BAB II

LANDASAN TEORI

2.1. Sistem Rekomendasi

Sistem rekomendasi merupakan suatu sistem yang dibangun dari minat *user* terhadap suatu *item* yang mana sistem memprediksikan suatu rekomendasi *item* kepada *user* (Sanjung, 2011). Sistem rekomendasi juga berarti suatu Teknik perangkat lunak yang dapat memberikan saran – saran akan *item* yang bermanfaat bagi *user* (Ricci, et al., 2011). Rekomendasi tersebut berhubungan dengan proses pengambilan keputusan, bahan kue mana yang akan diambil dan dibeli. Oleh karena itu diperlukan model rekomendasi yang tepat dalam pembelian bahan kue.

2.2 Bahan kue

Bahan kue merupakan bahan – bahan seperti krim, coklat, terigu, gula, coklat, mentega, dan dihias dengan hiasan yang digunakan untuk membuat kue. (Tenri, dkk., 2013). Toko Lucky sendiri memiliki 1879 bahan kue yang dibagi dalam 44 kategori. Bahan kue sebanyak ini diperlukan suatu rekomendasi agar lebih efektif ketika pembeli ingin melakukan pembelian maupun dari sisi toko agar *item* lebih banyak kejual sehingga mengurangi *item* yang lewat masa pakai.

2.3 Data Mining

Data mining merupakan proses semi otomatik yang menggunakan teknik statistik, matematika, *artificial Intelligence*, dan *machine learning,* untuk mengekstraksi dan mengindentifikasi informasi maupun pengetahuan yang potensial dan berguna, dan bermanfaat di dalam *database* besar (turban , 2005) *data*

mining juga berarti teknologi yang menggabungkan metoda analisis tradisional dengan algoritma canggih untuk memproses *data* dengan volume yang besar, sehingga informasi atau pengetahuan yang berguna dalam *database* dapat d*item*ukan (Safitri , 2012), untuk memroses informasi pada penjualan toko lucky dengan baik, diperlukan suatu algoritma untuk menemukan informasi yang dapat berguna bagi sistem, yaitu dengan algoritma FP-Growth.

2.2.1 Association Rule Mining

Untuk menghasilkan rekomendasi yang efisien, diperlukan Teknik *data mining* yang tepat, dimana teknik association rule merupakan salah satunya, merupakan teknik untuk menemukan aturan assosiatif anatara suatu kombinasi *item*, terkenal diaplikasikannya untuk menganalisa isi keranjang belanja di pasar swalayan, (dinus, 2018). Contoh aturan assosiatif dari analisa pembelian dari suatu pasar swalayan adalah dapat diketahuinya berapa besar kemungkinan seorang pelanggan membeli roti sama susu bersamaan, sehingga penempatan barang pun dapat dirancang atau promosi tertentu dapat dijalankan.

Dua parameter dapat menentukan penting tidaknya suatu association rule, yaitu support (nilai penunjang) yaitu persentase kombinasi *item* tersebut dalam *database* dan confidence (nilai kepastian) yaitu kuatnya hubungan antar *item* dalam aturan assosiatif, contoh bentuk association rule.

{ROTI, MENTEGA} -> {SUSU} {support = 40%, confidence = 50%}

Yang artinya : “50% dari transaksi di *database* yang memuat *item* roti dan mentega juga memuat *item* susu. Sedangkan 40% dari seluruh transaksi yang ada

di *database* memuat ketiga *item* itu." Atau dengan kata lain seorang juga membeli susu apabila membeli roti dan mentega dengan kemungkinan 50%. Analisis asosiasi didefinisikan suatu proses untuk menemukan semua assosiation rule yang memenuhi syarat minimum untuk support (minimum support) dan syarat minimum unutk confidence (minimum confidence).

