

**ANALISIS FAKTOR KEPUASAN DAN  
KETIDAKPUASAN PELANGGAN HOTEL PADA  
ULASAN *ONLINE* MENGGUNAKAN *TEXT MINING***

**Tugas Akhir**

**Diajukan untuk Memenuhi Salah Satu Persyaratan Mencapai Derajat  
Sarjana Komputer**



Dibuat Oleh:

**BRENDA EUGENIA LENAK**

**16 07 08842**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ATMA JAYA YOGYAKARTA  
2020**

## HALAMAN PENGESAHAN

Tugas Akhir Berjudul

**ANALISIS FAKTOR KEPUASAN DAN KETIDAKPUASAN PELANGGAN HOTEL PADA ULASAN  
ONLINE MENGGUNAKAN TEXT MINING**

yang disusun oleh

**BRENDA EUGENIA LENAK**

160708842

dinyatakan telah memenuhi syarat pada tanggal 03 Juli 2020

		Keterangan
Dosen Pembimbing 1	: Prof. Ir. A. Djoko Budiyanto, M.Eng., Ph.D.	Telah menyetujui
Dosen Pembimbing 2	: Iko Devi Sulstyaningtyas, S.Sos., M. Si.	Telah menyetujui
Tim Penguji		
Penguji 1	: Prof. Ir. A. Djoko Budiyanto, M.Eng., Ph.D.	Telah menyetujui
Penguji 2	: Dr. Pranowo, S.T., M.T.	Telah menyetujui
Penguji 3	: Yulus Harjoseputro, ST., MT.	Telah menyetujui

Yogyakarta, 03 Juli 2020

Universitas Alma Jaya Yogyakarta

Fakultas Teknologi Industri

Dekan

td

Dr. A. Teguh Siswanto, M.Sc

# PERNYATAAN ORISINALITAS & PUBLIKASI ILMIAH

Saya yang bertanda tangan di bawah ini:

Nama Lengkap : Brenda Eugenia Lenak  
NPM : 16 07 08842  
Program Studi : Informatika  
Fakultas : Teknologi Industri  
Judul Penelitian : Analisis Faktor Kepuasan dan Ketidakpuasan  
Pelanggan Hotel pada Ulasan *Online* Menggunakan  
*Text Mining*

Menyatakan dengan ini:

1. Tugas Akhir ini adalah benar tidak merupakan salinan sebagian atau keseluruhan dari karya penelitian lain.
2. Memberikan kepada Universitas Atma Jaya Yogyakarta atas penelitian ini, berupa Hak untuk menyimpan, mengelola, mendistribusikan, dan menampilkan hasil penelitian selama tetap mencantumkan nama penulis.
3. Bersedia menanggung secara pribadi segala bentuk tuntutan hukum atas pelanggaran Hak Cipta dalam pembuatan Tugas Akhir ini.

Demikianlah pernyataan ini dibuat dan dapat dipergunakan sebagaimana mestinya.

Palu, 22 Juni 2020

Yang menyatakan,

Brenda Eugenia Lenak

16 07 08842

## **HALAMAN PERSEMBAHAN**

**Semua akan indah pada waktu-Nya**



## KATA PENGANTAR

Puji dan syukur penulis haturkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan pembuatan tugas akhir “Analisis Faktor Kepuasan dan Ketidakpuasan Pelanggan Hotel pada Ulasan *Online* Menggunakan *Text Mining*” ini dengan baik.

Penulisan tugas akhir ini bertujuan untuk memenuhi salah satu syarat untuk mencapai derajat sarjana Teknik Informatika dari Program Studi Teknik Informatika, Fakultas Teknologi Industri di Universitas Atma Jaya Yogyakarta.

Penulis menyadari bahwa dalam pembuatan tugas akhir ini penulis telah mendapatkan bantuan, bimbingan, dan dorongan dari banyak pihak. Untuk itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Tuhan Yesus Kristus yang selalu membimbing dalam iman-Nya, memberikan berkat-Nya, dan menyertai penulis selalu.
2. Bapak Dr. A. Teguh Siswanto, M.Sc., selaku Dekan Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta.
3. Bapak Martinus Maslim, S.T., M.T., selaku Kepala Program Studi Teknik Informatika, Universitas Atma Jaya Yogyakarta
4. Bapak Prof. Ir. Djoko Budiyanto SHR A, M.Eng., Ph.D, selaku dosen pembimbing I sekaligus dosen pembimbing akademik penulis yang telah membimbing dan memberikan masukan serta motivasi kepada penulis untuk menyelesaikan tugas akhir ini.
5. Ibu Ike Devi Sulistyningtyas, S.Sos., M.Si., selaku dosen pembimbing II yang telah membimbing dan memberikan masukan serta motivasi kepada penulis untuk menyelesaikan tugas akhir ini.
6. Keluarga penulis terutama kedua orang tua yang telah mendoakan dan memberikan dukungan baik secara moral maupun materi kepada penulis selama berkuliah sehingga penulis boleh melewatinya dengan baik.

7. Para sahabat dan teman yang telah banyak membantu dan memberikan semangat kepada penulis dalam seluruh proses pengerjaan tugas akhir ini dan proses perkuliahan.
8. Semua pihak yang telah membantu penulis selama proses pengerjaan tugas akhir yang tidak dapat penulis sebutkan satu per satu.

Demikian laporan tugas akhir ini dibuat, dan penulis mengucapkan terima kasih kepada semua pihak. Semoga laporan ini dapat bermanfaat bagi pembaca.

Palu, 22 Juni 2020

Brenda Eugenia Lenak

16 07 08842



# DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS & PUBLIKASI ILMIAH.....	iii
HALAMAN PERSEMBAHAN.....	iv
KATA PENGANTAR .....	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
INTISARI.....	xi
BAB I. PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian .....	4
1.5. Metode Penelitian.....	4
BAB II. TINJAUAN PUSTAKA.....	7
BAB III. LANDASAN TEORI.....	10
BAB IV. ANALISIS DAN PERANCANGAN EKSPERIMEN .....	19
4.1. Deskripsi Problem.....	19
4.2. Analisis Kebutuhan Eksperimen.....	19
4.3. Perancangan Eksperimen .....	20
BAB V. HASIL EKSPERIMEN DAN PEMBAHASAN.....	33
5.1. Deskripsi Eksperimen .....	33
5.2. Hasil Eksperimen .....	33

5.3. Pembahasan Eksperimen.....	41
BAB VI. PENUTUP .....	47
6.1. Kesimpulan .....	47
6.2. Saran.....	48
LAMPIRAN.....	49
DAFTAR PUSTAKA .....	68





## DAFTAR GAMBAR

Gambar 3.1. Skala Bobot Sentimen .....	12
Gambar 3.2. Model <i>Community Management</i> .....	16
Gambar 4.1. Hasil Pengambilan Data .....	20
Gambar 4.2. Contoh <i>Stopwords</i> .....	23
Gambar 4.3. Hasil Proses Analisis Sentimen.....	27
Gambar 4.4. Hasil Penghitungan Frekuensi Kata Menggunakan <i>N-Gram</i> .....	29
Gambar 4.5. Hasil Proses Identifikasi Empat <i>Core</i> Bisnis Perhotelan .....	32
Gambar 5.1. Persentase Hasil Sentimen Analisis Hotel Bintang 4.....	34
Gambar 5.2. Persentase Hasil Sentimen Analisis Hotel Bintang 5.....	38

## DAFTAR TABEL

Tabel 3.1. Contoh <i>Sentiwords</i> .....	12
Tabel 3.2. Aturan Leksikal <i>Sentistrength</i> .....	13
Tabel 4.1. Contoh Data Mentah .....	21
Tabel 4.2. Tambahan <i>Stopwords</i> .....	23
Tabel 4.3. Contoh Hasil <i>Pre-Processing</i> Data.....	24
Tabel 4.4. Contoh Transformasi Data Analisis Sentimen Menggunakan <i>Sentistrength</i> .....	26
Tabel 4.5. Kode Empat <i>Core</i> Bisnis Perhotelan .....	30
Tabel 4.6. Contoh Kamus Empat <i>Core</i> Bisnis Perhotelan .....	30
Tabel 4.7. Contoh Proses Identifikasi Empat <i>Core</i> Bisnis Perhotelan Menggunakan Kamus.....	31
Tabel 5.1. 15 Kata Terbanyak Pada Ulasan Hotel Bintang 4.....	35
Tabel 5.2. Peringkat <i>Core</i> Bisnis Hotel Terkuat Pada Hotel Bintang 4.....	36
Tabel 5.3. 15 Kata Terbanyak Pada Ulasan Hotel Bintang 5.....	39
Tabel 5.4. Peringkat <i>Core</i> Bisnis Hotel Terkuat Pada Hotel Bintang 5.....	40
Tabel 5.5. 5 Faktor Teratas Kepuasan dan Ketidakpuasan Pelanggan Hotel Bintang 4 .....	43
Tabel 5.6. 5 Faktor Teratas Kepuasan dan Ketidakpuasan Pelanggan Hotel Bintang 5 .....	43

# INTISARI

## ANALISIS FAKTOR KEPUASAN DAN KETIDAKPUASAN PELANGGAN HOTEL PADA ULASAN *ONLINE* MENGGUNAKAN *TEXT MINING*

Intisari

Brenda Eugenia Lenak

16 07 08842

Selain dikenal sebagai kota pelajar, Yogyakarta juga dikenal sebagai kota wisata dengan banyak destinasi wisata yang meliputi wisata alam, budaya, hingga seni sehingga memiliki banyak sarana akomodasi seperti hotel. Wisatawan yang menggunakan layanan hotel tentu saja memiliki hak untuk memberikan nilai terhadap kualitas dari hotel tersebut. Pelanggan yang puas akan pelayanan hotel memiliki kemungkinan untuk kembali menggunakan layanan di hotel tersebut di kemudian hari bahkan merekomendasikan hotel tersebut ke orang lain ataupun sebaliknya sehingga harus menjadi perhatian pihak hotel. Maka penelitian ini bertujuan untuk mengidentifikasi faktor-faktor yang membuat pelanggan hotel puas dan tidak puas melalui ulasan yang mereka unggah.

Metode *Text Mining* dan Analisis Sentimen yaitu *Sentistrength* diterapkan pada ulasan yang dikumpulkan dari situs *TripAdvisor.com* untuk hotel-hotel bintang empat dan lima yang berada di DI Yogyakarta. Kumpulan ulasan ini akan analisis nilai sentimennya dan dibagi ke dalam kelas positif, netral dan negatif untuk melihat tingkat kepuasan pelanggan (*Customer Satisfaction Level*). Kemudian hasil masing-masing kelasnya akan diproses menggunakan metode *n-gram* untuk mengidentifikasi kata-kata yang paling sering disebutkan pelanggan sehingga akan terlihat beberapa faktor umum yang sering dibahas oleh pelanggan hotel. Selain itu pada penelitian ini juga akan dilakukan identifikasi kekuatan empat *core* utama bisnis perhotelan yang berfungsi sebagai pedoman bisnis perhotelan.

Temuan penelitian ini menunjukkan tingkat kepuasan pelanggan hotel dan lima faktor umum kepuasan dan ketidakpuasan pelanggan yang muncul pada masing-masing ulasan yang puas (positif) dan tidak puas (negatif). Selain itu, penelitian ini juga menunjukkan tingkat kekuatan dari empat *core* utama bisnis perhotelan yang ada menunjang kepuasan dan ketidakpuasan pelanggan pada ulasan.

Kata Kunci: Analisis Sentimen, *Customer Satisfaction*, *N-Gram*, *Text Mining*

Dosen Pembimbing I : Prof. Ir. Djoko Budiyanto SHR A, M.Eng., Ph.D

Dosen Pembimbing II : Ike Devi Sulistyningtyas, S.Sos., M.Si.

