kernel merupakan layer tempat keberadaan inti dari operating system android. Layer ini berisi file-file system yang mengatur system processing, memory, resource, drivers, dan sistem android lainnya. Inilah yang membuat file sistem pada Android mirip dengan file sistem pada sistem operasi berbasis Linux.

Kernel yang digunakan adalah kernel Linux versi 2.6, dan versi 3.x pada Android versi 4.0 ke atas. Kernel ini berbasis *monolithic*.

2.2.2 Fitur

Android memiliki fitur utama dalam membangun aplikasi yang sering digunakan sebagai berikut :

1. Multi-proses dan App Widget

   Sistem operasi android tidak melarang prosesor menjalankan lebih dari satu aplikasi dalam satu waktu. Sistem operasi android dapat mengatur aplikasi dan thread yang berjalan secara multitasking. Keuntungan yang didapat adalah ketika aplikasi berjalan dan berinteraksi dengan pengguna di layer depan sistem operasi, proses dari aplikasi lain dapat berjalan untuk melakukan pembaruan informasi. App Widget adalah mini aplikasi yang dapat embedded dalam aplikasi seperti home screen. *Multi-proses* dapat memberikan manfaat berupa *user experience* yang lebih banyak, namun pengguna fitur tersebut dapat menghabiskan banyak energi baterai jika penggunaan tidak benar [6].

2. Touch Gestures dan Multi-Touch

   Touchscreen adalah user interface intuitif yang digunakan banyak smartphine di dunia. Dengan fitur ini interaksi dapat dibuat lebih mudah karena cukup dengan menggunakan jari tangan. Multi-touch adalah kemampuan yang digunakan untuk interaksi memperbesar atau memutar objek [7].

3. Hard dan Soft Keyboard

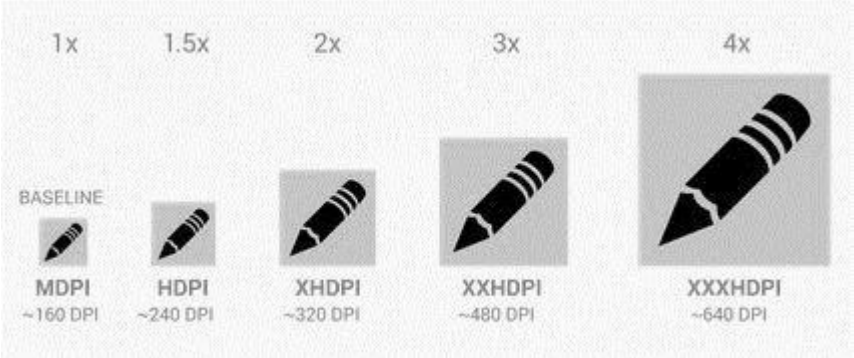
   Salah satu fitur pada perangkat smartphone adalah tombol fisik dan non fisik, tombol fisik digunakan untuk navigasi pendukung dalam pengoprasian android. Pengembang aplikasi tidak perlu secara manual untuk mengintegrasikan tombol tersebut dalam aplikasi [8].

2.2.3 Prinsip Desain

Android memiliki beberapa prinsip desain yang dapat menjadi acuan dalam membuat desain aplikasi android diantaranya berikut :

1. Multiple Assets

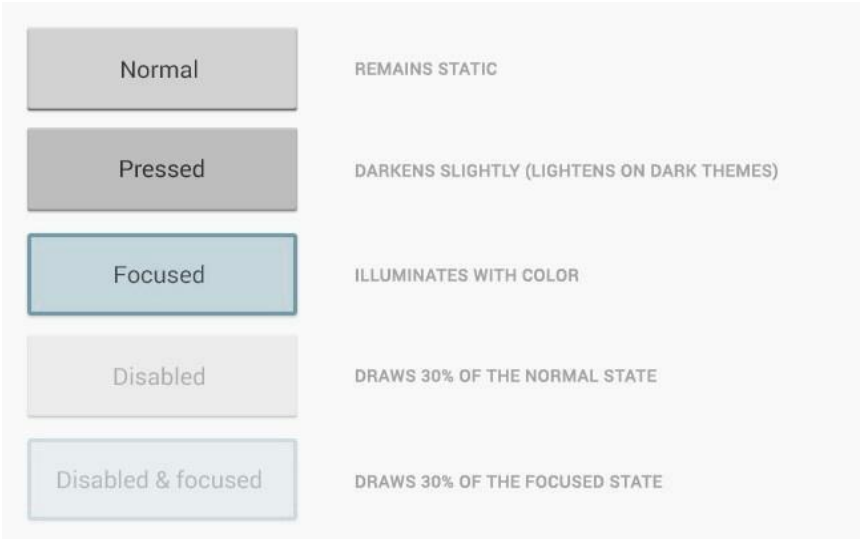
Android mendukung jutaan *smartphone,* tablet dan perangkat lain dalam berbagai ukuran layar dan ukuran, untuk itu *Multiple Assets* sangat disarankan digunakan untuk mengatasi fragmentasi pada android. Klasifikasi Ukuran Ikon, android menciptakan beberapa klasifikasi ukuran icon yaitu MDPI, HDPI, XHDPI, XXHDPI dan XXXHDPI [9].