2.3.1 Support

Support adalah presentase dari seluruh transaksi yang mengandung suatu *item* set. Sebagai contoh, produk mentega dan coklat muncul bersamaan dalam 500 transaksi dari total 1500 transaksi sehingga nilai *support* dari coklat dan mentega dalam transaksi tersebut adalah 30% ((500/1500) * 100%). Adapun rumus yang digunakan untuk menghitung nilai support seperti pada Rumus 2.1(Larose, 2014).

$$\text{Support (A)} = \frac{\text{Jumlah Transaksi Mengandung A}}{\text{Total Transaksi}} \quad \ldots(2.1)$$

2.3.2 Confidence

Confidence adalah perbandingan antara nilai *support* dari himpunan *items* yang mendahuiluinya dengan nilai *support* dari himpunan *items* yang terdapat di dalam *rule* tersebut. *Confidence* juga dapa diartikan sebagai suatu nilai kepastian yaitu kuatnya hubungan antar *item* dalam sebuah *itemset*. *Confidence* dapat dicari setelah pola frekuensi munculnya sebuah *item* d*item*ukan. Sebagai contoh, apabila tedapat rule "Jika membeli coklat dan keju maka konsumen membeli susu". Untuk menghitung nilai *confidence* tersebut digunakan Rumus 2.2 (Larose, 2014).

$$\text{Confidence} = P(B \mid A) = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Jumlah Transaksi Mengandung A}} \quad \ldots(2.2)$$

2.3 FP-Growth

Algoritma Frequent Pattern Growth atau biasa disingkat FP-Growth merupakan alternatif, sekaligus pengembangan algoritma apriori untuk menentukan himpunan *data* yang paling sering muncul (*frequent itemset*) (Anggraeni, 2014), dengan 2 tahapan utama, yaitu pembuatan tree dengan struktur *data* yang disebut FP-tree dari *data* transaksi yang telah disortir dari minimum *support* dan *confidence*. FP-tree yang telah dibentuk, diurutkan secara *descending* dari yang paling sering hingga paling jarang dibeli, yang kemudian dapat langsung diekstrak menjadi informasi rekomendasi (Kurniawan, 2017). Pemindaian *database* pada FP-Growth yang tidak berulang dalam proses *mining* karena tidak ada *candidate generation* sehingga lebih cepat (Arifin, 2017).

2.3.1 Tahapan Algoritma FP-Growth

Algoritma FP-Growth dibagi menjadi 2 tahap. Tahap pertama adalah tahap pembuatan *FP-Tree* dan tahap kedua adalah mengekstrak *itemset* yang yang sering muncul pada *FP-Tree*.

Tahapan pertama algoritma FP-Growth, input yang digunakan adalah *data* transaksi yang kemudian dicari nilai *support* kategori *item* dan menghilangkan *data* dengan *support* yang lebih kecil daripada minimum *support*. Proses ini dilakukan agar dalam pembentukan FP-Tree tidak memakan waktu dalam perhitungan *item* dengan frekuensi yang rendah. Kemudian *data* akan diurutkan secara dari yang paling sering dibeli hingga yang paling jarang dibeli, sehingga pembentukan *node* pada *FP-Tree* lebih efisien. FP-Tree memiliki root yang *null*. Setiap transaksi dalam *database* misalnya dengan kategori *item* {mentega, coklat, plastik} maka mentega

adalah *child* dari root, coklat adalah *child* dari mentega, plastik adalah *child* dari coklat. Jika suatu *item* sudah terdaftar sebagai child dari root maupun *parent*, maka *count* dari kategori tersebut akan ditambahkan. Hal yang sama akan dilakukan terhadap semua *data* transaksi hingga terbentuk keseluruhan FP-tree.

Tahap kedua dalam algoritma FP-growth adalah mengekstrak *data* dari *FP-tree* untuk mencari *items*et yang sering muncul. Prosedur dalam mengekstrak *FP-tree* adalah sebagai berikut.

a. Pada setiap *node* yang ada pada *FP-tree* akan dibaca nilai *count*. Setelah itu *node* tersebut akan dilihat *parent*-nya agar diketahui *item* apa saja yang dibeli bersamaan dengan barang tersebut sehingga didapatkan *itemset* dan jumlah *itemset*.
b. Setelah didapatkan *item* yang dibeli bersamaan dengan *item* pada *node*, *data* tersebut akan disimpan dan kembali dilakukan pengekstrakan *tree* terhadap node berikutnya.
c. Hasil ekstaksi *FP-tree* akan diurutkan berdasarkan dengan frekuensi tertinggi sehingga *user* mendapatkan rekomendasi kategori *item* yang memiliki pola yang paling sering dibeli bersamaan.

