BAB II

LANDASAN TEORI

2.1. *E-commerce*

Secara sederhana, *e-commerce* atau *e-commerce* adalah proses jual-beli dengan menggunakan internet. Namun, pada kenyataannya *e-commerce* tidak terbatas pada transaksi finansial, melainkan juga transaksi non-finansial seperti permintaan informasi atau tanya-jawab (Chaffey, 2011).

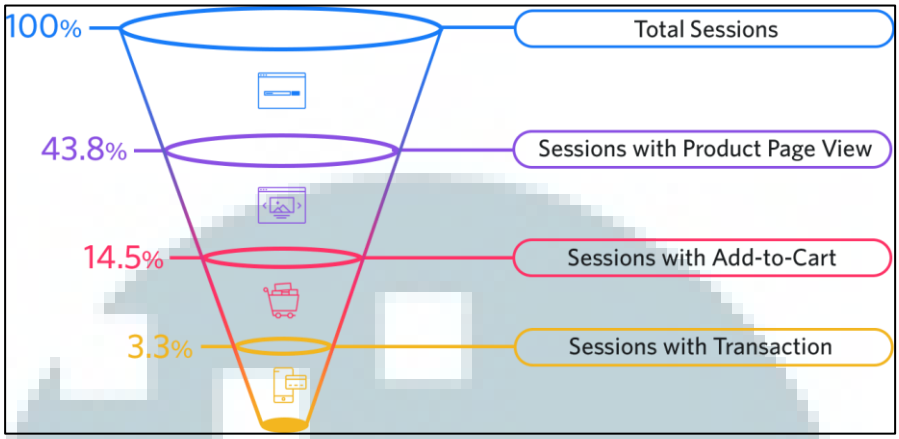
Terdapat tiga tipe utama dalam *e-commerce* yakni *Business-to-Consumer* (B2C), *Business-to-Business* (B2B), dan *Consumer-to-Consumer* (C2C) (Laudon & Traver, 2014). B2C menjual produk dari perusahaan langsung ke pembeli, B2B menjual produk dari perusahaan ke perusahaan, sedangkan C2C menjual produk dari konsumer ke konsumer.

2.2. *E-commerce Conversion Funnel*

E-commerce conversion funnel yang bila diterjemahkan berarti corong konversi pada situs *e-commerce* adalah urutan langkah yang dilalui oleh pengunjung situs sebelum akhirnya membeli produk atau terkonversi (Bulygo, 2018). Corong adalah metafora yang mengilustrasikan penurunan jumlah calon pembeli secara bertahap selama mereka melalui jalur konversi (Markus, 2017). Hal ini dikarenakan tidak semua pengunjung situs akan membeli barang. Bisa jadi dari 100 pengunjung hanya 70% yang melihat halaman produk, 50% yang memasukkan produk ke keranjang, dan 20% yang melakukan pembelian.

E-commerce conversion funnel umumnya dibagi menjadi empat tahapan yang lebih dikenal sebagai model AIDA. Keempat tahapan itu adalah *Awareness, Interest, Desire,* dan *Action* (Charlesworth, 2007). Tujuan dari dibangunnya corong ini adalah agar lebih banyak orang yang dapat melalui setiap tahapannya dan membeli produk. Tahap pertama, *Awareness*, adalah saat calon pembeli mengunjungi situs *e-commerce*. Di tahap ini, pengunjung mengetahui keberadaan situs atau toko dan melihat katalog produk, namun belum tertarik untuk mengetahui lebih lanjut tentang detail produk. Tahap kedua, *Interest,* adalah saat calon pembeli mengunjungi halaman detail produk karena ketertarikannya terhadap suatu produk. Tahap ketiga, *Desire,* adalah saat pengunjung memiliki keinginan untuk membeli produk dari toko tersebut yang biasanya ditandai dengan memasukkan produk ke keranjang belanja. Tahap terakhir, *Action*, adalah saat pembeli membeli barang dari toko tersebut.

