BAB II
LANDASAN TEORI

2.1. RSUD Majalengka

Rumah Sakit Umum Daerah Majalengka berdiri sekitar tahun 1911. Kemudian pada tahun 1962 seorang berkebangsaan Jerman, dr. Heinz Time Germal tercatat menjadi Direktur pertama Rumah Sakit Majalengka. Pada tahun 1965 dr. Heinz Time Germal kembali ke negerinya dan untuk sementara pimpinan Rumah Sakit Majalengka dipegang oleh Kepala Dinas Kesehatan Dati II Majalengka, yaitu dr. le Tiong Bie atau dr. Iwan Satibi. Pada saat itu, Rumah Sakit Umum Daerah Majalengka merupakan Rumah Sakit kelas D.

Pada tahun 1966 dr. Dadang Sulaeman Rusydi yang berasal dari Tasikmalaya diangkat menjadi Direktur RSUD Majalengka yang kedua dan beliau memimpin sampai tahun 1972. Setelah kepemimpinan dr. Sulaeman Rusyidi berakhir, kursi kepemimpinan diganti oleh dr. Muchyidin Hanafi Sutisna Sanjaya yang pada waktu itu juga menjabat sebagai Kepala Dinas Kesehatan Dati II Majalengka menggantikan dr. Iwan Satibi yang pensiun tahun 1980. Pada tahun 1984, dr. Muchyidin Hanafi Sutisna Sanjaya digantikan oleh Direktur penggantinya, yaitu dr. Gufron Amali yang memimpin sampai dengan tahun 1997.

Pada tahun 1988 di masa kepemimpinan beliau, dengan Surat Keputusan Menkes Nomor. 105/Menkes/SK/II/tahun 1988 RSUD Majalengka ditetapkan sebagai Rumah Sakit kelas C. Selanjutnya ditetapkan menerapkan Pola Pengelolaan Keuangan Badan Layanan Umum Daerah status BLUD Penuh dengan Keputusan Bupati nomor 48 tahun 2010 tanggal 22 Februari 2010 tentang Pengelolaan Keuangan Badan Layanan Umum Daerah.

Pada periode selanjutnya, pimpinan dipegang sementara oleh dr. H. Endang Suhenda sebagai pelaksana harian selama satu tahun. Pada pertengahan tahun 1997 dilantik Direktur definitif, yaitu dr. Irama Nirwana Lubis, M.Kes dan berakhir tahun 1998 karena beliau meninggal dunia selama masih menjabat dan selanjutnya Direktur dijabat lagi oleh dr. Endang Suhenda sebagai pelaksana teknis (PLT) yang berakhir sampai bulan Maret 2001, karena pada bulan tersebut dilantik lagi Direktur

definitif, yaitu dr. H. Maman Suparman Gani, MARS. Kepemimpinan dr. H. Maman Suparman Gani, MARS berakhir sampai tahun 2003 sehubungan dengan adanya mutasi Direktur karena dr. H. Maman Suparman Gani, MARS menjadi Direktur RSUD Cideres dan sebagai penggantinya adalah dr. H. Eddy Rudianto M.Kes yang semula menjadi Direktur RSUD Cideres.

Kepemimpinan dr. H. Eddy Rudianto, M.Kes berakhir pada tahun 2006 karena beliau memasuki masa pensiun dan kemudian digantikan oleh dr. H. Hamdi, M.Kes. Kepemimpinan beliau juga berakhir sampai pensiun yaitu tanggal 1 September 2008. yang langsung digantikan secara definitif oleh dr. H. Asep Suandi, M.Epid. Pada ahir tahun 2008 seiring adanya penggantian Bupati dari Bupati Tuti Hayati Anwar, SH, M.Si kepada H. Sutrisno, SE M.Si terjadi mutasi kepala SKPD pada bulan Februari Tahun 2009, dr. H. Asep Suandi. M.Epid di Mutasi ke RSUD Cideres dan RSUD Majalengka dijabat oleh H. Alimudin, S.Sos., MM terhitung tanggal 25 Februari 2009.

Saat ini RSUD Kabupaten Majalengka telah berkembang menjadi 205 Tempat Tidur, dengan 9 jenis pelayanan spesialis, memiliki unggulan dalam pelayanan KIA/Obstetri Ginekologi dan pelayanan pengobatan infeksi, direncanakan pada tahun 2018 akan meningkat menjadi kelas B agar dapat memenuhi kebutuhan pelayanan kesehatan rujukan bagi masyarakat Kabupaten Majalengka.

2.1.1. Logo Instansi

Logo berasal dari Bahasa Yunani yaitu Logos, yang berarti kata, pikiran, pembicaraan, akal budi. Sementara itu kata instansi dalam KBBI diartikan sebagai badan pemerintah umum. Suarianto Rustan mengemukakan bahwa logo bisa menggunakan elemen apa saja, berupa tulisan, logogram, gambar, ilustrasi, dan lain-lain (Rustan, 2009:12). Selanjutnya, Sularko, dkk. dalam buku *"How Do They Think"*, mengemukakan bahwa logo atau *corporate identity* atau *brand identity* adalah sebuah tanda yang secara langsung tidak menjual, tetapi memberi suatu identitas yang pada akhirnya sebagai alat pemasaran yang signifikan, bahwa logo mampu membantu membedakan suatu produk atau jasa dari kompetitornya (Sularko, et al., 2008:6). Suatu logo diperoleh maknanya dari suatu kualitas yang

disimbolkan, melalui pendekatan budaya perusahaan atau instansi serta historis atau aspirasi perusahaan. Secara keseluruhan logo merupakan instrumen rasa harga diri dan nilai-nilainya mampu mewujudkan citra positif dan dapat dipercaya. Adapun berikut ini merupakan logo dari RSUD Majalengka.



Gambar 2.1 Logo RSUD Majalengka

Adapun arti dan makna dari logo RSUD Majalengka pada Gambar 2.1 adalah sebagai berikut:

1. Huruf kapital M berwarna putih

   Huruf kapital M yang digambarkan seperti aliran air melambangkan nama Kabupaten Majalengka yang berarti RSUD Majalengka memberikan pelayanan yang berkesinambungan dan paripurna seperti air yang mengalir menyejukan untuk warga Majalengka dan sekitarnya.

2. Palang berwarna hijau

   Palang berwarna hijau merupakan lambang dari pelayanan kesehatan.

2.1.2. Visi dan Misi

Menurut Wibisono visi adalah serangkaian kata-kata bahkan rangkaian kalimat mengungkapkan impian, cita-cita, rencana, harapan sebuah perkumpulan, perusahaan, organisasi yang ingin dicapai di masa mendatang (Wibisono, 2006:43). Visi juga dapat dikatakan sebagai sesuatu yang sangat dibutuhkan untuk organisasi demi menjamin kesuksesan dan kelestarian organisasi atau perusahaan jangka panjang. Sementara misi merupakan sesuatu yang menentukan kebutuhan apa yang diinginkan oleh perusahaan dimana mereka berada, sekaligus berupaya dalam memuaskan pelanggannya (Prasetyo dan Benedicta, 2004:8).

Visi dari RSUD Majalengka adalah “Menjadi Rumah Sakit Terpercaya dan Pilihan Utama di Kabupaten Majalengka Tahun 2018”. Makna terpercaya dari visi tersebut adalah pelayanan rumah sakit yang prima, profesional, dan bermutu dengan mengutamakan pelayanan kepada masyarakat sehingga masyarakat/pelanggan mempercayakan pelayanan kesehatannya kepada RSUD Majalengka, dan pada akhirnya akan menjadi pilihan utama dari masyarakat Majalengka.

Sejalan dengan visi RSUD Majalengka, maka dirumuskan misi yang pada prinsipnya lebih bersifat tujuan jangka panjang dari suatu organisasi dan berfungsi memberikan tuntunan yang teguh dalam pengambilan keputusan. Dengan dasar pemikiran tersebut, maka RSUD Majalengka merumuskan misi yang sesuai dengan tugas pokoknya sesuai Peraturan Daerah Kabupaten Majalengka Nomor 10 Tahun 2009 yaitu:

“Melaksanakan upaya kesehatan secara berdaya guna dan berhasil guna dengan mengutamakan upaya penyembuhan dan pemulihan yang dilakukan secara serasi, terpadu dengan upaya peningkatan, pencegahan dan melaksanakan upaya rujukan, serta melaksanakan pelayanan yang bermutu sesuai standar pelayanan Rumah Sakit dalam rangka meningkatkan IPM Kabupaten Majalengka Tahun 2018”.