Dengan demikian dihasilkan pola dengan frekuensi tinggi dan informasi tersebut dapat digunakan oleh pengguna aplikasi untuk mendapatkan rekomendasi *item* apa lagi yang biasa dibeli dari *item* yang dilihat *user* berdasarkan *frequent pattern* sehingga diharapkan dapat meningkatkan *cross selling*. Konsumen mendapatkan pengalaman belanja baru tanpa barang yang tertinggal atau kombinasi *item* dari rekomendasi yang belum pernah dicoba.

```
Fp-Growth (Tree,α)
    For each(anItem in the header of Tree) do{
    β := anItem ∪ α
    Generate(β with support = anItem.support)
    Construct β's conditional base pattern
    And β's conditional FP-Tree Tree β
    if Treeβ ≠ ∅

}

Then call FP-growth(Tree β, β)
```

Gambar 2.1 Pseudocode FP-Growth (Purba, Siswanto 2017)

a. Tahap Pembangunan Conditional Pattern Base

Conditional Pattern base merupakan sub-*database* yang berisi prefix path (lintasan/jalur prefix) dan suffix pattern (pola akshiran) (Han, 2004). Pembangkitan conditional pattern didapatkan melalui penyimpanan struktur tree malalui penndekatan FP-Tree yang telah dibangun dan disusun sesuai prosedur sebelumnya. Untuk menemukan conditional pattern base dari contoh kasus di atas, maka perlu ditentukan tree dengan path yang berakhir dengan support count terkecil, dalam contoh kasus pada subbab 2.3.2

2.3.2 Tahap Pebangkitan Conditional FP-Tree

Support count dari setiap *item* pada setiap conditional pattern base akan dijumlahkan pada tahap ini, kemudian setiap *item* yang memiliki jumlah support count lebih besar sama dengan minimum support count akan dibangkitkan dengan conditional FP-Tree(Han,2004). Sebagai contoh untuk conditional FP-Tree, titik temu untuk produk Untuk lebih detailnya dapat dilihat melalui Tabel 2.1, dan Tabel 2.2.

Tabel 2.1 *Data* transaksi mentah

No Transaksi	ID Transaksi	Kategori *Item*
1	0159/IK/UTM/0119	SELAI, FILLING, GLAZE, GLAZE, GLAZE, GLAZE, GLAZE, GLAZE, MENTEGA, MENTEGA, MENTEGA
2	0161/IK/UTM/0119	COKLAT, KOTAK, SUSUBUBUK, SUSUBUBUK
3	0180/IK/UTM/0319	INSTANT, COKLAT, MESIS, COKLAT, GLAZE, MESIS, MESIS, GLAZE, COKLAT, MESIS, COKLAT, COKLAT, COKLAT, COKLAT, GLAZE,
4	0187/IK/UTM/0619	CASES, COKLAT, PREMIKS, COKLAT, PERISA, INSTANT
5	0188/IK/UTM/0619	KOTAK, MIKA, LILIN, LILIN, KOTAK, GULA

Tabel 2.1 merupakan sampel transaksi – transaksi yang diambil secara acak dari *database*, dengan mengambil kategori – kategori *item*nya saja, dimana dalam satu transaksi dapat mencapai 5 kategori yang berbeda, yang kemudian kategori – kategori ini diurutkan berdasarkan frekuensi munculnya dari transaksi, dimana apabila ada pengulangan kategori yang muncul dihitung 1 yang dapat dilihat pada Tabel 2.2 di bawah ini.

Tabel 2.2 Frekuensi Kategori

Rank	Kategori	Frekuensi
1	COKLAT	3
2	GLAZE	2
3	INSTANT	2
4	KOTAK	2
5	MENTEGA	1
6	KOTAK	1
7	SUSUBUBUK	1
8	INSTANT	1
9	GULA	1
10	MIKA	1
11	PERISA	1
12	PREMIKS	1
13	SELAI	1

14	LILIN	1

Tabel 2.2 merupakan frekuensi kategori – kategori yang muncul pada transaksi sampel yang terdapat pada tabel 2.1. setiap kategori yang muncul pada 1 transaksi, dimana ketika ada kategori yang sama pada satu transaksi terhitung 1, dan kemunculan – kemunculan kategori dijumlahkan.

