BAB 2
LANDASAN TEORI

2.1 Matematika

Matematika merupakan ilmu universal yang mendasari perkembangan teknologi modern, mempunyai peran penting dalam berbagai disiplin dan memajukan daya pikir manusia. Perkembangan pesat di bidang teknologi informasi dan komunikasi ini dilandasi oleh perkembangan matematika di bidang teori bilangan, aljabar, analisis, teori peluang dan matematika diskrit. Untuk menguasai dan mencipta teknologi di masa depan diperlukan penguasaan matematika yang kuat sejak dini [1].

2.2 Sniper

Sniper atau dengan kata lain disebut penembak runduk, adalah seorang prajurit infanteri yang secara khusus terlatih untuk mempunyai kemampuan membunuh secara tersembunyi dari jarak jauh menggunakan senapan.

Istilah ini muncul pada tahun 1770-an, pada prajurit – prajurit kolonial Inggris di India, dari kata *snipe* yaitu sejenis burung yang sangat sulit untuk didekati dan ditembak. Oleh karena itu mereka yang mahir memburu burung tersebut diberi julukan *sniper* [6].

2.3 Game

Dalam kamus bahasa Indonesia *game* adalah permainan. Sebuah permainan adalah sebuah sistem dimana pemain terlibat dalam konflik buatan, disini pemain berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau buatan, dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku pemain dan menentukan permainan. Permainan merupakan sebuah aktivitas rekreasi dengan tujuan bersenang-senang, mengisi waktu luang, atau berolahraga santai. Permainan biasanya dilakukan sendiri atau bersama-sama [7].

Permainan video (*video game*) adalah permainan yang menggunakan interaksi antarmuka pengguna melalui gambar yang dihasilkan oleh piranti video. Permainan video umumnya menyediakan sistem penghargaan misalnya skor yang dihitung berdasarkan tingkat keberhasilan yang dicapai dalam menyelesaikan tugas-tugas yang ada di dalam permainan. Sistem elektronik yang digunakan untuk menjalankan permainan video disebut *platform*, contohnya adalah komputer pribadi dan konsol permainan [7].

2.3.1 Game Edukasi

Game Edukasi adalah *game* yang khusus dirancang untuk mengajarkan user suatu pembelajaran tertentu, pengembangan konsep dan pemahaman dan membimbing mereka dalam melatih kemampuan mereka, serta memotivasi mereka untuk memainkannya [8].

Menurut Hurd dan Jenuings, perancangan *Education game* yang baik haruslah memenuhi kriteria dari education *game* itu sendiri. Berikut ini adalah beberapa kriteria dari sebuah *education game*, yaitu [8]:

1. Nilai Keseluruhan (*Overall Value*)

   Nilai keseluruhan dari suatu *game* terpusat pada desain dan panjang durasi *game*. Aplikasi ini dibangun dengan desain yang menarik dan interaktif. Untuk penentuan panjang durasi, aplikasi ini menggunakan fitur *timer*.
2. Dapat Digunakan(*Usability*)

   Mudah digunakan dan diakses adalah poin penting bagi pembuat *game*. Apliksi ini merancang sistem dengan *interface* yang *user friendly* sehingga *user* dengan mudah dapat mengakses aplikasi.
3. Keakuratan (*Accuracy*)

   Keakuratan diartikan sebagai bagaimana kesuksesan model/gambaran sebuah *game* dapat dituangkan ke dalam percobaan atau perancangannya. Perancangan aplikasi ini harus sesuai dengan model *game* pada tahap perencanaan.

4. Kesesuaian (*Appropriateness*)

   Kesesuaian dapat diartikan bagaimana isi dan desain *game* dapat diadaptasikan terhadap keperluan *user* dengan baik. Aplikasi ini menyediakan menu dan fitur yang diperlukan *user* untuk membantu pemahaman *user* dalam menggunakan aplikasi.