Menurut Dafe Chaffey dalam penelitiannya bersama *Smart Insight*, terdapat perbedaan rata-rata persentase konversi antar setiap tahapnya, namun yang pasti adalah jumlah calon pembeli yang memasuki tahap berikutnya berkurang disbanding tahap sebelumnya. Rata-rata persentase konversi dari setiap tahap pada bulan Juni 2018 adalah sebagai berikut (Chaffey D. , 2018):



Gambar 2.1. *Benchmark* Persentase *e-commerce* Pada Setiap Tahapnya

Penjelasan dari gambar di atas adalah 100% atau seluruh pengunjung dihitung sebagai pengguna yang memasuki corong pada bagian *awareness*. Lalu, rata-rata 43,8% dari jumlah pengunjung akan memiliki ketertarikan terhadap produk dan melihat halaman detail atau dengan kata lain memasuki corong tahap *interest*. Demikian seterusnya hingga 3,3% pengunjung membeli produk.

2.3. DFD

Data flow diagram (DFD) atau diagram arus data adalah gambaran grafis suatu sistem dengan menggunakan bentuk-bentuk dan simbol-simbol untuk menggambarkan bagaimana data mengalir melalui suatu proses yang saling berkaitan. DFD adalah cara paling alamiah untuk mendokumentasi proses (McLeod & Schell, 2004). DFD menggambarkan pembagian sistem ke modul yang lebih kecil. Salah satu keuntungan DFD adalah memudahkan pengertian sistem kepada orang awam yang kurang menguasai bidang teknologi sekalipun (Ladjamudin, 2005). DFD dibagi menjadi beberapa tingkatan yakni diagram konteks, diagram nol, dan diagram rinci.

2.2.1. Tingkatan DFD

Penjelasan setiap tingkatan dari DFD menurut Ladjamudin dalam bukunya "Analisis dan Desain Sistem Informasi" adalah sebagai berikut:

1. Diagram konteks

   Diagram konteks adalah diagram level tertinggi yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Pada tingkat ini, terdapat penggambaran seluruh masukkan dan keluaran data di sistem tersebut. Diagram konteks hanya terdiri dari satu proses dan tidak boleh ada penyimpanan data.

2. Diagram nol

   Di tingkat ini, dilakukan penggambaran menyeluruh mengenai sistem yang akan ditangani yakni mencakup fungsi-fungsi utama atau proses yang ada, aliran data, dan entitas luar. Pada level ini sudah dimungkinkan adanya penyimpanan data.

3. Diagram rinci

   Diagram rinci menggambarkan proses yang terjadi dalam setiap fungsi yang terdapat di diagram nol.

2.2.2. *Ballancing* DFD

Hal yang perlu diperhatikan untuk menjaga *balancing* dalam DFD:

1. Harus terdapat keseimbangan input dan output antara satu level dan level berikutnya. Aliran data yang masuk ke dalam dan keluar dari suatu proses harus sama dengan aliran data yang

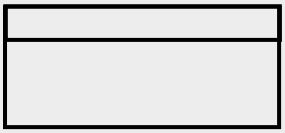
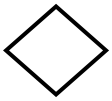
masuk ke dalam dan keluar dari rincian proses pada level di bawahnya.

2. Nama aliran data, penyimpanan data, dan entitas luar pada tiap level harus sama, jika objeknya sama.

2.2.3. Simbol DFD

Simbol DFD yang dipakai dalam penelitian kali ini adalah versi Yourdon dan DeMarco. Simbol komponennya antara lain yaitu entitas luar, aliran data, proses, penyimpanan data, *split/merge* dan *off-page connector*. Simbol dalam pemodelan fungsional dapat dilihat pada tabel 2.1.