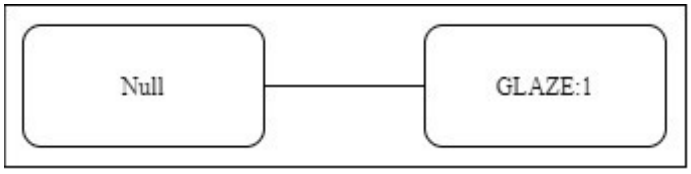
2.3.3 Tahap Pencarian Frequent *Items*et

Apabila conditional FP-Tree merupakan lintasan tunggal (single path), maka didapatkan frequent *items*et dengan melakukan kombinasi *item* untuk setiap conditional FP-Tree. Jika bukan lintasan tunggal, maka dilakukan pembangkitan FP-Growth secara rekursif (Han, 2004). Algoritma FP-Growth menemukan frequent *items*et yang berakhiran suffix tertentu dengan menggunakan metode divide and conquer untuk memecah problem menjadi subproblem yang lebih kecil (Han, 2004). Untuk frequent pattern *items*et pada contoh kasus di atas, didapatkan hasil seperti pada Tabel 2.3 dan dilanjutkan dengan pembangkitan fp-tree seperti pada Gambar 2.2 hingga 2.7

Tabel 2.3 Frekuensi Kategori

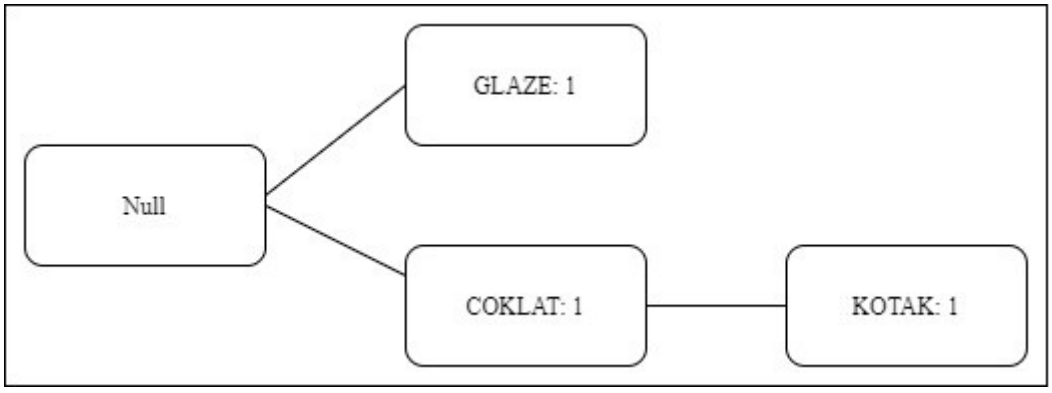
Transaksi	**Kategori**
0159/IK/UTM/0119	GLAZE
0161/IK/UTM/0119	COKLAT, KOTAK
0180/IK/UTM/0319	COKLAT, GLAZE, INSTANT
0187/IK/UTM/0619	COKLAT, INSTANT
0188/IK/UTM/0619	KOTAK

Tabel 2.3 merupakan sampel – sampel transaksi dengan kategori – kategori yang terdapat pada transaksi tersebut, dari kategori – kategori tersebut, membentuk *tree*, tipe *data* yang berbentuk seperti sebuah pohon seperti pada gambar 2.2 hingga gambar 2.6, dimana pada akhirnya membentuk suatu asosiasi.