Gambar 2. 2 Klasifikasi ukuran Ikon

2. Touch Feedback

Touch Feedback dalam android digunakan sebagai repon setiap objek yang ditekan pengguna. Hal ini bertujuan untuk memberi tahu pengguna objek mana yang berinteraksi dengan pengguna [10].



Gambar 2. 3 Touch Feedback

3. Pattern Gesture

Touch gesture berguna untuk meningkatkan *experience* pengguna dalam menggunakan aplikasi. Terdapat beberapa *gesture* yang didukung oleh android diantaranya adalah [11] :

1. Touch

   Touch adalah *gesture default* sebagai *event* untuk item tertentu yang paling sering digunakan dalam perangkat android, hampir seluruh kontrol utama aplikasi ini menggunakan *touch*.

2. Swipe or Drag

   Swipe adalah menyentuh sebuah titik pada layar dan menggerakkan jari yang tetap tersentuh pada layar ke titik lain pada layar.

3. Pich Open

   Teknik lain yang biasa digunakan adalah dengan menggunakan dua jari, di mana kedua jari tersebut menyentuh dua titik pada layar yang terpisah di mana ujung dari dua jari terseut tidak bersentuhan, kemudian kedua jari tersebut sambil teteap menyentuh layar bergerak saling mendekati.

4. Long Press

   Digunakan untuk seleksi data, dimungkinkan untuk memilih satu atau lebih item dalam sebuah tampilan dan menjalankan suatu fungsionalitas tertentu.

5. Double Touch

   Pada android, teknik ini biasanya dipakai untuk melakukan *zoom in* atau memperbesar dan *zoom out* atau memperkecil sebuah objek gambar.

6. Pich Close

   Dimana *spread* dilakukan dengan berawal dua jari bersentuhan pada ujungnya dan ditempelkan sebuah titik yang sama pada layar, kemudian kedua jari tersebut bergerak memisahkan atau menjauhi satu sama lain. Gerakan ini untuk menutup layar.

2.2.4 Android Life Cycle

Apikasi android terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android

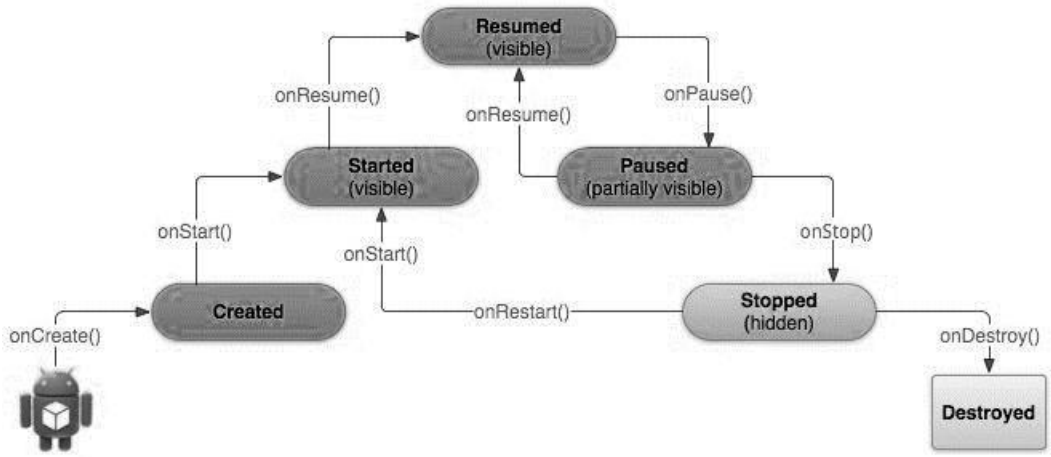
yang berbeda seperti ditunjukan pada tabel, klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan [12] :

Tabel 2. 1 Komponen Aplikasi Android

Functionality	Java Base Class	Example
Focused thing a user can do	Activity	Edit a note, play a game
Background process	Service	Play music, update weather icon
Receive messages	BroadcastReceiver	Trigger alarm upon event
Store and retrive data	ContentProvider	Open a phone contact

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan mencantumkan specified permission sebelum digunakan seperti komponen *Service, Broadcast Receiver, Content Provider*.

Android memiliki paradigma pemrograman lain tidak seperti paradigma pemrograman biasa di mana aplikasi yang dijalankan pada fungsi main(), sistem android menjalankan kode dalam method Activity dengan menerapkan metode *callback* tertentu yang sesuai dengan tahap tertentu dari siklus hidup. Setiap aplikasi yang berjalan dalam sistem operasi android memiliki siklus hidup yang berbeda dengan aplikasi dekstop atau web. Hal ini dikarenakan aplikasi mobile memiliki tingkat interupsi proses yang lumayan tinggi seperti ketika *handling* panggilan masuk aplikasi diharuskan menghentikan proses sementara. Penerapan siklus hidup juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai pengguna [12].



