



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Raut Wajah

Bagi manusia, ekspresi raut wajah memberikan indikator emosi yang penting. Emosi dasar diungkapkan dan diakui dengan cara yang sama di seluruh dunia (Young, dkk., 2002). Pada penelitian ini untuk menampilkan *filter* yang telah di *shuffle* dengan algoritma Fisher-Yates, aplikasi harus dapat menentukan emosi dari raut wajah pengguna terlebih dahulu. Aplikasi dapat menentukan emosi raut wajah pengguna marah, senang, sedih, dan terkejut. Emosi marah dan sedih pada raut wajah diklasifikasikan sebagai emosi negatif, sedangkan emosi senang dan terkejut diklasifikasikan sebagai emosi positif (Alves, dkk., 2008). Menurut Faigin (2017). Raut wajah seseorang dengan alis yang lebih rendah pada area wajah akan dinilai sebagai emosi marah (Young, dkk., 2002). Menurut Listverse (2013), emosi senang pada raut wajah dianggap sebagai ekspresi paling universal dengan arti yang hampir sama di semua budaya dengan ditunjukkan oleh senyuman dan mata berbentuk bulan sabit, emosi pada raut wajah ini bahkan telah muncul ketika menjadi bayi (Listverse, 2013). Menurut Psychmechanics (2015), Alis yang sedikit miring ke atas di atas hidung membentuk 'V' terbalik dan menghasilkan kerutan jenis 'tapal kuda' di dahi, kelopak mata atas sangat sedikit terkulai, bibir direntangkan secara horizontal dan sudut bibir ditekan menandakan raut wajah dengan emosi sedih. Menurut Listverse (2013), emosi terkejut pada raut wajah mudah diidentifikasi dengan mata melebar dan mulut yang menganga.

2.2 Algoritma Fisher-Yates

Algoritma Fisher-Yates merupakan metode pengacakan yang lebih baik atau dapat dikatakan sesuai untuk pengacakan angka, dengan waktu eksekusi yang cepat serta tidak memerlukan waktu yang lama untuk melakukan proses pengacakan (*shuffle*). Algoritma Fisher-Yates dibagi menjadi dua metode, yaitu original dan modern. Pada penelitian ini akan digunakan metode *modern* yang khusus untuk pengacakan dengan sistem komputerisasi. Fisher-Yates *modern* sedikit berbeda dalam kecepatannya, hal ini dapat dibedakan pada sifat algoritma Fisher-Yates *modern* yang memindahkan angka yang telah tereliminasi ke urutan *array* terakhir. Berikut langkah-langkah yang dilakukan dalam melakukan pengacakan dengan algoritma Fisher-Yates *modern* (Nugraha, dkk., 2012).

- a. Siapkan *array* berisi angka 1-N.
 - b. Pilih sebuah angka acak K diantara 1 sampai dengan jumlah angka yang belum dieliminasi.
 - c. Dihitung dari bawah, tukar angka yang akan dieliminasi dengan angka terakhir pada *array*, kemudian masukkan angka yang tereliminasi ke *array temp* (Result).
 - d. Ulangi langkah kedua dan ketiga sampai semua angka tereliminasi
 - e. Urutan angka yang terdapat pada *array temp* adalah hasil permutasi acak
- Sebagai contoh, perbedaan cara kerja algoritma Fisher Yates *original* dan *modern* dapat dilihat pada gambar-gambar berikut.

a. Algoritma Fisher-Yates Original

Untuk algoritma Fisher-Yates Original, langkah pertama yang dilakukan adalah dengan menyiapkan *array* berisi angka 1 sampai dengan N (*total length*).

Tabel 2.1. Ilustrasi Algoritma Fisher-Yates Original (1)

Range	Roll	Scratch	Result
		1 2 3 4 5 6 7 8	

Kemudian, ambil sebuah angka secara acak di antara 1 sampai dengan jumlah angka yang belum tereliminasi. Misalnya hasil pengambilan nilai secara acak adalah angka 3, maka angka 3 yang berada pada *array* ke-3 akan dieliminasi dan akan dimasukkan ke *array temp*.