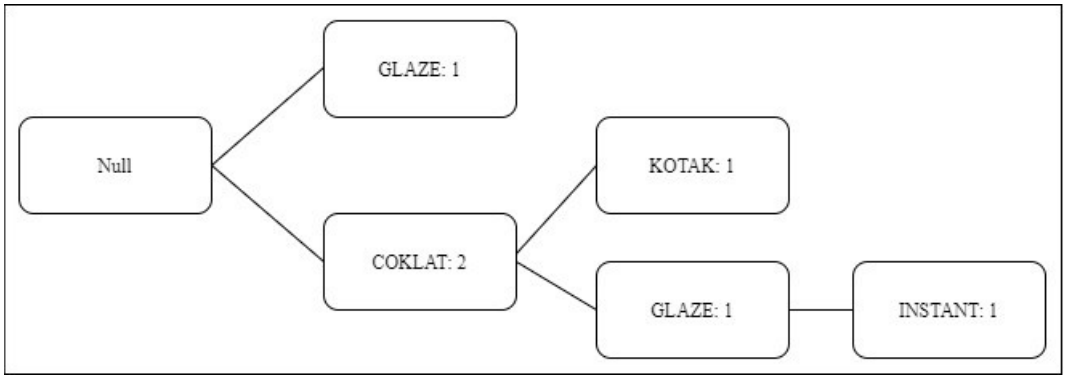
Gambar 2.2 Transaksi 1

Gambar 2.2 Transaksi 1 merupakan awal transaksi dengan 1 kategori, yaitu kategori GLAZE, dikarenakan tree masi null dan tidak ada kategori apapun, maka tree yang awalnya null, karena ditambahkan *node* GLAZE, dengan *support* sebanyak 1, sesuai dari proses ini GLAZE baru muncul 1 kali.



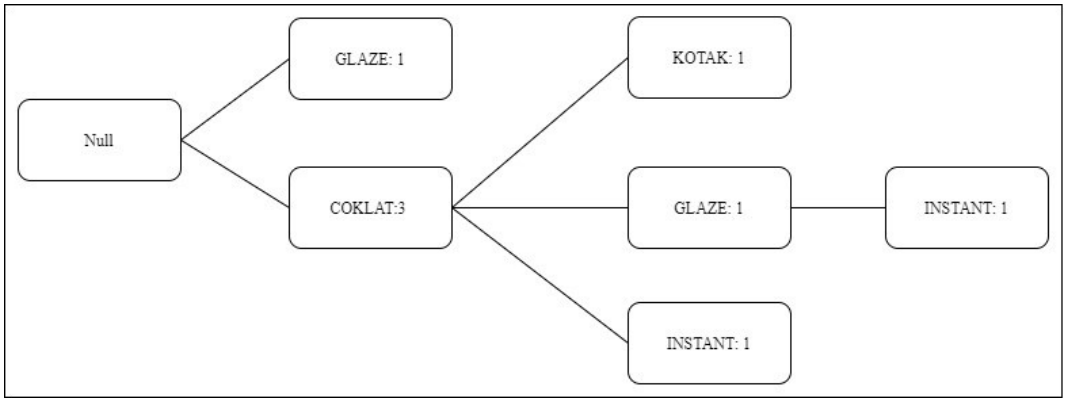
Gambar 2.3 Transaksi 2

Gambar 2.3 merupakan penambahan node dari tree yang sudah dimasukkan transaksi pertama, dimana transaksi kedua adanya kategori COKLAT dan diikuti dengan KOTAK, karena GLAZE tidak ada pada transaksi ini maka support pada GLAZE tidak bertambah, dan node baru terbentuk dengan kategori COKLAT, dan kemudian KOTAK cabang dari COKLAT.



