BAB II
TEORI PENUNJANG

Bab ini berisi penjelasan mengenai konsep dasar yang digunakan sebagai acuan atau penunjang dalam perancangan aplikasi serta tools yang digunakan untuk membangun aplikasi. Diantaranya Aplikasi, Promosi, Augmented Reality, Android, Android SDK, UML, Alpha dan Beta Testing, dan Skala likert.

2.1 Aplikasi

Aplikasi merupakan perangkat lunak yang siap pakai dengan menjalankan instruksi-instruksi dari *user* (pengguna) yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia. Penulis akan mengemukakan beberapa pendapat dari para pengembang tentang aplikasi, yaitu sebagai berikut : [8]

1. **Menurut Pranama**

   "Aplikasi adalah satu unit perangkat lunak yang sengaja dibuat untuk memenuhi kebutuhan akan berbagai aktivitas ataupun pekerjaan, seperti aktivitas perniagaan, periklanan, pelayanan masyarakat, game, dan berbagai aktivitas lainnya yang dilakukan oleh manusia."

2. **Menurut Verman dkk**

   "Aplikasi adalah perangkat intruksi khusus dalam komputer yang dirancang agar kita menyelesaikan tugas-tugas tertentu."

3. **Menurut Yuhefizar**

   "Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. Jadi aplikasi merupakan sebuah program yang dibuat dalam sebuah perangkat lunak dengan computer untuk memudahkan pekerjaan atau tugas-tugas seperti penerapan, penggunaan dan penambahan data yang dibutuhkan. "

2.2 Promosi

Promosi merupakan komunikasi dan penyampaian pesan yang dilakukan baik oleh perusahaan maupun perantara dengan tujuan memberikan informasi mengenai produk, harga dan tempat. Informasi itu bersifat memberitahukan,

membujuk, mengingatkan kembali kepada konsumen, para perantara atau kombinasi keduanya. Penulis akan mengemukakan beberapa pendapat dari para ahli pemasaran tentang penelitian promosi, yaitu sebagai berikut: [4]

1. **Menurut Djaslim Saladin dan Yevis Marty Oesman**

   "Promosi adalah suatu komunikasi informasi penjual dan pembeli yang bertujuan untuk merubah sikap dan tingkah laku pembeli, yang sebelumnya tidak mengenal menjadi mengenal sehingga menjadi pembeli dan mengingat produk tersebut".

2. **Menurut Buchari Alma**

   "Promosi adalah sejenis komunikasi yang memberi penjelasan dan meyakinkan calon konsumen mengenai barang dan jasa dengan tujuan untuk memperoleh perhatian, mendidik, mengingatkan dan meyakinkan calon konsumen".

2.3 Augmented Reality

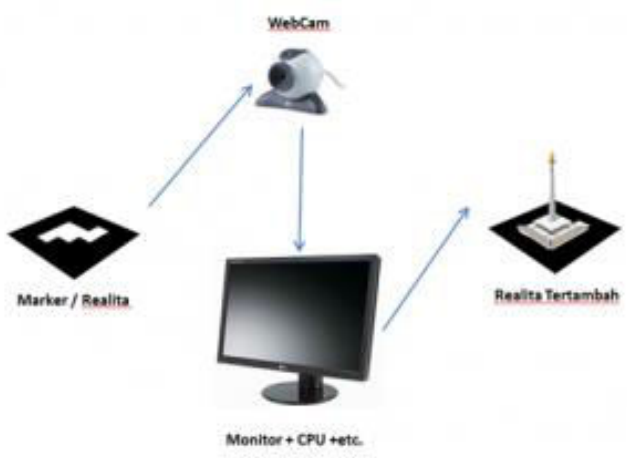
AR (*Augmented Reality*) atau dalam bahasa Indonesianya realitas tertambah adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata (*real time*). Tidak seperti realitas maya atau VR (*Virtual Reality*) yang sepenuhnya menggantikan kenyataan, realitas tertambah sekedar atau melengkapi kenyataan. *Virtual Reality* atau Realitas Maya adalah teknologi yang memungkinkan *user* dapat berinteraksi dengan suatu lingkungan berdimensi 3 yang disimulasikan oleh komputer terhadap suatu objek nyata atau imajinasi, sehingga membuat *user* seolah-olah terlibat secara fisik pada lingkungan tersebut.