Tabel 2.1. Simbol dalam DFD

Notasi	Keterangan
	Entitas eksternal
	Aliran data
	Proses
	Penyimpanan data
	*Split / Merge*
	*Off-page connector*

Penjelasan dari masing-masing elemen tersebut adalah sebagai berikut:

1. Entitas Luar

   Sesuatu yang berada di luar sistem namun dapat memberikan atau menerima data dari sistem. Disimbolkan dengan segi empat.

2. Aliran data

   Tempat mengalirnya informasi, dapat berupa masukan bagi sistem atau keluaran dari sistem. Digambarkan dengan panah di antara proses atau penyimpanan data.

3. Proses

   Merupakan hal yang dikerjakan oleh sistem, berfungsi untuk mengubah masukan menjadi data keluaran sesuai dengan spesifikasi yang diinginkan.

4. Penyimpanan data

   Tempat penyimpanan data dalam sistem, disimbolkan dengan sepasang garis sejajar.

5. *Split/Merge*

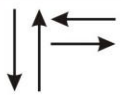
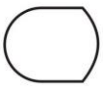
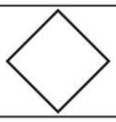
   Split berfungsi untuk memecah satu aliran data menjadi beberapa macam aliran data. Sedangkan *Merge* berfungsi untuk menggabungkan beberapa macam aliran data menjadi satu buah aliran data.

6. *Off-page connector*

   Komponen *off-page connector* digunakan untuk menghubungkan suatu aliran data yang di *split / merge* dengan proses yang berada satu tingkat di atasnya.

2.4. *Flowchart*

Flowchart adalah bagan-bagan dengan arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma (Ladjamudin, 2005). Simbol-simbol yang terdapat dalam *flowchart* dapat dilihat dalam gambar berikut:

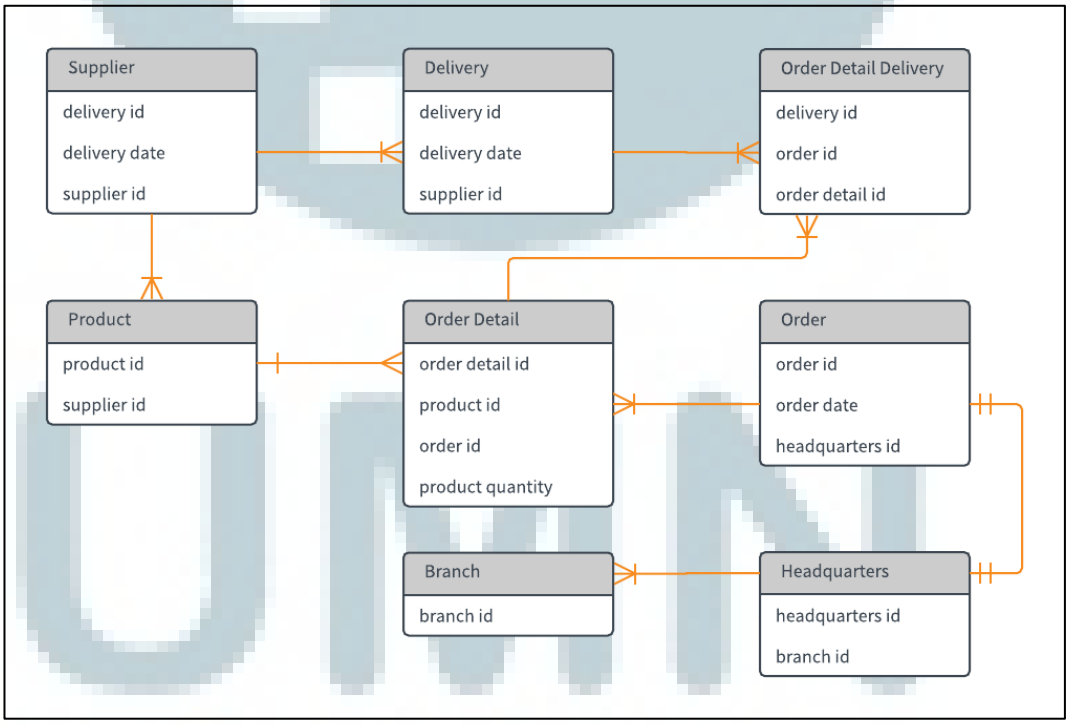
	**Flow Direction symbol** Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		**Simbol Manual Input** Simbol untuk pemasukan data secara manual on-line keyboard
	**Terminator Symbol** Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		**Simbol Preparation** Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	**Connector Symbol** Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		**Simbol Predefine Proses** Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	**Connector Symbol** Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		**Simbol Display** Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	**Processing Symbol** Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		**Simbol disk and On-line Storage** Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	**Simbol Manual Operation** Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		**Simbol magnetik tape Unit** Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	**Simbol Decision** Simbol pemilihan proses berdasarkan kondisi yang ada.		**Simbol Punch Card** Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	**Simbol Input-Output** Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		**Simbol Dokumen** Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Gambar 2.2. Simbol dalam *Flowchart*

2.5. ERD

Entity Relationship Diagram (ERD) merupakan sebuah teknik untuk menggambarkan struktur logis dari sebuah basis data dalam sebuah cara piktorial. ERD mendokumentasikan data dalam sistem dengan mengidentifikasi jenis entitas dan hubungannya (McLeod & Schell, 2004). Dengan demikian, ERD memberikan arti yang dapat dimengerti untuk mengkomunikasikan bahasan yang menonjol dari desain suatu basis data yang diberikan (Date, 2000).

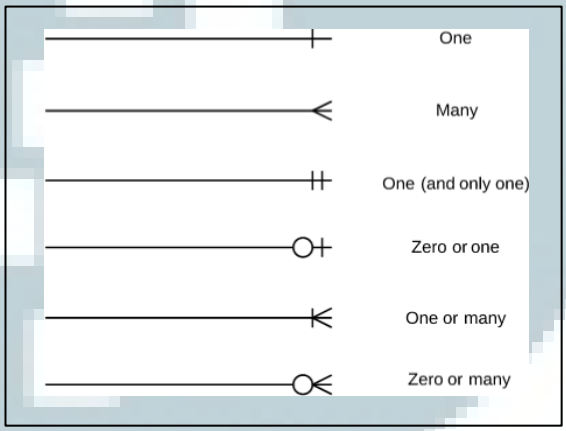
Terdapat beberapa cara untuk menggambarkan ERD seperti *Chen Notation, Crow's Foot style, Bachman Style, IDEF1X style,* dan *Barker Style*. Dalam laporan ini, notasi yang akan digunakan akan mengacu kepada *Crow's Foot Style*.



Gambar 2.3. Contoh ERD berdasar *Crow's Foot Style*

2.4.1. Kardinalitas ERD

Kardinalitas menunjukkan jumlah maksimum himpunan entitas yang dapat berelasi dengan himpunan entitas pada himpunan entitas yang lain, kardinalitas relasi merujuk kepada hubungan maksimal yang terjadi dari himpunan entitas yang lain dan begitu juga sebaliknya. Kardinalitas relasi yang terjadi diantara dua himpunan entitas dapat berupa satu ke satu, satu ke banyak, dan banyak ke banyak (Fathansyah, 2007).