Gambar 2. 4 Siklus Hidup Android

Siklus sesungguhnya berisi dengan masing-masing state yang sebenarnya adalah suatu method yang dimiliki kelas activity .Berikut penjelasan pada masing-masing state [12] :

1. Created

   Method ini dipanggil ketika activity pertama kali dibuat.

2. Started

   Method ini dipanggil ketika sebuah activity tampil ke pengguna.

3. Resumed

   Method ini dipanggil ketika activity yang berjalan pada saat itu dihentikan sementara (paused) dan activity sebelumnya dijalankan kembali(resumed). (Hasil dari method OnRestart()).

4. Paused

   Method ini dipanggil ketika activity di hentikan sementara (pause) dan berikutnya ketika dijalankan kembali akan berada dalam posisi resume dan memanggil method OnResume().

5. Stopped

   Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan service di background.

6. Destroyed

   Method ini dipanggil sebelum activity dihancurkan (destroy) oleh sistem (baik secara manual maupun untuk kepentingan pelonggaran memori.

2.3 MySQL

Structure Query Language dengan singkatan SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Dengan memanfaatkan bahasa web seperti HTML, dan PHP, SQL merupakan tool penting yang tidak dapat ditinggalkan untuk pengembangan aplikasi database berbasis web. Query dasar yang sering digunakan adalah *create*, *read*, *update* dan *delete*.

1. *Create* digunakan untuk menginputkan data kesebuah tabel di database, contoh querynya sebagai berikut :

   INSERT INTO anggota (nama_anggota, umur) VALUES (‘Asep’, 21)
2. *Read* digunakan untuk menampilakan data sesuai dengan ketentuan tertentu, contoh querynya sebagai berikut :

   SELECT * FROM anggota
3. *Update* digunakan untuk mengubah data sesuai dengan apa yang diinginkan, contoh querynya adalah sebagai berikut :

   UPDATE anggota SET umur = ‘21’ WHERE id_anggota = 1
4. *Delete* digunakan untuk menghapus data yang ingin dihilangkan, contoh querynya sebagai berikut :

   DELETE FROM anggota WHERE id_anggota =5

2.4 User Generated Content

User Generated Content adalah merujuk pada berbagai tipe konten materi atau media yang tersedia untuk umum dan diproduksi oleh end-user atau para non-media profesional. Kemajuan teknologi web memungkinkan konten. website tidak lagi dimonopoli oleh profesional media, melainkan dapat dibuat oleh para

penggunanya. Menggunakan *User Generated Content* (UGC) juga merupakan cara yang bagus untuk membangun hubungan dan loyalitas follower Anda. Jika mereka melihat Anda memperhatikan apa yang mereka katakan, mereka akan lebih bersemangat dengan merk Anda [13]. Berikut adalah klasifikasi dari beberapa *User Generated Content* (UGC) yang umum ditemukan di internet [14]:

1. Rating

   Hanya melihat rating yang terdapat di situs Amazon.com misalnya, mampu membuat banyak orang yang tadinya ragu untuk membeli sesuatu menjadi yakin akan pilihannya. Terutama jika rating atau review yang diberikan atas suatu produk berasal dari teman-temannya sendiri. Dapat dilihat pada gambar 2.5 Rating Amazon.com.



Gambar 2. 5 Rating Amazon.com

2. Status Update

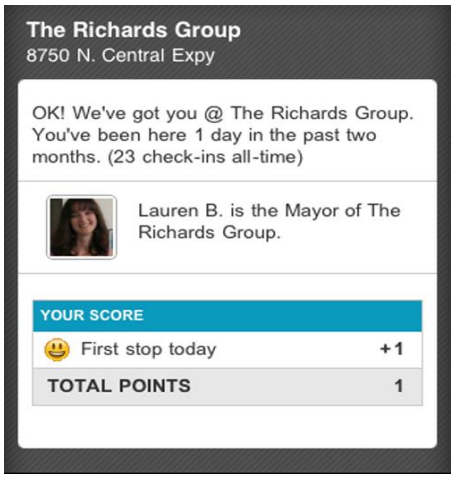
   Ini yang biasa ditemukan di Facebook dan Twitter, yang merupakan jawaban Anda atas pertanyaan mereka, "What's on your mind?" menggerakkan massa yang besar untuk melakukan sesuatu. Dapat dilihat pada gambar 2.6 Status Update.



Gambar 2. 6 Status Update

3. Check-Ins

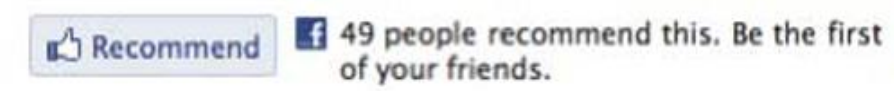
   Melalui fitur berbasis lokasi dari layanan situs media sosial seperti Foursquare, Facebook Places, dan Gowalla misalnya, orang-orang bisa melakukan check-in dan memberitahukan posisi mereka di mana pun. Sering kali fitur ini disandingkan dengan komentar dan review mengenai lokasi yang bersangkutan sehingga membuatnya lebih menarik. Dapat dilihat pada gambar 2.7 Check-Ins.