Tabel 2.2. Ilustrasi Algoritma Fisher-Yates Original (2)

Range	Roll	Scratch	Result
1-8	3	1 2 <u>3</u> 4 5 8 7	3

Selanjutnya, lakukan proses yang sama dengan mengambil angka secara acak diantara 1 sampai dengan jumlah angka yang belum tereliminasi. Misalnya hasil pengambilan nilai acak adalah 4, maka angka 5 pada posisi ke-4 akan tereliminasi dan dikirim ke *array temp*.

Tabel 2.3. Ilustrasi Algoritma Fisher-Yates Original (3)

Range	Roll	Scratch	Result
1-7	4	1 2 3 4 <u>5</u> 8 7	3 5

Proses yang sama akan terus dilakukan hingga semua angka tereliminasi. Hasil pengacakan dengan Algoritma Fisher-Yates Original dapat diambil dari *array temp* (kolom *result* pada tabel).

Tabel 2.4. Ilustrasi Algoritma Fisher-Yates Original (4)

Range	Roll	Scratch	Result
1 - 6	5	1 2 3 4 5 6 <u>7</u> 8	3 5 7
1 - 5	3	1 2 3 <u>4</u> 5 6 7 8	3 5 7 4
1 - 4	4	1 2 3 4 5 6 7 <u>8</u>	3 5 7 4 8
1 - 3	1	<u>1</u> 2 3 4 5 6 7 8	3 5 7 4 8 1
1 - 2	2	1 2 3 4 5 <u>6</u> 7 8	3 5 7 4 8 1 6
		1 2 3 4 5 6 7 8	3 5 7 4 8 1 6 2

b. Algoritma Fisher-Yates Modern

Pada algoritma Fisher-Yates Modern terdapat sedikit perubahan pada langkah-langkah yang dilakukan. Langkah pertama, siapkan *array* berisi angka 1 hingga dengan N (*total length*).

Tabel 2.5. Ilustrasi Algoritma Fisher-Yates Modern (1)

Range	Roll	Scratch	Result
		1 2 3 4 5 6 7 8	

Kemudian, ambil sebuah angka secara acak di antara 1 sampai dengan jumlah angka yang belum tereliminasi. Misalnya didapatkan angka 6, maka angka 6 akan ditukar dengan angka 8 yang berada pada posisi *array* terakhir dan dieliminasi.

Tabel 2.6. Ilustrasi Algoritma Fisher-Yates Modern (2)

Range	Roll	Scratch	Result
1-8	6	1 2 3 4 5 <u>8</u> 7	<u>6</u>

Contoh selanjutnya, misalnya didapat angka 2, maka angka 2 yang berada pada *array* ke-2 akan ditukar dengan angka 7 yang berada pada posisi *array* terakhir. Kemudian, angka 2 akan tereliminasi dan di kirim ke array temp.

Tabel 2.7. Ilustrasi Algoritma Fisher-Yates Modern (3)

Range	Roll	Scratch	Result
1-7	2	1 7 3 4 5 8	<u>2</u> 6

Proses yang sama akan terus dilakukan hingga semua angka tereliminasi. Urutan angka yang terdapat pada *array temp* adalah hasil dari pengacakan yang dilakukan.

Tabel 2.8. Ilustrasi Algoritma Fisher-Yates Modern (4)

Range	Roll	Scratch	Result
1 - 6	6	1 7 3 4 5	<u>8</u> 2 6
1 - 5	1	<u>5</u> 7 3 4	<u>1</u> 8 2 6
1 - 4	3	5 7 <u>4</u>	<u>3</u> 1 8 2 6
1 - 3	3	5 7	<u>4</u> 3 1 8 2 6
1 - 2	1	<u>7</u>	<u>5</u> 4 3 1 8 2 6
			<u>7</u> 5 4 3 1 8 2 6

2.3 Color Theory

Penelitian telah menunjukkan bahwa warna yang berbeda memiliki efek yang berbeda pula seperti psikologis, emosional dan fisik. Manusia dalam merespon sebuah warna berada pada tingkat sadar dan tidak sadar. Mereka adalah alat yang hebat untuk memanipulasi cara kita merasakan (Rhynard, 2012).