Secara garis besar perbedaan *Virtual Reality* dan *Augmented Reality* pada bagian interaksi terhadap pengguna, dimana pada *Augmented Reality* pengguna dapat berinteraksi secara langsung dengan cara mendengarkan, melihat, serta menyentuh aplikasi yang berjalan secara *real time*. Sehingga *Augmented Reality* seakan-akan membawa aplikasi tersebut kepada pengguna seperti aslinya,yang mengacu pada bahan yang dihasilkan seperti berupa teks, gambar maupun video.[16]

"Realitas tertambah / AR dapat diaplikasikan untuk semua indra, tidak hanya visual, termasuk pendengaran, sentuhan dan penciuman. Gunanya untuk memperkaya pengalaman penggunanya, membantu persepsi dan interaksi penggunanya dengan dunia nyata."[12] Teknologi ini biasanya digunakan pada bidang militer, medis, komunikasi, dan manufaktur yang mempunyai risiko besar dan membutuhkan tambahan benda-benda semu yang meniru benda-benda nyata sebelum diimplementasikan.

2.3.1 Cara kerja Augmented Reality

Augmented Reality bekerja berdasarkan deteksi citra (gambar), dan citra yang digunakan adalah marker. Prinsip kerjanya adalah kamera yang telah dikalibrasi akan mendeteksi marker yang diberikan, kemudian setelah mengenali dan menandai pola marker, *webcam* akan melakukan perbandingan apakah marker sesuai dengan *database* yang dimiliki atau tidak. Bila tidak, maka informasi marker tidak akan diolah, tetapi bila sesuai maka informasi marker akan digunakan untuk me-*render* dan menampilkan objek 3D atau animasi yang telah dibuat sebelumnya.



Gambar II.1 Cara Kerja Augmented Reality

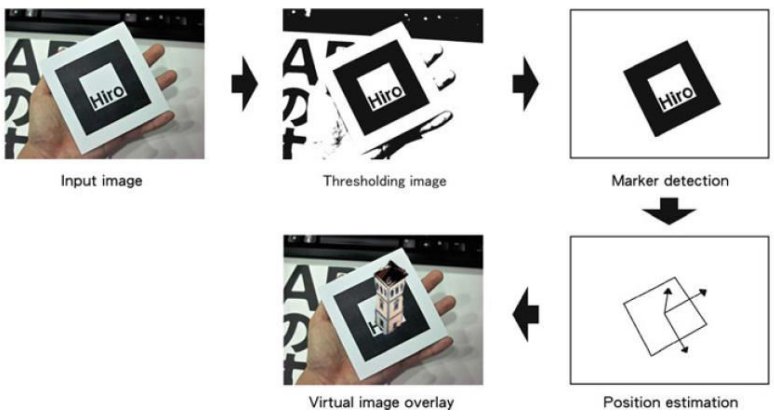
Pada umumnya Augmented Reality membutuhkan alat masukkan (*input device*) seperti kamera atau *Webcam*, alat keluaran (output device) seperti monitor atau *Head Mounted Display* (HMD), alat pelacak (*tracker*) agar benda maya tambahan berupa penanda (*marker*) yang dihasilkan berjalan secara *real-time* atau walaupun benda nyata yang menjadi induknya digeser-geser akan tetap muncul di

atas marker, dan komputer untuk menjalankan program AR. “Salah satu aspek yang paling penting dari *Augmented Reality* adalah untuk membangun teknik yang tepat untuk interaksi intuitif antara pengguna dan konten virtual AR aplikasi.”[5]

2.3.2 Metode Augmented Reality

Ada beberapa metode yang digunakan pada *Augmented Reality* salah satunya adalah *Marker Based Tracking*. Metode ini merupakan teknologi *Augmented Reality* yang dihadirkan dari gabungan teknologi *computer vision* dan *image processing* yang mencari informasi dari sebuah gambar secara langsung. Marker yang akan digunakan harus terlebih dahulu dibuat, dengan cara mendaftarkan gambar yang akan menjadi penanda agar bisa memunculkan objek 3D kedalam sebuah *database*. Dalam hal ini penulis menggunakan Vuforia sebagai pembuat *database* marker. Dimana marker tersebut dibutuhkan untuk memunculkan objek 3D yang akan dimunculkan setelah proses *tracking* dan *positioning* dilakukan.