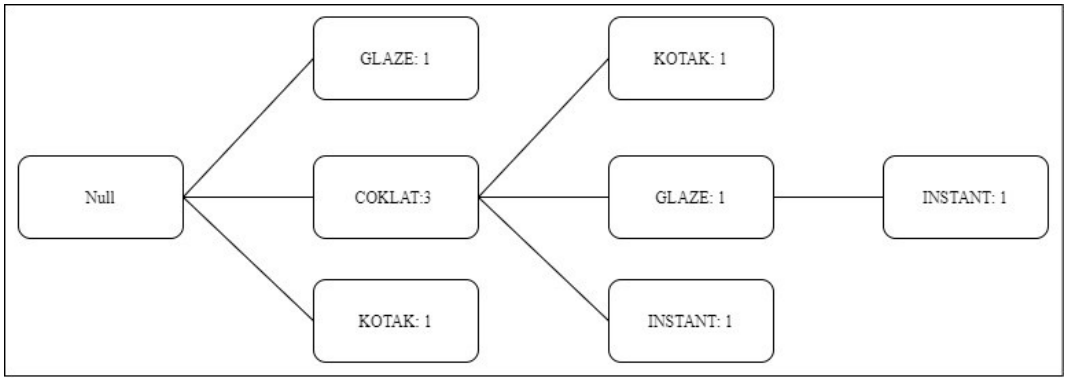
Gambar 2.4 Transaksi 3

Gambar 2.4 merupakan proses penambahan transaksi 3 ke *tree* yang sudah dimasukkan transaksi 1 dan transaksi 2, dimana pada transaksi ketiga ada kategori COKLAT, GLAZE, dan INSTANT. Kategori coklat telah ada tree sebelumnya maka tidak membuat cabang baru, cabang baru terjadi ketika kategori selanjutnya merupakan GLAZE, yang muncul setelah COKLAT maka dari itu membuat cabang baru sehabis COKLAT dan diikuti dengan INSTANT.



Gambar 2.5 Transaksi 4

Gambar 2.5 dimana transaksi 4 dimasukkan ke dalam *tree*, setelah 3 transaksi telah dimasukkan ke dalam tree. Transaksi ke empat sendiri ada kategori COKLAT, dan INSTANT, sehingga adanya penambahan support pada COKLAT, kemudian membuat node baru untuk kategori INSTANT.



Gambar 2.6 Transaksi 5

Gambar 2.6 transaksi 5 dimasukkan ke dalam tree, setelah empat transaksi dimasukkan ke dalam *tree* yang dapat dilihat pada Gambar 2.1 hingga Gambar 2.5. Kategori KOTAK, dengan membuat node baru dari null, dikarenakan transaksi hanya 1 kategori yaitu KOTAK, dan belum pernah muncul pertama kali.

2.4 Website

Menurut Collins English Dictionary (2014), *Website* atau situs *web* adalah suatu halaman web yang saling berhubungan yang umunya berada pada *server* yang berisikan informasi yang disediakan secara perorangan, kelompok, atau organisasi. Website dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa teks, gambar, video, audio, dan animasi lainnya yang tersedia melalui jalur koneksi internet (Abdulloh, 2015). Lebih jelasnya, website merupakan halaman-halaman yang berisi informasi yang ditampilkan oleh *internet browser*, seperti *Microsoft Edg*e, *Google Chrome* atau yang lainnya. Internet adalah jaringan yang digunakan untuk mengirimkan informasinya (Abdulloh, 2015). Sebuah halaman *web* merupakan berkas teks yang berisi aturan – aturan yang dikombinasikan sedemikian rumah yang berbasis HTML, maupun XHTML. Website juga dapat disisipi dengan *scripting* CSS

(*Cascasding Style Sheets*), ataupun javascript, website ditransmisikan melalui protocol HTTP atau HTTPS (Arif, 2017). Terdapat 2 website yang banyak digunakan, website statis dan website dinamis yang memiliki fungsi yang mirip tapi berbeda. Website statis atau kadang yang disebut website tradisional, karena pemrograman website pertama berupa website statis, website statis tidak dapat menjalankan fungsi – fungsi komplek seperti website dinamis dengan kelebihan website statis memiliki kecepatan *load* lebih cepat, dan keamanan yang baik. Website dinamis merupakan sebuah *website* yang memungkinkan penggunanya untuk berinteraksi secara langsung, dimana pengguna dapat menambah, memodifikasi, ataupun menghapus konten di dalam sebuah web tanpa harus membuka struktur kode dari web tersebut. (Esaputra, 2020). web dinamis dibangun dengan *front end* yang akan ditampilkan ke pada user, dan backend untuk mengelola interaksi – interaksi yang dilakukan. Pembangunan website menggunakan library ReactJS yang mana berbasis javascript untuk pembuatan *front end* pada aplikasi *web*, React. React memberikan Kemudahan pembangunan website dengan tanpa memuat ulang halaman, memberikan kecepatan, kesederhanaan, dan skalabilitas (Facebook, 2019). Didukung dengan ExpressJS yang merupakan *framework backend* yang membantu dalam proses *routing, rendering view*, dan mendukung *middleware*, sehingga pengembangan dengan javascript untuk *backend* menjadi lebih cepat. Dukungan penggunanan *MySQL* sebagai database website pun dapat terealisasi, dan pengembangan menjadi lebih cepat menggunakan ExpressJS (Equan, 2019).