Gambar 2. 7 Check-Ins

4. Like

   Fitur “Like” atau “Recommend” dari Facebook memungkinkan para pengguna untuk menunjukkan bahwa mereka menyukai situs, produk, atau jasa tertentu. Selain mengklik “Like”, profil si pengguna pun langsung terhubung seperti halnya melakukan aktivitas menambah teman (Add Friend). Dapat dilihat pada gambar 2.8 Like.



Gambar 2. 8 Like

2.5 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 Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman 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 [15].

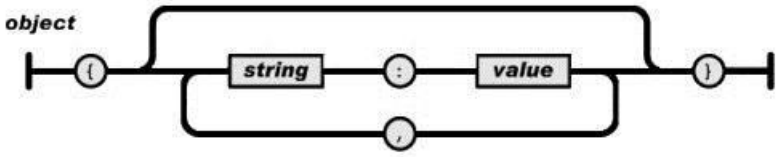
JSON terbuat dari dua struktur:

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).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini JSON menggunakan bentuk sebagai berikut [15] :

1. Objek

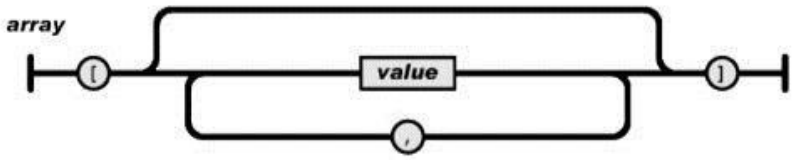
   Objek adalah sepasang nama / nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma). Objek biasanya digunakan untuk menyimpan data tunggal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



Gambar 2. 9 Objek JSON

2. Larik

Larik 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. 10 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, Bollean dan Nilai NULL [15].

2.6 Web Services

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (service) yang disediakan oleh suatu sistem yang menyediakan web service. Web service menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam Web Service dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detil pemrograman yang terdapat di dalamnya.

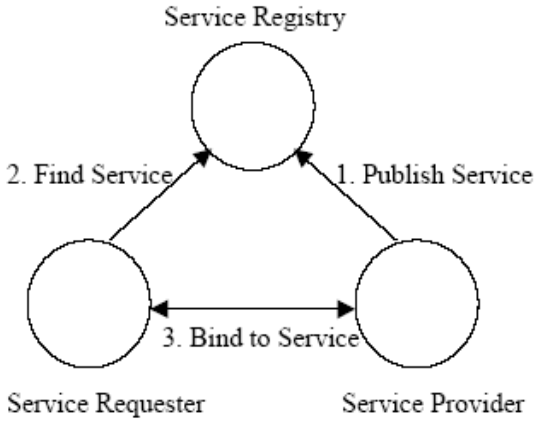
Beberapa alasan mengapa digunakannya web service adalah sebagai berikut:

1. Web service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. Web service memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. Web service cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. Web service berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall.

2.6.1 Arsitektur Web Service

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

1. Service Requester (peminta layanan)
2. Service Provider (penyedia layanan)
3. Service Registry (daftar layanan)



Gambar 2. 11 Arsitektur Web Service

- Service Provider: Berfungsi untuk menyediakan layanan/service dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia.
- Service Registry: Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/service yang telah di-register.

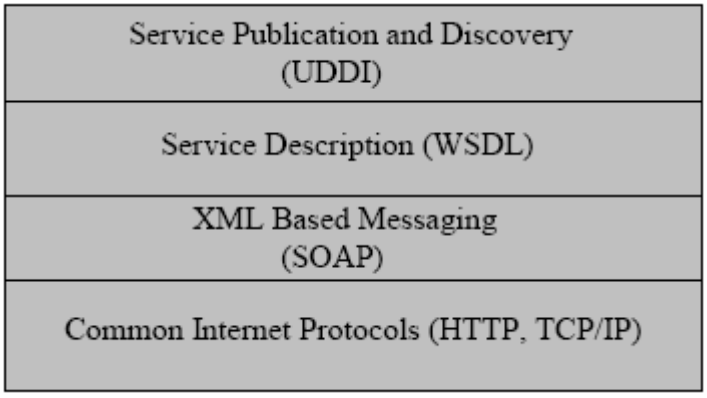
- Service Requestor: Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

2.6.2 Operasi-Operasi Web Service

Secara umum, web service memiliki tiga operasi yang terlibat di dalamnya, yaitu:

1. Publish/Unpublish: Menerbitkan/menghapus layanan ke dalam atau dari registry.
2. Find: Service requestor mencari dan menemukan layanan yang dibutuhkan.
3. Bind: Service requestor setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke service provider untuk melakukan interaksi dan mengakses layanan/service yang disediakan oleh service provider.

2.6.3 Komponen-Komponen Web Service



Gambar 2. 12 Komponen-Komponen Web Service

Web service secara keseluruhan memiliki empat layer komponen seperti pada gambar di atas, yaitu:

1. Layer 1: Protokol internet standar seperti HTTP, TCP/IP
2. Layer 2: Simple Object Access Protocol (SOAP), merupakan protokol akses objek berbasis XML yang digunakan untuk proses pertukaran data/informasi antar layanan.