2.1.3. Lokasi Bisnis

Rumah Sakit Umum Daerah Majalengka berlokasi di Jl. Kesehatan No. 77, Kelurahan Majalengka Wetan, Kecamatan Majalengka, Kabupaten Majalengka. Adapun batas-batas lahan yang dimiliki RSUD Majalengka yaitu :

Sebelah utara : Pemakaman umum
Sebelah selatan : Jl. Kesehatan (Permukiman Penduduk)
Sebelah timur : Permukiman penduduk
Sebelah barat : Gg. Marjali (Permukiman Penduduk)

Dengan luas lahan 41.240 m2 *catchment area* RSUD Majalengka meliputi jangkauan wilayah III Cirebon, Kuningan, Indramayu, Sumedang, Ciamis, dan Tasikmalaya. Adapun akses pelayanan tersebut meliputi:

Sebelah utara mencapai jangkauan wilayah Kecamatan Dawuan, Jatiwangi, serta wilayah perbatasan Kabupaten Indramayu dan Kabupaten Cirebon.

Sebelah selatan mencapai jangkauan wilayah Kecamatan Maja, Talaga, Cikijing dan perbatasan dengan Kabupaten Ciamis.

Sebelah Barat mencapai jangkauan wilayah Kecamatan Panyingkiran, Kadipaten, Jatutujuh, serta wilayah perbatasan dengan Kabupaten Sumedang dan Kabupaten Indramayu.

Sebelah Timur mencapai jangkauan wilayah Kecamatan Cigasong, Rajagaluh, Sindangwangi serta wilayah perbatasan dengan Kabupaten Cirebon dan Kabupaten Kuningan.

2.1.4. Tugas Pokok dan Fungsi

Berdasarkan Undang-Undang Rumah Sakit Bab III pasal 4 Rumah Sakit mempunyai tugas memberikan pelayanan kesehatan perorangan secara paripurna. Sedangkan menurut Peraturan Daerah Kabupaten Majalengka Nomor 10 Tahun 200 Tanggal 1 Desember tahun 2009 tentang Organisasi perangkat Daerah Kabupaten Majalengka. Rumah Sakit Umum Daerah Majalengka mempunyai tugas pokok melaksanakan upaya kesehatan secara berdaya guna dan berhasil guna dengan mengutamakan upaya penyembuhan, pemulihan yang dilakukan secara serasi, terpadu dengan upaya peningkatan dan pencegahan serta melaksanakan upaya rujukan, melaksanakan pelayanan yang bermutu sesuai standar pelayanan Rumah Sakit.

Untuk menjalankan tugas sebagaimana dimaksud dalam Undang-Undang Rumah Sakit Bab III Pasal 4, Rumah Sakit mempunyai fungsi:

a. Penyelenggaraan pelayanan pengobatan dan pemulihan kesehatan sesuai dengan standar pelayanan rumah sakit;
b. Pemeliharaan dan peningkatan kesehatan perorangan melalui pelayanan kesehatan yang paripurna tingkat kedua dan ketiga sesuai kebutuhan medis;
c. Penyelenggaraan pendidikan dan pelatihan sumber daya manusia dalam rangka peningkatan kemampuan dalam pemberian pelayanan kesehatan; dan

d. Penyelenggaraan penelitian dan pengembangan serta penapisan teknologi bidang kesehatan dalam rangka peningkatan pelayanan kesehatan dengan memperhatikan etika ilmu pengetahuan bidang kesehatan.

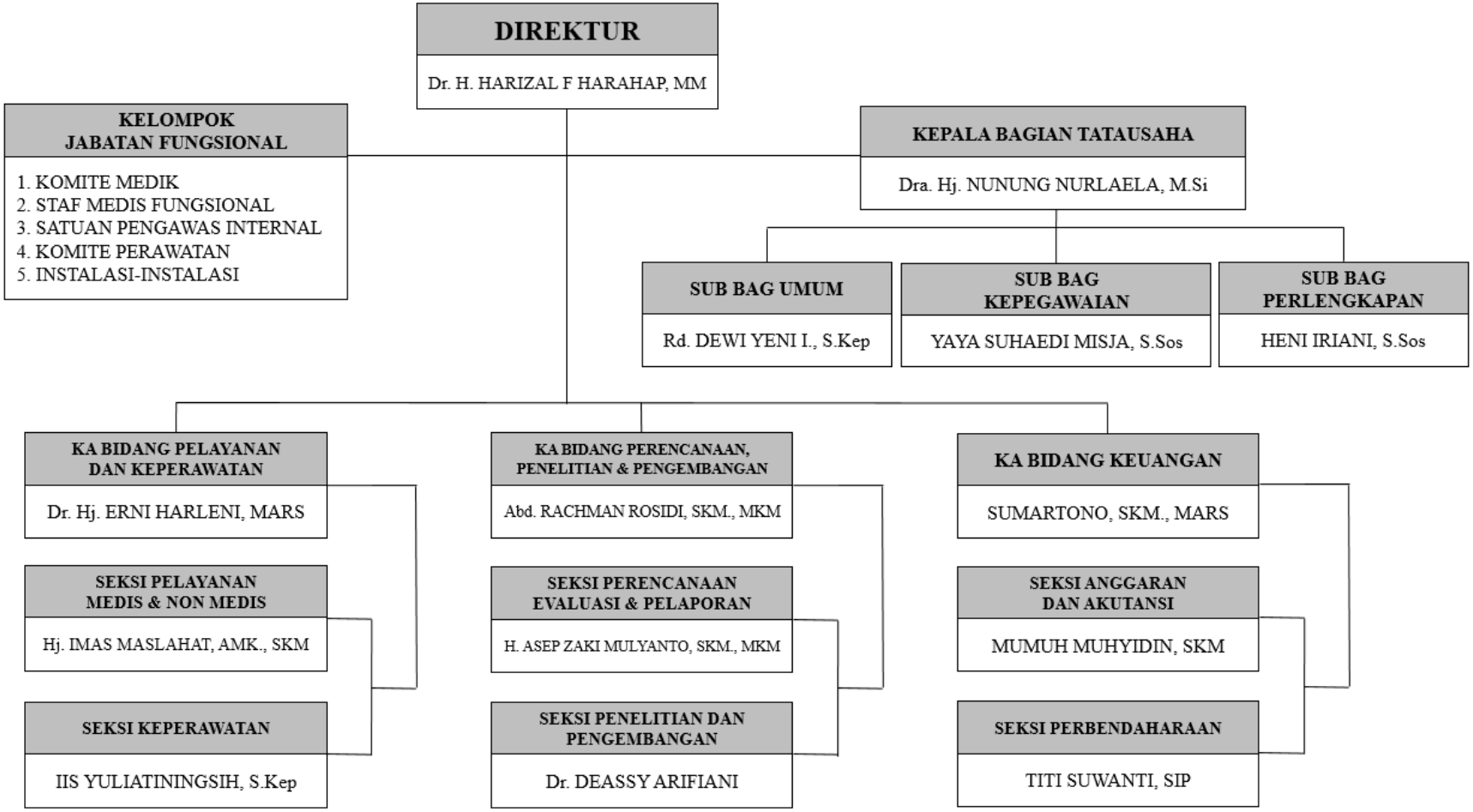
Adapun Rumah Sakit Umum Daerah Majalengka menyelenggarakan fungsi sebagai berikut:

a. Pelayanan Medis;
b. Penunjang medis dan non medis;
c. Pelayanan dan asuhan keperawatan;
d. Pelayanan rujukan;
e. Peningkatan Sumber Daya Manusia (SDM);
f. Penelitian dan pengembangan;
g. Pelayanan administrasi umum dan keuangan.

2.1.5. Struktur Organisasi

Struktur organisasi didefinisikan sebagai mekanisme-mekanisme formal dengan bagaimana organisasi dikelola (Handoko, 2003:169). Sedangkan menurut Robbins dan Coulter struktur organisasi dapat diartikan sebagai kerangka kerja formal organisasi yang dengan kerangka kerja itu tugas-tugas pekerjaan dibagi-bagi, dikelompokkan, dan dikoordinasikan (Robbins dan Coulter, 2007:284).

Berdasarkan Peraturan Daerah Kabupaten Majalengka Nomor 10 Tahun 2009 Tanggal 1 Desember tahun 2009 Tentang Organisasi perangkat Daerah Kabupaten Majalengka Susunan Organisasi RSUD Majalengka dapat dilihat pada Gambar 2.2 berikut ini.



Gambar 2.2 Struktur Organisasi RSUD Majalengka

2.2. Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan perangkat lunak. Berikut adalah beberapa konsep dasar dan teori terkait:

2.2.1. *Machine Learning*

Machine learning merupakan kemampuan komputer untuk melakukan pembelajaran tanpa harus menjelaskan secara eksplisit kepada komputer (Samuel, 1959). Dengan kata lain *machine learning* adalah bagaimana memberikan kemampuan kepada komputer untuk melakukan aktivitas belajar guna menyelesaikan masalah secara mandiri. Algoritma *machine learning* dapat dikelompokkan berdasarkan masukan dan keluaran yang diharapkan dari algoritma. Adapun tipe algoritma *machine learning* diantaranya adalah *supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, developmental learning algorithm, transduction,* dan *learning t*o *learn*.