Penelitian saat ini telah menyelidiki bagaimana makna psikologis yang tersirat oleh warna merah membiaskan proses ekspresi kemarahan. Merah telah terbukti membawa makna ancaman, bahaya dan kemarahan. Warna merah telah mendapatkan perhatian paling banyak karena digunakan dalam kehidupan sehari-hari untuk menyampaikan bahaya dan potensi ancaman dalam konteks yang berbeda. Mengingat hubungan erat antara warna merah, marah, dan agresi, warna merah sangat cocok untuk memfasilitasi pemrosesan ekspresi kemarahan. (Young, dkk., 2013). Menurut Naz Kaya (2004), warna hitam terlihat untuk membangkitkan emosi negatif seperti kesedihan, depresi dan ketakutan karena berhubungan dengan kematian, duka atau peristiwa tragis. Seperti warna hitam, warna abu-abu terkait dengan emosi negatif seperti perasaan sedih, depresi, kesepian dan ketakutan. Menurut Naz Kaya (2004), dalam penyelidikan asosiasi emosional anak-anak dengan warna, Boyatzis dan Varghese (1994) menemukan bahwa warna terang seperti biru dikaitkan dengan emosi positif misalnya, senang dan kuat. Menurut penelitian yang dilakukan Hermsen (2012), peserta diminta untuk mewarnai gambar yang menggambarkan berbagai emosi yang digambarkan dengan kalimat di bagian atas setiap gambar. Hasilnya, Hermsen melaporkan asosiasi emosi-warna berikut ini: marah dengan merah, kejutan serta kebahagiaan dengan oranye dan kuning.

2.4 iOS Development

iOS adalah sistem operasi perangkat bergerak yang dikembangkan dan didistribusikan oleh Apple Inc. Sistem operasi ini pertama diluncurkan tahun 2007 untuk iPhone dan iPod Touch, dan telah dikembangkan untuk mendukung perangkat Apple lainnya seperti iPad dan Apple TV. Tidak seperti Windows

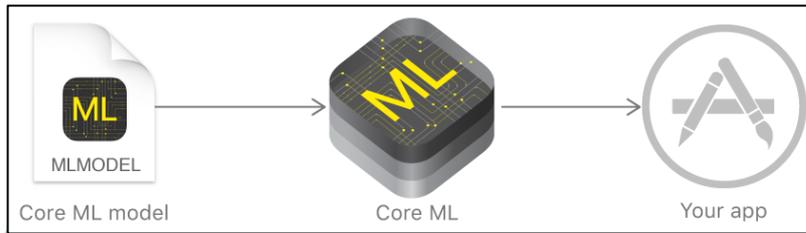
Phone (Windows CE) Microsoft dan Android Google, Apple tidak melisensikan iOS untuk diinstal di perangkat keras non-Apple. Hingga 12 September 2012, App Store Apple berisi lebih dari 700.000 aplikasi iOS, yang secara kolektif telah diunduh lebih dari 30 miliar kali. Sistem operasi ini memiliki pangsa pasar 14,9% untuk unit sistem operasi perangkat bergerak telepon cerdas yang dijual pada kuartal ketiga 2012, terbanyak setelah Android Google (IDC, 2016). Pada bulan Juni 2012, iOS mencakup 65% konsumsi data web perangkat bergerak (termasuk di iPod Touch dan iPad).

Xcode adalah sebuah *integrated development environment* (IDE) untuk macOS yang berfungsi untuk mengembangkan perangkat lunak macOS, iOS, watchOS, dan tvOS. Pertama kali dirilis pada tahun 2003, dan versi terakhirnya sekarang adalah versi 8 yang tersedia gratis di Mac App Store untuk pengguna macOS Sierra (Apple, 2018).

Swift adalah bahasa pemrograman yang kokoh dan intuitif yang diciptakan Apple untuk mengembangkan aplikasi untuk iOS, Mac, Apple TV, dan Apple Watch. Bahasa ini didesain untuk memberikan lebih banyak kebebasan kepada pengembang. Swift mudah digunakan dan bersifat sumber terbuka (*open source*) (Apple, 2018).