Tracking marker merupakan proses yang dilakukan oleh sistem ketika melakukan pemindaian gambar, terhadap *database* yang sudah ada. Dalam hal ini Vuforia Sdk sebagai penyedia layanan melakukan proses *computer vision*, dimana pada proses tersebut Vuforia menggunakan metode NFT (*Natural Feature Tracking*). *Natural Feature* merupakan salah satu metode untuk menstabilkan dan memperluas jangkauan dalam sistem pelacakan *Augmented Reality.*



Gambar II.2 Proses Tracking dari Teknologi Augmented Reality

Secara garis besar proses *tracking marker* dari teknologi *Augmented Reality* dapat dilihat dari gambar 2.2 dimana pada awal proses kamera akan melacak marker, yang dilanjutkan pada *tresholding image* (pemisahan warna menjadi 1 dan 0) dimana 1 untuk warna putih dan 0 untuk warna hitam, lalu dilanjutkan pencocokan *marker*, dan diakhiri dengan penempatan animasi objek 3D.

Ibanez, A. S., dan Figueras, J. P. [9] mengatakan bahwa "untuk mengenali gambar Vuforia memproses fitur-fiturnya dan membandingkan hasilnya ke *database* yang terkandung dalam dataset aplikasi. Jika hasil cocok, maka gambar tersebut dianggap telah dikenali. Oleh karena itu jika gambar tidak jelas, kemungkinan gambar tidak akan terdeteksi."

2.3.3 Vuforia

Vuforia merupakan *Software Development Kit* (SDK) yang memungkinkan pembangunan aplikasi *Augmented Reality* pada perangkat *mobile* dengan dukungan untuk Android, iOS dan ekstensi Unity3D yang akan dibangun. Vuforia memiliki arsitektur yang kuat untuk mengoptimalkan proses *tracking*, *registering* dan *graphics visualization* memungkinkan berbagai fitur untuk *tracking* dan *registering frame marker* (marker acuan), gambar dan teks .

"Arsitektur pada komponen utama Vuforia SDK terdiri dari 6 macam yaitu kamera, *image converter, tracker, video background renderer, application code,* dan *target resource*." [9]

1. Kamera

Kamera memastikan setiap *frame* ditangkap dan bisa melewati pelacak. Pengembang hanya memberitahu kapan kamera harus memulai atau berhenti menangkap sebuah gambar. lalu *frame* kamera secara otomatis sampai di perangkat, tergantung format dan ukuran gambarnya.

2. Image Converter

Format pixel converter tunggal mengkonversi antara format kamera ke format yang sesuai dengan OpenGL-ES *rendering* dan untuk pelacakan. Konversi ini

termasuk kedalam *down-sampling*. fungsinya untuk membuat gambar kamera dengan resolusi berbeda dalam konversi *stack frame*.

1. Tracker

Pelacak tunggal berisi algoritma visi komputer yang mendeteksi dan melacak benda di dunia nyata kedalam frame video kamera. Berdasarkan gambar kamera, algoritma yang berbeda mendeteksi target baru atau marker, dan memeriksa tombol virtual. Hasilnya disimpan dalam sebuah *state object* yang digunakan oleh *renderer* video background dan dapat diakses dari kode aplikasi. Itu pelacak dapat memuat beberapa set data, tetapi hanya satu saja yang dapat aktif dalam satu waktu.