3. Layer 3: Web Service Definition Language (WSDL), merupakan suatu standar bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.
4. Layer 4: Universal Description, Discovery and Integration (UDDI) adalah sebuah service registry bagi pengalokasian web service. UDDI mengkombinasikan SOAP dan WSDL untuk pembentukan sebuah registry API bagi pendaftaran dan pengenalan service.

2.7 Object Oriented Analysis Design

Object Oriented Analysis Design (OOAD) mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). Proses analisis bertujuan memahami masalah yaitu dengan memahami apa yang sebenarnya terjadi didunia nyata. Sementara proses desain bertujuan untuk memahami pemecahan masalah yang didapat dari proses analisis, yaitu dengan dengan mengusulkan secara detail sistem komputer yang seperti apa yang dibangun untuk mengatasi suatu masalah [16].

2.7.1 Unified Modeling Language

Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk mendengarkan dan mengvisualisasikan artifak dan proses analisis dan desain beroroentasi objek. UML menyediakan standar pada notasi dan diagram yang bisa digunakan untuk memodelkan suatu sistem. UML menjadi bahasa yang bisa digunakan untuk berkomunikasi dalam perspektif objek antar user dan developer, antara developer dan developer, antara developer analisis dan developer desain,dan antara developer desain dengan developer pemrograman [16]. UML merupakan suatu bahasa pemodelan yang terdiri banyak model diantaranya adalah :

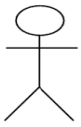
1. Use Case Diagram

   Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu,

misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Adapun komponen-komponen dalam use case diagram antaranya :

1. Actor

   Actor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi aktor ini bisa berupa orang, perangkat keras atau objek yang lain dalam sistem yang sama. Biasanya yang dilakukan oleh aktor adalah memberikan informasi pada sistem atau memerintakan sistem melakukan sesuatu. Dapat dilihat pada gambar 2.11 Actor.



Gambar 2. 13 Actor

2. Use Case

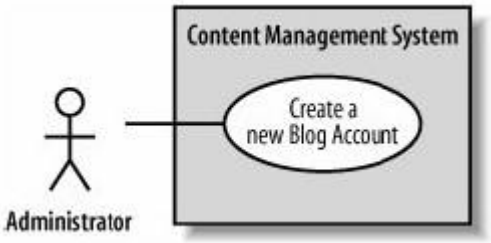
   Use Case Mengidentifikasi fitur kunci dari sistem. Tanpa fitur ini, sistem tidak akan memenuhi permintaan user/actor. Setiap use case mengekspresikan tujuan dari sistem yang harus dicapai. Diberi nama sesuai dengan tujuannya dan digambarkan dengan elips dengan nama di dalamnya. Dapat dilihat pada gambar 2.12 Use Case.



Gambar 2. 14 Register

3. System

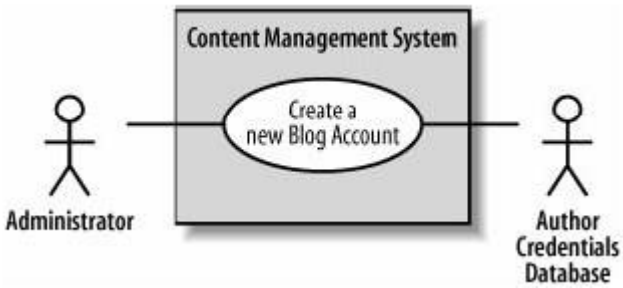
   Menyatakan batasan sistem dalam relasi dengan actor-actor yang menggunakannya (di luar sistem) dan fitur-fitur yang harus disediakan (dalam sistem). Digambarkan dengan segi empat yang membatasi semua use case dalam sistem terhadap pihak mana sistem akan berinteraksi. Dapat dilihat pada gambar 2.13 System.



Gambar 2. 15 System

4. Association

Mengidentifikasikan interaksi antara setiap actor tertentu dengan setiap use case tertentu. Digambarkan sebagai garis antara actor terhadap use case yang bersangkutan. Dapat dilihat pada gambar 2.14 Association.

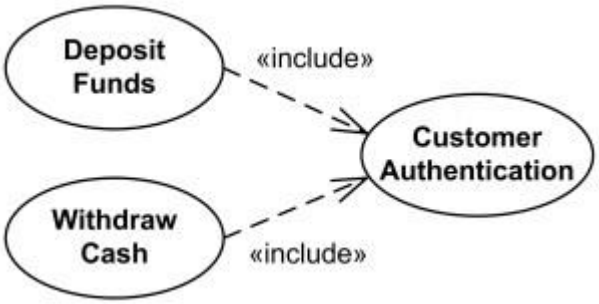


Gambar 2. 16 Association

5. Dependency

Dependency (Include)

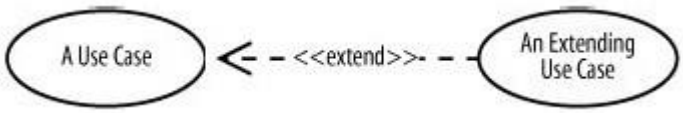
1. Mengidentifikasi hubungan antar dua use case di mana yang satu memanggil yang lain.
2. Digambarkan dengan garis putus-putus bermata panah dengan notasi <<include>> pada garis.
3. Jika pada beberapa use case terdapat bagian yang memiliki aktivitas yang sama maka bagian aktivitas tersebut biasanya dijadikan use case tersendiri dengan relasi dependensi setiap use case semula ke use case yang baru ini sehingga memudahkan pemeliharaan.



Gambar 2. 17 Include

Dependency (Extend)

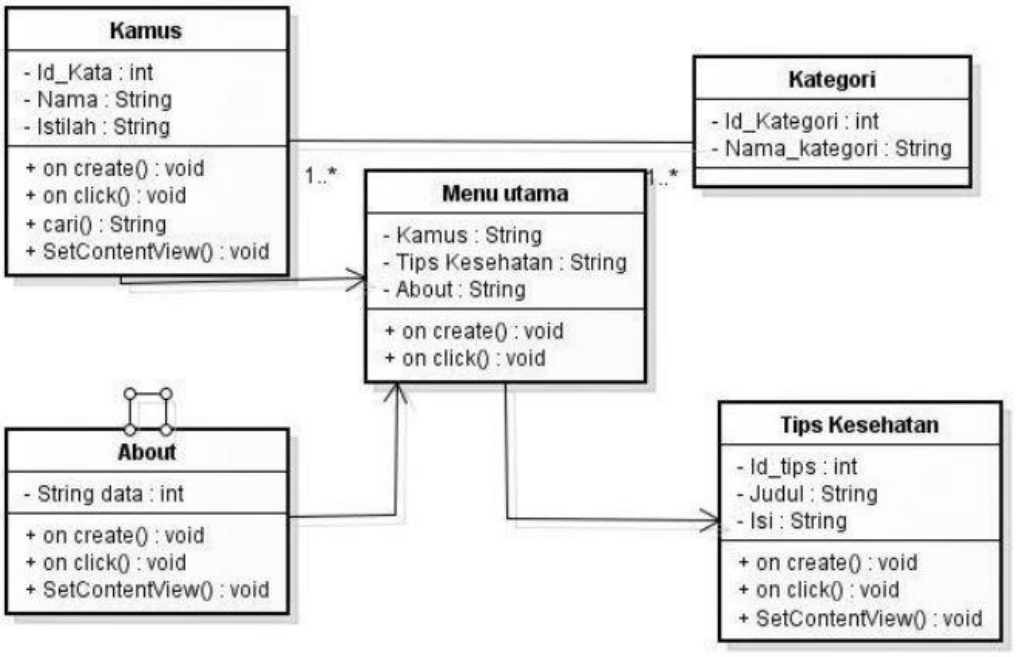
1. Digambarkan serupa dengan dependensi <<include>> kecuali arah panah berlawanan.
2. Jika pemanggilan memerlukan adanya kondisi tertentu maka berlaku dependensi <<extend>>.



Gambar 2. 18 Extend

2. Class Diagram

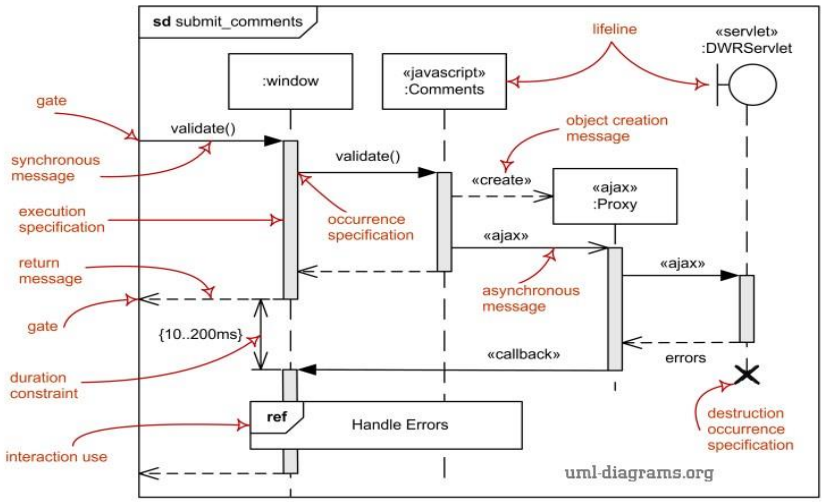
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time [17]. Dapat dilihat pada gambar 2.17 Class Diagram.



Gambar 2. 19 Class Diagram

2.6 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 atar 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 men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan [17]. Dapat dilihat pada gambar 2.18 Sequence Diagram.



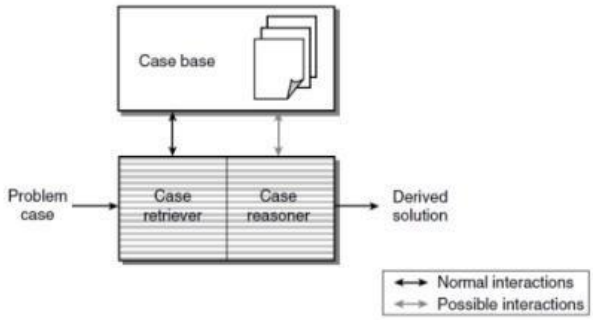
Gambar 2. 20 Sequence Diagram

2.8 Rekomendasi

Sistem rekomendasi merupakan model aplikasi dari hasil observasi terhadap keadaan dan keinginan pelanggan. Oleh karena itu sistem rekomendasi memerlukan model rekomendasi yang tepat agar yang direkomendasikan sesuai dengan keinginan pelanggan, serta mempermudah pelanggan mengambil keputusan yang tepat dalam menentukan produk yang akan digunakannnya [18].

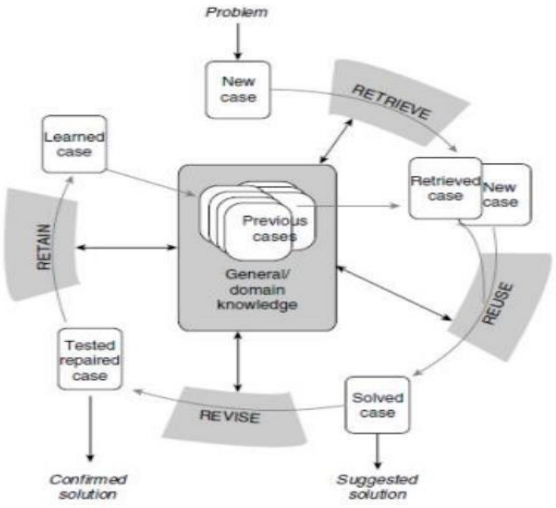
2.9 Case Based Reasoning (CBR)

Case Based Reasoning adalah metode penyelesaian masalah dengan menggunakan solusi masalah-masalah sebelumnya yang serupa [1]. CBR sendiri adalah metode yang umum digunakan manusia dalam menyelesaiakan masalah sehari-hari. Layaknya metode penyelesaian masalah lainnya, computer dapat meniru CBR [19].



Gambar 2. 21 Komponen Case Based Reasoning

Komponen penyelesaian masalah CBR memiliki dua bagian utama : case retriever dan case reasoner. Case retriever bertugas untuk mencari masalah-masalah yang mirip dengan masalah yang dimasukkan pengguna di dalam basis pengetahuan (case base), sementara case reasoner bertugas untuk mencari solusi masalah pengguna dengan melihat atau menyesuaikan solusi masalah yang ditemukan case retriever [19].



Gambar 2. 22 Siklus Metode Case Based Reasoning

Dalam eksekusinya, ada empat tahapan dalam proses CBR [19] :

1. *Retrieve*: mengambil kasus-kasus lama dari case base yang mirip dengan kasus yang dihadapi.
2. *Reuse*: menggunakan solusi kasus-kasus lama hasil retrieve tersebut untuk menyelesaikan kasus yang baru tersebut.
3. *Revise*: jika diperlukan, mengadaptasi solusi kasus lama agar sesuai dengan kondisi masalah baru.

4. *Retain*: menyimpan solusi hasil revise yang telah divalidasi ke dalam basis data, agar dapat digunakan untuk menyelesaikan masalah serupa di masa depan.

Tidak semua bagian *case retriever* dan *case reasoner* harus diotomatis. Ada banyak program aplikasi CBR yang hanya melakukan *case retriever* dan memberikan hasil *case retriever* ke pengguna. Pengguna sendiri yang selanjutnya menafsirkan data tersebut dan mengambil keputusan. Dalam hal ini proses *case retrieve* dikerjakan oleh komputer, namun *case reasoning*-nya diserahkan pada pengguna [20].

2.10 Algoritma Nearest Neighbor Retrieval

Algoritma Nearest Neighbor Retrieval (k-nearest neighbor atau k-NN) sebuah algoritma untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut [19].

K-Nearest Neighbor berdasarkan konsep ‘learning by analogy’. Data learning dideskripsikan dengan atribut numerik n-dimensi. Tiap data learning merepresentasikan sebuah titik, yang ditandai dengan c, dalam ruang n-dimensi. Jika sebuah data query yang labelnya tidak diketahui diinputkan, maka K-Nearest Neighbor akan mencari k buah data learning yang jaraknya paling dekat dengan data query dalam ruang n-dimensi. Jarak antara data query dengan data learning dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data query dengan semua titik yang merepresentasikan data learning dengan rumus Euclidean Distance.

Algoritma nearest neighbor retrieval menyimpan semua training sampel dan tidak. Membangun classifier sampai sampel baru (unlabeled) perlu diklasifikasikan, sehingga algoritma nearest neighbor retrieval sering disebut dengan instance-based atau lazy learners.

2.11 Metode Pengujian

Metode pengujian adalah cara atau teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan. Perangkat lunak dapat diuji dengan dua cara, yaitu:

1. Pengujian dengan menggunakan data uji untuk menguji semua elemen program (data internal, loop, logika, keputusan dan jalur). Data uji dibangkitkan dengan mengetahui struktur internal (kode sumber) dari perangkat lunak.
2. Pengujian dilakukan dengan mengeksekusi data uji dan mengecek apakah fungsional perangkat lunak bekerja dengan baik. Data uji dibangkitkan dari spesifikasi perangkat lunak.

2.11.1 Black Box Testing

Pengujian black box merupakan pendekatan komplementer dari teknik white box, karena pengujian black box diharapkan mampu mengungkap kelas kesalahan yang lebih luas dibandingkan teknik white box. Pengujian black box adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian black box merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan. Pengujian black box berusaha menemukan kesalahan dalam kategori :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi.

Tidak seperti pengujian white box yang dilakukan pada awal proses pengujian, pengujian black box cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian black box memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi.

Dengan mengaplikasikan teknik black box, maka dapat menarik serangkaian test case yang memenuhi kriteria berikut ini:

1. Test case yang mengurangi, dengan harga lebih dari satu, jumlah test case tambahan yang harus didesain untuk mencapai pengujian yang dapat dipertanggungjawabkan.
2. Test case yang memberi tahu sesuatu mengenai kehadiran atau ketidakhadiran kelas kesalahan daripada memberi tahu kesalahan yang berhubungan hanya dengan pengujian spesifik yang ada.

2.12 Skala Likert

Skala *likert* digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang kejadian atau gejala sosial. Dengan menggunakan skala likert, maka variabel dijabarkan menurut urutan variabel, sub variabel, indikator, deskriptor. Dan deskriptor ini dapat dijadikan titik tolak untuk membuat butir instrumen berupa pernyataan atau pertanyaan yang perlu dijawab oleh responden.

Item item likert menyediakan repon dengan kategori yang berjenjang. Biasanya banyaknya jenjang adalah lima, yaitu sangat setuju, setuju, ragu ragu, tidak setuju dan sangat tidak setuju. Setiap kategori respon selanjutnya diberi skor. Untuk item positif skor terbesarnya adalah lima, skor terendah adalah satu dan sebaliknya jika item negatif. Untuk menghitung total skor tiap responden adalah dengan cara menjumlahkan skor skor item yang diperoleh reponden. Oleh karena itu, prosedur penskalaan likert sering disebut sebagai *Likert's Summeted Rating* [21].

Langkah langkah menyusunnya adalah sebagai berikut :

1. Tentukan secara tegas sikap terhadap topik apa yang akan diukur.
2. Tentukan secara tegas sub varibel/dimensi yang menyusun sikap tersebut, kognitif, afektif dan konatif.

3. Susun pernyataan/pertanyaan atau item yang merupakan alat pengukur dimensi yang menyusun sikap yang akan diukur sesuai dengan indikator.
4. Setiap item diberi respon yang sifatnya tertutup.
5. Untuk setiap respon, jawaban diberi skor berdasarkan kriteria sebagai berikut: apabila item positif maka angka terbesar diletakkan pada respon sangat setuju, sedangkan bila item negatif maka angka terbesar diletakkan pada respon sangat tidak setuju.
6. Untuk mengetahui posisi setiap responden tentang suatu variabel, tentukan skor maksimal dan skor minimal yang mungkin dicapai oleh responden.

Langkah langkah untuk mengetahui bagaimana sikap tiap responden terhadap kualitas produk adalah :

1. Menentukan skor maksimal, yaitu skor jawaban terbesar dikali banyak item.
2. Menentukan skor minimal, yaitu skor jawaban terkecil dikali banyak item.
3. Menentukan nilai median, yaitu hasil penjumlahan skor maksimal dengan skor minimal dibagi dua.
4. Menentukan nilai kuartil 1, yaitu hasil penjumlahan skor minimal dengan median dibagi dua.
5. Menentukan nilai kuartil 3, yaitu hasil penjumlahan skor maksimal dengan median dibagi dua.
6. Membuat skala yang menggambarkan skor minimal, nilai kuartil 1, median, nilai kuartil 3 dan skor maksimal.
7. Mencari batas batas skor untuk masing masing kategori sikap.
8. Membuat tabel distribusi frekuensi sikap tiap responden terhadap kualitas produk.

Langkah langkah untuk mengetahui bagaimana sikap responden secara keseluruhan terhadap kualitas produk adalah :

1. menentukan skor maksimal, yaitu skor maksimal yang diperoleh tiap responden dikali banyaknya responden.
2. Menentukan skor minimal, yaitu skor minimal yang diperoleh tiap responden dikali banyaknya responden.

3. Menentukan nilai median, yaitu hasil penjumlahan total skor maksimal dengan total skor minimal dibagi dua.
4. Menentukan nilai kuartil 1, yaitu hasil penjumlahan total skor minimal dengan median dibagi dua.
5. Menentukan nilai kiartil 3, yaitu hasil penjumlahan total skor maksimal dengan median dibagi dua.
6. Membuat skala yang menggambarkan total skor minimal, nilai kuartil 1, median, kuartil 3 dan total skor maksimal.
7. Mencari batas batas skor untuk masing masing kategori sikap.
8. Menentukan skor total yang diperoleh seluruh responden.
9. Mengintrepretasikan skor total responden dengan skala pada point.