

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Digital Marketing**

Digital Marketing adalah kegiatan pemasaran termasuk *branding* yang menggunakan berbagai media berbasis web salah satunya yaitu sosial media[6]. Digital Marketing memiliki beberapa karakteristik seperti:

1. Upaya meningkatkan interaktivitas bisnis dengan pelanggan yang bergantung pada teknologi.
2. Sebuah dialog elektronik (teknologi interaktif) untuk memberikan akses informasi kepada pelanggan (komunitas, individu), dan sebaliknya.
3. Upaya melakukan semua kegiatan bisnis melalui internet untuk tujuan penelitian, analisis dan perencanaan untuk menemukan, menarik, dan mempertahankan pelanggan.
4. Upaya meningkatkan akselerasi jual beli barang dan jasa (tertentu), informasi dan ide melalui internet.

#### **2.2 Buzz Marketing**

*Buzz Marketing* adalah kegiatan pemasaran yang memanfaatkan para *buzzer* untuk terus membicarakan suatu produk agar bisa menjadi viral atau trending di media social[4]. Ada tiga hal penting pada *Buzz Marketing*[7], yaitu:

##### 1. *Noise*

Konsumen saat ini sulit menentukan pilihannya karena banyak iklan yang dilihat setiap harinya. Konsumen menjadi bingung untuk menentukan produk yang diinginkannya, sehingga mereka lebih tertarik untuk mendengarkan rekomendasi produk dari orang lain dan juga sekelompok teman.

##### 2. *Skepticism*

Konsumen pada umumnya ragu terhadap kebenaran informasi yang diterimanya. Hal ini terjadi karena konsumen pernah mengalami suatu

kekecewaan pada produk. Lalu konsumen tersebut mencoba berpaling ke produk lain dengan cara mencari informasi kepada sekelompok teman tentang produk yang direkomendasikan sesuai yang mereka butuhkan.

### 3. *Connectivity*

Konsumen selalu melakukan interaksi serta berkomunikasi satu sama lain hampir setiap hari dan akhirnya berkomentar tentang suatu sama lain pengalaman mereka terhadap penggunaan produk atau jasa.

## 2.3 *Buzzer Twitter*

*Buzzer Twitter* adalah pengguna akun *Twitter* dengan ratusan bahkan sampai jutaan pengikut, sehingga melakukan kegiatan penyebaran informasi yang lebih luas. *Buzzer* secara harfiah berarti perangkat yang mengeluarkan suara keras untuk menarik perhatian. Analogi *Buzzer Twitter* mengacu pada akun yang berdampak besar, aktif *men-tweet* konten sehingga tweet mereka sering di-*retweet*, dan aktif berinteraksi dengan orang lain di media sosial *Twitter*[8]. *Buzzer* adalah akun media sosial nyata atau anonim yang membagikan konten kepada publik dari media sosial dengan tujuan untuk membuat topik pada konten yang dibagikan menjadi viral.

Berikut adalah indikator *Buzzer* yang dirangkum dari beberapa sumber[8][9][10][11] yaitu:

#### 1. Populer

- Jumlah pengikut yang banyak (umumnya memiliki 3000 pengikut)
- Memiliki jaringan yang luas

#### 2. Aktif

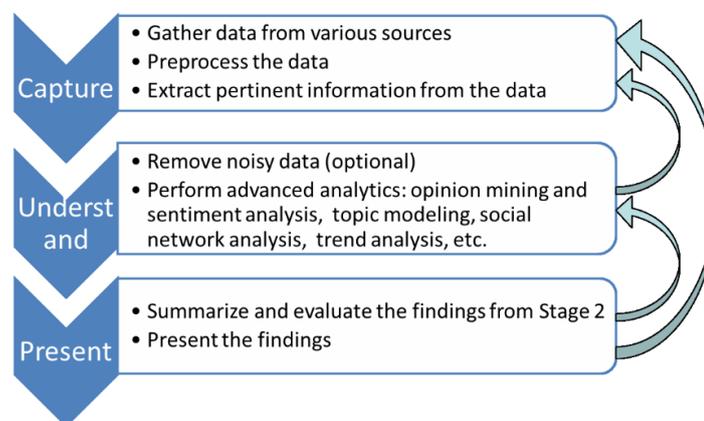
- Jumlah interaksi yang besar (*retweet, reply, like, quote*)
- Ikut memberikan, memperluas, dan mempromosikan suatu informasi.