2. Video Background Renderer

Video background renderer membuat gambar kamera dapat disimpan dalam *state* objek. Kinerja video *Background Renderer* dioptimalkan untuk perangkat tertentu.

3. Application Code

Pengembang aplikasi harus menginisialisasi semua komponen di atas dan melakukan 3 kunci langkah dalam kode aplikasi. Untuk setiap *frame* yang diproses, *state object* akan diperbarui dan aplikasi akan membuat metode yang akan dipanggil. Pengembang aplikasi harus :

- Mengantrikan *state object* untuk pendeteksian target yang baru, marker atau memperbarui status dari elemen.
- Memperbaharui logika aplikasi dengan input data yang baru.
- Membuat lapisan dari tampilan *Augmented*

4. Target Resource

Target Resource dibuat menggunakan *Target Management System* secara *on-line*. Dataset yang sudah di-download berisi file konfigurasi XML yang memungkinkan pengembang untuk mengkonfigurasi fitur *trackable* dan file biner tertentu yang berisi dataset *trackable*. Asset ini akan dikompile oleh pengembang

aplikasi ke dalam paket installer aplikasi dan digunakan pada saat *run-time* oleh Vuforia SDK.

2.3.4 Unity 3D

Unity 3D merupakan sebuah *game engine,* yaitu software pengolah gambar, grafik, suara, input, dan lain-lain yang ditunjukan untuk membuat game. Unity 3D merupakan *game engine* multiplatform yang mampu di-*publish* secara *standalone* (.exe), berbasis web, android, Ios, XBOX, maupun PS3, dengan catatan mendapat lisensi.

Kelebihan Unity dibanding dengan *game engine* lainya adalah kemampuan membuat game cross platform. Dengan Unity 3D, game yang Anda buat dapat dimainkan di berbagai perangkat, seperti *smartphone* dan game console. Unity sendiri dapat membuat berbagai macam game, seperti RPG (*Role Playing Game*), shooter, racing, dan lain sebagainya. Berikut adalah fitur-fitur yang ada pada Unity 3D. [11]

1. **Rendering**

   Untuk meningkatkan kualitas pemetaan atau tokoh dalam *game*, Unity 3D mendukung penggunaan *software* pengolahan gambar lain, seperti 3ds Max, Maya, Softimage, Blender, Modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks, dan Allegorithmic Substance.

2. **Scripting**

   Scripting yang digunakan pada Unity dibangun menggunakan Mono 2.6. Mono 2.6 merupakan implementasi open source dari .Net Framework. Bahasa pemrograman yang didukung oleh Unity 3D, antara lain JavaScript, C#, dan Boo (menggunakan sintaks Phyton).

3. **Asset Store**

   Asset store merupakan tempat untuk mendapatkan asset yang digunakan untuk menunjang pembuatan game. *Asset* pada Unity 3D dibagi menjadi dua, yaitu eksternal dan internal. *Asset* eksternal merupakan asset yang ditambahkan dari sumber di luar Unity 3D, seperti 3D model, Texture, Sound Effect. Asset internal merupakan asset yang sudah ada di dalam Unity, seperti Materials, Shaders, Cube Maps, Physics Materials, dan Prefabs.

1. **Platform**

   Seperti yang dibahas sebelumnya, Unity 3D dapat menjalankan secara cross platform. Platform yang didukung Antara lain Xbox One, Blacberry 10, Windows 8, Windows Phone 8, Windows, Mac, Linux, Android, iOs, Unity Web Player, Adobe Flash, PlayStation 3, Xbox 360, Wii U dan Wii.

2.4 Android

Menurut Nazaruddin Safaat yang dikutip oleh Gugy Munzy [13] dalam penelitianya, "Android adalah sistem operasi berbasis Linux bagi telepon seluler seperti telepon pintar dan komputer tablet." Android juga menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam piranti gerak.