Adapun *supervised learning* merupakan algoritma yang belajar berdasarkan sekumpulan contoh pasangan masukan dan keluaran yang diinginkan. Algoritma ini mengamati contoh-contoh tersebut dan kemudian menghasilkan sebuah model yang mampu memetakan masukan yang baru menjadi keluaran yang tepat (Russell, et al., 1995). Sementara *unsupervised learning* mempunyai tujuan untuk mempelajari dan mencari pola-pola menarik pada masukan yang diberikan (Murphy, 2012). Meskipun tidak disediakan keluaran yang tepat secara eksplisit. Salah satu algoritma *unsupervised learning* yang paling umum digunakan adalah *clustering* atau pengelompokan (Russell, et al., 1995).

Semi-supervised learning ini menggabungkan antara *supervised* dengan *unsupervised* untuk menghasilkan suatu fungsi. Dalam bukunya yang berjudul "*Artificial Intelligence* A *Modern Approach*", (Russell, et al., 1995) menyebutkan bahwa algoritma pembelajaran semi terarah diberikan contoh masukan dan keluaran yang tepat dalam jumlah sedikit dam sekumpulan masukan yang keluarannya belum diketahui. Algoritma ini harus membuat sebuah rangkaian kesatuan antara dua tipe algoritma untuk dapat menutupi kelemahan pada masing-masing algoritma.

Tipe lain dari *machine learning* adalah *reinforcement learning*. Tipe ini mengajarkan bagaimana cara bertindak untuk menghadapi suatu masalah, yang suatu tindakan itu mempunyai dampak. Sedangkan *developmental learning algorithm* atau pembelajaran berkembang adalah sebuah bidang yang bertujuan untuk mempelajari mekanisme pengembangan, arsitektur, dan batasan yang memungkinkan dibuatnya metode pembelajaran yang berlaku seumur hidup, serta bersifat terbuka terhadap kemampuan dan pengetahuan untuk dipasangkan pada mesin. Selanjutnya *transduction*, tipe ini hampir mirip dengan *supervised learning*, tapi tidak secara jelas untuk membangun suatu fungsi melainkan mencoba memprediksi output. Sementara itu tipe *machine learning* yang terakhir adalah *learning to learn*, yaitu tipe *machine learning* yang menggunakan algoritma untuk mempelajari yang sebelum-sebelumnya.

2.2.2. Analisis Sentimen

Menurut Rozi, Pramono, dan Dahlan analisis sentimen atau *opinion mining* merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini (Rozi, et al., 2012). Sementara itu Bing Liu dalam bukunya yang berjudul *Sentiment Analysis and Opinion Mining* mengemukakan bahwa analisis sentimen yang juga disebut *opinion mining* adalah bidang studi yang menganalisis opini atau pendapat publik, menganalisis sentimen publik, menafsirkan, menganalisis atau memprediksi sikap, dan emosi terhadap entitas seperti produk, jasa, organisasi, individu, masalah, peristiwa, topik, dan atribut (Liu, 2012:8).

Oleh karena itu, dapat disimpulkan bahwa inti dari analisis sentimen adalah proses memahami/menafsirkan sikap atau emosi terhadap suatu entitas seperti produk, jasa, dan masalah/peristiwa untuk mendapatkan informasi. Analisis sentimen dilakukan untuk melihat pendapat atau kecendrungan opini terhadap sebuah masalah atau peristiwa oleh seseorang, apakah cendrung beropini positif atau beropini negatif. Dalam perkembangannya pengkategorian kalimat menjadi positif atau negatif menjadikannya sebuah bahan untuk evaluasi. Seperti contoh analisis sentimen terhadap pengaduan pelayanan sebuah instansi tertentu. Hasil dari

analisis dapat digunakan sebagai bahan evaluasi agar pelayanan di instansi tersebut menjadi lebih baik.

Besarnya pengaruh dan manfaat analisis sentimen, menyebabkan penelitian ataupun aplikasi mengenai analisis sentimen berkembang pesat, bahkan di dalam *Handbook of Natural Language Processing* dengan judul *Sentiment Analysis and Subjectivity* oleh Bing Liu dikemukakan fakta bahwa di Amerika ada 20-30 perusahaan yang memfokuskan pada layanan analisis sentimen (Liu, 2010:1). Penggunaan analisis sentimen juga semakin bermanfaat seiring berkembangnya penggunaan media sosial. Sentimen atau opini masyarakat terhadap peristiwa ataupun individu seringkali tercurahkan pada media sosial. Hal ini menyebabkan pentingnya analisis sentimen untuk melihat kecendrungan masyarakat dalam menyikapi peristiwa atau individu tersebut. Contoh lain manfaat dari analisis sentimen adalah dalam dunia usaha, diantaranya untuk melakukan pemantauan terhadap sebuah produk. Secara cepat dapat digunakan sebagai alat bantu untuk melihat respon masyarakat terhadap produk tersebut. Sehingga dapat segera diambil langkah-langkah strategis berikutnya (Nur dan Santika, 2012:1).

Adapun pada penelitian ini proses atau alur yang dilakukan dalam membangun sebuah analisis sentimen dapat dilihat pada Gambar 2.3 berikut ini.



Gambar 2.3 Alur Analisis Sentimen

Preprocessing yang dilakukan pada analisis sentimen ini berfungsi untuk merubah format, bentuk dan atau struktur data ke bentuk yang lebih mudah, ringan, dan efektif untuk kemudian dilakukan pengklasifikasian. Sebelum dilakukan pengklasifikasian menggunakan algoritma *naïve bayes classifier*, dilakukan terlebih dahulu pembobotan menggunakan metode TF-IDF.

2.2.3. *Text Mining*

Text mining merupakan bidang ilmu baru yang mencoba untuk mengumpulkan/mendapatkan informasi yang berguna dari teks/dokumen asli yang belum dimanipulasi (Witten, 2003:1). Sementara itu, menurut salah seorang

Professor *University of California Berkeley*, Marti Hearst menyebutkan bahwa *text mining* adalah proses untuk menemukan pengetahuan baru, yang belum pernah diketahui, secara otomatis oleh komputer dari sumber-sumber tertulis yang berbeda (Hearst, 2003). Dapat dikatakan bahwa *text mining* adalah proses menganalisa teks untuk mendapatkan suatu informasi yang berguna untuk tujuan tertentu.

Text mining itu sendiri merupakan varian baru hasil perkembangan dari data *mining*. Pada intinya proses kerja *text mining* sama dengan proses kerja data *mining* pada umumnya, perbedaan hanya terletak pada data yang di proses. Dalam proses *text mining* ada istilah *preprocessing* data, yaitu proses pendahuluan yang diterapkan terhadap data teks. Proses *text mining* pada umumnya melakukan tahapan *tokenizing*, *filtering*, dan *stemming*. Tahap *tokenizing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyususnnya. Setelah input dilakukan proses *tokenizing*, maka tahap selanjutnya dilakukan *filtering*. Yaitu tahap mengambil kata-kata penting dari hasil tokenisasi. Tahap selanjutnya adalah tahap *stemming* yaitu tahap mencari dasar kata dari setiap kata hasil *filtering*. Setiap kata yang memiliki imbuhan seperti imbuhan awalan dan akhiran maka akan diambil kata dasarnya (Herwansyah, 2009:2).

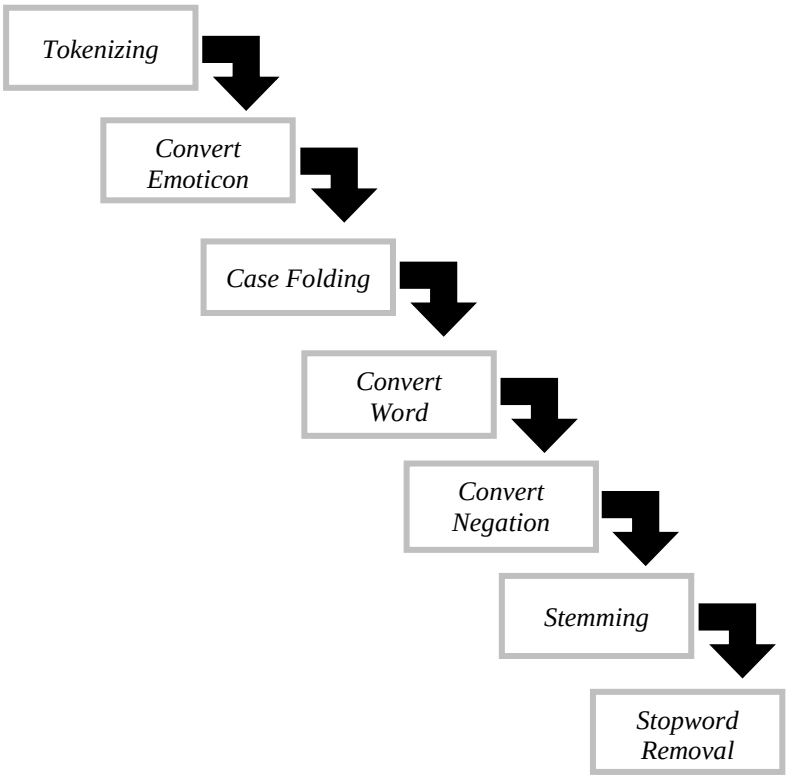
Sesuai dengan namanya, *text mining* fokus kepada data yang berbentuk dokumen atau teks, dengan memanfaatkan informasi yang terdapat pada teks, di mana sumber datanya berasal dari dokumen atau teks tertentu, dengan tujuan untuk mencari kata-kata yang mewakili isi dokumen atau teks, sehingga dapat dilakukan analisa hubungan antar dokumen atau teks tersebut.

2.2.3.1. *Preprocessing*

Sebelum dilakukan proses pengklasifikasian, dilakukan terlebih dahulu proses *preprocessing* terhadap data yang akan digunakan sebagai bahan untuk klasifikasi nantinya. Vijayarani mengemukakan dalam penelitiannya bahwa *preprocessing* sangat berperan penting dalam pengaplikasian *text mining*, karena merupakan proses utama untuk mempermudah pemrosesan serta untuk menghasilkan informasi yang akurat (Vijayrani et al., 2015:9).

Tujuan *preprocessing* itu sendiri adalah untuk mentransformasikan data kedalam suatu format data yang lebih mudah untuk diproses sesuai dengan

kebutuhan penelitian. Tahapan *text preprocessing* yang dilakukan disesuaikan dengan bentuk data serta hasil data *preprocessing* yang dibutuhkan. Pada penelitian ini, tahapan *preprocessing* yang dilakukan adalah *tokenizing*, *convert emoticon*, *case folding*, *convert word*, *convert negation*, *stemming*, dan *stopword removal*. Berikut ini merupakan gambaran proses atau alur yang dilakukan pada *preprocessing*, dapat dilihat pada Gambar 2.4.



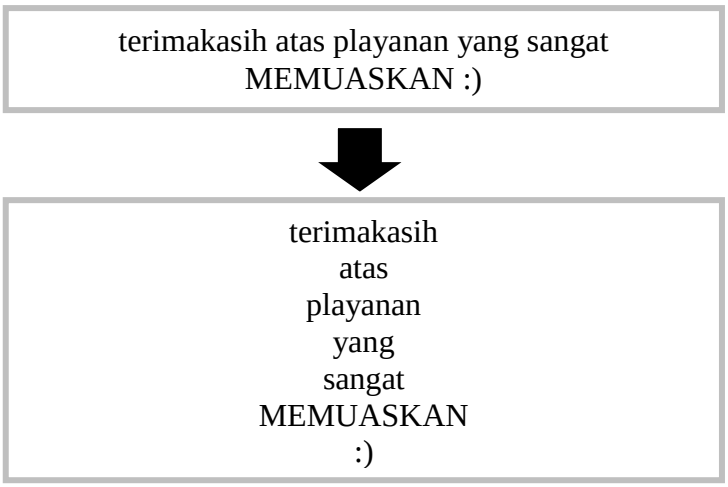
Gambar 2.4 Alur *Preprocessing*

Seluruh tahapan *preprocessing* tersebut memiliki fungsi dan peran yang berbeda. Untuk mendapatkan dataset yang berdimensi lebih kecil dari data sebelumnya, terstruktur, serta bersih dari *noise*, maka keseluruhan tahap tersebut berkesinambungan.

2.2.3.1.1. *Tokenizing*

Tokenizing adalah proses untuk membagi teks yang dapat berupa kalimat, paragraf atau dokumen, menjadi token-token atau bagian-bagian tertentu (Manning, et al., 2008). *Tokenizing* merupakan sebuah proses yang dilakukan untuk menjadikan sebuah kalimat menjadi lebih bermakna dengan cara memecah kalimat

tersebut menjadi kata. *Tokenizing* yang dilakukan adalah memisahkan setiap kata yang dipisahkan oleh tanda pemisah antara kata seperti spasi dan tanda pemisah lainnya. Contohnya dapat dilihat pada Gambar 2.5 berikut ini.



Gambar 2.5 Gambaran Proses *Tokenizing*

2.2.3.1.2. *Convert Emoticon*

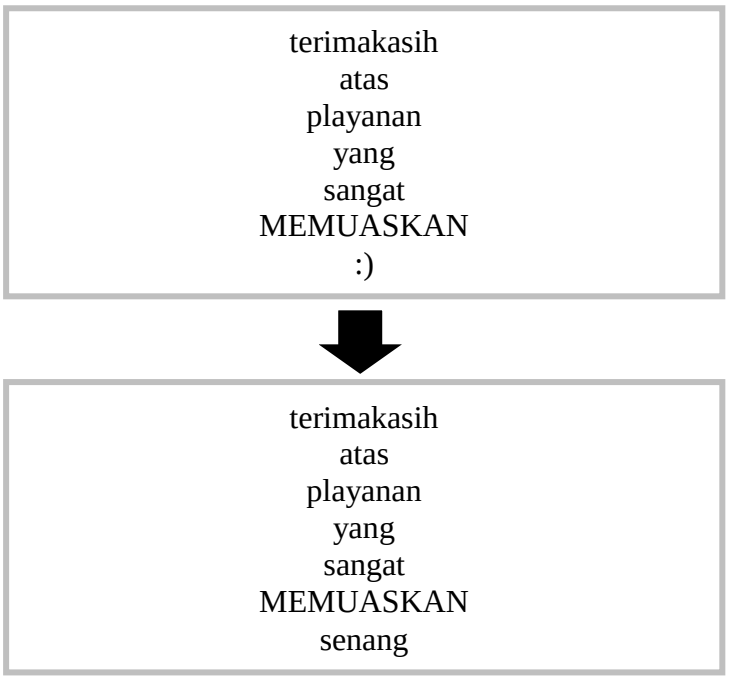
Emoticon merupakan salah satu cara untuk mengekspresikan ungkapan perasaan secara tekstual. Biasanya dalam pengaduan pelayanan *emoticon* digunakan untuk mengekspresikan perasaan pengadu dalam pelayanannya atau sebagai bentuk mengekspresikan persetujuan atau pertidaksetujuan suatu kalimat. Pada penelitian sebelumnya yang dilakukan oleh Sunni dan Widyantoro disebutkan bahwa *convert emoticon* dalam suatu kalimat dianggap penting dan memiliki kontribusi dalam menentukan nilai sentimen suatu kalimat (Sunni dan Widyantoro, 2012). Setiap *emoticon* dikonversikan ke dalam *string* yang sesuai dengan setiap ekspresi *emoticon tersebut.* Namun, tidak semua akan diimplementasikan, karena tidak semua *emoticon* sering digunakan oleh pengguna SMS. Untuk lebih jelasnya dapat dilihat pada Tabel 2.1 berikut ini.

Tabel 2.1 Daftar *Convert Emoticon*

Emoticon	**Konversi**
>:] :-) :) :o) :] :3 :c) :> =] 8) =) :} :^)	senang
>:D :-D :D 8-D 8D x-D xD X-D XD =-D =D =-3 =3	tertawa
>:[:-(:(:-c :c :-< :< :-[:[:{	sedih

D:< D: D8 D; D= DX v.v D-‘:	horor
>:P :-P :P X-P x-p xp XP :-p :p =p :-b :b	lidah
>:o >:O :-O :O o_O o.O 8-0	kaget
>:\ >:/ :-/ :-. :/ :\ =/ =\ :S	kesal
:| :-|	datar

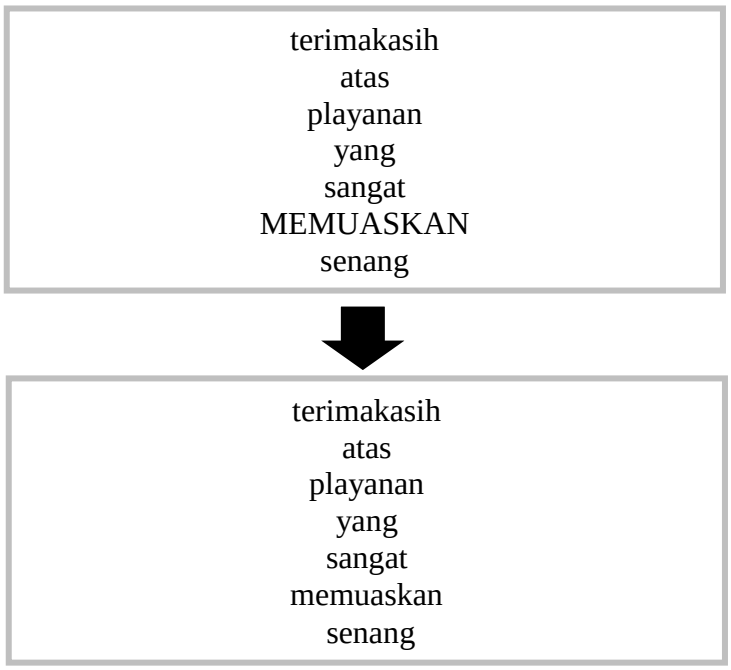
Berikut ini merupakan contoh dari *convert emoticon*, untuk lebih jelasnya dapat dilihat pada Gambar 2.6.



Gambar 2.6 Gambaran Proses *Convert Emoticon*

2.2.3.1.3. *Case Folding*

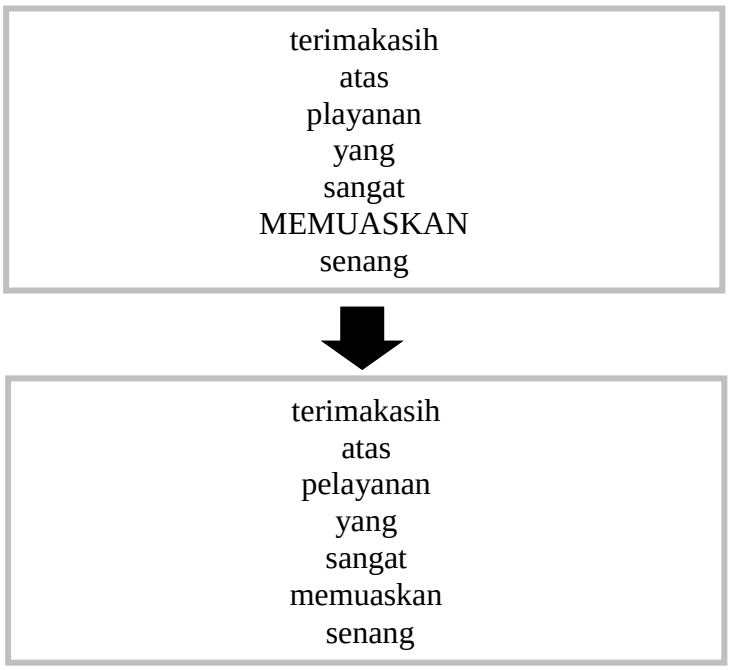
Case folding merupakan tahapan *prepocessing* yang berfungsi untuk menyeragamkan karakter pada dokumen atau teks. Pada proses ini, semua huruf dirubah menjadi huruf kecil atau *lowercase* (Sunni dan Widyantoro, 2012:202). Untuk lebih jelasnya dapat dilihat pada Gambar 2.7 berikut ini.



Gambar 2.7 Gambaran Proses *Case Folding*

2.2.3.1.4. *Convert Word*

Convert word merubah kata yang tidak baku menjadi baku. Beberapa kata yang muncul dalam pesan sms cendrung tidak baku, dan tidak ada dalam kamus. Oleh karenanya, untuk kata yang tidak ada di dalam kamus, akan ditranslasikan ke kata terdekat dengan menggunakan kamus yang dibuat dengan melihat pola kemunculan kata-kata tidak baku tersebut (Sunni dan Widyantoro, 2012:202). Untuk lebih jelasnya dapat dilihat pada Gambar 2.8 berikut ini.



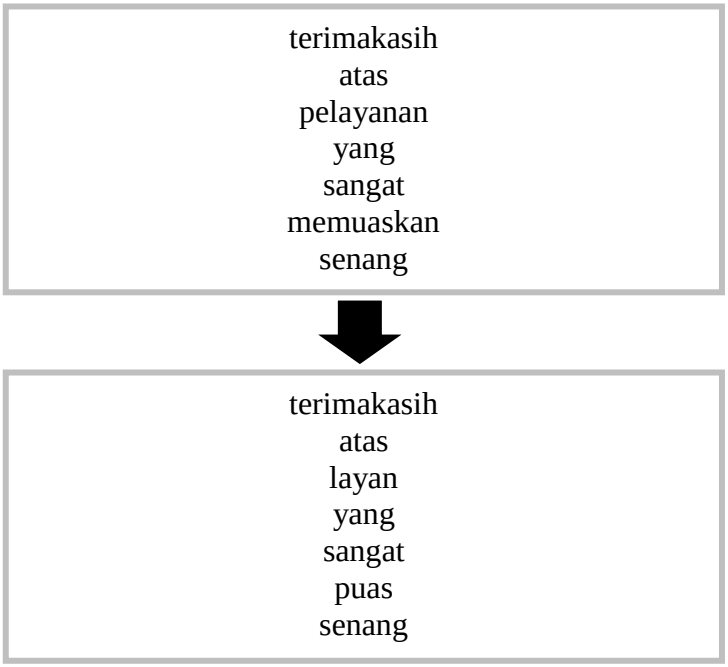
Gambar 2.8 Gambaran Proses *Convert Word*

2.2.3.1.5. *Convert Negation*

Kata-kata dalam suatu kalimat yang bersifat negasi, akan merubah nilai sentimen suatu kalimat pengaduan. *Convert negation* akan dilakukan jika terdapat kata negasi sebelum kata yang bernilai positif, maka kata tersebut akan diubah nilainya menjadi negatif dan berlaku pula untuk sebaliknya. Sunni dan Widyantoro dalam penelitiannya menyebutkan kata yang bersifat negasi diantaranya adalah "tidak", "tdk", "tak", "enggak", "nggak", "gak", "ga", "g", "gg", "kaga" (Sunni dan Widyantoro, 2012:202). Contohnya kata "nyaman" merupakan kata kunci positif, namun jika sebelumnya diikuti kata "gak" atau "tidak" maka nilainya akan diubah menjadi negatif.

2.2.3.1.6. *Stemming*

Stemming merupakan proses untuk mencari *root* dari kata. Pencarian *root* sebuah kata atau biasa disebut dengan kata dasar dapat memperkecil hasil indeks tanpa harus menghilangkan makna (Utomo, 2013:42). Sebagai contoh, kata pelayan, melayani, terlayani, dilayani akan distem ke kata dasarnya yaitu "layan". Contohnya dapat dilihat pada Gambar 2.9 berikut ini.



Gambar 2.9 Gambaran Proses *Stemming*

Algoritma *stemming* untuk satu bahasa berbeda dengan algoritma untuk bahasa lainnya. Proses *stemming* pada teks berbahasa Indonesia lebih rumit karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan kata dasar dari sebuah kata. Ledy Agusta dalam penelitiannya menyebutkan, proses *stemming* pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan sufiks. Sedangkan pada teks berbahasa Indonesia, selain sufiks, prefiks, dan konfiks juga dihilangkan (Agusta, 2009:196). Adapun algoritma yang biasa digunakan dalam proses *stemming* teks berbahasa Indonesia diantaranya adalah Porter, Nazief & Adriani, Ahmad Yusoff, Arifin & Setiono, Jelita Asian dan lain-lain. Berikut ini merupakan perbandingan dari beberapa algoritma *stemming* dari penelitian yang telah dilakukan sebelumnya.

Tabel 2.2 Perbandingan Algoritma *Stemming*

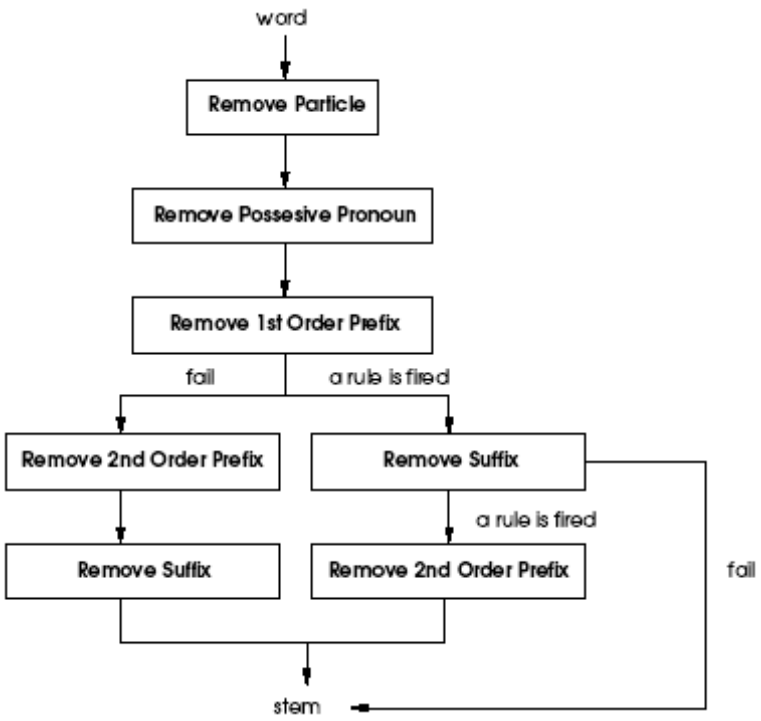
No	Peneliti	Judul	Kelebihan	Kekurangan
1.	Ledy Agusta	Perbandingan Algoritma Stemming Porter dengan Nazief &	1. Algoritma Porter membutuhkan waktu yang lebih singkat	1. Algoritma Nazief & Adriani membutuhkan

		Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia		waktu yang lebih lama
			2. Algoritma Nazief & Adriani memiliki persentase keakuratan lebih besar	2. Algoritma Porter memiliki persentase keakuratan lebih kecil
2.	Asriko Adipathy, Yanuar Firdaus A. W., Angelina Prima Kurniati	Analisis dan Imlementasi Perbandingan Stemming dengan Menggunakan Algoritma Jelita Asian dan Arifin & Setiono Pada Information Retrieval	1. Algoritma Asian Jelita memiliki performasi lebih baik	1. Algoritma Arifin & Setiono memiliki performasi yang kurang baik
			2. Algoritma Asian Jelita memiliki kekuatan yang lebih baik dalam proses *indexing*	2. Algoritma Arifin & Setiono tidak memiliki kekuatan yang baik dalam proses *indexing*
3.	Yonissa Qamila Intan Atazsu, Angelina Prima Kurniati, Moch Arif Bijaksana	Analisis dan Implementasi Perbandingan Stemming Menggunakan Algoritma Nazief & Adriani dengan Algoritma Ahmad Yusoff Sembok Pada Information Retrieval	1. Algoritma Nazief & Adriani memiliki performasi lebih baik	1. Algoritma Ahmad Yusoff Sembok memiliki performasi yang kurang baik
			2. Algoritma Nazief & Adriani memiliki kekuatan (*stemmer strength*) yang lebih tinggi	2. Algoritma Ahmad Yussof Sembok memiliki kekuatan (*stemmer strength*) yang lebih rendah
			3. Algoritma Nazief & Adriani menghasilkan keakuratan yang lebih baik	3. Algoritma Ahmad Yussof Sembok menghasilkan keakuratan yang kurang baik

Penelitian-penelitian tersebut tentunya dapat menghasilkan kesimpulan yang berbeda-beda mengingat data dan parameter yang digunakan masing-masing penelitian berbeda. Pada penelitian ini proses *stemming* yang dilakukan adalah menggunakan algoritma porter karena dirasa sesuai dengan data yang diolah. Selain itu, merujuk pada penelitian yang dilakukan oleh Ledy Agusta terdapat beberapa saran dalam penelitian yang dapat meningkatkan tingkat keakurasian dalam proses *stemming*. Hal ini yang menjadi pertimbangan, karena selain waktu dalam proses *stemming* yang lebih cepat bila dibandingkan algoritma lain, akurasi juga menjadi lebih baik ketika menjalankan saran yang diberikan.

2.2.3.1.6.1. Algoritma Porter

Algoritma porter ditemukan oleh Martin Porter pada tahun 1980. Porter sendiri merupakan algoritma penghilangan akhiran *morphological* dan infleksional yang umum dari bahasa Inggris. Algoritma tersebut digunakan untuk *stemming* bahasa Inggris, kemudian karena proses *stemming* bahasa Inggris berbeda dengan bahasa Indonesia maka, dikembangkan algoritma khusus untuk bahasa Indonesia. Algoritma porter untuk bahasa Indonesia berdasarkan *English Porter Stemmer* dikembangkan oleh W.B. Frakes pada tahun 1992. Karena bahasa Inggris datang dari kelas yang berbeda, beberapa modifikasi telah dilakukan untuk membuat algoritma porter dapat digunakan sesuai dengan bahasa Indonesia (Tala, 2003). Desain dari algoritma porter untuk bahasa Indonesia dapat dilihat pada Gambar 2.10 berikut ini.



Gambar 2.10 Desain Algoritma Porter Bahasa Indonesia

Pada Gambar 2.10 terlihat beberapa langkah *removal* dalam desain algoritma porter untuk bahasa Indonesia. Adapun penjelasan aturan lebih lengkapnya dapat dilihat pada Tabel 2.3 sampai dengan Tabel 2.7. Aturan pertama adalah menghapus partikel seperti: -kah, -lah.

Tabel 2.3 Kelompok Aturan Pertama: *Inflectional Particles*

Suffix	*Replacement*	*Measure Condition*	*Additional Condition*	*Examples*
kah	NULL	2	NULL	bukukah → buku
lah	NULL	2	NULL	adalah → ada
pun	NULL	2	NULL	bukupun → buku

Aturan kedua adalah menghapus kata ganti (*Possesive Pronoun*), seperti -ku, -mu, -nya.

Tabel 2.4 Kelompok Aturan Kedua: *Inflectional Possesive Pronouns*

Suffix	*Replacement*	*Measure Condition*	*Additional Condition*	*Examples*
ku	NULL	2	NULL	bukuku → buku
mu	NULL	2	NULL	bukumu → buku
nya	NULL	2	NULL	bukunya → buku

Langkah selanjutnya adalah menghapus awalan pertama seperti terlihat pada Tabel 2.5 berikut ini.

Tabel 2.5 Kelompok Aturan Ketiga: *First Order of Derivational Prefixes*

Suffix	*Replacement*	*Measure Condition*	*Additional Condition*	*Examples*
meng	NULL	2	NULL	mengukur → ukur
meny	s	2	V…	menyapu → sapu
men	NULL	2	NULL	menduga → duga
mem	p	2	V…	memilah → pilah
mem	NULL	2	NULL	membaca → baca
me	NULL	2	NULL	merusak → rusak
peng	NULL	2	NULL	pengukur → ukur
peny	s	2	V…	penyapu → sapu
pen	NULL	2	NULL	penduga → duga
pem	p	2	V…	pemilah → pilah
pem	NULL	2	NULL	pembaca → baca
di	NULL	2	NULL	diukur → ukur
ter	NULL	2	NULL	tersapu → sapu
ke	NULL	2	NULL	kekasih → kasih

Melanjutkan langkah sebelumnya, jika pada langkah sebelumnya awalan kata tidak ditemukan atau tidak dihapus maka akan dilanjutkan dengan menghapus awalan kedua. Setelah awalan kedua dihapus maka, akhiran akan dihapus dan kata tersisa diasumsikan sebagai kata dasar (*root* word). Sementara jika pada awalan pertama kata ditemukan dan dihapus, maka akan dilanjutkan dengan menghapus akhiran kata.

Tabel 2.6 Kelompok Aturan Keempat: *Second Order of Derivational Prefixes*

Suffix	*Replacement*	*Measure Condition*	*Additional Condition*	*Examples*
ber	NULL	2	NULL	berlari → lari
bel	NULL	2	ajar	belajar → ajar
be	NULL	2	K*er…	bekerja → kerja
per	NULL	2	NULL	perjelas → jelas
pel	NULL	2	ajar	pelajar → ajar
pe	NULL	2	NULL	pekerja → kerja

Ketika menghapus akhiran tidak ditemukan, maka kata tersebut diasumsikan sebagai kata dasar (*root word*). Sementara jika akhiran ditemukan maka akan

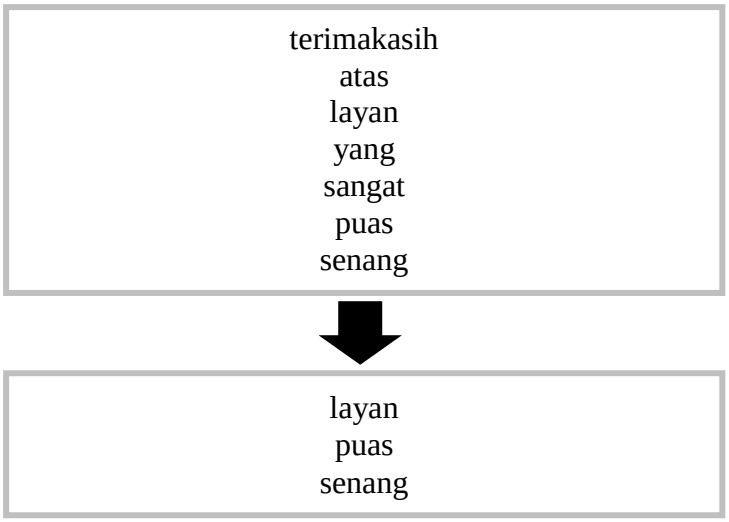
dilanjutkan dengan menghapus awalan kedua dan kata terakhir diasumsikan sebagai kata dasar (*root word*) seperti terlihat pada Tabel 2.7 berikut ini.

Tabel 2.7 Kelompok Aturan Kelima: *Derivational Suffixes*

Suffix	*Replacement*	*Measure Condition*	*Additional Condition*	*Examples*
kan	NULL	2	prefix {ke, peng}	tarikkan → tarik (meng)ambilkan → ambil
an	NULL	2	prefix {di, meng, ter}	makanan → makan (per)janjian → janji
i	NULL	2	prefix {ber, ke, peng}	tandai → tanda (men)dapati → dapat

2.2.3.1.7. *Stopword Removal*

Stopword removal adalah tahapan selanjutnya pada proses *text preprocessing*. *Stopword list* adalah kata-kata yang semestinya dihilangkan, jika kata pada data set terdapat pada *stopword list* maka kata akan dihilangkan. Tetapi jika tidak terdapat di dalamnya maka proses akan berlanjut tanpa menghilangkan kata pada dokumen (Wijaya, 2015:2). Hal tersebut biasa disebut dengan *stopword blacklist.* Sebaliknya, kita dapat mengisi *stopword list* dengan *whitelist.* Dalam artian sebaliknya, kata-kata yang akan diolah hanya yang termasuk kedalam *list* saja. Kita hanya perlu memasukan kata-kata yang sekiranya berpengaruh untuk diolah. Gambaran dari proses *stopword removal* dapat dilihat pada Gambar 2.11 berikut ini.



Gambar 2.11 Gambaran Proses Stopword Removal

2.2.3.2. TF-IDF *Features*

Setiap dokumen atau kalimat pesan diwakili oleh vektor dengan pengenalan elemen-elemen yang dikenali dari tahap ekstraksi dari dokumen (Wijaya, 2015:2). Vektor yang terdiri dari bobot setiap pemberhentian yang menggunakan dasar perhitungan pada metode TF-IDF. TF-IDF adalah metode pembobotan yang mengaitkan antara *term frequency* (TF) dan *invers document frequency* (IDF). Langkah awal pada pembobotan TF-IDF adalah menemukan nomor kata yang diketahui sebagai bobot atau *term frequency* di setiap dokumen atau kalimat pesan setelah dilakukan pengalian oleh *invers document frequency*. Metode TF-IDF dapat dirumuskan sebagai berikut:

a. TF (*Term Frequency*)

Term Frequency adalah cara pembobotan term atau kata yang paling sederhana (Junianto, 2014). Bobot kata t pada dokumen/teks pesan diberikan dengan:

(2.1)

$$w_{ij} = tf_{ij} * idf$$

b. IDF (*Inverse Document Frequency*)

Jika TF memperhatikan kemunculan kata dalam dokumen/teks pesan, IDF memperhatikan kemunculan kata pada kumpulan dokumen/teks pesan (Junianto, 2014). Faktor IDF pada suatu kata t diberikan oleh:

(2.2)

$$idf = log\left(\frac{N}{df_j}\right)$$

Dimana w_{ij} adalah bobot kata i pada dokumen/teks pesan j, sementara N adalah jumlah dokumen/teks pesan, dan *term frequency* atau tf_{ij} adalah jumlah dari kemunculan kata i pada dokumen/teks pesan j, df_j (*document frequency*) adalah jumlah dokumen j yang berisi kata i.

2.2.4. Klasifikasi

Klasifikasi merupakan proses pencarian sekumpulan model atau fungsi yang menggambarkan dan membedakan kelas data. Tujuan dari klasifikasi adalah untuk memprediksi kelas dari suatu objek yang belum diketahui kelasnya (Mulyanto, 2009:204). Dalam penelitian ini terdapat beberapa proses pengklasifikasian. Seperti mengklasifikasi pesan pengaduan yang masuk kedalam pesan positif atau negatif dan mengklasifikasi pesan pengaduan yang masuk kedalam kelas atau bidang yang sesuai dengan isi pesannya. Klasifikasi sendiri memiliki dua proses yaitu membangun model klasifikasi dari sekumpulan kelas data yang sudah didefinisikan sebelumnya (*training data set*) dan menggunakan model tersebut untuk klasifikasi data *test* serta mengukur akurasi dari model.

Proses pengklasifikasian itu sendiri dapat dilakukan dengan berbagai metode seperti *Naïve Bayes Classifier*, *Support Vector Machine*, *K-Nearest Neighbour Classifier*, *Neural Network*, dll. Dalam penelitian kali ini dilakukan klasifikasi menggunakan metode *naïve bayes classifier* yang sudah teruji akurasi serta komputasinya dari berbagai penelitian sebelumnya.

2.2.4.1. *Naïve Bayes Classifier*

Naive bayes classifier adalah metode pembelajaran mesin yang memiliki model dalam bentuk probabilitas atau peluang (Sunni dan Widyantoro, 2015:203). Tergantung pada situasi yang tepat dari model probabilitas, *naive bayes classifier* dapat dilatih sangat efisien dalam *supervised learning*. Kelebihan dari *naive bayes classifier* adalah hanya memerlukan sejumlah kecil data pelatihan untuk mengestimasi parameter (ciri-ciri dan varian dari variabel) yang diperlukan untuk klasifikasi (Chandra, *et al*., 2016). Karena variabel diasumsikan independen, hanya varian dari variabel-variabel untuk setiap kelas yang perlu ditentukan. Selain dari pada itu, menurut Akhmad Pandhu Wijaya kelebihan lainnya dari *naïve bayes classifier* adalah kemampuannya mengklasifikasi dokumen dengan kesederhanaan dan kecepatan komputasinya (Wijaya, 2015:1).

Metode *naive bayes classifier* menempuh dua tahap dalam proses klasifikasi teks, yaitu tahap pelatihan dan tahap klasifikasi. Data latih yang berupa pasangan pesan dan kelas bidang atau sentimennya dijadikan sebagai sumber pembentukan

model analisis. Setiap fitur yang merepresentasikan pesan dihitung probabilitas kemunculannya di masing-masing kelasnya. Ketika data *testing* pesan baru akan diklasifikasikan, maka nilai probabilitas dari setiap fitur untuk masing-masing kelas dikalikan. Nilai probabilitas yang paling besar diantara masing-masing kelas dijadikan kelas pesan baru yang masuk tersebut. Adapun formula untuk menentukan kelas dari pesan baru ditunjukkan oleh persamaan (2.3) berikut ini (Sunni dan Widyantoro, 2015:203).

(2.3)

$$V_{MAP} = arg \max_{V_j \in V} \mathrm{P}(v_j) \prod_{i=1}^{n} P(w_i|v_j)$$

Dimana:

w_i = atribut atau fitur ke-i

v_j = kelas ke-j (contoh: positif atau negatif)

V = himpunan kelas

V_{MAP} = kelas suatu pesan

2.2.5. *Confusion Matrix*

Pada bidang *machine learning* dan statistik klasifikasi secara khusus, *confusion matrix* juga dikenal dengan *error matrix* yaitu sebuah metode yang memungkinkan memvisualisasikan suatu kinerja algoritma, salah satunya adalah *supervised learning* (Stehman, 1997). Setiap kolom dari matriks merepresentasikan prediksi dari setiap kelas atau *predicted class*, sementara setiap baris merepresentasikan kelas sebenarnya *actual class* (Powers, 2007).

Confusion matrix berisi informasi tentang klasifikasi aktual dan yang telah terprediksi yang dilakukan oleh sistem klasifikasi. Pada umumnya dievaluasi dengan menggunakan data matriks. Metode klasifikasi akan dievaluasi terutama pada bagian akurasi dari hasil klasifikasi. Akurasi sebuah klasifikasi berpengaruh terhadap performa dari suatu klasifikasi. Berikut ini contoh tabel yang menunjukan *confusion matrix* untuk klasifikasi dua kelas.