5. Relevan (*Relevance*)

   Relevan artinya dapat mengaplikasikan isi *game* ke target *user*. Agar dapat relevan terhadap *user*, sistem harus membimbing mereka dalam pencapaian tujuan pembelajaran.

6. Objektifitas (*Objectives*)

   Objektifitas menentukan tujuan *user* dan kriteria dari kesuksesan atau kegagalan. Dalam aplikas ini objektivitas adalah usaha untuk mempelajari hasil dari permainan.

7. Umpan Balik (*Feedback*)

   Untuk membantu pemahaman *user* bahwa permainan (*performance*) mereka sesuai dengan objek *game* atau tidak, *feedback* harus disediakan. Aplikasi ini menyajikan animasi dan efek suara yang mengindikasikan kesuksesan atau kegagalan permainan.

2.3.2 First Person Shooter (FPS)

First Person Shooter (FPS) adalah salah satu dari banyak jenis *game*, sesuai judulnya *game* jenis ini mengambil pandangan orang pertama pada *game*-nya sehingga seolah-olah kita sendiri yang berada dalam *game* tersebut, kebanyakan *game* ini mengambil setting peperangan *game* dengan senjata-senjata militer (di indonesia, *game* jenis ini sering di sebut *game* tembak-tembakan). Contoh *game* jenis ini antara lain *Duke Nukem 3D, Quake, Blood, Unreal Tournament, Half-Life, Counter-Strike, Halo, Perfect Dark, TimeSpliters, Call of Duty, System Shock, dan Golden Eye 007*[9].



GAMBAR 2. 1 Contoh Game FPS

2.4 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri dan digunakan oleh berbagai macam perangkat *mobile*. Awalnya, *Google Inc*. membeli *Android Inc*., pendatang baru yang membuat peranti lunak untuk telepon seluler. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan *hardware*, *software*, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, *dan Nvidia*. Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode – kode Android di bawah lisensi *Apache*. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari *Google* atau *Google Mail Services* (GMS) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai *Open Handset Distribution* (OHD) [10]



GAMBAR 2. 2 Logo Android

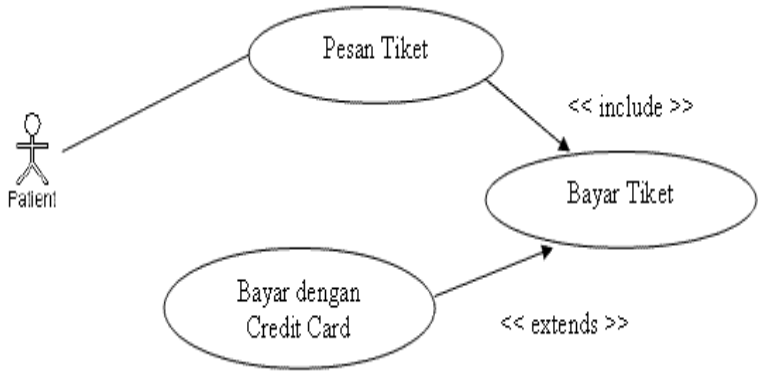
2.5 Unified Modeling Language (UML)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unfied Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks - teks pendukung. UML hanaya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek [11].

2.5.1 Use Case Diagram

Use case Diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah system dan siapa saja yang berhak menggunakan fungsi-fungsi itu [11].

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami.

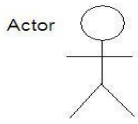


GAMBAR 2. 3 Contoh Use Case Diagram

Elemen – elemen diagram *use-case* adalah:

1. Aktor

   Aktor adalah pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain. Aktor adalah sesuatu atau seseorang yang berinterkasi dengan sistem. Aktor adalah tipe (kelas) bukan instan. Aktor merepresentasikan peran bukan perilaku individu dari sistem. Aktor mempunyai nama, nama yang dipilih seharusnya menyatakan peran aktor.