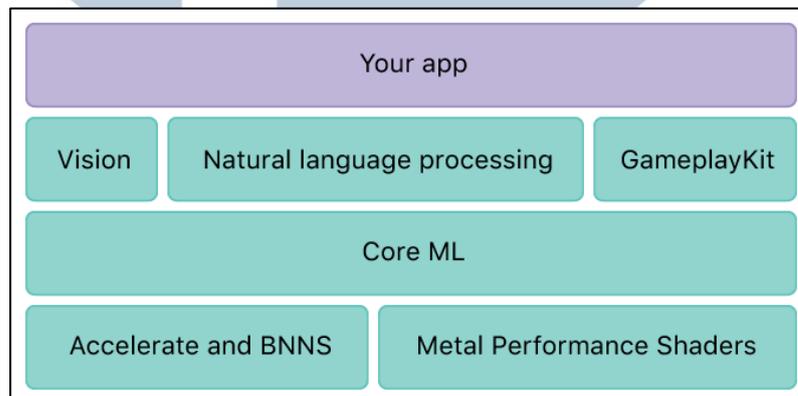
2.5 Core ML

Core ML adalah *framework* yang memungkinkan aplikasi untuk terintegrasi dengan beragam jenis model *machine learning*. Selain mendukung *deep learning* yang mendalam dengan lebih dari 30 *layer types*, *framework* ini mendukung model standar seperti *tree ensembles*, SVMs, dan *generalized linear models* (Apple, 2018).



Gambar 2.1 Mekanisme Kerja Core ML (Apple, 2017)

Dengan Core ML, *developer* dapat mengintegrasikan *model machine learning* yang telah dilatih kedalam aplikasi. *Model machine learning* yang telah dilatih adalah hasil penerapan algoritma *machine learning* untuk set dari *training data*. *Model* akan membuat prediksi berdasarkan input data baru. Sebagai contoh, sebuah model yang telah di *training* untuk menentukan harga sebuah rumah mungkin dapat memprediksi menghitung harga rumah berdasarkan jumlah dari kamar tidur dan kamar mandi.



Gambar 2.2 Arsitektur Aplikasi dengan Core ML (Apple, 2017)

Core ML adalah fondasi untuk *domain-specific frameworks* dan *functionality*. Core ML mendukung *Vision framework* untuk melakukan analisis foto, *Foundation* untuk *natural language processing*, dan *GameplayKit* untuk mengevaluasi *learned decision trees*. Core ML sendiri dibangun diatas *low-level primitives* seperti *Accelerate and BNNS* dan *Metal Performance Shaders*. Kinerja Core ML dioptimalkan untuk on-device performance yang meminimalkan *memory*

footprint dan konsumsi daya. Menjalankan Core ML secara terisolasi pada perangkat memastikan privasi data pengguna dan menjamin bahwa aplikasi tetap berfungsi dan responsif ketika koneksi jaringan tidak tersedia.

2.6 Vision

Vision adalah framework yang memungkinkan aplikasi untuk dapat melakukan *face tracking*, *face detection*, *text detection*, *rectangle detection*, *barcode detection*, *object tracking* dan *image registration* (Apple, 2018).



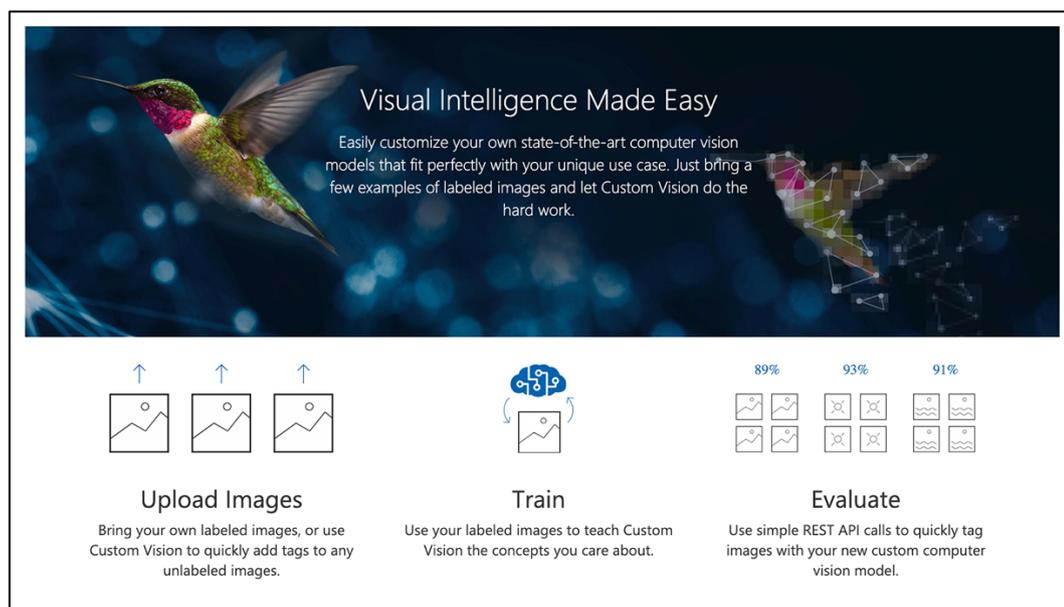
Gambar 2.3 Mekanisme Kerja Vision (MoveoApps, 2018)