Gambar 2.4. Notasi kardinalitas dalam ERD

2.6. RAD

RAD atau *Rapid Apllication Development* adalah metodologi pengembangan perangkat lunak yang mengabungkan antara model prototipe dan model iteratif. RAD adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik *incremental* atau bertingkat (SDLC RAD Model, 2017). RAD hampir sama dengan model *waterfall*, bedanya siklus pengembangan yang ditempuh model ini tergolong pendek (Association, 2014). Tahap-tahap dalam RAD antara lain (E.Perry, 2006):

1. Perencanaan kebutuhan

   Pada tahap ini, dilakukan identifikasi kebutuhan sistem yang akan dibuat dengan cara berdiskusi dengan klien (pemilik usaha) dan melakukan *case-based reasoning* (CBR) dengan membandingkan fitur pada situs B2C produk mode untuk membantu merumuskan kebutuhan fitur (Maher., Balachandran, & Zhang, 2014). Data-data yang diperoleh kemudian akan diseleksi untuk menentukan fitur mana saja yang akan diwujudkan dari alternatif-alternatif yang ada dan membuat batasan sistem. Hasil yang didapatkan dari tahapan ini adalah rumusan kebutuhan sistem.

2. Perancangan

   Perancangan sistem mempunyai dua tujuan utama yakni memberikan gambaran umum kebutuhan informasi kepada pemakai dan memberi gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya. Perancangan sistem yang akan dibangun dilakukan baik itu dari sisi desain layout atau tampilan (nilai artistik & estetika nya) dan dari sisi teknis.

3. Konstruksi dan pengujian

   Pada tahap ini terjadi proses pembuatan situs dan pengetesan. Situs yang telah selesai dibuat akan melalui tahap pengetesan yakni memastikan fungsi-fungsi berjalan dengan baik, data yang tampil sesuai dengan yang diharapkan, serta pemrosesan yang terjadi dalam progam berjalan dengan lancar.

4. Operasi dan Perawatan Sistem

   Setelah sistem diimplementasi dengan berhasil, sistem akan dioperasikan dan dirawat. Sistem perlu dirawat karena beberapa hal yaitu sistem mengandung kesalahan yang perlu diperbaiki, sistem mengalami perubahan karena permintaan baru dari pemakai sistem, atau sistem mengalami perubahan karena perubahan lingkungan luar.

2.7. *Black-box Testing*

Black-box testing adalah testing yang berfokus pada spesifikasi program yang dibuat tanpa pengetahuan tentang struktur internal program (Srinivassan Desikan, 2006). Pengujian *black-box* berusaha untuk menemukan kesalahan dalam kategori:

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan antar muka
3. Kesalahan dalam akses basis data
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

Terdapat beberapa metode pengujian yang terdapat dalam *black-box testing* seperti *equivalence partitioning, boundary value analysis / limit testing, comparison testing, sample testing, robustness testing, behaviour testing, requirement testing, performance testing, endurance testing,* dan *cause-effect relationship testing.*

Pada penelitian kali ini, metode yang akan digunakan adalah *equivalence partitioning*, *boundary value analysis* dan *robustness testing.* Berikut penjelasan dari metode *black-box* yang dipilih (Sukamto, 2017):

1. *Equivalence Partitioning*

   Membagi *input* menjadi kelas-kelas data yang dapat digunakan untuk membuat kasus uji. *Input* dibagi menjadi kondisi yang valid atau tidak, dapat berupa nilai numerik, rentang nilai, kumpulan nilai yang berkaitan, atau *boolean*. Contoh, pada program *input* nilai, rentang yang valid adalah 0 – 100. Kelas yang terbentuk adalah dua kelas yang tidak valid (< 0 dan > 100) dan satu kelas valid (0 – 100).

2. *Boundary Value Analysis*

   Boundary Value Analysis atau BVA adalah komplemen dari *equivalence partitioning.* Pada BVA dilakukan pemilihan elemen dalam kelas *equivalance* agar kasus uji yang dijalankan tidak terlalu banyak. Jika mengambil contoh sebelumnya, maka uji kasus yang dapat diambil dari setiap kelas adalah -1, 0, 1, 99, 100, 101 (a-1, a, a+1, b-1, b, b+1).

3. *Robustness Testing*

   Data masukan dipilih dari luar spesifikasi yang telah didefinisikan. Tujuan dari pengujuan ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid. Misalnya, memasukkan teks atau karakter pada program *input* nilai.

2.8. *User Acceptance Test*

User Acceptance Testing atau UAT merupakan proses verifikasi bahwa solusi yang dibuat dalam sistem sudah sesuai untuk pengguna. Proses ini berbeda dengan pengujian sistem, UAT memastikan bahwa solusi dalam sistem tersebut akan bekerja untuk pengguna sesuai dengan persyaratan yang harus dipenuhi (Cimperman, 2006). Evaluasi UAT diimplementasikan dalam daftar pernyataan yang dapat dipilih kesesuaiannya dengan pengalaman responden menggunakan skala Likert. Skala Likert memudahkan responden untuk menjawab setiap pertanyaan berdasarkan skala respon yang terukur (Punch, 2009). Akan digunakan skala 1 – 4 pada setiap pernyataan dengan ketentuan sebagai berikut:

1 = Sangat tidak setuju 3 = Setuju

2 = Tidak setuju 4 = Sangat Setuju

2.9. Skala Likert

Skala Likert merupakan suatu skala psikometrik yang umum digunakan dalam kuesioner maupun survei (Russell & Cohn, 2012). Responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Berikut urutan proses pencarian skor dengan skala Likert (Riduwan, 2009):

Nilai Indeks Maksimal = Jumlah Responden x Nilai Tertinggi x Jumlah Soal

Nilai Indeks Minimum = Jumlah Responden x Nilai Terendah x Jumlah Soal

Nilai Indeks = ((F1 x 1) + (F2 x 2) + … + (Fn x n))

Keterangan:

a. n adalah skala Likert
b. Fn adalah frekuensi jawaban responden yang memilih skala n

Digunakan metode tiga kotak (*three box method*) untuk menentukan pengelompokan hasil. Cara mengetahui rentang pada pengelompokan tiga kotak adalah dengan membagi rentang antara nilai indeks maksimal dan nilai indeks minimum dengan tiga. Misal nilai indeks maksimal adalah 96 dan nilai indeks minimum adalah 24, maka rentang yang didapat adalah 72 (96 – 24). Kemudian, rentang ini dibagi dengan 3 sehingga menghasilkan 24 (72/3). Lalu, hasil pengelompokan didapat dari menambah nilai minimum dengan rentang hingga mencapai nilai maksimum. Contoh hasil metode tiga kotak:

24 – 48 = Rendah

49 – 72 = Sedang

73 – 96 = Tinggi

Terakhir, dari nilai indeks yang telah dicari sebelumnya, dilakukan pengecekan angka indeks tersebut masuk ke dalam pengelompokan yang mana.

2.10. *Welch Two Sample T-Test*

T-Test dilakukan untuk menentukan apakah rata-rata antara kedua kelompok sama (Stephanie, 2018). Sebuah *t-test* mengajukan pertanyaan, "Apakah perbedaan antara cara dua sampel cukup signifikan untuk mengatakan bahwa beberapa karakteristik lain dapat menyebabkannya?". Dalam t-test ini, hipotesis nol yang

diajukan adalah bahwa tidak ada perbedaan yang signifikan antara kedua kelompok. Setelah itu, data akan diproses untuk menemukan varians antara dua sampel. *T-test* yang digunakan pada penelitian ini adalah *Welch Two Sample t-test* atau *t-test* tidak setara. Rumus dari *Welch Two Sample t-test* adalah sebagai berikut:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \quad (1)$$

Di mana $\bar{X}_1, s_1^2$ dan N_1 secara berurutan adalah sampel sampel pertama, varians sampel dan ukuran sampel. Dalam perhitungan, var.equal ditetapkan sebagai *false* yang artinya penghitungan tidak akan mengasumsikan bahwa kedua kumpulan data memiliki varian yang sama (standar deviasi). Dalam *t-test* ini, hipotesis nol yang diajukan adalah bahwa tidak ada perbedaan yang signifikan antara kedua kelompok.