Jadwal Sidang Tugas Akhir : Jumat, 3 Juli 2020



# **BAB I. PENDAHULUAN**

## **1.1. Latar Belakang**

Selain dikenal sebagai kota pelajar, Yogyakarta juga dikenal sebagai kota wisata. Hal ini ditunjukkan dengan banyaknya destinasi wisata di Yogyakarta yang meliputi wisata alam, budaya, dan desa/kampung wisata. Berdasarkan data Statistik Kepariwisataaan DIY Tahun 2018, jumlah objek wisata di DIY sebanyak 185 objek wisata [1]. Objek-objek wisata tersebut telah menjadi magnet yang kuat untuk berkunjung ke Yogyakarta. Dan berdasarkan data tersebut juga tercatat bahwa total kunjungan wisatawan ke objek-objek wisata tersebut mencapai 25.515.788 orang pertahun [1].

Sebagai kota wisata, adanya hotel sebagai sarana akomodasi para wisatawan tentu saja sangat penting. Hotel menyediakan tempat tinggal sementara untuk para wisatawan untuk beristirahat dan menikmati kegiatan mereka. Berdasarkan data Statistik Kepariwisataaan DIY Tahun 2018 juga menyebutkan bahwa ada kenaikan jumlah wisatawan yang menggunakan jasa akomodasi hotel bintang dan non bintang pada tahun 2018 sebesar 8,79% dibanding tahun 2017. Sehingga dapat diperkirakan bahwa minat wisatawan untuk menggunakan jasa akomodasi akan terus meningkat dari tahun ke tahun. Karena itu, hotel sebagai penyedia akomodasi harus bersiap dan mampu untuk menyediakan kebutuhan wisatawan dan juga mampu memperbaharui diri dan memperbaiki kesalahan yang ada untuk memberikan kesan yang baik dan nyaman pada wisatawan [1].

Sejalan dengan cepatnya arus globalisasi, perkembangan industri perhotelan pun berkembang sangat cepat, tidak hanya di negara kita tetapi di seluruh negara [2]. Hotel-hotel di dunia saat ini tengah berlomba-lomba untuk menyediakan fasilitas akomodasi terbaik untuk para pelanggannya. Hal ini tentu saja untuk membuat pelanggan puas dan merasakan pengalaman tak terlupakan ketika menggunakan jasa mereka. Pelanggan yang puas akan pengalamannya menggunakan layanan ini adalah pelanggan yang akan kembali menggunakan

layanan yang sama untuk waktu yang akan datang [3]. Tidak hanya kembali menggunakan layanan yang sama, pelanggan yang puas juga dapat merekomendasikannya kepada orang lain [4]. Hal ini juga sejalan dengan hasil penelitian dari Dwi dan Rochman [5] bahwa kualitas layanan mempunyai pengaruh langsung yang positif dan signifikan terhadap kepercayaan juga kepuasan pelanggan bahkan juga berlaku sebaliknya. Karena itulah, menurut Jay dan Dwi [4] kepuasan pelanggan saat ini dianggap sebagai salah satu hasil terpenting dari semua kegiatan pemasaran di perusahaan yang berorientasi pasar seperti hotel.

Pelanggan merupakan jantung dari sebuah perusahaan berorientasi pasar seperti hotel karena pelangganlah yang menentukan kelangsungan hidup perusahaan tersebut ke depannya [6]. Kepuasan pelanggan adalah kunci dalam menciptakan hubungan jangka panjang antara perusahaan (Hotel) dengan pelanggannya [7]. Hotel mencoba untuk berkomunikasi dengan pelanggan lewat berbagai media untuk mengetahui apakah pelanggan puas atau tidak. Kegiatan untuk mengukur kepuasan pelanggan saat ini dibutuhkan oleh pihak perusahaan untuk melihat apakah layanan yang mereka punya berhasil membuat pelanggan puas ataukah tidak sebagai strategi dalam mempertahankan hubungan jangka panjang ini [7]. Karena itulah tanggapan-tanggapan dari pelanggan (baik yang di dapatkan secara langsung maupun secara daring) merupakan bagian yang tidak dapat dipisahkan dari kegiatan pengukuran kepuasan [6].

Seiring berkembangnya situs penyedia ulasan secara daring, penelitian pun sudah banyak dilakukan dalam bidang analisis sentimen [8]. Analisis sentimen dan *text mining* merupakan dua hal yang tidak dapat dipisahkan terutama jika melakukan analisis sentimen pada media daring [9]. Analisis sentimen digunakan untuk mengukur tingkat sentimen suatu data teks yang dalam penelitian kali ini diambil dari data teks ulasan pelanggan yang diunggahnya secara daring (*Online Review*). Pengukuran ini menggunakan pembobotan terhadap suatu kalimat, kata, bahkan pasangan kata yang ada untuk menentukan kelas sentimen kata (Positif, Netral, dan Negatif). Kegiatan pengukuran

sentimen ini, digunakan untuk mengetahui dan menganalisis pendapat dan kesan dari pelanggan setelah menggunakan layanan hotel. Hal ini dapat dilihat dari seberapa baik hotel memberikan pelayanan kepada pelanggan-pelanggannya yang kemudian akan menjadi faktor penentu tingkat kepuasan dari pelanggan [10].

Penelitian akan berfokus pada pengukuran sentimen pelanggan hotel dari ulasan yang diunggah oleh pelanggan hotel pada situs *TripAdvisor.com* yang kemudian akan diproses dan diolah untuk dapat menghasilkan kelas tingkatan kepuasan pelanggan hotel yang berada di DI Yogyakarta sebagai salah satu kota pariwisata di Indonesia. Kemudian data diolah berdasarkan kelasnya (Positif, Netral, dan Negatif) untuk melihat faktor-faktor umum yang dibicarakan pelanggan yang mempengaruhi tingkatan kepuasan tersebut. Setelah itu data akan diidentifikasi dengan empat *core* bisnis utama perhotelan untuk melihat kekuatan dari keempat *core* bisnis ini dalam mempengaruhi kepuasan pelanggan. Penelitian ini juga akan melakukan analisis berdasarkan hasil-hasil yang didapatkan untuk dapat menjadi saran kepada pihak hotel dalam upaya peningkatan dan perbaikan kualitas.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan di atas, maka dapat dirumuskan masalah yaitu bagaimana menerapkan *text mining* dalam menemukan dan menganalisis faktor-faktor kepuasan dan ketidakpuasan pelanggan hotel di Yogyakarta berdasarkan ulasan *online* yang diunggah pelanggan di situs *TripAdvisor.com*?

## **1.3. Batasan Masalah**

Berisikan rumusan masalah yang di atas, maka ada beberapa batasan masalah yang dilakukan, yaitu :

1. Analisis faktor kepuasan dan ketidakpuasan pelanggan hotel yang dibuat adalah khusus untuk hotel-hotel berbintang empat dan lima yang berada di Yogyakarta.

2. Data yang diambil berupa ulasan *online* berbahasa Indonesia.
3. Analisis dibuat menggunakan data ulasan *online* hotel-hotel bintang empat dan lima yang ada di Yogyakarta yang diambil dari situs *TripAdvisor.com*.

#### 1.4. Tujuan Penelitian

Berdasarkan rumusan masalah yang di atas, maka tujuan penelitian ini adalah untuk menemukan dan menganalisis faktor kepuasan dan ketidakpuasan pelanggan hotel di Yogyakarta berdasarkan ulasan *online* yang diunggah pelanggan di situs *TripAdvisor.com* menggunakan *text mining*.

#### 1.5. Metode Penelitian

Penelitian ini menggunakan data yang diambil dari situs *TripAdvisor.com*. Data berupa ulasan hotel-hotel bintang empat dan lima yang berada di DI. Yogyakarta.

Berikut adalah metode yang digunakan dalam penelitian ini:

1. Pengumpulan data dari situs web *TripAdvisor.com*

Data yang diambil yaitu data teks dari ulasan setiap hotel bintang empat dan lima di DI. Yogyakarta.

2. *Pre-processing* data

Beberapa proses *pre-processing* atau pembersihan data yang diterapkan yaitu *Case Folding* (Mengubah seluruh huruf pada data menjadi *lowercase*), *Filtering* yang terdiri dari menghapus tanda baca, angka, *URL* dan *hashtag*, *stopwords*, dan spasi berlebih, dan *Normalization* (Memperbaiki beberapa kata singkatan atau tidak lengkap).

3. Analisis Sentimen menggunakan metode *Sentistrength*

Data akan dikategorisasi berdasarkan nilai sentimennya menggunakan metode *Sentistrength* menjadi tiga kelas yaitu kelas positif, netral, dan negatif kemudian akan dihitung persentasenya dan



selanjutnya dianalisis. Analisis sentimen ini digunakan untuk menganalisis kesan yang diberikan oleh pelanggan berdasarkan pengalaman mereka sendiri dalam menggunakan layanan hotel.

4. Penghitungan frekuensi kata menggunakan metode *n-gram*

Data akan dihitung frekuensi dari masing-masing kata untuk setiap kelas menggunakan metode *n-gram*. Penghitungan ini dilakukan untuk mendapatkan kata-kata yang paling sering muncul pada masing-masing kelas dan digunakan sebagai bahan analisis faktor-faktor umum kepuasan pelanggan. Selain itu, hasil dari metode ini akan digunakan juga untuk penyusunan kamus / leksikon baru untuk tahap identifikasi empat *core* bisnis perhotelan.

5. Identifikasi kekuatan empat *core* bisnis Perhotelan (*The Four Main Bussiness Quadrants*)

Data akan kembali diolah untuk melakukan identifikasi kekuatan empat *core* bisnis hotel menggunakan pendekatan berbasis *leksikon* (kamus). Selanjutnya data akan dianalisis untuk mendapatkan *core* bisnis terkuat yang mempengaruhi kepuasan pelanggan.

6. Penarikan kesimpulan berdasarkan hasil eksperimen

Membuat kesimpulan tentang pengetahuan faktor kepuasan dan ketidakpuasan pelanggan hotel berdasarkan hasil-hasil yang telah didapatkan sebelumnya. Lalu melakukan analisis bagaimana pihak hotel dapat menggunakan pengetahuan yang terbentuk untuk meningkatkan maupun memperbaiki kualitas layanan ke depannya agar pelanggan bisa mendapatkan kepuasan yang maksimal.

## 1.6. Sistematika Penulisan

### BAB I PENDAHULUAN

Bab ini berisi konsep dasar dalam pembuatan dan penulisan tugas akhir.

## **BAB II TINJAUAN PUSTAKA**

Bab ini berisi beberapa penelitian terdahulu beserta tabel pembandingnya yang menyangkut penelitian yang dilakukan penulis.

## **BAB III LANDASAN TEORI**

Bab ini berisi penjelasan tentang teori-teori yang digunakan penulis pada penelitian yang dilakukan.

## **BAB IV ANALISIS DAN PERANCANGAN EKSPERIMEN**

Bab ini berisi deskripsi problem, analisis kebutuhan, dan perancangan eksperimen yang dilakukan.

## **BAB V HASIL EKSPERIMEN DAN PEMBAHASAN**

Bab ini berisi deskripsi eksperimen, hasil eksperimen, dan pembahasan eksperimen yang dilakukan.

## **BAB VI PENUTUP**

Bab ini berisi kesimpulan dan saran yang didapatkan dari penelitian yang dilakukan.

## BAB II. TINJAUAN PUSTAKA

Perkembangan teknologi telekomunikasi tahun ke tahun bertujuan untuk mendukung kelancaran komunikasi masyarakat. Akan tetapi, masyarakat memiliki kebutuhan yang mengharapkan penyedia layanan telekomunikasi dapat meningkatkan layanan mereka. Kepuasan masyarakat yang dalam hal ini sebagai pengguna layanan merupakan fokus utama dari penyedia layanan telekomunikasi agar tidak pindah ke penyedia jasa lainnya. Seperti yang dilakukan oleh penelitian sebelumnya oleh Umi dkk [11] untuk dapat mengetahui tingkat kepuasan dari pengguna, maka diperlukan analisis sentimen pada *Twitter* pengguna yang menyangkut penyedia layanan telekomunikasi tersebut. Analisis sentimen ini mengelompokkan polaritas dari teks dalam kalimat untuk diklasifikasikan sebagai pendapat yang positif atau negatif dan menggunakan metode *SVM*.