Dalam mekanisme kerja Vision, terdapat 3 kategori kelas utama, yaitu: VNRequest, VNRequestHandler dan VNSequenceHandler, dan VNObservation. VNRequest merupakan *abstract superclass* untuk *request* analisis foto. VNRequest menjadi jembatan untuk melakukan sebuah fungsi (*request*) pada Vision. VNImageRequestHandler dan VNSequenceRequestHandler akan melakukan eksekusi dari *request* yang telah diberikan beserta *completion block*, VNImageRequestHandler dan VNSequenceRequestHandler ini dapat mengeksekusi lebih dari satu request pada gambar yang diberikan. VNObservation merupakan hasil *array* yang berisi observasi dari hasil *request*.

Dengan Core ML Framework, Vision dapat menggunakan *model machine learning* yang telah di *training* untuk mengklasifikasikan input data, seperti, menerapkan klasifikasi *model* kepada foto dan melakukan *preprocess image* tersebut untuk membuat tugas *machine learning* lebih mudah dan dapat diandalkan.

2.7 Custom Vision

Custom Vision adalah *tool* yang dibuat oleh Microsoft untuk membangun *custom image classifiers*. *Tool* ini memudahkan serta mempercepat untuk membangun, menyebarkan, dan mengimprove *image classifiers*. *Tool* ini menyediakan REST API dan antarmuka web untuk mengunggah gambar dan melakukan *training* (Microsoft Azure, 2018). Custom Vision Service melakukan *image classification* tetapi tidak sampai *object detection*. Ini berarti bahwa Custom Vision Service mengidentifikasi apakah suatu gambar adalah objek tertentu, tetapi tidak di mana objek itu berada di dalam gambar. Custom Vision dirancang untuk memudahkan dalam membangun *classifier*, dan untuk membantu meningkatkan kualitas *classifier* dari waktu ke waktu. Custom Vision dapat meng-*export model* yang telah di *training* kedalam *CoreML model* yang akan digunakan pada penelitian ini. Custom Vision akan memberikan hasil *training performance* yang menampilkan nilai *precision* serta nilai *recall* sebagai indikator seberapa bagus dari sebuah *model* (Roth, 2018).