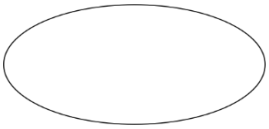


GAMBAR 2. 4 Aktor

2. *Use-case*

   Use-case adalah cara spesifik penggunaan sistem oleh aktor. Ciri – ciri dari *use-case* adalah:

   - Pola perilaku harus dipenuhi oleh sistem
   - Sekuen transaksi terhubung yang dilakukan aktor dan sistem
   - Memberikan sesuatu yang berharga bagi aktor

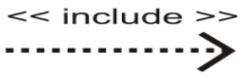


GAMBAR 2. 5 Use-case

3. Hubungan Antar *Use-case*

Keterhubungan antar *use-case* dengan *use-case* lain berupa generalisasi antara *use-case,* yaitu:

- *Include*, perilaku *use-case* merupakan bagian dari *use-case* lain.



GAMBAR 2. 6 Include

- *Extend*, perilaku *use-case* memperluas perilaku *use-case* yang lain.



GAMBAR 2. 7 Extend

2.5.2 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi [11].

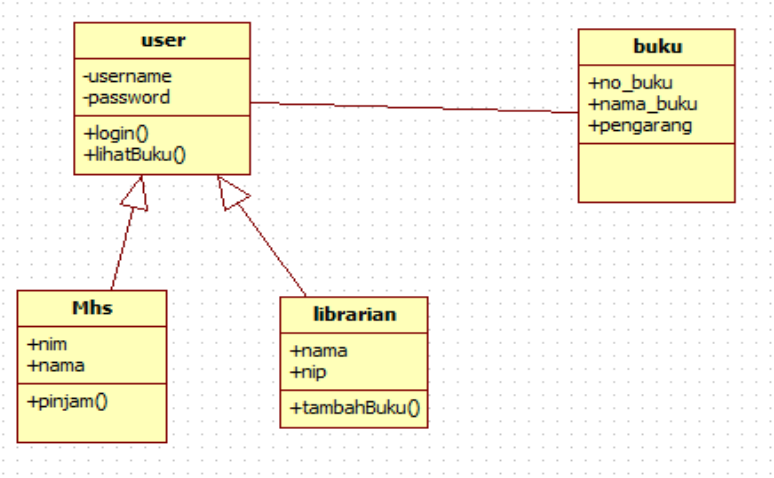
a. Atribut

   Atribut merupakan variabel – variabel yang dimiliki oleh suatu kelas.

b. Metode

   Metode atau operasi adalah fungsi – fungsi yang dimiliki oleh suatu kelas.

Class diagram dibuat agar programmer membuat kelas – kelas sesuai rancangan di dalam *class diagram* agar antara dokumentasi perancangan dan perangkat lunak sinkron [10].

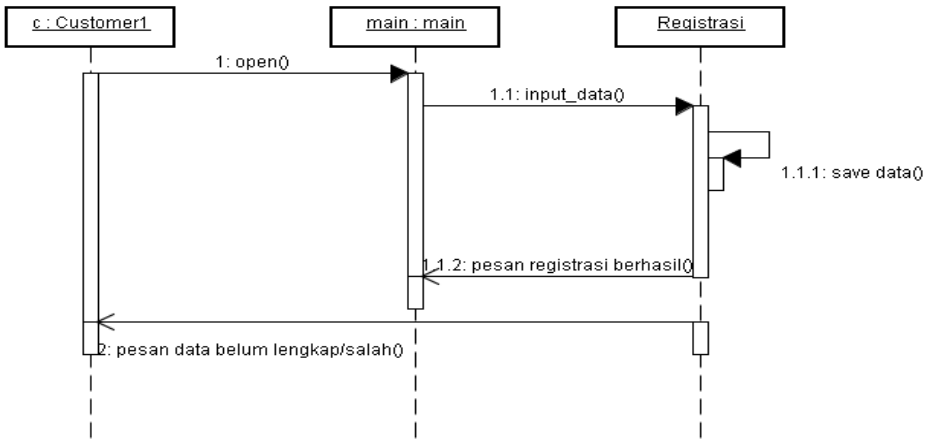


GAMBAR 2. 8 Contoh Class Diagram

2.5.3 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki oleh kelas yang diinstansiasi menjadi objek itu. Membuat *sequence diagram* juga dibutuhkan untuk melihat *scenario* yang ada pada *use case.*

Banyaknya diagram *sequence* yang harus dibuat adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram*, sehingga semakin banyak *use case* yang dibuat maka *sequence diagram* yang dibuat semakin banyak juga [11].