Selain Metode *SVM*, banyak penelitian analisis sentimen yang menggunakan metode lain seperti metode *Naïve Bayes* dan metode Algoritma Genetik (GA). Seperti pada penelitian Pradana dkk [10] yang meneliti mengenai analisis sentimen masyarakat terhadap suatu film, juga Risa [12] yang melakukan penelitian analisis sentimen pada ulasan jasa maskapai penerbangan. Opini yang dihimpun dari *Twitter* mengenai film dan ulasan jasa maskapai penerbangan dari <http://www.airlinequality.com>. Tujuan penelitian ini untuk mengetahui bagaimana pendapat masyarakat mengenai suatu film yang diproduksi dan pendapat masyarakat mengenai kualitas jasa maskapai penerbangan. Sama seperti pada penelitian di atas, penelitian ini membagi ulasan ke dalam positif dan negatif, sehingga dapat membantu konsumen dalam memiliki jasa maskapai penerbangan yang terbaik.

Penelitian selanjutnya yang dilakukan oleh Dinda [13] yang juga menggunakan metode *Naïve Bayes*, yaitu melakukan analisis sentimen pada ulasan restoran dengan teks bahasa Indonesia. Penelitian ini bertujuan untuk mengintegrasikan metode pemilihan fitur *Genetic Algorithm* dalam menganalisa sentimen pada ulasan restoran dengan teks bahasa Indonesia menggunakan

algoritma *Naïve Bayes*. Hal ini untuk membantu para calon pembeli dalam pengambilan keputusan dalam mencari restoran agar bisa mengurangi waktu dalam membaca ulasan dan komentar suatu restoran.

Selain dikombinasikan dengan metode *Genetic Algorithm*, metode *Naïve Bayes* juga digunakan dalam penelitian Andi Taufik [14] dengan mengintegrasikannya bersama metode *Particle Swarm Optimization* untuk menganalisis sentimen pada ulasan hotel juga setahun kemudian penelitian dilakukan oleh Agung Nugroho [15] yang menganalisis sentimen pada media sosial mengenai isu kenaikan tarif dasar listrik di *Twitter* yang juga menggunakan *Naïve Bayes* dengan dikombinasikan fitur *N-Gram*. Andi menganalisis ulasan hotel yang didapatkannya dari situs *TripAdvisor.com* yang kemudian melihat besar akurasi yang didapatkan dengan menggabungkan kedua metode tersebut. Penelitian Andi ini menghasilkan peningkatan yang baik jika menggunakan kedua metode tersebut untuk analisis ulasan hotel. Sedangkan Agung juga menghasilkan peningkatan yang baik dengan menggabungkan metode *N-Gram* pada *Naïve Bayes*.

Selain Andi, beberapa penelitian lain yang juga menganalisis sentimen pada ulasan hotel yaitu penelitian yang dilakukan oleh Katerina dkk [16] yang menganalisis ulasan hotel di Sarasota dan Shawn dkk [17] yang menganalisis ulasan hotel di Moskow. Katerina menggunakan Metode *Text Mining* yang digunakannya dengan *tools PASW modeler* sehingga dapat menghasilkan 10 peringkat topik besar masing-masing faktor kepuasan dan ketidakpuasan pelanggan. Lalu Shawn menggunakan metode *statistic Regression Linear* untuk mengklasifikasikan ulasan menjadi masing-masing 5 topik besar beserta tingkat kepuasannya yaitu rendah, sedang, dan tinggi.

Penelitian selanjutnya yaitu penelitian yang dilakukan oleh Ernitza dan Walid [18] yang melakukan analisis untuk mengetahui pendapat pengguna tentang fitur dari suatu aplikasi yang didapatkan dari *App Store*. Data yang didapatkan merupakan ulasan pengguna dari 7 aplikasi yang ada di *Apple App Store* dan *Google Play Store*. Penelitian ini menggunakan metode sentimen analisis berbasis leksikan atau menggunakan kamus yaitu metode klasifikasi *sentistrength* untuk mengklasifikasi ulasan ke dalam kelas positif dan negatif. Hasil yang didapatkan

berupa fitur-fitur yang paling besar frekuensi katanya pada masing-masing kelasS.

Lalu, penelitian lainnya yaitu penelitian yang dilakukan En-Gir Kim dan Se-Hak Chun [19] yang menganalisis ulasan *online* dari pengguna beberapa merek mobil untuk mengetahui apa saja kelebihan dan kekurangan dari merek-merek mobil tersebut. Data diambil dari situs *Edmunds.com* dan diolah menggunakan *Statistic method* dengan *tools R Programming*. Penelitian ini menghasilkan masing-masing lima kelebihan dan kekurangan dari tiga merek mobil tertentu.

Penelitian selanjutnya dilakukan oleh penulis yaitu untuk menganalisis faktor-faktor kepuasan dan ketidakpuasan pelanggan hotel pada ulasan *online* di Yogyakarta. Penelitian ini untuk menemukan dan menganalisis faktor-faktor kepuasan dan ketidakpuasan apa saja yang menyebabkan kepuasan ataupun ketidakpuasan pelanggan dari pengalaman mereka menggunakan layanan hotel-hotel di Yogyakarta. Data yang digunakan yaitu data ulasan *online* hotel-hotel bintang empat dan lima di Yogyakarta yang dikumpulkan dari situs web *TripAdvisor.com*. Penelitian ini menggunakan metode *Text Mining* dan Sentimen Analisis dengan menggunakan algoritma *Sentistrength* dan menghasilkan masing-masing lima faktor kepuasan dan ketidakpuasan pelanggan hotel dan juga analisis kekuatan empat *core* utama bisnis perhotelan.

## BAB III. LANDASAN TEORI

### 3.1 *Text Mining* atau Penambangan Teks

*Text mining* merupakan penambangan data berupa teks. Teknik ini secara umum merujuk pada proses penggalian pola-pola yang menarik dan penting atau juga pengetahuan dari dokumen teks yang tidak terstruktur. Penambangan teks melakukan asosiasi satu bagian teks dengan lainnya berdasarkan aturan-aturan tertentu. Ini dapat dianggap sebagai bentuk perpanjangan dan spesifik dari teknik *data mining* atau penambangan data. Karena bentuk paling alami dari penyimpanan informasi adalah teks, maka penambangan teks ini diyakini memiliki potensi komersial yang lebih tinggi dibandingkan dengan penambangan data biasanya. Faktanya, penelitian terbaru menunjukkan bahwa 80% dari informasi sebuah perusahaan terkandung dalam dokumen teks [20].

Sama seperti penambangan data, penambangan teks juga mengalami masalah yang sama yaitu jumlah data yang besar, dimensi yang tinggi, dan data yang kotor. Akan tetapi penambangan teks ini juga merupakan tugas yang jauh lebih kompleks dibandingkan dengan penambangan data karena melibatkan penambangan data teks yang secara inheren tidak terstruktur dan juga tidak jelas minimal data yang semi-struktur sedangkan penambangan data utamanya memproses data yang terstruktur. Dikarenakan data yang terstruktur ditujukan agar dapat dengan mudah diproses oleh komputer atau program secara otomatis, juga *pre-processing* data pada penambangan data jauh lebih mudah daripada penambangan teks. Teks diciptakan bukan untuk mesin, tetapi untuk digunakan oleh manusia untuk menyimpan informasi. Karena itu, sulit membuat mesin untuk dapat mengerti suatu informasi dalam teks [20].

Penambangan teks merupakan bidang yang multidisiplin, yang melibatkan pengambilan informasi, analisis teks, ekstraksi informasi, pengelompokkan, kategorisasi, visualisasi, teknologi basis data, pembelajaran mesin (*machine learning*) dan juga penambangan data. Penambangan teks dapat divisualisasikan menjadi dua fase, yaitu pemurnian teks yang mengubah

dokumen teks bentuk bebas menjadi bentuk peralihan yang dipilih, dan juga penyulingan pengetahuan yang menyimpulkan pola atau pengetahuan dari bentuk peralihan itu sendiri [21].

Penambangan teks yang lebih baru menggunakan teknik yang dikembangkan di bidang pencarian informasi, statistik, dan pembelajaran mesin. Bentuk paling sederhana dari penambangan teks dapat dianggap sebagai pencarian informasi, juga disebut dengan pencarian teks atau dokumen seperti yang dilakukan oleh mesin pencari biasa. Namun, penambangan teks yang lebih tepat terdiri dari bidang-bidang seperti klasifikasi teks otomatis menurut beberapa set kategori tetap, *text clustering*, *automatic summarization*, ekstraksi topik dari teks atau kelompok teks dan juga analisis tren topik dalam aliran teks. Sedangkan pencarian informasi dan bentuk penambangan teks lainnya sering menggunakan *word stemming* [22].

### 3.2 Sentimen Analisis

*Sentiment analysis* atau analisis sentimen merupakan bidang penelitian baru yang lahir dari bidang *Natural Language Processing (NLP)*, yang memiliki tujuan untuk mendeteksi, mengekstraksi, dan mengklasifikasikan pendapat dan sentimen yang terdapat dalam suatu data teks. Analisis ini mempelajari sentimen, pendapat, sikap, evaluasi, penilaian dan juga emosi yang terdapat di dalam data. Berbagai bidang ilmu menggunakan analisis ini untuk mengetahui dan mempelajari lebih lanjut mengenai sentimen orang terhadap suatu kualitas layanan tertentu, individu tertentu, organisasi, masalah, ataupun suatu topik acara tertentu [23].

Ada beberapa kriteria yang biasa digunakan untuk klasifikasi teks dalam analisis sentimen yaitu polaritas sentimen dalam positif, negatif, dan netral, kemudian ada setuju atau tidak setuju, lalu kabar baik atau buruk, dukungan atau oposisi, pro dan kontra, atau bagus atau jelek, dan sebagainya [23]. Untuk mengimplementasikannya, digunakan berbagai pendekatan klasifikasi sentimen juga berbagai *tools* sudah tersedia. Contoh pendekatan yang sering digunakan saat ini yaitu pendekatan berbasis leksikon atau menggunakan kamus.

### 3.3 *Sentistrength*

*Sentistrength* merupakan salah satu metode algoritma atau program yang digunakan dalam analisis sentimen dan *opinion mining*. Karena merupakan salah satu metode dalam analisis sentimen, *sentistrength* digunakan untuk mengukur kekuatan sentimen dari suatu data teks untuk mendeteksi polaritas sentimen [24]. Inti dari *sentistrength* sendiri adalah penggunaan leksikon atau kamus sentimen (*Lexicon based*) yang ada untuk menentukan nilai sentimen teks tersebut [25]. Pada awalnya program *sentistrength* dibuat untuk menganalisis teks dalam Bahasa Inggris dengan menggunakan kamus sentimen Bahasa Inggris sehingga kamus ini diadaptasi dalam Bahasa Indonesia.

Program *sentistrength* akan mencocokkan setiap kata dari data inputan dengan yang ada dikamus sentimen lalu akan menampilkan skor sentimennya baik sentimen positif maupun negatif. Skala bobot skor sentimen seperti pada gambar 3.1. Semakin tinggi nilainya maka akan semakin kuat sentimen yang dimilikinya baik untuk positif maupun negatif. Angka 0 tidak digunakan dan angka 1 dan -1 menunjukkan tidak adanya sentimen baik positif maupun negatif [24]. Kata-kata pada yang ada pada kamus sentimen terdiri dari kata sifat, kata kerja, kata benda, kata keterangan yang memiliki kekuatan sentimen seperti pada contoh tabel 3.1. [26].