#### 3. Kreatif

- Kemampuan produksi konten seperti pengetahuan jurnalistik dan pembedaan atau pemilihan informasi.

## 2.4 Social Media Analytics Process

*Social Media Analytics* adalah monitoring, analisis, mengukur, dan menafsirkan interaksi digital dan hubungan orang, topik, ide dan konten lainnya yang terjadi melalui media sosial[12][13]. *Social Media Analytics* berfokus kepada pengembangan dan alat evaluasi informatika dan kerangka kerja untuk mengumpulkan, memantau, menganalisis, meringkas, dan memvisualisasikan di data media sosial dengan tujuan untuk memfasilitasi percakapan dan interaksi untuk mengekstrak pola dan kecerdasan yang berguna[5]. Pada Social Media Analytics memiliki tiga tahapan seperti pada Gambar 2.1 Social Media Analytics Process.



Gambar 2.1 Social Media Analytics Process

Tahapan pada Social Media Analytics :

### 1. *Capture*

Tahapan ini adalah tahapan pengumpulan data dan praprosesan data sehingga data siap untuk diolah pada langkah selanjutnya. Pengumpulan data yang dilakukan adalah proses data hasil *crawling* pada media sosial. Praprosesan data dilakukan untuk menyiapkan data sehingga siap dipakai untuk tahapan selanjutnya, tahapan pada praprosesan data antara lain seperti melakukan seleksi data berdasarkan kebutuhan data yang ingin digunakan, tahapan ini bertujuan agar data yang digunakan bersih dari data yang tidak diperlukan dan membuat data menjadi lebih terstruktur.

### 2. *Understand*

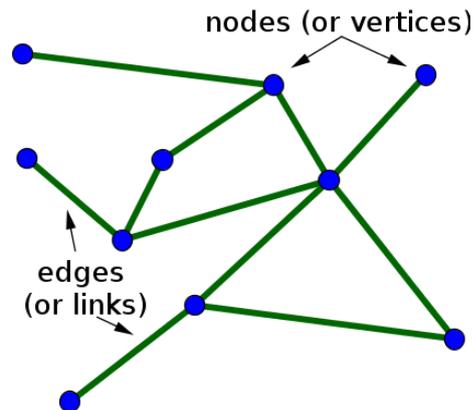
Pada tahapan ini akan dilakukan penerapan model analisis terhadap data yang sebelumnya telah disiapkan dari hasil praprosesan data. Pada model analisis diantaranya seperti *sentiment analysis*, *topic modelling*, *social network analysis*, *trend analysis*, dan lain - lain.

### 3. *Present*

Pada tahapan *Present* adalah tahap hasil dari pemodelan pada tahap sebelumnya dengan menyimpulkan dan mengevaluasi informasi dari hasil analisa yang dilakukan, penyampaian informasi dapat berupa visualisasi agar pengetahuan yang disampaikan dapat lebih dimengerti oleh orang lain.

## 2.5 *Social Network Analysis*

*Social Network Analysis* adalah ilmu yang mempelajari hubungan manusia dengan menggunakan teori graf. *SNA* berfokus pada pola hubungan antara aktor serta menggambarkan hubungan jaringan tersebut[14]. Individu atau orang digambarkan sebagai nodes atau titik, sedangkan edges atau links menggambarkan relasi yang terjadi antar individu Implementasi *Social Network Analysis* mampu menggambarkan relasi atau hubungan antar aktor melalui visualisasi berbentuk graf. Relasi dalam *Social Network Analysis* dapat diolah dalam bentuk kalkulasi yang menghasilkan *centrality* dalam sebuah jaringan sosial sesuai dengan posisi masing-masing aktor di dalam struktur jaringan tersebut. Berikut adalah contoh visualisasi graph sederhana pada *Social Network Analysis* pada Gambar 2.2 Visualisasi Graph sederhana pada Social Network Analysis.



Gambar 2.2 Visualisasi Graph sederhana pada Social Network Analysis.

### 2.4.1 Centrality

*Centrality* merupakan ukuran atau metrik yang digunakan *didalam Social Network Analysis*. *Centrality* dalam *Social Network Analysis* ada empat jenis yaitu *degree centrality*, *betweenness centrality*, *closeness centrality*, *eigenvector centrality*, dan *page rank centrality*.

#### 1. Degree Centrality

*Degree Centrality* merupakan ukuran centrality yang paling mudah untuk dihitung. *Degree Centrality* berguna dalam menilai *node* yang sentral untuk menyebarkan informasi dan mempengaruhi orang lain dalam lingkungan mereka[14]. *Degree Centrality* dihitung dengan rumus dibawah berikut.

$$CD(n_i) = d(n_i) \quad (2.1)$$

dan untuk memperoleh nilai yang telah dinormalisasikan akan menjadi

$$CD(n_i) = \frac{d(n_i)}{n - 1} \quad (2.2)$$

Keterangan :

$d(n_i)$  merupakan links atau ties yang ada dalam suatu jaringan.

#### 2. Betweenness Centrality

*Betweenness Centrality* didefinisikan sebagai ukuran tingkat kepentingan seseorang berdasarkan kemampuannya dalam menghubungkan banyak

orang lain didalam jejaring sosialnya. Perhitungan *Betweenness Centrality* dapat dihitung dengan rumus berikut.

$$CB(ni) = \sum \frac{\sigma(u, v|ni)}{\sigma(u, v)} \quad (2.3)$$

Keterangan :

$\sigma(u, v|ni)$  merupakan banyaknya jalur terpendek dari titik u ke titik v yang melewati titik ni.

$\sigma(u, v)$  mewakili jumlah jalur terpendek antara titik u dan titik v.

### 3. *Closeness Centrality*

*Closeness Centrality* merupakan ukuran tingkat kepentingan seseorang berdasarkan jarak rata – rata menuju seluruh orang lainnya didalam jejaring sosial. *Closeness Centrality* dapat dihitung dengan menggunakan rumus.

$$CC(ni) = \frac{n - 1}{\sum d(ni, nj)} \quad (2.4)$$

$$CC(ni) = \frac{n - 1}{N - 1} \frac{n - 1}{\sum d(ni, nj)} \quad (2.4)$$

Keterangan :

N adalah banyaknya titik yang ada.

Sedangkan,  $d(ni, nj)$  mewakili banyaknya jalur terpendek antara titik ni dan nj.

### 4. *Eigenvector Centrality*

*Eigenvector Centrality* adalah ukuran tingkat kepentingan seseorang berdasarkan akumulasi tingkat kepentingan orang – orang yang berada disekitarnya.

$$Ax = \lambda x \quad (2.5)$$

Keterangan :

A adalah matriks ketetanggaan dari graf G dengan nilai eigen  $\lambda$  . Berdasarkan teorema Perron-Frobenius, ada solusi unik , yang semua entrinya positif, jika  $\lambda$  adalah nilai eigen terbesar dari matriks ketetanggaan A

### 5. Page Rank

*Page Rank Centrality* merupakan suatu pengukuran yang bertolak belakang terhadap peran seorang node jika dibandingkan pengukuran menggunakan *Degree Centrality*. Pada Pagerank seorang aktor memiliki kedudukan tinggi jika aktor tersebut memiliki ikatan sebagai penerima (in-links) paling banyak didalam jaringan. Perbedaan utama konsep *Degree Centrality* dan Pagerank adalah Degree Centrality berfokus pada ikatan yang keluar (out-links), sedangkan Pagerank fokus pada ikatan yang masuk (in-links)[15].

$$P_d(i) = \frac{d_i(i)}{n - 1} \quad (2.7)$$

Keterangan :

$P_D(i)$  adalah Page Rank.

$d(i)$  adalah jumlah ikatan masuk dari I.

$n$  adalah jumlah seluruh anggota populasi node dalam jaringan.

## 2.6 Twitter

*Twitter* adalah situs web yang dimiliki dan dioperasikan oleh Twitter Inc., yang menyediakan jejaring sosial *microblogging* di mana pengguna dapat

mengirim dan membaca tweet. *Microblog* adalah sejenis alat komunikasi online yang memungkinkan pengguna memperbarui status pemikiran dan tindakan orang tentang topik atau fenomena tertentu. *Tweet* ditampilkan di halaman profil pengguna sebagai teks hingga 140 karakter. *Tweet* dapat dilihat secara publik, tetapi pengirim dapat membatasi pesan hanya ke daftar teman mereka. Pengguna dapat melihat *tweet* dari pengguna lain yang disebut pengikut.

## 2.7 *Twitter API*

*Application Programming Interface (API)* adalah fitur yang dimaksudkan untuk menggantikan bahasa yang digunakan untuk panggilan sistem dengan bahasa yang lebih terstruktur dan dapat dipahami oleh programmer. *Twitter* menyediakan *API (Application Programming Interface)* sehingga memudahkan setiap orang untuk mengambil data dari *Twitter*[16]. Fungsi yang dibuat dengan *API* membuat panggilan sistem bergantung pada sistem operasi. Mungkin nama system callnya sama dengan nama *API*-nya. *Search Tweet API* dirancang untuk memudahkan pengguna mengelola kueri pencarian di konten *Twitter*. Ini memungkinkan pengguna untuk mencari *tweet* berdasarkan kata kunci tertentu atau *tweet* yang lebih spesifik berdasarkan nama pengguna *Twitter* mereka. *API* pencarian juga menyediakan akses ke data tentang topik populer.

## 2.8 *Tweepy Crawler*

*Tweepy Crawler* adalah modul *library* dalam bahasa pemrograman *Python* yang melakukan tugas mengekstrak data dari *Twitter* melalui *Twitter API*[17]. *Crawler* adalah mesin atau sistem yang bertanggung jawab untuk ekstraksi data. Sederhananya, *crawler* adalah proses mengunjungi dan membaca web, mengambil informasi, mengindeks setiap kata dalam dokumen, dan menambahkannya ke database menggunakan sistem yang telah ditentukan.

## 2.9 *Application Programming Interface (API)*

*Application Programming Interface* atau *API* merupakan integrasi dari dua bagian sistem aplikasi. *Application programming interface (API)* merupakan suatu dokumentasi yang terdiri dari interface, fungsi, kelas, struktur dan sebagainya untuk membangun perangkat lunak[18]. Dengan adanya *API* ini, maka memudahkan programmer untuk membongkar suatu *software*, kemudian dikembangkan atau

diintegrasikan dengan perangkat lunak yang lain. *API* dapat dikatakan sebagai penghubung dari suatu aplikasi dengan aplikasi lain yang memungkinkan programmer menggunakan sistem fungsi. Proses ini dikelola melalui sistem operasi. Keunggulan dari *API* ini adalah memungkinkan suatu aplikasi dengan aplikasi lain dapat saling terhubung dan juga berinteraksi langsung[19]. . Metode-metode interaksi menggunakan dasar *HTTP* seperti pada Tabel 2.1 *Request Method API*

Tabel 2.1 *Request Method API*

No.	<i>Request Method</i>	Keterangan
1	<i>GET</i>	Mendapatkan ( <i>read</i> ) sebuah sumber daya ( <i>resource</i> ) yang diidentifikasi dengan <i>URI (Uniform Resource Identifier)</i>
2	<i>POST</i>	Mengirimkan sumber daya ( <i>resource</i> ) ke server. Digunakan untuk membuat ( <i>create</i> ) sumber daya baru
3	<i>DELETE</i>	Mengirimkan sumber daya ( <i>resource</i> ) ke server. Digunakan untuk memasukkan ( <i>insert</i> ) atau memperbarui ( <i>update</i> ) sumber daya yang tersimpan.
4	<i>PUT</i>	Mengirimkan sumber daya ( <i>resource</i> ) ke server. Digunakan untuk memasukkan ( <i>insert</i> ) atau memperbarui ( <i>update</i> ) sumber daya yang tersimpan

### 2.10 Regular Expression

*Regular Expression* atau *Regex*, adalah ekspresi untuk menemukan pola kalimat atau string. Pada level rendah, *Regular Expression* dapat menemukan fragmen kata. Pada tingkat tinggi, *Regular Expression* dapat memanipulasi data dengan mengambil, menghapus, dan memodifikasi data *string*[20]. *Regular Expression* adalah string teks yang secara khusus dikodekan sebagai kumpulan string yang cocok dengan suatu pola. *Regular Expression* dapat diartikan sebagai string yang berisi kombinasi beberapa metakarakter umum dan khusus atau *metasequences*. *Regex* memiliki dua fungsi utama yakni mencari dan mengganti, mencari suatu pola tertentu dalam teks lalu menggantinya menjadi pola yang lain.

*Regex* mempunyai modul didalam bahasa pemrograman Python[21]. Pada python memiliki fungsi, metakarakter, urutan khusus, dan pengaturan seperti pada Tabel 2.2 Fungsi *Regex*, Tabel 2.3 Metakarakter *Regex*, Tabel 2.4 Aturan Khusus *Regex*, dan Tabel 2.5 Set atau Pengatur *Regex*.

Tabel 2.2 Fungsi *Regex*

No	Fungsi	Deskripsi
1	findall	Mengembalikan daftar yang berisi semua kecocokan.
2	search	Mengembalikan objek Pertandingan jika ada kecocokan di mana saja dalam string.
3	split	Mengembalikan daftar di mana string telah dibagi di setiap pertandingan.
4	sub	Mengganti satu atau banyak kecocokan dengan string.

Tabel 2.3 Metakarakter *regex*

No.	Karakter	Deskripsi	Contoh
1	[]	Satu set karakter.	"[a-m]"
2	\	Memberi sinyal urutan khusus (juga dapat digunakan untuk menghindari karakter khusus).	"\d"
3	.	Semua karakter (kecuali karakter baris baru).	"he.o"
4	^	Mulai dengan	"^hello"
5	\$	Diakhiri dengan	"planet\$"
6	*	Nol atau lebih kejadian.	"he.*o"
7	+	Satu atau lebih kejadian.	"he.+o"
8	?	Satu atau nol kejadian.	"he.?o"
9	{}	Persis jumlah kejadian yang ditentukan.	"he.{2}o"
10		Entah atau	"falls stays"
11	()	Tangkap dan kelompokkan	

Tabel 2.4 Aturan Khusus *regex*

No	Karakter	Deskripsi	Contoh
1	\A	Mengembalikan kecocokan jika karakter yang ditentukan berada di awal string	"\AThe"
2	\b	Mengembalikan kecocokan di mana karakter yang ditentukan berada di awal atau di akhir kata	r"\bain" r"ain\b"
3	\B	Mengembalikan kecocokan di mana karakter yang ditentukan ada, tetapi BUKAN di awal (atau di akhir) kata	r"\Bain" r"ain\B"
4	\d	Mengembalikan kecocokan di mana string berisi angka (angka dari 0-9)	"\d"

5	\D	Mengembalikan kecocokan di mana string tidak mengandung angka	"\D"
6	\s	Mengembalikan kecocokan di mana string berisi karakter spasi putih	"\s"
7	\S	Mengembalikan kecocokan di mana string TIDAK mengandung karakter spasi putih	"\S"
8	\w	Mengembalikan kecocokan di mana string berisi karakter kata apa pun (karakter dari a hingga Z, angka dari 0-9, dan garis bawah _ karakter)	"\w"
9	\W	Mengembalikan kecocokan di mana string TIDAK mengandung karakter kata apa pun	"\W"
10	\Z	Mengembalikan kecocokan jika karakter yang ditentukan berada di akhir string	"Spain\Z"

Tabel 2.5 Set atau pengatur *regex*

No	Pengatur	Deskripsi
1	[arn]	Mengembalikan kecocokan di mana salah satu karakter yang ditentukan (a, r, atau n) ada
2	[a-n]	Mengembalikan kecocokan untuk setiap karakter huruf kecil, menurut abjad antara a dan n
3	[^arn]	Mengembalikan kecocokan untuk karakter apa pun kecuali a, r, dan n
4	[0123]	Mengembalikan kecocokan di mana salah satu digit yang ditentukan (0, 1, 2, atau 3) ada
5	[0-9]	Mengembalikan kecocokan untuk setiap digit antara 0 dan 9
6	[0-5][0-9]	Mengembalikan kecocokan untuk angka dua digit dari 00 dan 59

7	[a-zA-Z]	Mengembalikan kecocokan untuk karakter apa pun menurut abjad antara a dan z, huruf kecil ATAU huruf besar
8	[+]	Dalam set, +, *, .,  , (), \$, {} tidak memiliki arti khusus, jadi [+] berarti: mengembalikan kecocokan untuk setiap karakter + dalam string

### 2.11 Normalisasi *Min-Max Scaller*

Min-max scaller normalization adalah metode normalisasi dengan melakukan transformasi linear terhadap data sehingga menghasilkan keseimbangan nilai pada data[22]. Metode ini dapat dirumuskan seperti pada rumus dibawah ini.

$$Z = \frac{X - \min}{\max - \min} \quad (2.8)$$

Dimana:

X adalah nilai datanya.

min adalah nilai terkecil dari atribut tersebut.

max adalah nilai terbesar dari atribut tersebut.