2.5 Skala Likert

Menurut Sugiyono (2012), skala likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau kelompok orang tentang fenomena. Skala likert merupakan skala yang dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tetnang suatu gejala atau fenomena. Perhitungan skor pada skala likert dibagi menjadi lima kriteria, Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), dan Sangat Tidak Setuju (STS)

Tabel 2.4 Kriteria Skala Likert (Sugiyono, 2012)

Pernyataan	Skor	Bobot Interval
Sangat Setuju	5	Skor >=80%
Setuju	4	Skor >=60%
Netral/Biasa Saja	3	Skor >=40%
Tidak Setuju	2	Skor >=20%
Sangat Tidak Setuju	1	Skor <20%

Tabel 2.4 merupakan Skala Likert dengan pernyataan Sangat Setuju yang direpresentasikan dengan angka 5 dan yang paling rendah angka 1 merepresentasikan Sangat Tidak Setuju dalam pembentukan skor kuesioner penelitian ini. Persentasi skor pada suatu kuesioner dihitung berdasarkan rumus yang dijelaskan oleh Sugiyono (2012).

Persentase Skor = (((Sangat Setuju *5) + (Setuju *4) + (Netral *3) + (Tidak Setuju *2) + (Sangat Tidak Setuju *1))/(5*Jumlah Responden))*100% …(2. 3)

2.6 EUCS (End *User* Computing Satisfaction)

Tampilan situs web merupakan suatu essensial, dimana citra yang positif dapat terbangun dari sini sehingga kepercayaan konsumen dapat berasal dari sini, suasana menyenangkan bagi *user* mengambil keputusan untuk membeli melalui

daya Tarik emosional ini.(Fitriansyah, 2018). Rumusan suatu sistem berhasil ketika 6 faktor, yaitu kualitas sistem, kualitas informasi, kualitas layanan, penggunaan sistem, kepuasan pengguna, dan manfaat sistem yang memuaskan *user* (Delone Mclean 2016).

Adanya instrumen pengukuran *End User Computing Satisfaction* (EUCS) yang dirumuskan oleh Doll, dan *User Information Satisfaction* (UIS) yang dirumuskan oleh Ives, dapat mengukur kepuasan pengguna. Berdasarkan penilitian Seddon & Yip dibandingkannya kedua metode tersebut didapatkan bahwa metode EUCS lebih baik dibandingkan dengan metode UIS (Fitriansyah, 2018). Variabel – variabel yang menjadi fokus pada metode EUCS yaitu, *content*, *accuracy*, *format*, *ease of use*, dan *timeliness* dapat terlihat pada gambar 2.

Penelitian ini menggunakan *data* primer yang diperoleh secara langsung dari responden penelitian dengan cara menyebar kuesioner, kuesioner bersifat tertutup dimana diberikan alternatif jawaban, diisi oleh pelanggan toko lucky sebagai responden. Sampling yang digunakan dalam penelitian ini adalah probability sampling dan untuk menentukan jumlah sampel digunakan teknik stratified random sampling.

Menurut Arikunto (2010), realibilitas menunjuk pada suatu pengertian bahwa suatu instrumen dapat dipercaya atau diandalkan untuk digunakan sebagian alat pengumpul *data* karena instrument tersebut baik. *Cronbach Alpha* adalah rumus yang digunakan untuk menguji realibilitas dari suatu instrumen.

Nilai realibilitas di atas 0,90 merupakan nilai untuk realibilitas sempurna jika nilai realibilitas di antara 0,70 – 0,90, nilai tersebut menunjukkan reliabilitas tinggi. Nilai reliabilitas di antara 0,50 – 0,70 moderat, dan nilai reliabilitas dibawah 0,50 menunjukkan nilai reliabilitas yang rendah (Ridho,2017).

$$r_{11} = \left[\frac{k}{(k-1)}\right]\left[1 - \frac{\in Si}{st}\right] \quad \ldots(2.4)$$

Keterangan:

r_{11}= Nilai reabilitas

$\in Si$ = Jumlah varians skor setiap pertanyaan

St = Varians total

K = Jumlah pertanyaan