-1 (not negative) to -5 (extremely negative)  
1 (not positive) to 5 (extremely positive)

Gambar 3.1. Skala Bobot Sentimen

Tabel 3.1. Contoh *Sentiwords*

Adil :5	Gagap : -2	Gesit : 3	Licik : -4	Mulia : 4
Alibi : -4	Emosi : -3	Heboh : 2	Layu :-3	Mesra : 3
Bau : -4	Dewasa : 4	Hina : -4	Kaya : 3	Nikmat : 4
Berani : 4	Dendam : -3	Ironis : -2	Kalap : -4	Padat : 1
Cantik : 5	Cemas : -3	Jauh : -2	Jelas : 4	Pamer : -4



Setelah itu program akan menghitung skor akhir sentimen yang akan menentukan kelas sentimen suatu teks. Cara penghitungan skor akhir sentimen ini yaitu program akan menyimpan skor tertinggi dari masing-masing skor positif dan negatif lalu akan menambahkan kedua hasil tersebut. Hasil penambahan ini akan menghasilkan skor akhir sentimennya yaitu antara rentang -4 hingga 4. Akan tetapi, sebelum melakukan penghitungan skor akhir sentimen ini ada beberapa aturan linguistik tambahan yang dibuat untuk mengatasi beberapa masalah dalam bahasa [25]. Beberapa aturan ini antara lain adanya kata negasi (*negating words*) yang berfungsi untuk membalik polaritas sentimen, kata pendukung (*booster words*) untuk menambahkan nilai skor sentimen, emotikon dan kata ungkapan (*idiom words*) untuk memberikan skor tertentu pada elemen emotikon karena dinilai mengandung sentimen, kata tanya yang digunakan untuk mengabaikan kata negatif pada teks jika mengandung kata tanya. Contoh aturan-aturan leksikal ini seperti pada tabel 3.2.

Tabel 3.2. Aturan Leksikal *Sentistrength*

Aturan	Contoh	Hasil
<i>Negating Words</i>	Baik = 4 Kasar = -3	Tidak Baik = -4 Bukan Kasar = 3
<i>Booster Words</i>	Bijak = 4 Jahat = -4	Lebih Bijak = 5 Agak Jahat = -3
Emotikon	^_^ = 4 :'( = -3	-
<i>Idiom Words</i>	Anak Emas = 4 Kambing Hitam = -4	-
<i>Question Words</i>	Apakah kamu jahat[-4]?	Skor =1 [netral]

### 3.4 Kepuasan Pelanggan (*Customer Satisfaction*)

*Customer Satisfaction* atau kepuasan pelanggan adalah suatu metrik yang digunakan oleh pihak layanan pelanggan profesional suatu perusahaan untuk

mengukur perasaan pelanggan mengenai adanya interaksi terbaru antara perusahaan dengan pelanggannya [7]. Kepuasan pelanggan ini juga dapat merujuk pada pengukuran seberapa puas seorang pelanggan dengan perlakuan perusahaan kepada mereka. Mengukur kepuasan pelanggan berarti memfokuskan pelanggan sebagai jantung dari bisnis suatu perusahaan [3]. Ini dapat membantu perusahaan dalam membentuk identitas *brand* mereka. Pelanggan yang sangat puas akan mampu menyebarkan kalimat atau kata-kata yang baik pada orang lain mengenai bisnis perusahaan tersebut [3]. Setelah itu perusahaan akan menikmati citra perusahaan yang sangat positif. Citra yang positif ini memungkinkan untuk membuat perusahaan menikmati perlindungan dari pelanggan setia hingga mampu menarik pelanggan-pelanggan baru [27].

Kepuasan pelanggan adalah kunci dari menjalin hubungan jangka panjang suatu perusahaan dengan pelanggannya [7]. Pelanggan yang puas dengan pengalamannya adalah pelanggan yang akan kembali ke bisnis perusahaan tersebut dalam waktu yang dekat [3]. Karena itu, perusahaan perlu mempertahankan kepuasan pelanggannya. Jika perusahaan dapat mempertahankan kepuasan pelanggan pada fokus pemikiran dan secara konstan memperhatikannya dalam pengembangan, perusahaan mampu mengurangi tingkat kesempatan dalam membuat sesuatu yang salah kepada pelanggan [7]. Ketika pelanggan memberikan ulasan dan penilaian mereka, perusahaan dapat mengetahui apa yang harus mereka tingkatkan maupun mereka perbaiki.

Ketika mencoba mengerti dan mengidentifikasi kebutuhan dan ekspektasi pelanggan, perusahaan tidak boleh hanya mendengar opini dari pelanggannya saja tetapi dari semua pelanggan baik itu pelanggan perusahaan tersebut maupun bukan. Pelanggan yang bukan pelanggan perusahaan tersebut memiliki kemungkinan untuk tertarik dengan perusahaan tersebut. Selain itu, perusahaan juga harus mendengarkan ulasan yang positif maupun negatif tidak hanya salah satu. Karena pelanggan yang tidak puas adalah pelanggan yang memberi perusahaan arahan dalam perkembangan perusahaan [3]. Ada beberapa tipe pengukuran yang berbeda yang biasanya digunakan dalam mengukur kepuasan pelanggan yaitu sebagai berikut [3]:

- *Ad-hoc measurement*

Pengukuran ini merupakan pengukuran yang dilakukan pada suatu waktu T dan dibuat sesuai dengan pesanan atau permintaan tertentu untuk memberikan solusi yang disesuaikan. Pengukuran ini dibuat khusus berdasarkan masalah atau peristiwa tertentu sehingga terbatas. Akan tetapi kepuasan pelanggan adalah konsep yang berkembang dan tidak terbatas sehingga pengukuran ini tidak fleksibel dan hanya memberikan pandangan yang tetap mengenai peristiwa khusus.

- *Barometric measurement*

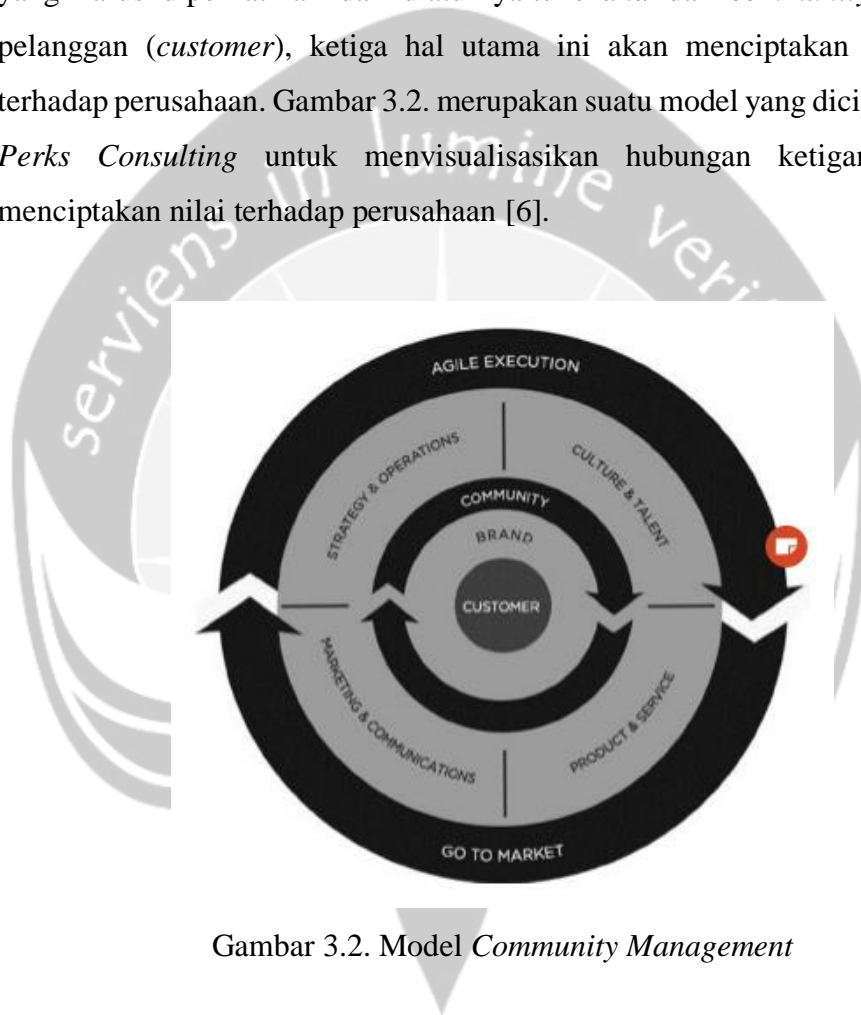
Pengukuran ini merupakan pengukuran yang reguler dan sudah sering digunakan. Pengukuran ini memungkinkan untuk mengukur perubahan kepuasan pelanggan dari waktu ke waktu dan dapat dilakukan setiap kuartal, semester, dll. Akan tetapi indikator dan implementasi operasional pengukuran ini memerlukan perhatian khusus sehingga pengukuran tetap stabil dari waktu ke waktu.

- *Continuous measurement*

Pengukuran ini merupakan jenis studi terakhir yang mungkin dilakukan dan direkomendasikan. Pengukuran ini menawarkan nilai tambah dan memungkinkan perusahaan untuk melihat lebih spesifik seperti “apa yang menyenangkan” atau “apa yang tidak menyenangkan” untuk melakukan perbaikan. Pada pengukuran ini kuesioner tidak akan diberikan dalam satu waktu tetapi dapat diisi kapan saja sehingga pelanggan dapat mengekspresikan diri kapanpun mereka ingin melakukannya. Perusahaan dapat mengikuti perubahan sehari-hari kepuasan pelanggan mereka. Akan tetapi, pengukuran ini membutuhkan alat yang fleksibel yang dapat dengan mudah diintegrasikan ke dalam pengukuran ini.

### 3.5 The Four Main Business Quadrants

Perusahaan sukses menaruh pelanggan sebagai pusat dari segalanya. Dengan mengerti kebutuhan dan keinginan pelanggan, perusahaan dapat menciptakan solusi untuk masa yang akan datang dan memastikan kelangsungan hidup perusahaan tersebut. Setelah pelanggan, ada hal utama lain yang harus diperhatikan dan diatur yaitu *brand* dan *community*. Bersama pelanggan (*customer*), ketiga hal utama ini akan menciptakan suatu nilai terhadap perusahaan. Gambar 3.2. merupakan suatu model yang diciptakan oleh *Perks Consulting* untuk memvisualisasikan hubungan ketiganya dalam menciptakan nilai terhadap perusahaan [6].



Gambar 3.2. Model *Community Management*

Berdasarkan model yang dibuat oleh *Perks Consulting* yaitu pelanggan sebagai jantung atau pusat artinya segala yang dilakukan perusahaan selalu berdasarkan pelanggan. Perusahaan menyediakan produk dan layanan yang berdasarkan kebutuhan pelanggan. Dengan *brand* yang unik juga menarik dan *community managing* yang baik, perusahaan dapat memperoleh impresi yang baik dari pelanggan. Akan tetapi, tidak hanya itu perusahaan juga perlu

menyusun strategi bisnis yang tepat agar mampu memenangkan hati pelanggan. Perusahaan harus memperhatikan empat *core* bisnis yang mereka jalankan untuk menyusun strategi yang tepat [6]. Keempat *core* ini yaitu :

1. *Product and Service* (Produk dan Layanan)

Dalam sebuah perusahaan, produk dan layanan haruslah selalu berorientasi dari pelanggan yang setiap waktu terus berubah dan berkembang. Produk dan layanan ini harus sesuai dengan yang pelanggan butuhkan dan inginkan. Sehingga perusahaan harus mampu untuk terus berinovasi dan berkreasi untuk menyediakan produk dan layanan yang sesuai dengan ekspektasi pelanggan. Tidak hanya produk utamanya, tetapi seluruh produk termasuk pendukung seperti *cover* produk juga harus diperhatikan. Sehingga perusahaan harus memiliki mekanisme yang dapat membantu perusahaan untuk memahami kebutuhan dan keinginan pelanggan.

2. *Marketing and Communications* (Pemasaran dan Komunikasi)

Pemasaran pada dasarnya merupakan upaya yang dilakukan perusahaan untuk menggerakkan pergerakan pembelian produk atau jasa yang ditawarkan. Perusahaan perlu hadir dan berinteraksi langsung dengan pelanggan untuk menarik perhatian pelanggan sehingga memicu keinginan untuk membeli dan menggunakan produk dan layanan perusahaan tersebut. Karena itu, perusahaan harus tahu strategi pemasaran untuk mencari tahu siapa target penjualannya, apa yang diinginkan, bagaimana membangun relasi, dan apa yang diharapkan. Selain itu, untuk memasarkan produk dan layanannya perusahaan perlu memperhatikan bagaimana mereka berkomunikasi sehingga pelanggan mudah memahami pesan yang ingin disampaikan seperti menggunakan suatu media perantara, atau iklan, atau bentuk lainnya.