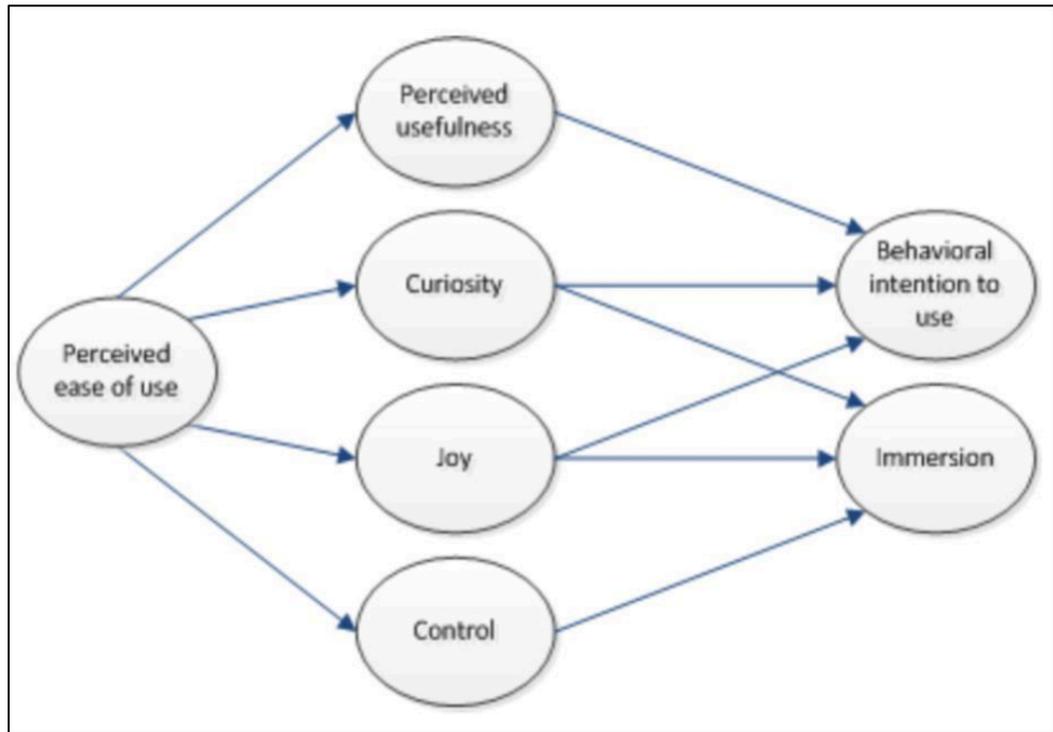


Gambar 2.4 Header Web Custom Vision (Microsoft, 2018)

2.8 Hedonic Motivation System Adoption Model

Hedonic Motivation System Adoption Model (HMSAM) merupakan model yang digunakan untuk mengukur sebuah sistem informasi dalam memenuhi kebutuhan intrinsik pengguna. HMSAM merupakan sebuah model pengembangan dari Heijden (2004) yang digabungkan dengan *cognitive absorption* oleh Lowry yang dinamakan Hedonic Motivation System Adoption Model (HMSAM) (Lowry, dkk., 2013). Hasil yang didapatkan Lowry (2013) disimpulkan bahwa kemudahan yang dirasakan ketika menggunakan sistem (*perceived ease of use*) bukan merupakan faktor yang mempengaruhi secara langsung pada keinginan seseorang menggunakan sistem informasi (*behavioral intention to use*) dan perasaan seseorang terbawa suasana ketika menggunakan sistem (*immersion*) tetapi faktor *perceived usefulness*, *curiosity*, *joy*, dan *control* yang berdampak langsung pada *behavioral intention to use* dan *immersion*. Pada HMSAM, keinginan seseorang menggunakan sistem informasi (*behavioral intention to use*) ditentukan dari seberapa besar orang tersebut melihat keuntungan yang didapatkan (*percieved of usefulness*), rasa ingin tahu (*curiosity*) dan perasaan senang yang didaptkkan ketika menggunakan sistem (*joy*), sedangkan untuk perasaan seseorang terbawa suasana ketika menggunakan sistem (*immersion*) dipengaruhi oleh rasa ingin tahu (*curiosity*), perasaan senang yang didapatkan ketika menggunakan sistem (*joy*), dan perasaan seseorang mendapatkan kendali dalam menggunakan sistem. Model HMSAM dapat dilihat pada gambar 2.5.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2.5 Hedonic Motivation System Adoption Model (HMSAM)
(Lowry, P. B. dkk, 2013)

2.9 Skala Likert

Menurut Sugiyono (2012) skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam memberi jawaban, responden dapat memberi tanda *checklist* atau tanda silang pada jawaban yang dipilih sesuai pertanyaan. Skala Likert umumnya menggunakan 5 angka penelitian, yaitu sebagai berikut.

- a. Sangat Setuju (SS) dengan *score* 5.
- b. Setuju (S) dengan *score* 3.
- c. Ragu-ragu (RR) dengan *score* 3.
- d. Tidak Setuju (TS) dengan *score* 2.
- e. Sangat Tidak Setuju (STS) dengan *score* 1.

Menurut Sugiyono (2012), untuk menentukan persentase skor yang didapatkan dari pengambilan data kuesioner dengan skala Likert menggunakan Rumus (2.1).

$$PS = \frac{(SS * 5) + (S * 4) + (RR * 3) + (TS * 2) + (STS * 1)}{(5 * \text{Jumlah Responden})} \times 100\%$$

Keterangan:

PS = Persentase Skor

SS = Sangat Setuju

S = Setuju

RR = Ragu-ragu

TS = Tidak Setuju

STS = Sangat Tidak Setuju

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA