BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Tinjauan Perusahaan adalah untuk mengetahui keadaan perusahaan diantaranya adalah sejarah perusahaan, struktur organisasi serta visi dan misi.

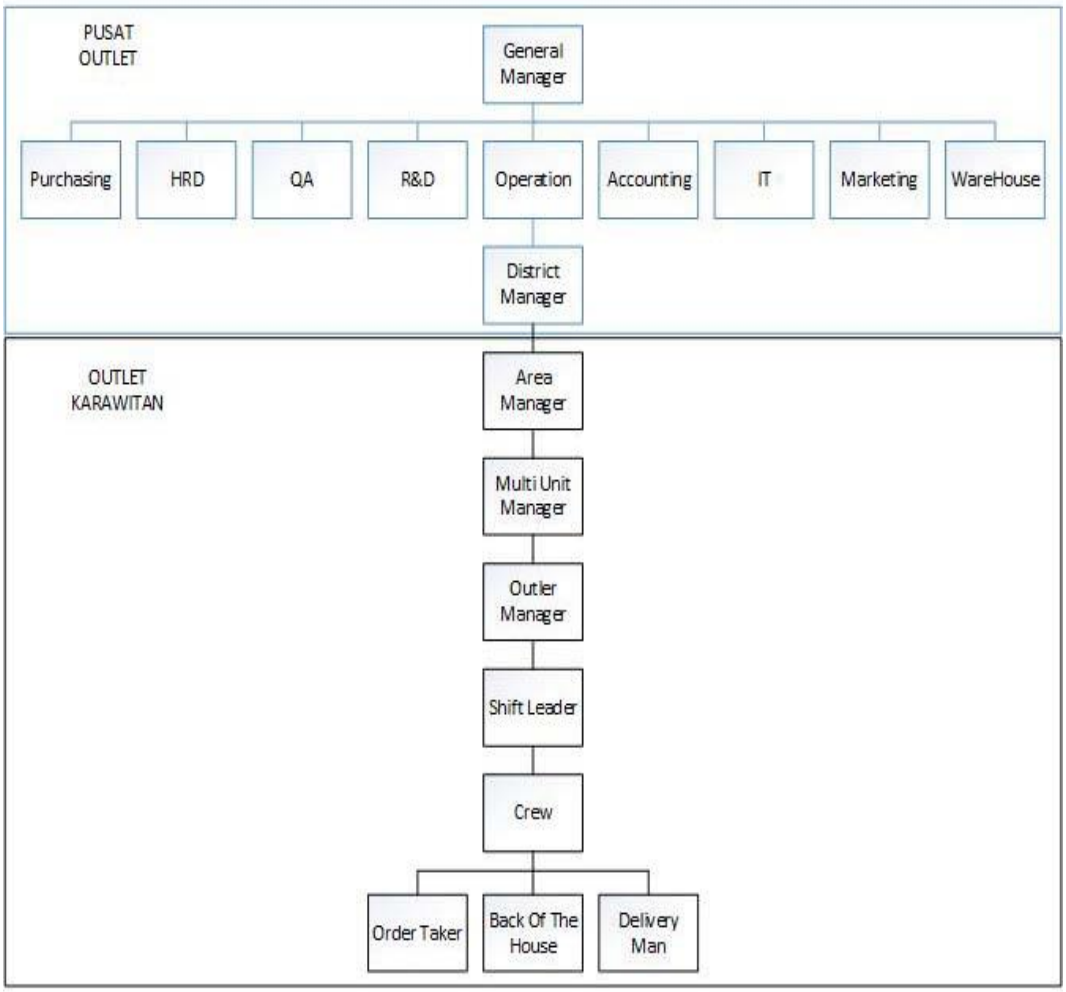
2.1.1 Sejarah Perusahaan

Pizza Hut adalah perusahaan yang bergerak di bidang kuliner, menjual berbagai macam produk seperti pizza, pasta, snack, flatbread dan lain lainya. Pizza Hut hadir di Indonesia untuk pertama kalinya pada tahun 1984, dan merupakan restoran pizza pertama di Indonesia dibawah PT Sarimelati Kencana. Sebagai salah satu produsen makanan siap saji terbesar, *Pizza Hut* memiliki kewajiban untuk meningkatkan pelayanannya. Pelayanan untuk mengantarkan pizza langsung kepada pembeli pun menjadi agendanya. kemudian didirikankanlah oleh perusahaan PT Sari Melati Kencana anak dari perusahaan *Pizza Hut* yaitu *Pizza Hut Delivery* atau yang lebih dikenal dengan PHD.

PHD (*Pizza hut delivery)* adalah anak perusahaan dari pizza hut yang dikelola oleh PT Sari Melati Kencana yang bergerak di bidang kuliner, menjual berbagai macam produk hampir sama dengan pizza hut namun yang membedakan yaitu dari topingan dan menu lainya, serta PHD mengkhususkan pembelian produk dalam sistem *delivery* dengan jaminan tiba di rumah *customer* tidak lebih dari 30 menit. PHD juga menerapkan sistem takeaway yaitu *customer* bisa melakukan pembelian produk langsung datang ke outlet terdekat. PHD untuk memuaskan pelangganya(*customer*) menyediakan tempat untuk berbagi pengalaman konsumen berupa penilaian terhadap outlet PHD, yang opininya dapat disalurkan melalui website www.halopizzaindonesia.com, hasil penilaian opini yang didapat dapat dijadikan bahan evaluasi bagi outlet perusahaan sebagai masukan untuk perusahaan agar tetap maju dan berkembang.

2.1.2 Struktur Organisasi Perusahaan

Struktur organisasi di *Pizza Hut Delivery* dapat dilihat dari gambar 2.1 berikut:



Gambar 2.1 Struktur Organisasi Perusahaan

Sistem organisasi sangat diperlukan untuk pengaturan tugas, tanggung jawab, dan wewenang dalam perusahaan. Berikut ini dijabarkan uraian singkat mengenai struktur organisasi di *Pizza Hut Delivery*.

1. General Manager

Bertanggung jawab atas implementasi kebijakan perusahaan dan memastikan berjalannya peraturan perusahaan serta kesesuaiannya dengan objektif dan strategi perusahaan sesuai target bisnis perusahaan secara menyeluruh.

2. *Purchasing*

Departemen ini bertanggung jawab menyusun sistem pembelian untuk ingredient/bahan baku Ambil dan lokal yang dapat menjamin konsistensi ingredient/bahan baku sesuai dengan spesifikasi, sertifikat halal, dan pemasoknya. Selain itu, sesuai dengan daftar ingredient/bahan baku yang sudah disahkan oleh LPPOM/MUI.

3. *Research and Development (R&D)*

Departemen ini bertanggung jawab dalam mengembangkan atau menciptakan produk baru maupun menyempurnakan produk yang sudah ada, pengawasan kualitas untuk memenuhi kebutuhan pelanggan, menjamin kualitas produk yang dihasilkan sesuai dengan persyaratan pelanggan, serta membantu dan menjamin keberlangsungan semua sistem mutu yang dijalankan perusahaan. Ingredient/bahan baku yang digunakan untuk pengembangan produk baru harus sudah mendapatkan atau dalam proses sertifikasi halal.

4. *Quality Assurance (QA)*

Departemen QA bertanggung jawab dalam melakukan proses registrasi sertifikasi halal, pembuatan resep menu, melakukan pengawasan terhadap sanitasi lingkungan restoran, peralatan dan para karyawan, melakukan evaluasi performance supplier dengan menjalankan *food safety* dan quality system audit, dan melakukan evaluasi performance restoran atau melaksanakan audit internal terkait pengawasan proses produksi.

5. *Human Resource Development (*HRD*)*

Departemen HRD bertanggung jawab terhadap segala sesuatu yang berhubungan dengan recruitment (penerimaan tenaga kerja), pengembangan karyawan, hubungan industrial, meliputi surat peringatan karyawan, sanksi karyawan, pemutusan hubungan kerja (PHK), serta biaya keuangan, seperti penggajian karyawan dan pembiayaan pemeliharaan peralatan dan gedung. Departemen HRD juga bertanggung jawab terhadap penjadwalan kegiatan kantor, pembuatan surat-surat, pemeliharaan gedung kantor, kendaraan, serta kondisi dalam dan luar bangunan. Selain itu, bertanggung jawab terhadap hubungan

dengan pihak luar dan keseluruhan aspek hukum yang berhubungan dengan perusahaan, seperti perizinan dengan pemerintah, sewa tanah dan gedung.

6. Operation

Departemen ini bertanggung jawab melakukan proses produksi sesuai dengan standar yang telah ditetapkan, melakukan proses dokumentasi administrasi keluar masuk ingredient/bahan baku, mengendalikan dan memonitor seluruh sistem di restoran, serta menyusun dan melaksanakan sistem penyimpanan ingredient yang dapat menjamin sistem halal. Dalam pelaksanaan tugasnya, operation terbagi menjadi beberapa bagian, antara lain:

*a. District Manager (*DM*)*

Membuat kebijakan umum di setiap area yang dipimpinnya.

b. *Area Manager* (AM)

Memimpin beberapa outlet pada satu area dan bertanggung jawab tugas dan wewenang kepada District Manager.

c. *Multi Unit Manager* (MUM)

Membawahi 4-5 outlet dalam suatu wilayah. Memeriksa laporan dari *Outlet Manager* berupa laporan prestasi bisnis harian, dan laporan SOS. Laporan – laporan tersebut akan dianalisis oleh *Multi Unit Manager*(MUM)

d. *Outlet Manager* (OM)

Memegang penuh 1 outlet dan menjaga kelancaran aktivitas kerja. Selain itu melakukan pemesanan barang dan bahan baku pada supplier. Tugas lain dari OM adalah memberikan laporan-laporan administrasi kepada MUM dan AM.

e. *Shift Leader* (SL)

Memimpin dan menjamin kelancaran proses kerja pada shift yang dipimpinnya. Selain bertanggung jawab kepada OM tugas dari shift leader adalah melakukan pemesanan barang dan bahan baku pada supplier.

f. *Crew Trainer* (CT)

Memberikan pelatihan pada calon karyawan PHD.

g. Crew

Menjalankan kegiatan operasional sesuai tugasnya. Terdiri dari tiga bagian yaitu:

h. Order Taker

Pegawai yang bertugas menerima order dari customer dan menerima pembayaran. Tugas lainnya adalah menghitung persediaan barang dan bahan baku.

i. Back Of The House

Pegawai yang bertugas menyiapkan dan membuat produk yang dipesan oleh customer. Tugas lainnya adalah menghitung persediaan barang dan bahan baku.

j. Delivery Man

Pegawai yang bertugas mengirim produk yang dipesan oleh customer pada pembilan secara delivery. Tugas lainnya adalah menghitung persediaan barang dan bahan baku.

k. Accounting

Departemen ini bertanggung jawab atas keuangan perusahaan meliputi pengaturan aliran keuangan perusahaan, penyusunan sistem akuntansi, penyusunan anggaran perusahaan, menangani atau memproses masalah perpajakan, penyusunan laporan keuangan, melakukan analisis keuangan, serta melakukan kegiatan pembukuan yang terkait dengan administrasi kantor.

l. *Information Technology* (IT)

Departemen ini bertanggung jawab terhadap berlangsungnya kegiatan informasi baik secara internal maupun eksternal (pihak luar), melalui penyusunan sistem teknologi informasi, pengadaan perangkat komunikasi seperti komputer, hardware, software, dan jaringan (network) internet serta intranet, mengoperasikan dan memelihara infrastruktur IT, meliputi jaringan internet dan intranet, server, dan data center, serta melakukan pengembangan sistem layanan IT, termasuk pengembangan aplikasi sistem informasi.

m. Marketing

Departemen ini bertanggung jawab terhadap keseluruhan kegiatan pemasaran dan penjualan produk-produk perusahaan, menentukan strategi pemasaran dalam pencapaian target penjualan, menetapkan target-target penjualan, terus membina hubungan baik dengan konsumen, distributor maupun pihak advertising agency, dan menerima keluhan dari pelanggan untuk diselesaikan.

n. Warehouse

Departemen ini bertanggung jawab terhadap penerimaan, penyimpanan serta pengeluaran barang. Departemen ini juga bertugas membuat pemesanan barang pada purchasing, menjaga kestabilan keluar masuknya barang, melakukan dokumentasi terkait sistem penggudangan meliputi keluar masuknya barang, menjaga kualitas barang, berkoordinasi dengan divisi lain tentang pemakaian barang, melakukan cek stok fisik barang dengan stock card, serta melakukan penataan barang agar mudah dicari dan diambil.

2.1.3 Visi dan Misi

Visi dan misi Pizza Hut Delivery dirangkum dalam satu kalimat, yaitu "*To be Indonesia's leading mid casual dining restaurant, offering great experience, and the best pizza meal at affordable value*" yang artinya menjadi pelopor kelas menengah kasual di Indonesia yang menawarkan pengalaman luar biasa dan pizza terbaik dengan harga terjangkau. *Pizza Hut Delivery* memiliki visi untuk menjadi yang terunggul pada tingkat restoran kelas menengah di Indonesia yang dicapai lewat misi menawarkan kenyamanan suasana yang terbaik dan menyajikan pizza terbaik dengan harga yang terjangkau.

Pizza Hut juga memiliki nilai-nilai organisasi yang dijadikan sebagai dasar dalam menjalankan organisasi serta membangun relasi dengan pelanggan, mitra usaha, dan pemegang saham. Keempat nilai tersebut antara lain :

1. Integritas, yaitu jujur dalam berpikir dan bekerja, dapat dipercaya, tulus, dan bersikap profesional saat berhubungan dengan rekan kerja, pelanggan, dan para supplier.
2. Keunggulan, yaitu melakukan pekerjaan yang lebih dari sekedar panggilan tugas dan melakukan lebih dari apa yang diharapkan.
3. Pertumbuhan Usaha, yaitu mengembangkan diri dan memperoleh keuntungan dengan cara menjadi "*casual dining restaurant*" yang terbaik. Personil harus berjuang untuk meningkatkan kemampuan dan pengetahuan, berbagai keterampilan dan belajar bersama dengan rekan kerja sehingga bisa berkembang bersama, baik secara individu maupun organisasi.

4. Keuntungan, yaitu sedapat mungkin memberikan keuntungan kepada para pemegang saham dengan pengawasan dan peningkatan usaha penjualan.

2.2 Landasan Teori

Landasan teori ini merupakan bagian yang akan membahas tentang uraian pemecahan masalah yang akan ditemukan pemecahannya melalui pembahasan-pembahasan secara teoritis. Teori-teori yang akan dikemukakan merupakan dasar-dasar penulis untuk meneliti masalah – masalah yang akan dihadapi dalam penelitian. Mulai dari pembahasan mengenai sumber data dimana data itu diperoleh, kemudian penjelasan mengenai tahapan *preprocessing*, setelah data diperoleh kemudian dilakukan proses pembobotan (*Term Weighting)* Penjelasan teori – teori disini berkaitan dengan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan sistem analisis sentimen dengan menggunakan metode K-Nearest Neigbor.

2.2.1 Analisis Sentimen

Analisis sentimen merupakan suatu cara dalam menganalisis pendapat, penilaian, dan sikap terhadap suatu entitas seperti produk, karakter orang, pelayanan, peristiwa dan topik. Salah satu contoh dari pengaplikasian analisis sentimen yaitu suatu perusahaan ingin menilai kinerja perusahaan tersebut dan perusahaan tersebut menyediakan layanan untuk menerima opini dari konsumen mengenai kinerja perusahaan tersebut. Analisis sentimen digunakan untuk mengelompokkan opini positif dan negatif dari customer yang datang kepada perusahaan. Fokus utama dari analisis sentimen adalah untuk menyatakan mana yang termasuk opini positif dan mana yang termasuk opini negatif. Salah satu contoh penggunaan analisis sentimen dalam dunia nyata adalah identifikasi kecenderungan pasar dan opini pasar terhadap suatu objek barang. [4]

Selain mengekstraksi sentimen, hal yang biasa orang ingin tahu adalah, kapan terjadi perubahan sentimen dan apa yang menyebabkan sentimen tersebut berubah. Hal ini menjadi penting, karena dengan mengetahui apa yang menyebabkan sentimen berubah, pihak yang bersangkutan bisa mengambil keputusan dengan lebih baik. Misal saja ketika mengetahui suatu topik atau kejadian menyebabkan sentimen turun, maka pihak yang bersangkutan akan

menghindari kejadian serupa untuk meningkatkan sentimen. Kebutuhan-kebutuhan tersebut biasanya muncul ketika suatu pihak ingin mendapatkan sentimen publik yang baik atau melakukan pencitraan. Kebutuhan seperti ini biasa dimiliki oleh tokoh-tokoh publik, atau lebih khusus lagi tokoh politik seperti calon gubernur, calon presiden, menteri, atau ketua partai. Hal ini juga memungkinkan individu untuk mendapatkan sebuah pandangan tentang sesuatu (review) pada skala global.

2.2.2 *Preprocessing*

Preprocessing merupakan proses menggali, mengolah, mengatur informasi dengan cara menganalisis hubungannya, aturan-aturan yang ada di data tekstual semi terstruktur atau tidak terstruktur. Untuk memudahkan informasi yang diinginkan maka dilakukan langkah transformasi data ke dalam suatu format yang sesuai dengan kebutuhan pemakai. Proses ini disebut *preprocessing* dokumen. Setelah dalam bentuk yang lebih terstruktur dengan adanya proses diatas data dapat dijadikan sumber data yang dapat diolah lebih lanjut. tahapannya terdiri dari *case folding, convert emoticon, tokenizing, stopword removal dan stemming*.

1. Case Folding

Pada tahap ini, semua huruf akan diubah menjadi huruf kecil. langkah-langkah pada tahap *case folding* adalah sebagai berikut :

1. Memeriksa ukuran setiap karakter dari awal sampai akhir karakter.
2. Jika ditemukan karakter yang menggunakan huruf kapital (*uppercase*), maka huruf tersebut akan diubah menjadi huruf kecil (*lowercase*).

2. Convert Negation

Convert Negation merupakan proses konversi kata-kata negasi yang terdapat pada suatu opini, karena kata negasi mempunyai pengaruh dalam merubah nilai sentimen pada suatu tweet. Jika terdapat kata negasi maka akan disatukan dengan kata setelahnya. Kata - kata negasi tersebut meliputi kata “bukan”, “tidak”, “tak”, “ga”,”gak”, “enggak”, “jangan”, dan ”nggak”.

Langkah – langkah pada tahap *convert negation* adalah sebagai berikut :

1. Kata yang digunakan adalah hasil dari *case folding*

2. Jika ditemukan opini yang mengandung kata – kata negasi maka akan disatukan kata negasi tersebut dengan kata setelah kata negasi tersebut.

3. Tokenizing

Tokenizing merupakan tahap pemotongan kalimat berdasarkan tiap kata yang menyusunnya. Proses ini melakukan penguraian deskripsi yang semula berupa kalimat-kalimat menjadi kata-kata dan menghilangkan simbol seperti titik(.), tanda seru(!), tanda tanya (?), koma(,), spasi, *emoticon.*

Langkah-langkah pada tahap *tokenizing* adalah sebagai berikut:

1. Kata yang digunakan adalah hasil dari *convert negation*.
2. Memotong setiap kata dalam kalimat berdasarkan pemisah kata yaitu spasi.
3. Menghilangkan simbol seperti titik(.), tanda seru(!), tanda tanya (?), koma(,), spasi, *emoticon.*

4. Stopword Removal

Stopword didefinisikan sebagai *term* yang tidak berhubungan dengan subyek utama dari *database* meskipun kata tersebut sering kali hadir di dalam dokumen dan kata yang dianggap tidak dapat memberikan pengaruh dalam menentukan suatu kategori sentimen. Kata-kata tersebut dimasukkan kedalam daftar stopword yang biasanya berupa :

1. Kata ganti orang. Hanya dapat digunakan untuk mengganti nomina orang, nama orang, atau hal-hal lain yang dipersonifikasikan. Misalnya : ia, Saudara, Bapak, Ibu, Tuan, Nyonya, Mba, Mr, Mrs, karyawan, karyawati, pegawai dsb
2. Kata ganti penanya. Misalnya : apa, kapan, mengapa, siapa, bagaimana, berapa, di mana, ke mana, di dsb
3. Kata ganti petunjuk. Misalnya : ini, itu dsb
4. Kata ganti penghubung. Misalnya : yang, dan, atau dsb
5. Kata irrelevant. Misalnya : salah satu, karena, sangat, juga, agak, dengan, harus, dari, dgn, dg, yg, oke dsb

Untuk lebih lengkapnya daftar kata *stopword removal* dicatumkan dalam lampiran.

Langkah-langkah pada stopword removal adalah sebagai berikut:

1. Kata hasil tokenizing akan dibandingkan dengan daftar stopword.

Dilakukan pengecekan apakah kata sama dengan daftar stopword atau tidak.

2. Jika kata sama dengan yang ada pada daftar stopword, maka akan dihilangkan.

5. Stemming

Stemming merupakan tahap untuk mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*root word*) dengan menggunakan aturan-aturan tertentu. Dengan menggunakan stemming dapat mengurangi variasi kata yang sebenarnya memiliki kata dasar yang sama. Salah satu algoritma stemming yaitu Algoritma Nazief dan Adriani.

Algoritma stemming Nazief dan Adriani dikembangkan berdasarkan aturan morfologi Bahasa Indonesia yang mengelompokkan imbuhan menjadi awalan (*prefix*), sisipan (*infix*), akhiran (*suffix*) dan gabungan awalan-akhiran (*confixes*). Algoritma ini menggunakan kamus kata dasar dan mendukung recoding, yakni penyusunan kembali kata-kata yang mengalami proses stemming berlebih. Algoritma stemming Nazief dan Adriani menggunakan morfologi imbuhan sebagai berikut :

1. Inflection suffixes merupakan kumpulan akhiran (suffixes) yang tidak merubah kata dasar. Misalnya kata ‘makan’+’-lah’-> ’makanlah’. Inflection suffixes terbagi menjadi :

a. Particles (P) seperti : ‘-lah’ dan ‘-kah’. Contoh : tidurlah, siapakah.

b. Possesive pronouns (PP) seperti : ‘-ku’,’-mu’ dan ‘nya’. Contoh : kataku, katamu, katanya.

2. Derivation suffixes (DS) merupakan kumpulan akhiran (suffixes) yang langsung menempel pada kata dasar, seperti : ‘-i’, ‘-an’ dan ‘kan’. Misalnya kata dasar ‘singkir’ ditambah derivation suffix ‘-kan’ menjadi ‘singkirkan’.

3. Derivation prefixes (DP) merupakan himpunan awalan (prefixes) yang menempel langsung pada kata dasar maupun terhadap kata yang telah mempunyai sampai dua derivation prefixes. Seperti : ‘mem-‘ dan ‘per-‘. Misalnya derivation prefixes ‘mem-’ dan ‘per-’ + ‘kerjakannya’ akan menjadi ‘memperkerjakannya’.

Langkah – langkah *stemming* menggunakan Algoritma Nazief dan Adriani adalah sebagai berikut [5]:

1. Kata yang belum di-stemming dicari pada kamus. Jika kata itu langsung ditemukan, berarti kata tersebut adalah kata dasar. Kata tersebut dikembalikan dan algoritma dihentikan.
2. Hilangkan inflectional suffixes terlebih dahulu. Jika hal ini berhasil dan suffix adalah partikel (“lah” atau ”kah”), langkah ini dilakukan lagi untuk menghilangkan inflectional possessive pronoun suffixes (“ku”, “mu” atau”nya”).
3. Derivational suffix (“-i”, “-an” dan “kan”) kemudian dihilangkan. Lalu langkah ini dilanjutkan lagi untuk mengecek apakah masih ada derivational suffix yang tersisa, jika ada maka dihilangkan. Jika tidak ada lagi maka lakukan langkah selanjutnya.
4. Kemudian derivational prefix (“di-“,”ke-“,”se-“,”te-“,”be-“,”me-“ dan “per-“ “) dihilangkan. Lalu langkah ini dilanjutkan lagi untuk mengecek apakah masih ada derivational prefix yang tersisa, jika ada maka dihilangkan. Jika tidak ada lagi maka lakukan langkah selanjutnya.
5. Setelah tidak ada lagi imbuhan yang tersisa, maka algoritma ini dihentikan kemudian kata dasar tersebut dicari pada kamus, jika kata dasar tersebut ketemu berarti algoritma ini berhasil tapi jika kata dasar tersebut tidak ketemu pada kamus, maka dilakukan recoding.
6. Jika semua langkah telah dilakukan tetapi kata dasar tersebut tidak ditemukan pada kamus juga maka algoritma ini mengembalikan kata yang asli sebelum dilakukan *stemming*.

Proses *stemming* dalam penelitian ini merupakan proses terakhir dari tahap *preprocessing*, setelah opini hasil *preprocessing* sudah dilakukan maka dilakukan pembobotan kata agar opini bisa diklasifikasikan menggunakan metode KNN.