### 3. *Culture and Talent* (Budaya dan SDM)

Perusahaan juga harus memiliki budaya dan kualitas SDM yang baik dan sesuai. Budaya yang dimaksud adalah iklim yang dibangun sehingga menciptakan suatu kesan tertentu dan dapat menjadi ciri perusahaan tersebut. Seperti misalnya perusahaan yang disiplin, kuno, modern, dan lainnya. Budaya ini dapat dibangun dari sumber daya yang ada di dalamnya sehingga dengan adanya SDM yang berkualitas dan sesuai dengan kriteria maka akan membantu perusahaan memperkuat budaya tersebut. Sehingga SDM merupakan aset yang sangat bernilai dan penting. Contohnya hotel yang berada di bawah afiliasi perusahaan Perancis akan berbeda dengan hotel yang berada di bawah afiliasi perusahaan Amerika karena adanya perbedaan budaya seperti bagaimana cara mereka menyapa, cara mereka melayani, ataupun bagaimana cara mereka berpenampilan.

### 4. *Strategy and Operations* (Strategi dan Pengoperasian)

Strategi merupakan landasan (*blueprint*) menuju kesuksesan. Hal ini berarti bagaimana perusahaan berupaya untuk memenangkan pasar. Strategi ini yang menjadi landasan perusahaan mengambil keputusan menciptakan produk yang inovatif, menentukan bagaimana berkomunikasi dengan pelanggan dan sebagainya. Sedangkan pengoperasiannya menyangkut faktor-faktor yang mendukung dan menyertai langkah-langkah strategi seperti faktor finansial, aturan dan kebijakan, prosedur-prosedur, lokasi, dan sebagainya. Dengan strategi dan pengoperasiannya, perusahaan menjadi paham siapa kompetitornya dan bagaimana bersaing dengan kompetitornya sehingga dapat memenangkan pasar.

## BAB VI. PENUTUP

### 6.1. Kesimpulan

Berdasarkan penelitian mengenai analisis faktor kepuasan dan ketidakpuasan pelanggan hotel menggunakan metode *text mining* dan analisis sentimen: *Sentistrength* yang dilakukan pada hotel bintang empat dan lima di DI Yogyakarta ini, dapat disimpulkan bahwa rata-rata pelanggan yaitu sebanyak 86,5% pelanggan yang menggunakan layanan hotel ini sudah merasa puas dengan layanan yang diberikan. Kemudian, disimpulkan juga faktor produk dan layanan merupakan faktor utama yang menyebabkan pelanggan puas maupun tidak puas yang diikuti dengan beberapa faktor-faktor pendukung lainnya. Faktor-faktor pendukung ini yaitu faktor keramahan dan keterampilan staf, fasilitas pendukung, kestrategisan lokasi dan keindahan arsitektur hotelnya. Lalu ada beberapa tambahan pada faktor ketidakpuasan yaitu harga yang ditawarkan, prosedur pemesanan, dan juga arsitektur bangunan hotel.

Pelanggan memiliki ekspektasi dan penghargaan yang tinggi atas faktor produk dan layanan ini. Sehingga ketika menggunakan layanan hotel, fokus utama pelanggan adalah faktor produk dan layanan hotel tersebut. Faktor ini merupakan salah satu *core* dari empat *core* utama bisnis perhotelan yang terkuat dan sangat mempengaruhi sehingga perlu diberi perhatian lebih dari pihak hotel. Pihak hotel dapat memanfaatkan temuan dari penelitian ini yaitu berupa faktor-faktor kepuasan dan ketidakpuasan pelanggan untuk dapat memperbaiki maupun meningkatkan kualitas layanan mereka secara tepat sesuai dengan kebutuhan dan ekspektasi pelanggannya untuk internal hotel. Selain itu, untuk keperluan eksternal pihak hotel dapat menggunakan hasil temuan penelitian yaitu faktor-faktor kepuasan sebagai faktor-faktor kekuatan dan kelebihan hotel dalam melayani pelanggannya sehingga kekuatan ini dapat membangun citra yang positif di masyarakat. Sedangkan untuk faktor-faktor ketidakpuasan, pihak hotel dapat menjadikannya sebagai bentuk janji hotel kepada pelanggan bahwa kelima faktor ini merupakan faktor yang akan diperbaiki dan ditingkatkan

sehingga pelanggan tidak perlu ragu dan khawatir untuk menggunakan layanan dihotel tersebut agar tidak menjadi citra yang negatif tetapi dapat menjadi citra yang positif.

## 6.2.Saran

Setelah penelitian dilakukan, ada beberapa saran yang dapat dilakukan untuk penelitian selanjutnya yang juga meneliti *text mining* khususnya analisis sentimen menggunakan metode *sentistrength*, yaitu sebagai berikut :

1. Analisis sentimen dapat dilakukan pada data ulasan teks dari berbagai *platform* lain termasuk media sosial.
2. Pada bagian *pre-processing*, dapat dilakukan lebih banyak normalisasi kata untuk berbagai kata yang merupakan singkatan, salah ketik, dan sebagainya karena dapat lebih membantu ketika proses pengolahan data.
3. Kamus/Leksikon sentimen Bahasa Indonesia yang digunakan ketika klasifikasi sentimen dapat ditambah dan diperkaya lagi sehingga menghasilkan hasil klasifikasi yang lebih baik lagi.



# LAMPIRAN

## 1. *Source Core Data Scraping (Pengumpulan Data)*

```
# Use pip to install the package
import requests
import pandas
from bs4 import BeautifulSoup
import sys

# Simple GUI
print('Which hotel shall we scrape today?')
uri = input('>>> ')
print()

print('Enter the name of extracted csv:')
csvName = input('>>> ')
print()

print('Okay, let\'s go')
print()

if(('https://www.tripadvisor.' in uri) == False):
    print('Are you sure about that?')
    print('Run me again.')
    sys.exit()

elif(uri == ''):
    print('Okay, maybe later.')
    sys.exit()

elif(csvName == ''):
    print('Please be serious.')
    sys.exit()

# Temp Var
reviewHotelNameList = []
reviewNameList = []
reviewDateList = []
reviewTitleList = []
reviewCommentList = []

print()
print(uri)
print()

#-----
# Request
req = requests.get(uri)
soup = BeautifulSoup(req.text, 'html.parser')

# Check if there was an review or not
if(soup.find('div', {'class': 'hotels-hotel-review-about-noreviews-NoReviewsCTA__noReviewsYet--10L25'}) != None):
```

```

    res = soup.find('div', {'class': 'hotels-hotel-review-about-
noreviews-NoReviewsCTA__noReviewsYet--10L25'})

    if(res.text == 'There are no reviews for this property
yet.'):
        print('No Review, not interest')
        print('Bye')
        sys.exit()

# Get the max page number
if(soup.find_all('a', {'class': 'pageNum
cx_brand_refresh_phase2'}) != []):
    res = soup.find_all('a', {'class': 'pageNum
cx_brand_refresh_phase2'})
    print(res)

    maxPage = int(res[-1].text)

elif(soup.find_all('a', {'class': 'pageNum
cx_brand_refresh_phase2'}) == []):
    if(soup.find('a', {'class': 'pageNum
cx_brand_refresh_phase2'}) != None):
        res = soup.find('a', {'class': 'pageNum
cx_brand_refresh_phase2'})
        print(res)

        maxPage = int(res.text)
    else:
        maxPage = 1

# This is the function
def scrape_data(uri, soup, maxPage):
    # Pagination Page
    currentPage = 0
    # Loop
    while currentPage in range(0, maxPage):

        getHotelName = soup.find('h1', {'class': 'hotels-hotel-
review-atf-info-parts-Heading_heading--2ZOcD'})
        print('Hotel: ' + getHotelName.text)

        # Get Max Review List
        maxReview = soup.find_all('a', {'class': 'ui_header_link
social-member-event-MemberEventOnObjectBlock__member--35-jC'})

        # Pagination Review
        currentReviewPage = 0
        while currentReviewPage in range(0, len(maxReview)):
            # Divider
            print('----- Review -----')
            str(currentReviewPage + 1) + ' -----')

            getReviewName = soup.find_all('a', {'class':
'ui_header_link
social-member-event-
MemberEventOnObjectBlock__member--35-jC'})[currentReviewPage]

```

```

print('Name: ' + getReviewName.text)

getReviewDate = soup.find_all('div', {'class':
'social-member-event-MemberEventOnObjectBlock__event_type--
3njyv'})[currentReviewPage].find()
if('.com' in uri):
    getReviewDateUpdate =
getReviewDate.text.replace(getReviewName.text + ' wrote a review
', '')
elif('.co.id' in uri):
    getReviewDateUpdate =
getReviewDate.text.replace(getReviewName.text + ' menulis ulasan
', '')
print('Date: ' + getReviewDateUpdate)

getReviewTitle = soup.find_all('div', {'class':
'location-review-review-list-parts-ReviewTitle__reviewTitle--
2G09Z'})[currentReviewPage].find('span')
print('Title: ' + getReviewTitle.text)

getReviewComment = soup.find_all('q', {'class':
'location-review-review-list-parts-
ExpandableReview__reviewText--
gOmRC'})[currentReviewPage].find('span')
print('Comment: ' + getReviewComment.text)

# Divider
print('-----')

# Push to csv in every iteration
# It will help you if there was a problem in the
midle of scraping
# But the downside is maybe it will take more
resource
reviewHotelNameList.append(getHotelName.text)
reviewNameList.append(getReviewName.text)
reviewDateList.append(getReviewDateUpdate)
reviewTitleList.append(getReviewTitle.text)
reviewCommentList.append(getReviewComment.text)

createTable = {
    'Hotel Name': reviewHotelNameList,
    'Reviewer Name': reviewNameList,
    'Review Date': reviewDateList,
    'Review Title': reviewTitleList,
    'Review Comment': reviewCommentList
}

dataFrame = pandas.DataFrame(createTable,
columns=['Hotel Name', 'Reviewer Name', 'Review Date', 'Review
Title', 'Review Comment'])
dataFrame.to_csv(r'./' + csvName + '.csv')
# Move to the next review
currentReviewPage += 1

# Move to the next page

```

```

        currentPage += 1

        # Request to next page
        uriMove = uri.replace('Reviews-', 'Reviews-or' +
str(currentPage*5) + '-')
        print(uriMove)
        # Divider
        print('----- Page ' + str(currentPage) + '-----
-----')
        print()
        # Request to next page
        req = requests.get(uriMove)
        soup = BeautifulSoup(req.text, 'html.parser')

# Fetch the data
scrape_data(uri, soup, maxPage)
#END of the Loop

# Change the uri so we can get English(.com) and Bahasa(.co.id)
reviews
if('.com' in uri):
    # Replace the domain to .co.id and fetch
    uri = uri.replace('.com', '.co.id')

    # Space
    print()
    print('Change url to .co.id *Review in Bahasa')
    print(uri)
    print()

    # New Request
    req = requests.get(uri)
    soup = BeautifulSoup(req.text, 'html.parser')

    # Check if there was a review or not
    if(soup.find('div', {'class': 'hotels-hotel-review-about-
noreviews-NoReviewsCTA_noReviewsYet--10L25'}) != None):

        res = soup.find('div', {'class': 'hotels-hotel-review-
about-noreviews-NoReviewsCTA_noReviewsYet--10L25'})

        if(res.text == 'There are no reviews for this property
yet.'):
            print('No Review, not interest')
            print('Bye')
            sys.exit()

    # Get the max page number
    if(soup.find_all('a', {'class': 'pageNum
cx_brand_refresh_phase2'}) != []):
        res = soup.find_all('a', {'class': 'pageNum
cx_brand_refresh_phase2'})
        print(res)

        maxPage = int(res[-1].text)

```

```

        elif(soup.find_all('a', {'class': 'pageNum
cx_brand_refresh_phase2'}) == []):
            if(soup.find('a', {'class': 'pageNum
cx_brand_refresh_phase2'}) != None):
                res = soup.find('a', {'class': 'pageNum
cx_brand_refresh_phase2'})
                print(res)

                maxPage = int(res.text)
            else:
                maxPage = 1

        scrape_data(uri, soup, maxPage)

elif('.co.id' in uri):
    # Replace the domain to .com and fetch
    uri = uri.replace('.co.id', '.com')

    # Space
    print()
    print('Changing to .com *Review in English')
    print(uri)
    print()

    # New Request
    req = requests.get(uri)
    soup = BeautifulSoup(req.text, 'html.parser')

    # Check if there was an review or not
    if(soup.find('div', {'class': 'hotels-hotel-review-about-
noreviews-NoReviewsCTA__noReviewsYet--10L25'}) != None):

        res = soup.find('div', {'class': 'hotels-hotel-review-
about-noreviews-NoReviewsCTA__noReviewsYet--10L25'})

        if(res.text == 'There are no reviews for this property
yet.'):
            print('No Review, not interest')
            print('Bye')
            sys.exit()

        # Get the max page number
        if(soup.find_all('a', {'class': 'pageNum
cx_brand_refresh_phase2'}) != []):
            res = soup.find_all('a', {'class': 'pageNum
cx_brand_refresh_phase2'})
            print(res)

            maxPage = int(res[-1].text)

        elif(soup.find_all('a', {'class': 'pageNum
cx_brand_refresh_phase2'}) == []):
            if(soup.find('a', {'class': 'pageNum
cx_brand_refresh_phase2'}) != None):
                res = soup.find('a', {'class': 'pageNum
cx_brand_refresh_phase2'})

```

```

        print(res)

        maxPage = int(res.text)
    else:
        maxPage = 1

    scrape_data(uri, soup, maxPage)

# Push to the csv for final product
# If the program manage to scrape all the data without problem
createTable = {
    'Hotel Name': reviewHotelNameList,
    'Reviewer Name': reviewNameList,
    'Review Date': reviewDateList,
    'Review Title': reviewTitleList,
    'Review Comment': reviewCommentList
}

dataFrame = pandas.DataFrame(createTable, columns=['Hotel Name',
'Reviewer Name', 'Review Date', 'Review Title', 'Review
Comment'])
# Set the Index from 0 to 1
dataFrame.index += 1
dataFrame.to_csv(r'./' + csvName + '.csv')

print()
print('Mission Complete')

```

## 2. Source Code Pre-Processing Data

```

# Load
library("tm")
library("SnowballC")
library("wordcloud")
library("RColorBrewer")
library("stringr")

setwd("F:/EKSPERIMEN")
docs<-readLines("B4-MentahTranslated.csv")

# Load the data as a corpus
docs <- Corpus(VectorSource(docs))

#Cleaning the text
# Convert the text to lower case
docs <- tm_map(docs, content_transformer(tolower))

#Remove punctuation
docs <- tm_map(docs, toSpace, "[[:punct:]]")

# Remove URL
removeURL <- function(x) gsub("http[[:alnum:]]*", " ", x)
docs <- tm_map(docs, removeURL)

#Replacing "/", "@", and "|" with space:

```

```

toSpace <- content_transformer(function (x , pattern )
gsub(pattern, " ", x))
docs <- tm_map(docs, toSpace, "/" )
docs <- tm_map(docs, toSpace, "@")
docs <- tm_map(docs, toSpace, "\\|")

#Remove numbers
docs <- tm_map(docs, toSpace, "[[:digit:]]")

# add stopwords & remove from corpus
myStopwords = readLines("stopwords.csv")
docs <- tm_map(docs, removeWords, myStopwords)

#Replace words
docs <- tm_map(docs, gsub, pattern="sgt", replacement="sangat")

# Eliminate extra white spaces
docs <- tm_map(docs, stripWhitespace)

# Save Result
dataframe <- data.frame(text = unlist(sapply(docs, `[]`)),
stringsAsFactors = F)
write.csv(dataframe, "F:/EKSPERIMEN/B4-Hasil.csv")

```

### 3. Source Code Sentistrength

```

# coding: utf-8
import re
from collections import OrderedDict
import numpy as np
import pandas as pd
import csv

class sentistrength:

    def __init__(self, config=dict()):
        self.negasi = [line.replace('\n','') for line in
open("negatingword.txt").read().splitlines()]
        self.tanya = [line.replace('\n','') for line in
open("questionword.txt").read().splitlines()]
        #create sentiment words dictionary
        self.sentiwords_txt = [line.replace('\n','').split(":")
for line in open("sentiwords_id.txt").read().splitlines()]
        self.sentiwords_dict = OrderedDict()
        for term in self.sentiwords_txt:
            self.sentiwords_dict[term[0]] = int(term[1])
        #create emoticon dictionary
        self.emoticon_txt = [line.replace('\n','').split(" | ")
for line in open("emoticon_id.txt").read().splitlines()]
        self.emoticon_dict = OrderedDict()
        for term in self.emoticon_txt:
            self.emoticon_dict[term[0]] = int(term[1])
        #create idioms dictionary
        self.idioms_txt = [line.replace('\n','').split(":") for
line in open("idioms_id.txt").read().splitlines()]

```

```

self.idioms_dict = OrderedDict()
for term in self.idioms_txt:
    self.idioms_dict[term[0]] = int(term[1])
#create boosterwords dictionary
self.boosterwords_txt =
[line.replace('\n', '').split(":") for line in
open("boosterwords_id.txt").read().splitlines()]
self.boosterwords_dict = OrderedDict()
for term in self.boosterwords_txt:
    self.boosterwords_dict[term[0]] = int(term[1])
self.negation_conf = config["negation"]
self.booster_conf = config["booster"]
self.ungkapan_conf = config["ungkapan"]
self.consecutive_conf = config["consecutive"]
self.repeated_conf = config["repeated"]
self.emoticon_conf = config["emoticon"]
self.question_conf = config["question"]
self.exclamation_conf = config["exclamation"]
self.punctuation_conf = config["punctuation"]
self.mean_conf = False
self.dfObj = pd.DataFrame(columns = ['Reviews' ,
'Result', 'Max_Pos' , 'Max_Neg', 'Kelas'])

def senti(self,term):
try:
    return self.sentiwords_dict[term]
except:
    return 0

def emosikon(self,term):
try:
    return self.emoticon_dict[term]
except:
    return 0

def ungkapan(self,term):
try:
    return self.idioms_dict[term]
except:
    return 0

def booster(self, term):
try:
    return self.boosterwords_dict[term]
except:
    return 0

def cek_negationword(self, prev_term, prev_term2):
#jika kata sebelumnya (index-1) adalah kata negasi,
negasikan nilai -nya
    if prev_term in self.negasi or prev_term2+" "+prev_term
in self.negasi:
        # print prev_term
        self.score = -abs(self.score) if self.score>0 else
abs(self.score)

```



```

def cek_boosterword(self,term):
    booster_score = self.booster(term)
    if booster_score !=0 and self.score>0: self.score +=
booster_score
    if booster_score !=0 and self.score<0: self.score -=
booster_score

def cek_consecutive_term(self, prev_term):
    if self.prev_score>0 and self.score >=3: self.score+=1
    if self.prev_score<0 and self.score <=-3: self.score-=1

def cek_ungkapan(self, bigram,trigram, i):
    bigram = ' '.join(bigram)
    trigram = ' '.join(trigram)
    ungkapan_score = self.ungkapan(bigram)
    if ungkapan_score==0:
        ungkapan_score = self.ungkapan(trigram)
    if ungkapan_score!=0:
        self.score = ungkapan_score
        self.prev_score = 0
        self.pre_max_pos[i-1] = 1
        self.pre_max_neg[i-1] = -1
        self.max_pos = self.pre_max_pos[i-2] #if
len(self.pre_max_pos)>1 else 1
        self.max_neg = self.pre_max_neg[i-2] #if
len(self.pre_max_neg)>1 else -1
        self.sentence_score[i-1] =
re.sub(r'\[\d\]', '', self.sentence_score[i-1])

def cek_repeated_punctuation(self, next_term):
    if re.search(r'!{2,}',next_term) and self.score >=3:
self.score+=1
    if re.search(r'!{2,}',next_term) and self.score <=-3:
self.score-=1

def remove_extra_repeated_char(self, term):
    return re.sub(r'([A-Za-z])\1{2,}',r'\1',term)
def plural_to_singular(self, term):
    return re.sub(r'([A-Za-z])+\1', r'\1',term)
def classify(self):
    result = "neutral"
    try:
        if self.mean_conf:
            mean_p = np.mean(self.mean_pos)
            mean_n = np.mean(self.mean_neg)
            print (mean_p, mean_n)
            if mean_p > mean_n:
                result = "positive"
            elif mean_p < mean_n and not self.is_tanya:
                result = "negative"
            elif mean_p < mean_n and self.is_tanya:
                result = "neutral"
        else:
            if abs(self.sentences_max_pos) >
abs(self.sentences_max_neg):
                result = "positive"

```

```

        elif abs(self.sentences_max_pos) <
abs(self.sentences_max_neg):
            result = "negative"
        elif abs(self.sentences_max_pos) ==
abs(self.sentences_max_neg):
            result = "neutral"
    except:
        print("error", self.sentences_max_pos,
self.sentences_max_neg)
        return result
    def cek_neutral_term(self, terms, i):
        if terms[i-1] in self.neutral_term or terms[i+1] in
self.neutral_term: self.score=1

    def main(self, sentence):
        self.neutral_term = ['jika', 'kalau']
        sentences = sentence.split('.')
        self.sentences_max_neg = -1
        self.sentences_max_pos = 1
        self.sentences_score = []
        self.sentences_text = []
        for sentence in sentences:
            self.max_neg = -1
            self.max_pos = 1
            self.mean_neg = [1]
            self.mean_pos = [1]
            self.sentence_score=[]
            terms = sentence.split()
            # terms = re.split(r'[\s,.]', sentence)
            terms_length = len(terms)
            self.is_tanya = False
            self.sentence_text = ''
            # print self.max_pos, self.max_neg
            #SEMUA KALIMAT YANG MEMILIKI TANDA SERU MEMILIKI +ve
minimal 2
            if self.exclamation_conf and
re.search('!', sentence): self.max_pos = 2
            self.prev_score = 0
            self.pre_max_pos = []
            self.pre_max_neg = []
            for i, term in enumerate(terms):
                # repeated_term = ''
                is_extra_char = False
                plural = ''
                self.score = 0
                # if re.search(r'[A-Za-z\-.]+', term):
                # print term
                if re.search(r'([A-Za-z])\1{3,}', term):
                    is_extra_char = True
                    # repeated term =term
                term = self.remove_extra_repeated_char(term)
                if re.search(r'([A-Za-z])+\1', term):
                    plural = term
                    term = self.plural_to_singular(term)
                #GET SENTI SCORE#
                self.score = self.senti(term)

```

```

        # print "senti score",term, self.score

        #NEGATION HANDLER#
        if self.negation_conf and self.score !=0 and
i>0:self.cek_negationword(terms[i-1],terms[i-2])
        # print "negation score",term, self.score

        #BOOSTERWORD HANDLER#
        if self.booster_conf and self.score !=0 and i>0
and i<=(terms_length-1):self.cek_boosterword(terms[i-1])
        if self.booster_conf and self.score !=0 and i>=0
and i<(terms_length-1):self.cek_boosterword(terms[i+1])
        # print "booster score",term, self.score

        #IDIOM/UNGKAPAN HANDLER#
        if self.ungkapan_conf and i>0 and
i<=(terms_length-1):self.cek_ungkapan([terms[i-
1],term],[terms[i-2],terms[i-1],term],i)
        # if self.ungkapan_conf and i>=0 and
i<(terms_length-1):self.cek_ungkapan([term,terms[i+1]])
        # print "idiom score",term, self.score

        #CONSECUTIVE SENTIMENT WORD#
        if self.consecutive_conf and i>0 and
i<=(terms_length-1) and self.score
!=0:self.cek_consecutive_term(terms[i-1])
        # print "consecutive score",term, self.score

        #+1 SENTI SCORE IF REPEATED CHAR ON
POSITIVE/NEGATIVE +2 IF NEUTRAL TERM
        if self.repeated_conf and is_extra_char==True
and self.score>0: self.score+=1
        if self.repeated_conf and is_extra_char==True
and self.score<0: self.score-=1
        if self.repeated_conf and is_extra_char==True
and self.score==0: self.score=2
        # print "repeat char score", term, self.score
        if self.punctuation_conf and i>=0 and
i<(terms_length-1): self.cek_repeated_punctuation(terms[i+1])
        # CEK APAKAH TERDAPAT KATA TANYA
        if self.question_conf and (term in self.tanya or
re.search(r'\?',term)):self.is_tanya = True
        # CEK neutral term
        if self.score!=0 and i>1 and i<(terms_length-2):
self.cek_neutral_term(terms,i)
        # if self.score!=0 and i>0 and i<(terms_length-
4): self.cek_neutral_term(terms,i)
        if self.emoticon_conf and self.score==0:
self.score = self.emosikon(term)

        self.prev_score = self.score
        if self.mean_conf and self.score>0:
self.mean_pos.append(self.score)
        if self.mean_conf and self.score<0:
self.mean_neg.append(abs(self.score))
        #GET MAX SCORE +ve/-ve