"Sifat Android yang terbuka telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain". [13]

Nazaruddin Safaat juga memaparkan berbagai versi Android, Berikut adalah versi-versi Android menurut Nazaruddin Safaat :

1. Android Versi 1.1
2. Android versi 1.5 (Cupcake)
3. Android versi 1.6 (Donut)
4. Android versi 2.0/2.1 (Eclair)
5. Android versi 2.2 (Froyo: Frozen Yoghurt)
6. Android versi 2.3 (Gingerbread)
7. Android versi 3.0/3.1 (Honeycomb)
8. Android versi 4.0 (ICS: Ice Cream Sandwich)
9. Android versi 4.1 (Jelly Bean)

Versi Android yang penulis sarankan untuk penggunaan aplikasi maupun dalam pembuatan aplikasi dengan minimal versi Android 4.0 adalah Android versi 4.1 (Jelly Bean), Android versi 4.4 (KitKat), Android versi 5.0 (Lollipop),

Android versi 6.0 (Marshmallow). Bagian ini diperlukan untuk Analisis Kebutuhan Fungsional pada Bab 3 sebagai kebutuhan perangkat lunak.

2.5 Android SDK Manager

Android Software Development Kit Manager adalah salah satu emulator Android. SDK Manager dapat menjalankan aplikasi-aplikasi yang terdapat pada Android karena bersifat emulator. Android SDK juga merupakan *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis Google Android. Nazaruddin Safaat juga memaparkan bahwa "Android SDK merupakan tools API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada platform Android yang menggunakan bahasa pemrograman Java".

2.6 UML (Unified Modeling Language)

"UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem". [6]

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (message) antar objects.
4. *Collaboration Diagram* untuk memodelkan interaksi antar objects.
5. *State Diagram* untuk memodelkan perilaku objects di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku Use Case dan objects di dalam system.

1. *Class Diagram* untuk memodelkan struktur kelas.
2. *Object Diagram* untuk memodelkan struktur object.
3. *Component Diagram* untuk memodelkan komponen object.
4. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

Berikut adalah UML yang penulis gunakan dalam penelitian ini yaitu use case diagram, sequence diagram, collaboration diagram dan class diagram.

1. Use Case Diagram

Use case diagram digunakan untuk memodelkan semua bisnis proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambar-kan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.

2. Sequence Diagram

Diagram Class dan diagram Object merupakan suatu gambaran model statis. Namun ada juga yang bersifat dinamis, seperti *Diagram Interaction*. *Diagram sequence* merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

3. Class Diagram

Class diagram menggambarkan struktur statis class di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu class tergantung/menggunakan *class* yang lain), *specialed* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (group bersama sebagai satu unit). sebuah sistem biasanya mempunyai beberapa *class* diagram.

4. Collaboration Diagram

Diagram *Collaboration* juga merupakan diagram *interaction*. Diagram membawa informasi yang sama dengan diagram *Sequence*, tetapi lebih

memusatkan atau memfokuskan pada kegiatan obyek dari waktu pesan itu dikirimkan. Setiap pesan pada diagram *Collaboration* mempunyai angka yang terurut. Pesan yang tingkatannya tertinggi adalah angka 1. Pesan yang berada pada tingkat yang sama memiliki prefix yang sama, namun suffix berbeda bergantung pada posisinya; hanya untuk angka 1, 2, dan seterusnya

2.7 Alpha dan Beta Testing

Tujuan Alpha Testing atau pengujian alpha adalah untuk mengidentifikasi dan menghilangkan sebanyak mungkin masalah sebelum akhirnya sampai ke *user*, serta dilakukan setelah *software* selesai oleh pengembang. Sedangkan Pengujian beta atau beta testing adalah istilah yang digunakan untuk menggambarkan proses pengujian eksternal di mana *software* tersebut dikirim ke kelompok atau memilih pelanggan yang menggunakannya dalam lingkungan dunia nyata. Pengujian beta biasanya dilakukan menjelang akhir dari siklus pengembangan produk dan idealnya sebagai validasi bahwa perangkat lunak siap dirilis untuk pelanggan yang sesungguhnya. [15]

2.7.1 Metode Black-box dan White-box Testing

Menurut Ron Patton[14], Black-box dan White-boxt testing merupakan dua istilah yang digunakan penguji perangkat lunak untuk menggambarkan bagaimana pendekatan pengujian mereka. Dalam pengujian black-box, penguji hanya tahu apa yang seharusnya dilakukan oleh *software*, namun penguji tidak dapat melihat bagaimana hal itu bisa beroperasi. Sedangkan dalam pengujian white-box, *software tester* memiliki akses ke kode program dan dapat memeriksanya sebagai petunjuk untuk membantu pengujian sehingga bisa melihat ke dalam box. Berdasarkan hal ini, *tester* dapat menentukan nomor-nomor tertentu lebih atau kurang cenderung gagal dan dapat menyesuaikan pengujian yang didasarkan pada informasi tersebut.

Ron Patton juga menyebutkan bahwa ada risiko untuk pengujian white-box. Sangat mudah gagal untuk pengujian obyektif perangkat lunak, karena Anda mungkin menyesuaikan pengujian atau tes untuk mencocokkan operasi kode ini. Hal ini juga yang membuat penulis memilih menggunakan metode *Black box*,

agar pengujian berfokus pada persyaratan fungsional perangkat lunak untuk melihat apakah program aplikasi menghasilkan output yang diinginkan atau tidak.

2.8 Skala Likert

Skala Likert umumnya digunakan untuk mengukur sikap atau respon seseorang terhadap suatu objek. Pengungkapan sikap dengan menggunakan skala Likert sangat popular di kalangan para ahli psikologi sosial dan para peneliti. Hal ini dikarenakan selain praktis, skala Likert yang dirancang dengan baik pada umumnya memiliki reliabilitas yang memuaskan.

Skala Likert berwujud kumpulan pertanyaan-pertanyaan sikap yang ditulis, disusun dan dianalisis sedemikian rupa sehingga respon seseorang terhadap pertanyaan tersebut dapat diberikan angka (skor) dan kemudian dapat diinterpretasikan. Skala Likert tidak terdiri dari hanya satu stimulus atau satu pernyataan saja melainkan selalu berisi banyak item (*multiple item measure*).

"Sejauhmana suatu skala Likert akan berfungsi seperti yang diharapkan, yaitu mengungkapkan sikap individu atau sikap kelompok manusia dengan cermat dan akurat, banyak tergantung pada kelayakan pertanyaan-pertanyaan sikap dalam skala itu sendiri. Oleh karena itu pernyataan-pernyataan yang dibuat untuk mengukur sikap harus dirancang dengan hati-hati. Stimulus harus ditulis dan dipilih berdasarkan metode konstruksi yang benar dan skor terhadap respon seseorang harus diberikan dengan cara-cara yang tepat."[15]

Berikut adalah langkah-langkah yang dapat ditempuh untuk mengetahui bagaimana sikap tiap responden terhadap kualitas produk. [10]

1. Menentukan skor maksimal, yaitu skor jawaban terbesar di kali 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.

1. Buatlah skala yang menggambarkan skor minimal, nilai kuartil 1, median, kuartil 3, dan skor maksimal.
2. Carilah batas-batas skor untuk masing-masing kategori sikap.
3. Buatlah tabel distribusi frekuensi sikap tiap responden terhadap kualitas produk.

Keterangan pencarian rentang nilai pada frekuensi :

- Kategori sikap sangat positif, yaitu daerah yang dibatasi oleh kuartil 3 dan skor maksimal (kuartil 3 ≤ x ≤ skor maksimal).
- Kategori sikap positif, yaitu daerah yang dibatasi oleh median dan kuartil 3 (median ≤ x < skor maksimal).
- Kategori sikap negative, yaitu daerah yag dibatasi oleh kuartil 1 dan median (kuartil 1 ≤ x < median).
- Kategori sikap sangat negative, yaitu daerah yang dibatasi oleh skor minimal dan kuartil 1 median (skor minimal ≤ x < kuartil 1).

Skala Likert tidak mengijinkan adanya pernyataan *item* netral. Jadi pernyataan yang ada dalam skala Likert hanya dua, pernyataan item posistif dan *item* negatif.