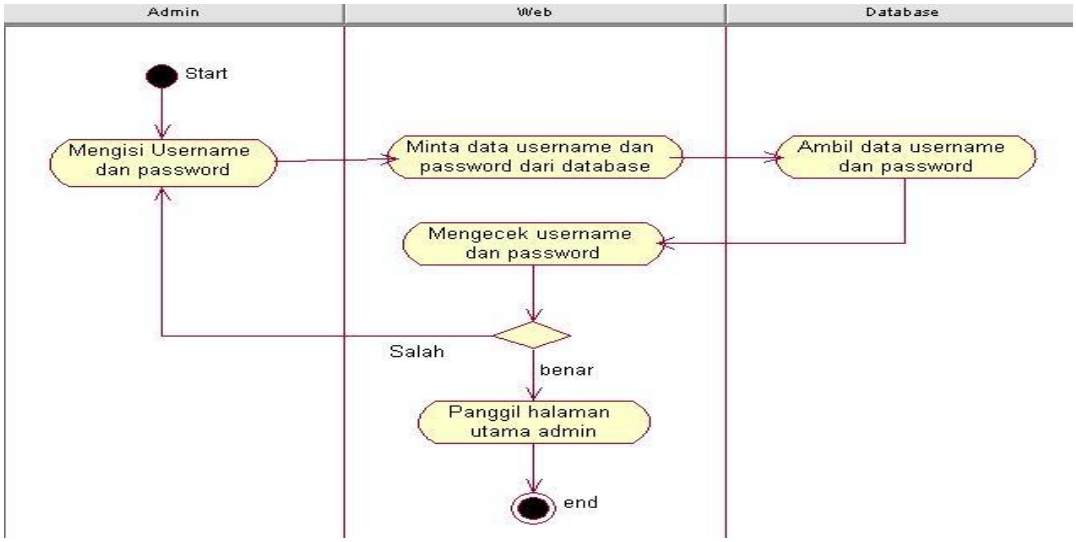
GAMBAR 2. 9 Contoh Sequence Diagram

2.5.4 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang berada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa *activity diagram* menggambarkan aktivitas bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan sistem [11].

Activity diagram juga banyak digunakan untuk mendefinisikan hal-hal berikut [11].

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokkan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didfinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.



GAMBAR 2. 10 Contoh Activity Diagram

2.6 Unity

Unity merupakan sebuah *game engine*, yaitu *software* pengolah gambar, grafik, suara, input, dan lain-lain yang ditujukan untuk membuat *game*. *Unity* merupakan *game engine multiplatform* yang mampu dipublikasi secara *standalone* (.exe), berbasis web, android, IoS, XBOX, maupun PS3, dengan catatan mendapatkan lisensi.

Unity adalah *game engine* yang dibuat oleh *Unity Technology*. Kelebihan *unity* dibandingkan *game engine* lainnya adalah kemampuan membuat *game cross platform*. Unity dapat membuat berbagai macam *game*, seperti *Role Playing Game (RPG), shooter, racing,* dan lain sebagainya [12].



GAMBAR 2. 11 Logo Unity

2.6.1 Sejarah dan Perkembangan Unity

Dalam beberapa tahun perkembangannya, sebelum dirilis, *Unity* telah diluncurkan pertama kali sebagai versi pra – rilis dengan *GooBall* sebuah *video game* yang didesain khusus untuk *Apple Macintosh*. *GooBall*, dengan *unity* pra – rilis, telah diluncurkan atau diumumkan pada bulan Maret tahun 2005, sementara itu *Unity* diluncurkan secara resmi sebagai aplikasi yang bersifat komersial pada dua bulan setelahnya yaitu bulan Juni tahun 2005. Satu tahun kemudian yaitu tahun 2006, aplikasi pengembang *game* ini telah menjadi nominasi untuk *Apple design awards* dalam kategori *"Best OS X Graphics"* [12].

Unity juga disebut sebagai aplikasi pengembang *multiplatform*, yang mana artinya *unity* mendukung untuk mengembangkan aplikasi *game* dan aplikasi yang lain untuk beberapa *platforms* seperti *game console, Mobile Phone platforms, Windows dan OS X* [12].

Sejak *unity* secara resmi dirilis sebagai *Unity* versi 1.0.1, banyak pembaharuan (*update*), *upgrades* dan fitur yang telah ditambahkan selama tahun perilisannya tersebut dan *Unity* terus berkembang secara terus – menerus. *Unity* hadir dengan beberapa pilihan perizinan *(license)*, dari yang versi gratis untuk *Unity* dasar dan *Unity pro* yang dijual dengan harga $1200. Baik *Unity* yang versi gratis dan *Unity* yang versi pro menawarkan banyak fitur yang dapat digunakan [12].

2.6.2 Fitur – Fitur Unity

Banyak kelebihan *game engine unity*, dapat dilihat dari beberapa fitur berikut ini [12].

1. *Rendering*

   Graphics engine yang digunakan adalah *Direct3D (Windows, Xbox 360), OpenGL (Mac, Windows, Linux, PS3), OpenGL ES (Android, iOS), dan proprietary APIs (Wii).* Ada pula kemampuan untuk *bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render-to-texture and full-screen post-processing effects.*

2. *Scripting*

   Script game engine dibuat dengan *Mono 2.6*, sebuah implementasi *open-source* dari *.NET Framework*. Programmer dapat menggunakan *UnityScript* (bahasa terkustomisasi yang terinspirasi dari *sintax ECMAScript*, dalam bentuk *JavaScript*), *C#*, atau *Boo* (terinspirasi dari sintax bahasa pemrograman *phyton*). Dimulai dengan dirilisnya versi 3.0, *Unity* menyertakan versi *MonoDevelop* yang terkustomisasi untuk *debug script*.

3. *Asset Tracking*

   Unity juga menyertakan *Server Unity Asset* sebuah solusi terkontrol untuk *developer game asset* dan *script*. Server tersebut menggunakan *PostgreSQL* sebagai *backend*, sistem *audio* dibuat menggunakan *FMOD library* (dengan kemampuan untuk memutar *Ogg Vorbis compressed audio*), *video playback* menggunakan *Theora codec*, *engine* daratan dan vegetasi (dimana mendukung *tree billboarding*, *Occlusion Culling* dengan Umbra), *built-in lightmapping* dan *global illumination* dengan *Beast, multiplayer networking* menggunakan *RakNet*, dan navigasi *mesh* pencari jalur *built-in*.

4. *Platforms*

   Unity support pengembangan ke berbagai *platform*. Didalam *project*, *developer* memiliki kontrol untuk mengirim keperangkat *mobile, web browser, desktop, and console. Unity* juga mengijinkan spesifikasi kompresi tekstur dan pengaturan resolusi di setiap platform yang didukung. Saat ini *platform* yang didukung adalah *BlackBerry 10, Windows 8, Windows Phone 8, Windows, Mac, Linux, Android, iOS, Unity Web Player, Adobe Flash, PlayStation 3, Xbox 360, Wii U* dan *Wii*. Meskipun tidak semua terkonfirmasi secara resmi, *Unity* juga mendukung *PlayStation Vita* yang dapat dilihat pada *game Escape Plan* dan *Oddworld: New 'n' Tasty.*

5. *Asset Store*

   Diluncurkan November 2010, *Unity Asset Store* adalah sebuah *resource* yang hadir di *Unity editor*. *Asset store* terdiri dari koleksi lebih dari 4,400 *asset packages*, beserta *3D models*, *textures* dan *materials*, sistem *particle*, musik dan efek suara, tutorial dan project, *scripting package*, *editor extensions* dan *online service*.

6. *Physics*

   Unity juga memiliki *suport built-in* untuk *PhysX physics engine* (sejak *Unity* 3.0) dari *Nvidia* dengan penambahan kemampuan untuk simulasi *real-time cloth* pada *arbitrary and skinned meshes, thick ray cast,* dan *collision layers.*

2.7 Skala Likert

Skala Likert adalah skala pengukuran yang digunakan untuk mengukur sikap seseorang, dengan menempatkan kedudukan sikapnya pada kesatuan perasaan kontinum yang berkisar dari “Sangat positif” hingga ke “sangat negatif" terhadap sesuatu (objek psikologis). Objek psikologis ini perlu ditentukan secara tegas, mengingat pembuatan sekumpulan item yang akan memperlihatkan opini terhadap objek, berdasarkan komponen-komponen dari objek psikologis tadi.

Secara umum semua item sikap dapat dipandang sebagai pernyataan-pernyataan yang berkaitan dengan opini, namun mengukur sikap dalam *skala likert* ini, menurut Daniel J, Mueller dibedakan ke dalam item kepercayaan (kognitif), item perasaan (afektif) dan item kecenderungan perilaku (konatif).

Item-item *likert* menyediakan respon dengan kategori yang berjenjang dari “sangat setuju” hingga “sangat tidak setuju”. Biasanya banyaknya jenjang kategori adalah lima, yaitu “sangat setuju”, “setuju”, “ragu-ragu”, “tidak setuju”, dan “sangat tidak setuju”.

Setiap kategori respon, jawaban responden selanjutnya perlu diberi skor. Untuk item positif, skor terbesar adalah 5, diberikan kepada respon “sangat

setuju", skor 4 diberikan kepada respon "setuju", skor 3 diberikan kepada respon "ragu-ragu, skor 2 diberikan kepada respon "tidak setuju", dan skor 1 diberikan kepada respon "sangat tidak setuju". Untuk menghitung total skor setiap responden adalah dengan cara menjumlahkan skor-skor item yang diperoleh responden. Oleh Karena itu, prosedur penskalaan likert sering disebut sebagai metode *summeted rating*.

TABEL 2. 1 Skor Skala Likert

Option	Skor Item Positif	Skor Item Negatif
Sangat Setuju	5	1
Setuju	4	2
Ragu-Ragu	3	3
Tidak Setuju	2	4
Sangat Tidak Setuju	1	5

Dalam menyusun penskalaan dengan metode *Likert's Summeted Rating*, dapat ditempuh langkah-langkah sebagai berikut.

1. Tentukan secara tegas sikap terhadap topik yang akan diukur.
2. Tentukan secara tegas, dimensi yang menyusun sikap tersebut, kognitif, afektif, atau konatif.
3. Susun pernyataan-pernyataan atau item yang merupakan alat pengukur dimensi yang menyusun sikap yang akan diukur, sesuai dengan indicator
4. Setiap item diberi pilihan respon yang sifatnya tertutup (*closed questionare*). Pilihan respons yang bias digunakan adalah 3, 5, 7, 9 atau 11. Dalam praktek, yang banyak digunakan adalah 5 pilihan respon.
5. Untuk setiap pilihan respon, jawaban diberi skor berdasarkan kriteria sebagai berikut. Apabila pilihan positif, maka angka terbesar diletakan pada respon "sangat setuju", sedangkan bila item negative, maka angka terbesar diletakan pada respon "sangat tidak setuju".

Untuk mengetahui posisi setiap responden tentang suatu variabel, tentukan skor maksimal dan skor minimal yang mungkin dicapai oleh setiap responden [13].