```

```

        self.max_pos= self.score if self.score >
self.max_pos else self.max_pos
        self.max_neg= self.score if self.score <
self.max_neg else self.max_neg
        #insert score info current term
        self.pre_max_pos.append(self.max_pos)
        self.pre_max_neg.append(self.max_neg)
        # print self.pre_max_pos, self.pre_max_neg
        if plural != '': term = plural
        self.sentence_text += ' {}'.format(term)
        if self.score != 0:term = "{} [{}]".format(term,
self.score)

        self.sentence_score.append(term)

        self.sentences_text.append(self.sentence_text)
        self.sentences_score.append("
".join(self.sentence_score))
        if self.is_tanya:
            self.max_neg = -1
            self.sentences_max_pos = self.max_pos if
self.max_pos > self.sentences_max_pos else
self.sentences_max_pos
            self.sentences_max_neg = self.max_neg if
self.max_neg < self.sentences_max_neg else
self.sentences_max_neg
            # print self.sentences_max_pos,
self.sentences_max_neg
            sentence_result = self.classify()
            self.dfObj = self.dfObj.append({'Reviews' :
self.sentences_text , 'Result' : self.sentences_score, 'Max_Pos'
: self.sentences_max_pos, 'Max_Neg' : self.sentences_max_neg,
'Kelas' : sentence_result} , ignore_index=True)
            print(self.dfObj);
            self.dfObj.to_csv('F:/EKSPERIMEN/Tanpa
Stopwords/Brenda/B5/B5-Hasil(New).csv', header=True,
index=True);
            return {"classified_text":".
".join(self.sentences_score),"tweet_text":".
".join(self.sentences_text),"sentence_score":self.sentences_sco
re,"max_positive":self.sentences_max_pos,"max_negative":self.se
ntences_max_neg,"kelas":sentence_result}
            #return {"classified_text":".
".join(self.sentences_score)}
            # print self.sentences_text
            #return {"classified_text":".
".join(self.sentences_score),"tweet_text":".
".join(self.sentences_text),"sentence_score":self.sentences_sco
re,"max_positive":self.sentences_max_pos,"max_negative":self.se
ntences_max_neg,"kelas":sentence_result}

            #return {"RESULT : " : self.sentences_max_pos +
self.sentences_max_neg,
            # "max_positif": self.sentences_max_pos,
            # "neg_positif": self.sentences_max_neg,
            # "kelas": sentence_result}
            # return {"classified_text":".

```

```

".join(self.sentences_score)}
    #return{sentece_result}

config = dict()
config["negation"] = True
config["booster"] = True
config["ungkapan"] = True
config["consecutive"] = True
config["repeated"] = True
config["emoticon"] = True
config["question"] = True
config["exclamation"] = True
config["punctuation"] = True
senti = sentistrength(config)

list2=[line.strip() for line in open("F:/EKSPERIMEN/Tanpa
Stopwords/Brenda/B5/B5-Bersih(New).csv","r", errors='ignore')];
for p in list2: print(senti.main(p))

```

#### 4. Source Code N-gram

```

setwd("F:/EKSPERIMEN/Tanpa Stopwords/Brenda/B4/T6")
docs <- readLines("B4-FullyPositif.csv")
docs <- Corpus(VectorSource(docs))

dtm <- TermDocumentMatrix(docs)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing = TRUE)
d <- data.frame(word = names(v),freq=v)
head(d, 15)

# Save Result
Dataframe <- data.frame(text = unlist(sapply(v, `(``)),
stringsAsFactors = F)
write.csv(dataframe, "F:/EKSPERIMEN/Tanpa
Stopwords/Brenda/B4/B4-FreqFullyPositif(New).csv")

```

#### 5. Source Code Identifikasi Empat Core Bisnis Perhotelan

```

import re
from collections import OrderedDict
import numpy as np
import pandas as pd
import csv

class sentistrength:

    def __init__(self, config=dict()):
        self.negasi = [line.replace('\n','') for line in
open("negatingword.txt").read().splitlines()]
        self.tanya = [line.replace('\n','') for line in
open("questionword.txt").read().splitlines()]
        #create sentiment words dictionary

```

```

        self.sentiwords_txt = [line.replace('\n','').split(":")
for
                                line
                                in
open("kamus_main_business.txt").read().splitlines()]
        self.sentiwords_dict = OrderedDict()
        for term in self.sentiwords_txt:
            self.sentiwords_dict[term[0]] = int(term[1])
        #create emoticon dictionary
        self.emoticon_txt = [line.replace('\n','').split(" | ")
for line in open("emoticon_id.txt").read().splitlines()]
        self.emoticon_dict = OrderedDict()
        for term in self.emoticon_txt:
            self.emoticon_dict[term[0]] = int(term[1])
        #create idioms dictionary
        self.idioms_txt = [line.replace('\n','').split(":")
for line in open("idioms_id.txt").read().splitlines()]
        self.idioms_dict = OrderedDict()
        for term in self.idioms_txt:
            self.idioms_dict[term[0]] = int(term[1])
        #create boosterwords dictionary
        self.boosterwords_txt =
[line.replace('\n','').split(":")
for line
in
open("boosterwords_id.txt").read().splitlines()]
        self.boosterwords_dict = OrderedDict()
        for term in self.boosterwords_txt:
            self.boosterwords_dict[term[0]] = int(term[1])
        self.negation_conf = config["negation"]
        self.booster_conf = config["booster"]
        self.ungkapan_conf = config["ungkapan"]
        self.consecutive_conf = config["consecutive"]
        self.repeated_conf = config["repeated"]
        self.emoticon_conf = config["emoticon"]
        self.question_conf = config["question"]
        self.exclamation_conf = config["exclamation"]
        self.punctuation_conf = config["punctuation"]
        self.mean_conf = False
        self.dfObj = pd.DataFrame(columns = ['Reviews' ,
'Result', 'Max_Pos' , 'Max_Neg', 'Kelas'])

    def senti(self,term):
        try:
            return self.sentiwords_dict[term]
        except:
            return 0

    def emosikon(self,term):
        try:
            return self.emoticon_dict[term]
        except:
            return 0

    def ungkapan(self,term):
        try:
            return self.idioms_dict[term]
        except:
            return 0

```

```

def booster(self, term):
    try:
        return self.boosterwords_dict[term]
    except:
        return 0

def cek_negationword(self, prev_term, prev_term2):
    #jika kata sebelumnya (index-1) adalah kata negasi,
    negasikan nilai -nya
    if prev_term in self.negasi or prev_term2+" "+prev_term
    in self.negasi:
        # print prev_term
        self.score = -abs(self.score) if self.score>0 else
abs(self.score)

def cek_boosterword(self,term):
    booster_score = self.booster(term)
    if booster_score !=0 and self.score>0: self.score +=
booster_score
    if booster_score !=0 and self.score<0: self.score -=
booster_score

def cek_consecutive_term(self, prev_term):
    if self.prev_score>0 and self.score >=3: self.score+=1
    if self.prev_score<0 and self.score <=-3: self.score-=1

def cek_ungkapan(self, bigram,trigram, i):
    bigram = ' '.join(bigram)
    trigram = ' '.join(trigram)
    ungkapan_score = self.ungkapan(bigram)
    if ungkapan_score==0:
        ungkapan_score = self.ungkapan(trigram)
    if ungkapan_score!=0:
        self.score = ungkapan_score
        self.prev_score = 0
        self.pre_max_pos[i-1] = 1
        self.pre_max_neg[i-1] = -1
        self.max_pos = self.pre_max_pos[i-2] #if
len(self.pre_max_pos)>1 else 1
        self.max_neg = self.pre_max_neg[i-2] #if
len(self.pre_max_neg)>1 else -1
        self.sentence_score[i-1] =
re.sub(r'\[\d\]', '', self.sentence_score[i-1])

def cek_repeated_punctuation(self, next_term):
    if re.search(r'!{2,}',next_term) and self.score >=3:
self.score+=1
    if re.search(r'!{2,}',next_term) and self.score <=-3:
self.score-=1

def remove_extra_repeated_char(self, term):
    return re.sub(r'([A-Za-z])\1{2,}',r'\1',term)
def plural_to_singular(self, term):
    return re.sub(r'([A-Za-z])\1', r'\1',term)
def classify(self):
    result = "neutral"

```

```

try:
    if self.mean_conf:
        mean_p = np.mean(self.mean_pos)
        mean_n = np.mean(self.mean_neg)
        print (mean_p, mean_n)
        if mean_p > mean_n:
            result = "positive"
        elif mean_p < mean_n and not self.is_tanya:
            result = "negative"
        elif mean_p < mean_n and self.is_tanya:
            result = "neutral"
    else:
        if abs(self.sentences_max_pos) >
abs(self.sentences_max_neg):
            result = "positive"
        elif abs(self.sentences_max_pos) <
abs(self.sentences_max_neg):
            result = "negative"
        elif abs(self.sentences_max_pos) ==
abs(self.sentences_max_neg):
            result = "neutral"
    except:
        print("error",self.sentences_max_pos,
self.sentences_max_neg)
        return result
    def cek_neutral_term(self,terms,i):
        if terms[i-1] in self.neutral_term or terms[i+1] in
self.neutral_term: self.score=1

def main(self,sentence):
    self.neutral_term = ['jika','kalau']
    sentences = sentence.split('.')
    self.sentences_max_neg = -1
    self.sentences_max_pos = 1
    self.sentences_score = []
    self.sentences_text = []
    for sentence in sentences:
        self.max_neg = -1
        self.max_pos = 1
        self.mean_neg = [1]
        self.mean_pos = [1]
        self.sentence_score=[]
        terms = sentence.split()
        # terms = re.split(r'[\s,.]',sentence)
        terms_length = len(terms)
        self.is_tanya = False
        self.sentence_text = ''
        # print self.max_pos, self.max_neg
        #SEMUA KALIMAT YANG MEMILIKI TANDA SERU MEMILIKI +ve
minimal 2
        if self.exclamation_conf and
re.search('!',sentence): self.max_pos = 2
        self.prev_score = 0
        self.pre_max_pos = []
        self.pre_max_neg = []
        for i,term in enumerate(terms):