2.2.3 Pembobotan(*Term Weighting*)

Term weighting merupakan tahapan untuk memberikan suatu nilai/bobot pada term yang terdapat pada suatu dokumen yang telah berhasil diekstrak. Metode yang akan digunakan untuk melakukan pembobotan terhadap term adalah pembobotan TF-IDF. Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah cara pemberian bobot hubungan suatu kata (term) terhadap dokumen. Untuk dokumen tunggal tiap kalimat dianggap sebagai dokumen. Metode ini menggabungkan dua

konsep untuk perhitungan bobot, yaitu *Term frequency* (TF) merupakan frekuensi kemunculan kata (t) pada kalimat (d). *Document frequency* (DF) adalah banyaknya kalimat dimana suatu kata (t) muncul. Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen. Pada Metode ini pembobotan kata dalam sebuah dokumen dilakukan dengan mengalikan nilai TF dan IDF. Pembobotan diperoleh berdasarkan jumlah kemunculan term dalam kalimat (TF) dan jumlah kemunculan term pada seluruh kalimat dalam dokumen (IDF).

Nilai IDF sebuah term dihitung menggunakan persamaan di bawah:

$$\mathbf{W}_{d.t} = \mathbf{TF}_{d.t} * \mathbf{IDF}_{t} \text{..... (2.1)}$$

Keterangan :

W = bobot kalimat ke-d d = Kalimat ke-d

TF = *term frequency* t = Kata(term) ke-t

IDF = *inverse document frequency*

Menghitung bobot (W) masing-masing dokumen dengan persamaan di bawah:

$$\mathbf{IDF} = \log\left(\frac{N}{Dfi}\right) \text{..... (2.2)}$$

Keterangan :

IDF = *inverse document frequency*

N = Jumlah kalimat yang berisi term(t)

Dfi = Jumlah kemunculan term terhadap D

Pembobotan kata dilakukan setelah melalui tahap *preprocessing*, nilai dari hasil pembobotan kata maka akan digunakan untuk menghitung nilai kemiripan antar dokumen(*Cosine Similarity*) yang dimana merupakan tahap dalam pengklasifikasian opini menggunakan metode KNN.

Berikut merupakan langkah – langkah dalam pembobotan :

1. Buat Susunan per record kata – kata baik opini data latih ataupun data uji, kata – kata yang muncul lebih dari 1 hanya dituliskan 1x
2. Isi angka 1 apabila katanya muncul di setiap (tf) baik data latih atau data uji, apabila tidak muncul isi angka 0

3. Hitung jumlah Kata – kata yang muncul di semua data latih dan uji(df)
4. Hitung Idf menggunakan persamaan (2.2)

Kalikan masing – masing W dengan masing – masing Tf sesuai dengan pasanganya. Contoh : Tf(i) *W(i)

2.2.4 Klasifikasi

Secara harfiah klasifikasi merupakan penggolongan atau pengelompokkan. Ada beberapa pengertian mengenai klasifikasi, namun jika berbicara berhubungan dengan text mining, klasifikasi yaitu proses pencarian sekumpulan model atau fungsi yang menggambarkan dan membedakan kelas data,yang tujuanya untuk memprediksi kelas dari suatu obyek yang belum diketahui kelasnya. Ada 2 proses klasifikasi yaitu membangun model klasifikasi dari sekumpulan kelas data yang sudah didefinisikan sebelumnya atau biasa disebut data training set dan menggunakan model tersebut untuk klasifikasi tes data serta mengukur akurasi dari model. Klasifikasi dapat dimanfaatkan dalam berbagai aplikasi seperti diagnosa medis, selective marketing, pengajuan kredit perbankan, email dan analisis sentimen. Model klasifikasi dapat disajikan dalam berbagai macam model klasifikasi seperti *decision trees, naïve bayes classifier, k-nearest neighbor classifier, neural network* dan lain-lain

2.2.4.1. *K-Nearest Neighbor*

K-Nearest Neighbor (KNN) adalah suatu metode yang menggunakan algoritma supervised dimana hasil dari query instance yang baru diklasifikan berdasarkan mayoritas dari kategori pada KNN. Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru bedasarkan atribut dan training sample. Classifier tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik query, akan ditemukan sejumlah k obyek atau (titik training) yang paling dekat dengan titik query. Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari k obyek.. algoritma KNN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari query instance yang baru.

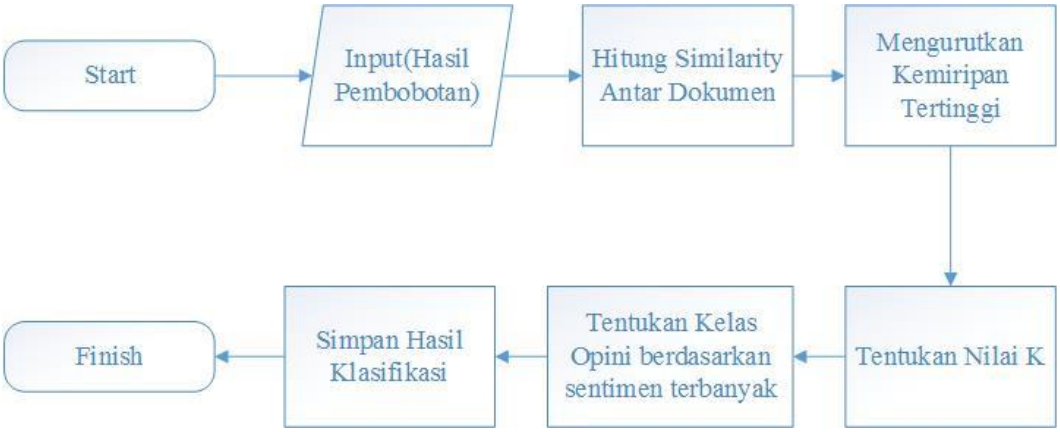
Algoritma metode KNN sangatlah sederhana, bekerja berdasarkan jarak terpendek dari query instance ke *training sample* untuk menentukan KNN-nya. Training sample diproyeksikan ke ruang berdimensi banyak, dimana masing-

masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi *training sample*. Pada *fase training*, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data training sample. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk testing data (yang klasifikasinya tidak diketahui). Jarak dari vektor baru yang ini terhadap seluruh vektor training sample dihitung dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Nilai k yang terbaik untuk algoritma ini tergantung pada data *training*. k dipilih dengan nilai ganjil akan mendapatkan hasil yang lebih baik dibandingkan dengan k dengan nilai genap, karena akan relevan terhadap kasus, dengan kategori yang sama nilai jumlahnya. nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Ketepatan algoritma KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik.

KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap training data yang memiliki banyak noise dan efektif apabila training data-nya besar. Sedangkan, kelemahan KNN adalah KNN perlu menentukan nilai dari parameter k (jumlah dari tetangga terdekat), training berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap *query instance* pada keseluruhan *training sample*. [6]

Adapun algortima k-NN dapat digambarkan dalam diagram alir K-*Nearest Neighbor* sebagai berikut.



Gambar 2.2 Diagram alir K-Nearest Neighbor

1. Hitung jarak antara data sampel (data uji) dengan data latih yang telah dibangun. Salah satu persamaan dalam menghitung jarak kedekatan dapat menggunakan persamaan 2.3 *Cosine Similarity*.
2. Menentukan parameter nilai k = jumlah tetanggaan terdekat bebas.
3. Mengurutkan jarak terkecil dari data uji
4. Pasangkan kategori sesuai dengan kesesuaian
5. Cari jumlah terbanyak dari tetanggaan terdekat Kemudian tetapkan kategori.

Jarak yang digunakan dalam penelitian ini adalah *Cosine Similarity.* [7]

$$Cos(i,k) = \frac{\sum_k (d_i d_k)}{\sqrt{\sum_k d^2{}_{ik}}\sqrt{\sum_k d^2{}_{jk}}} \text{ ..... (2.3)}$$

Keterangan :

$\sum_k (d_i d_k)$: Vector dot produk dari i dan k

$\sqrt{\sum_k d^2{}_{ik}}$: Panjang vector i

$\sqrt{\sum_k d^2{}_{jk}}$: Panjang vector k

i : data uji ke-i

k : data latih ke-j

2.2.5 *Confusion Matrix*

Confusion Matrix [8] berisi informasi tentang klasifikasi aktual dan yang telah diprediksi yang dilakukan oleh sistem klasifikasi. Kinerja sistem tersebut

umumnya dievaluasi dengan menggunakan data dalam matriks. Tabel berikut menunjukkan confusion matrix untuk klasifikasi dua kelas.

Metode ini menggunakan tabel matriks seperti pada Tabel 2.3 jika data set hanya terdiri dari dua kelas, kelas yang satu dianggap sebagai puas dan yang lainnya tidak puas.

Tabel 2.1 Confusion Matrix

		Diklasifikasikan sebagai (*Predicted Class*)	
		Puas	Tidak Puas
Klasifikasi yang benar (*Actual Class*)	Puas	True Puas	False Tidak Puas
	Tidak Puas	False Puas	True Tidak Puas

True puas adalah jumlah *record* puas yang diklasifikasikan sebagai puas, *false* puas adalah jumlah *record* puas yang diklasifikasikan sebagai tidak puas, *false* tidak puas adalah jumlah *record* tidak puas yang diklasifikasikan sebagai puas, *true* tidak puas adalah jumlah *record* tidak puas yang diklasifikasikan sebagai tidak puas. Setiap kolom dari confusion matrix merupakan contoh di kelas yang telah diprediksi, sedangkan setiap baris mewakili contoh di kelas yang sebenarnya. Setelah didapat *true* puas, *false* tidak puas, *true* tidak puas dan *false* tidak puas, selanjutnya hitung untuk menghitung nilai akurasinya. Berikut persamaan dari akurasi.

$$\text{Akurasi} = \frac{TruePuas + TrueTidakPuas}{TruePuas + FalsePuas + TrueTidakPuas + FalseTidakPuas} \quad .....(2.4)$$

2.2.6. K-Fold Cross-Validation

Dalam pendekatan *K-fold cross-validation*, setiap record digunakan beberapa kali dalam jumlah yang sama untuk training dan tepat sekali untuk *testing*. Untuk mengilustrasikan metode ini, anggaplah seperti mempartisi data ke dalam dua

subset yang berukuran sama. Pertama, pilih satu dari kedua subset tersebut untuk training dan satu lagi untuk testing. Kemudian dilakukan pertukaran fungsi dari subset sedemikian sehingga subset yang sebelumnya sebagai training set menjadi test set demikian sebaliknya. Pendekatan ini dinamakan *two-fold cross validation*. Total error diperoleh dengan menjumlahkan error –error untuk kedua proses tersebut.

Setiap record digunakan tepat satu kali untuk training dan satu kali untuk testing. Metode *k-fold cross-validation* mengeneralisasi pendekatan ini dengan mensegmentasi data ke dalam k partisi berukuran sama. Selama proses, salah satu dari partisi dipilih untuk testing, sedangkan sisanya digunakan untuk training. Prosedur ini diulangi k kali sedemikian sehingga setiap partisi digunakan untuk testing tepat satu kali. Total error ditentukan dengan menjumlahkan error untuk semua k proses tersebut [9].

Metode *k-fold cross-validation* menetapkan k = N, ukuran dari data set. Metode ini dinamakan pendekatan *leave-one-out*, setiap test set hanya mengandung satu record. Pendekatan ini memiliki keuntungan dalam pengunaan sebanyak mungkin data untuk training. *Test set* bersifat *mutually exclusive* dan secara efektif mencakup keseluruhan data set. Kekurangan dari pendekatan ini adalah banyaknya komputasi untuk mengulangi prosedur sebanyak N kali.

K-fold cross-validation adalah salah satu teknik untuk mengevaluasi keakuratan model, dengan ciri-ciri [10]:

1. Mempartisi data secara random ke dalam k buah himpunan/*fold* yaitu D1, D2, ..Dk. Setiap kelompok mempunyai jumlah yang hampir sama.
2. Pada perulangan i, gunakan Di sebagai data uji dan himpunan lainnya sebagai data pelatihan Contoh :
    a. Pada perulangan ke-1 : D1 sebagai data uji dan D2 s.d .Dk sebagai data pelatihan
    b. Pada perulangan ke-2 : D2 sebagai data uji dan D1, D3 s.d. Dk sebagai data pelatihanan dan seterusnya
c. Melakukan training dan pengujian sebanyak k kali
d. Menghitung keakuratan dengan rumus.

Sebagai gambaran jika kita melakukan 3-*Fold Cross-Validation* maka desain data eksperimenya adalah sebagai berikut :

Dataset :

Tabel 2.2 Data Set

K1	K2	K3

Data Subset :

Tabel 2.3 Data Subset

Eksperimen ke	Training	Testing
1.	K2, K3	K1
2.	K1, K3	K2
3.	K1, K2	K3

Untuk mendapatkan nilai akurasi ataupun ukuran penilaian lainnya dari hasil subset yang kita lakukan, dapat diambil nilai rataan dari seluruh subset tersebut.

2.3. Object Oriented Programming

OOP (*Object Oriented Programming*) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP yaitu untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, dan objek merupakan gabungan dari beberapa objek yang lebih kecil. Adapun konsep pada pemrograman berorientasi objek ini yaitu:

1. *Class*

Class yaitu penggambaran dari sebuah objek atau benda, sifat objek, dan juga apa yang bisa dilakukan oleh objek tersebut.

2. *Object*

Object adalah suatu bentuk dari class, dimana strukturnya menggabungkan data dan prosedur untuk bekerja bersama-sama.

3. Abstraction

Abstarction adalah suatu cara untuk melihat suatu object dalam bentuk yang lebih sederhana. Dengan Abstraction, suatu sistem yang kompleks dapat dipandang sebagai kumpulan subsistem-subsistem yang lebih sederhana,

4. Encapsulation

Encapsulation yaitu membungkus suatu objek sehingga pada saat kita akan memakai objek tersebut kita tidak perlu lagi tahu tentang detail bagaimana suatu action itu terjadi.

5. Polymorphisms

Polymorphisms merupakan suatu object bisa bertindak lain terhadap *message/method* yang sama. Dalam OOP, diterapkan tipe polymorphism melalui proses yang disebut overloading. Dapat dilakukan dalam implementasi metode yang berbeda pada sebuah object yang mempunyai nama yang sama.

6. Inheritance

Inheritance yaitu salah satu bentuk teknis untuk membuat sebuah class lebih spesifik lagi. Penggunaan inheritance dalam OOP untuk mengklasifikasikan objects dalam program sesuai karakteristik umum dan fungsinya. Hal ini akan membuat pekerjaan bersama object lebih mudah dan lebih intuitif. Hal ini juga membuat programming lebih mudah karena memungkinkan untuk mengkombinasikan karakteristik umum kedalam object parent dan mewariskan karakteristik ini ke *child object.*

2.4. Unified Modeling Languange

UML adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). Diagram – diagram yang ada pada di UML diantaranya adalah sebagai berikut :

1. *Use Case Diagram*

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk use case dan actor. Actor merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. Fungsi *Use case diagram* yaitu Menjelaskan fasilitas – fasilitas yang ada sesuai dengan kebutuhan.

2. *Activity Diagram*

Activity diagram adalah proses yang menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, dan gambaran – gambaran kemungkinan yang terjadi, dan bagaimana alir berakhir.

3. *Sequence Diagram*

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case. Interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Fungsi *sequence diagram* untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

4. *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.