```



```

# repeated_term = ''
is_extra_char = False
plural = ''
self.score = 0
# if re.search(r'[A-Za-z\-.]+',term):
# print term
if re.search(r'([A-Za-z])\1{3,}',term):
    is_extra_char = True
    # repeated_term =term
term = self.remove_extra_repeated_char(term)
if re.search(r'([A-Za-z])\1',term):
    plural = term
    term = self.plural_to_singular(term)
#GET SENTI SCORE#
self.score = self.senti(term)
# print "senti score",term, self.score

#NEGATION HANDLER#
if self.negation_conf and self.score !=0 and
i>0:self.cek_negationword(terms[i-1],terms[i-2])
# print "negation score",term, self.score

#BOOSTERWORD HANDLER#
if self.booster_conf and self.score !=0 and i>0
and i<=(terms_length-1):self.cek_boosterword(terms[i-1])
if self.booster_conf and self.score !=0 and i>=0
and i<(terms_length-1):self.cek_boosterword(terms[i+1])
# print "booster score",term, self.score

#IDIOM/UNGKAPAN HANDLER#
if self.ungkapan_conf and i>0 and
i<=(terms_length-1):self.cek_ungkapan([terms[i-
1],term],[terms[i-2],terms[i-1],term],i)
# if self.ungkapan_conf and i>=0 and
i<(terms_length-1):self.cek_ungkapan([term,terms[i+1]])
# print "idiom score",term, self.score

#CONSECUTIVE SENTIMENT WORD#
if self.consecutive_conf and i>0 and
i<=(terms_length-1) and self.score
!=0:self.cek_consecutive_term(terms[i-1])
# print "consecutive score",term, self.score

#+1 SENTI SCORE IF REPEATED CHAR ON
POSITIVE/NEGATIVE +2 IF NEUTRAL TERM
if self.repeated_conf and is_extra_char==True
and self.score>0: self.score+=1
if self.repeated_conf and is_extra_char==True
and self.score<0: self.score-=1
if self.repeated_conf and is_extra_char==True
and self.score==0: self.score=2
# print "repeat char score", term, self.score
if self.punctuation_conf and i>=0 and
i<(terms_length-1): self.cek_repeated_punctuation(terms[i+1])
# CEK APAKAH TERDAPAT KATA TANYA
if self.question_conf and (term in self.tanya or

```

```

re.search(r'\?',term)):self.is_tanya = True
    # CEK neutral term
    if self.score!=0 and i>1 and i<(terms_length-2):
self.cek_neutral_term(terms,i)
    # if self.score!=0 and i>0 and i<(terms_length-
4): self.cek_neutral_term(terms,i)
    if self.emoticon_conf and self.score==0:
self.score = self.emosikon(term)

        self.prev_score = self.score
        if self.mean_conf and self.score>0:
self.mean_pos.append(self.score)
        if self.mean_conf and self.score<0:
self.mean_neg.append(abs(self.score))
        #GET MAX SCORE +ve/-ve
        self.max_pos= self.score if self.score >
self.max_pos else self.max_pos
        self.max_neg= self.score if self.score <
self.max_neg else self.max_neg
        #insert score info current term
        self.pre_max_pos.append(self.max_pos)
        self.pre_max_neg.append(self.max_neg)
        # print self.pre_max_pos, self.pre_max_neg
        if plural !='': term = plural
        self.sentence_text += ' {}'.format(term)
        if self.score != 0:term = "{} [{}]".format(term,
self.score)
        self.sentence_score.append(term)

        self.sentences_text.append(self.sentence_text)
        self.sentences_score.append("
".join(self.sentence_score))
        if self.is_tanya:
            self.max_neg = -1
            self.sentences_max_pos = self.max_pos if
self.max_pos > self.sentences_max_pos else
self.sentences_max_pos
            self.sentences_max_neg = self.max_neg if
self.max_neg < self.sentences_max_neg else
self.sentences_max_neg
            # print self.sentences_max_pos,
self.sentences_max_neg
            sentence_result = self.classify()
            #self.dfObj = self.dfObj.append({'Reviews' :
self.sentences_text , 'Result' : self.sentences_score, 'Max_Pos'
: self.sentences_max_pos, 'Max_Neg' : self.sentences_max_neg,
'Kelass' : sentence_result} , ignore_index=True)
            self.dfObj = self.dfObj.append({'Reviews' :
self.sentences_text , 'Result' : self.sentences_score, 'Max_Pos'
: self.sentences_max_pos} , ignore_index=True)
            print(self.dfObj);
            self.dfObj.to_csv('F:/EKSPERIMEN/Tanpa
Stopwords/Brenda/B5/B5-NetralAkhir.csv', header=True,
index=True);
            return {"classified_text":".
".join(self.sentences_score),"tweet_text":".

```

```

".join(self.sentences_text), "sentence_score":self.sentences_score, "max_positive":self.sentences_max_pos, "max_negative":self.sentences_max_neg, "kelas":sentence_result}
        #return {"classified_text":".
".join(self.sentences_score)}
        # print self.sentences_text
        #return {"classified_text":".
".join(self.sentences_score), "tweet_text":".
".join(self.sentences_text), "sentence_score":self.sentences_score, "max_positive":self.sentences_max_pos, "max_negative":self.sentences_max_neg, "kelas":sentence_result}

        #return {"RESULT : ": self.sentences_max_pos +
self.sentences_max_neg,
        # "max_positif": self.sentences_max_pos,
        # "neg_positif": self.sentences_max_neg,
        # "kelas": sentence_result}
        # return {"classified_text":".
".join(self.sentences_score)}
        #return{sentence_result}

config = dict()
config["negation"] = False
config["booster"] = False
config["ungkapan"] = False
config["consecutive"] = False
config["repeated"] = False
config["emoticon"] = False
config["question"] = False
config["exclamation"] = False
config["punctuation"] = False
senti = sentistrength(config)

list2 = [line.strip() for line in open("F:/EKSPERIMEN/Tanpa
Stopwords/Brenda/B5/B5-NetralBersih.csv", "r", errors =
'ignore')];
for p in list2: print(senti.main(p))

```

## DAFTAR PUSTAKA

- [1] D. P. DIY, “STATISTIK KEPARIWISATAAN DIY 2018,” 2019. <https://visitingjogja.com/19962/statistik-pariwisata-diy-2018/>.
- [2] D. Affandi, “PENGARUH KUALITAS PELAYANAN TERHADAP KEPUASAN TAMU DI DEPARTEMEN REKREASI SARI ATER HOTEL & RESORT SUBANG Dodi Affandi Sekolah Tinggi Pariwisata ( STP ) Bandung Jalan Dr . Setiabudhi 186 Bandung Jawa Barat,” *BARISTA*, vol. 3, pp. 83–97, 2016.
- [3] C. for Business, “MEASURING CUSTOMER SATISFACTION,” .
- [4] J. Kandampully and D. Suhartanto, “Customer loyalty in the hotel industry: The role of customer satisfaction and image,” *Int. J. Contemp. Hosp. Manag.*, vol. 12, no. 6, pp. 346–351, 2000, doi: 10.1108/09596110010342559.
- [5] D. Kartikasari and F. Rochman, “Pengaruh Kualitas Layanan terhadap Kepuasan dan Kepercayaan di Rumah Sakit Bunda Kandangan Surabaya,” *E-Jurnal Manaj. Univ. Brawijaya*, vol. 3, no. 10, 2014.
- [6] L. Perkins, *The Community Manager’s Playbook: How to Build Brand Awareness and Customer Engagement*. 2015.
- [7] R. B. Vukmir, *The Ultimate Guide to Customer satisfaction*. 2018.
- [8] D. A. Muthia, “Komparasi Algoritma Text Mining Untuk Klasifikasi Review Hotel,” *J. Tek. Komput. AMIK BSI*, vol. IV, no. 2, pp. 69–74, 2018, doi: 10.31294/jtk.v4i2.3461.
- [9] R. Siringoringo, “Text Mining dan Klasterisasi Sentimen Pada Ulasan Produk Toko Online,” *Tek. Inform. Univ. Prima Indones. Medan*, vol. 2, pp. 1–6, 2019.
- [10] P. Antinasari, R. S. Perdana, and M. A. Fauzi, “Analisis Sentimen Tentang Opini Film Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes Dengan Perbaikan Kata Tidak Baku,” *J. Pengemb. Teknol. Inf.*

*dan Ilmu Komput.*, vol. 1, no. 12, pp. 1733–1741, 2017.

- [11] U. Rofiqoh, R. S. Perdana, and M. A. Fauzi, “Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexion Based Feature,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 1, no. 12, pp. 1725–1732, 2017, [Online]. Available: <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/628>.
- [12] R. Wati, “Penerapan Algoritma Genetika Untuk Seleksi Fitur Pada Analisis Sentimen Review Jasa Maskapai Penerbangan Menggunakan Naive Bayes,” *J. Evolusi*, vol. 4, no. 1, pp. 1689–1699, 2016, doi: 10.1017/CBO9781107415324.004.
- [13] D. A. Muthia, “Analisis Sentimen Pada Review Restoran Dengan Teks Bahasa Indonesia Menggunakan Algoritma Naive Bayes,” *Jurnal Ilmu Pengetah. Dan Teknol. Komput.*, vol. 2, no. 2, pp. 39–45, 2017, doi: 10.1515/HUMOR.2006.009.
- [14] A. Taufik, “Optimasi Particle Swarm Optimization Sebagai Seleksi Fitur Pada Analisis Sentimen Review Hotel Berbahasa Indonesia Menggunakan Algoritma Naive Bayes,” *J. Tek. Komput. AMIK BSI*, vol. III, no. 2, pp. 40–47, 2017.
- [15] A. Nugroho, “Analisis Sentimen Pada Media Sosial Twitter Menggunakan Naive Bayes Classifier Dengan Ekstraksi Fitur N-Gram,” *J. Sains Komput. Inform.*, vol. 2, no. 2, pp. 200–209, 2018.
- [16] K. Berezina, A. Bilgihan, C. Cobanoglu, and F. Okumus, “Understanding Satisfied and Dissatisfied Hotel Customers: Text Mining of Online Hotel Reviews,” *J. Hosp. Mark. Manag.*, vol. 25, no. 1, pp. 1–24, 2016, doi: 10.1080/19368623.2015.983631.
- [17] S. Mankad, H. S. Han, J. Goh, and S. Gavirneni, “Understanding online hotel reviews through automated text analysis,” *Serv. Sci.*, vol. 8, no. 2, pp. 124–138, 2016, doi: 10.1287/serv.2016.0126.
- [18] E. Guzman and W. Maalej, “How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews,” *Ieee*, pp. 153–162, 2015,

- [Online]. Available: <https://mast.informatik.uni-hamburg.de/wp-content/uploads/2014/06/FeatureSentiments.pdf>.
- [19] E. G. Kim and S. H. Chun, "Analyzing online car reviews using text mining," *Sustain.*, vol. 11, no. 6, 2019, doi: 10.3390/su11061611.
- [20] A. Tan, "Text Mining : The state of the art and the challenges Concept-based," *Proc. PAKDD 1999 Work.*, pp. 65–70, 2011, doi: 10.1007/978-1-60761-987-1\_19.
- [21] M. J. Kim, K. Ohk, and C. S. Moon, "Trend analysis by using text mining of journal articles regarding consumer policy," *New Phys. Sae Mulli*, vol. 67, no. 5, pp. 555–561, 2017, doi: 10.3938/NPSM.67.555.
- [22] A. Kao and S. Poteet, "Text mining and natural language processing," *ACM SIGKDD Explor. Newsl.*, vol. 7, no. 1, pp. 1–2, 2005, doi: 10.1145/1089815.1089816.
- [23] A. D'Andrea, F. Ferri, P. Grifoni, and T. Guzzo, "Approaches, Tools and Applications for Sentiment Analysis Implementation," *Int. J. Comput. Appl.*, vol. 125, no. 3, pp. 26–33, 2015, doi: 10.5120/ijca2015905866.
- [24] M. Thelwall, "Gender bias in sentiment analysis," *Online Inf. Rev.*, vol. 42, no. 1, pp. 45–57, 2017, doi: 10.1108/OIR-05-2017-0139.
- [25] H. Saif, Y. He, M. Fernandez, and H. Alani, "Contextual semantics for sentiment analysis of Twitter," *Inf. Process. Manag.*, vol. 52, no. 1, pp. 5–19, 2016, doi: 10.1016/j.ipm.2015.01.005.
- [26] D. Vilares, M. Thelwall, and M. A. Alonso, "The megaphone of the people? Spanish SentiStrength for real-time analysis of political tweets," *J. Inf. Sci.*, vol. 41, no. 6, 2015, doi: 10.1177/0165551510000000.
- [27] G. C. C. Cheng Boon Liat, Shaheen Mansori and B. C. Imrie, "Hotel Service Recovery and Service Quality: Influences of Corporate Image and Generational Differences in the Relationship between Customer Satisfaction and Loyalty," *J. Glob. Mark.*, vol. 30, no. 1, pp. 42–51, 2017.