Tabel 2.8 *Confusion Matrix*

		Predicted Class	
		Positive	*Negative*
Actual Class	*Positive*	*True Positives*	*False Negatives*
	Negative	*False Positives*	*True Negatives*

Keterangan:

True Positives : Jumlah *record* positif yang diklasifikasikan sebagai positif

False Positives : Jumlah *record* positif yang diklasifikasikan sebagai negatif

False Negatives : Jumlah *record* negatif yang diklasifikasikan sebagai positif

True Negatives : Jumlah *record* negatif yang diklasifikasikan sebagai negatif.

Setiap kolom dari tabel *confusion matrix* merupakan contoh di kelas yang telah diprediksi, sedangkan setiap baris mewakili contoh di kelas yang sebenarnya. Setelah mendapatkan nilai untuk masing-masing kelas, selanjutnya adalah menghitung nilai akurasi dan *precision*. *Precision* adalah ukuran terhadap suatu kelas yang telah diprediksi. Adapun persamaannya dapat dilihat pada persamaan (2.4) dan (2.5) berikut ini.

(2.4)

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN}$$

(2.5)

$$Akurasi = \frac{TP}{TP + FP}$$

2.2.6. *Short Message Service* (SMS)

Short Message Service (SMS) adalah layanan komponen pesan teks dari telepon, web, atau sistem komunikasi mobile lainnya. SMS menggunakan protokol komunikasi standar untuk memungkinkan telepon tetap atau perangkat ponsel untuk bertukar pesan teks singkat (Kelly, 2012).

Sebuah pesan sms maksimal terdiri dari 140 bytes, dengan kata lain sebuah pesan bisa memuat 140 karakter 8-bit, 160 karakter 7-bit atau 70 karakter 16-bit untuk bahasa Jepang, bahasa Mandarin, dan bahasa Korea yang memakai Hanzi

(Aksara Kanji/Hanja). Selain 140 bytes ini ada data-data lain yang termasuk. Adapula beberapa metode untuk mengirim pesan yang lebih dari 140 bytes, tetapi seorang pengguna harus membayar lebih dari sekali.

Pesan-pesan sms dikirim dari sebuah telepon genggam ke pusat pesan (SMSC dalam bahasa Inggris), disini pesan disimpan dan mencoba mengirimnya selama beberapa kali. Setelah sebuah waktu yang telah ditentukan, biasanya 1 hari 2 hari, lalu pesan dihapus. Seorang pengguna bisa mendapatkan konfirmasi dari pusat pesan ini.

2.2.6.1. Short Message Service (SMS) *Gateway*

Dikutip dari situs techopedia, *Short Message Service* (SMS) *Gateway* adalah mekanisme interaksi atau pengolahan pesan SMS sehingga dapat dikirim dan diterima. SMS *gateway* memfasilitasi dan merampingkan proses pesan teks untuk organisasi, dan sering melakukan beberapa konversi ke format yang berbeda (Techopedia, 2012).

SMS *Gateway* berfungsi sebagai penghubung yang melakukan *relay* sms antara ESME (*External Short Message Entitiy*) dan SMSC dan sebaliknya. Komunikasi antara ESME dan SMS *Gateway* dapat menggunakan protokol SMPP atau dengan HTTP, sementara ke SMSC menggunakan Protokol SMPP (*Short Message Peer to Peer*).

ESME adalah entitas luar yang dapat berupa *server* aplikasi penyedia layanan (*Application Service Provider*) yang dimiliki oleh *content provider*, aplikasi perbankan, *server polling*, dan lain-lain yang dapat menerima pesan, memproses pesan dan mengirim *respons* atas pesan yang masuk , serta perangkat lain seperti *email gateway*, WAP *proxy server*, *voice mail server*.

SMPP adalah protokol yang cukup populer yang terbuka, protokol standar industri yang dirancang untuk menyediakan antarmuka komunikasi data yang fleksibel untuk transfer data 20 pesan singkat antar entitas SMS, seperti ESME dan SMSC. SMPP yang banyak digunakan sekarang adalah SMPP v3.4. SMPP cukup populer dipakai karena memberikan kapasitas yang besar dan kecepatan transfer sms yang cepat.

Dalam komunikasi dengan menggunakan SMPP protokol, ada yang berfungsi sebagai SMPP *server* dan yang lain sebagai SMPP *client*. SMS *gateway* selain berfungsi sebagai SMPP *server* juga dapat berfungsi sebagai SMPP Client.

[ESME] ————————> [SMSGW]

ESME = SMPP *Client*, SMSGW = SMPP *Server*

[SMPP] ————————> [SMSC]

SMSGW = SMPP *Client*, SMSC = SMPP *Server*

Sebuah SMPP *client* setidaknya memerlukan 4 buah parameter untuk dapat berkomunikasi dengan SMPP *server*, seperti: IP Address SMPP *Server*, nomor *port*, *username* dan password.

Beberapa keuntungan dalam menggunakan SMS *gateway* adalah:

1. Pengetikan pesan lebih mudah menggunakan piranti masukan *keyboard*.
2. Kemudahan dan kecepatan manipulasi konten pesan (dapat menggunakan *copy paste*).
3. Kemudahan dalam memasukan data nomor telepon tujuan.
4. Kemudahan penggunaan hanya satu nomor perusahaan, sehingga mudah diingat dan mempermudah dalam sosialasi.

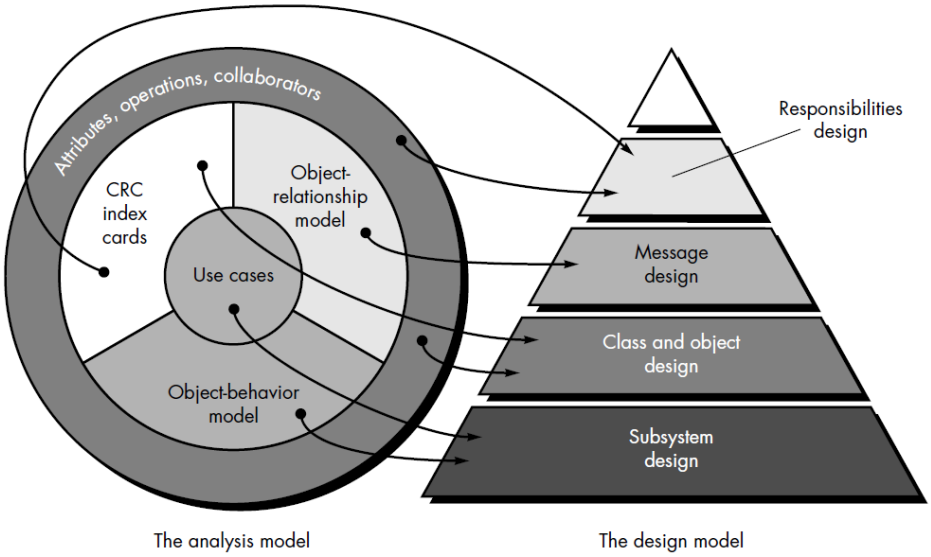
2.2.7. *Object Oriented Analisis and Design* (OOAD)

Roger S. Pressman dan Bruce R. Maxim berpendapat bahwa *Object Oriented Analysis Design* merupakan pendekatan pemodelan sistem dimana sekelompok objek berinteraksi dan setiap objek mewakili beberapa entitas kepentingan dalam sistem yang dimodelkan (Pressman dan Maxim, 2015). Analisis dan desain berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Konsep *Object Oriented Analysis Design* mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD).

OOA merupakan metode analisis yang memeriksa *requirements* (syarat/keperluan yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan. Sedangkan

OOD adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

OOD menerjemahkan model OOA dari dunia nyata kedalam model implementasi spesifik yang dapat direalisasikan di dalam perangkat lunak (Pressman dan Maxim, 2015). Hubungan antar keduanya dapat dilihat pada Gambar 2.12 berikut ini.



Gambar 2.12 Hubungan Antara OOA dan OOD

Proses OOD dapat digambarkan sebagai piramida yang terdiri dari empat lapisan. Lapisan dasar berfokus pada desain subsistem yang mengimplementasikan fungsi-fungsi sistem mayor. Lapisan kelas mengkhususkan keseluruhan arsitektur objek dan hirarki kelas yang diperlukan untuk mengimplementasi suatu sistem. Lapisan pesan menunjukan bagaimana kolaborasi antar objek akan direalisasi, dan lapisan tanggung jawab mengidentifikasikan atribut dan operasi yang menandai masing-masing kelas. Metodologi pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama, yaitu:

1. *Encapsulation*

Encapsulation merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat

mengaksesnya. Data terlindung dari prosedur atau objek lain, kecuali prosedur yang berada dalam objek itu sendiri.

2. *Inheritance*

Inheritance adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek induk diturunkan kepada anak objek, demikian seterusnya. Inheritance mempunyai hubungan secara hirarki. Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya.

3. *Polymorphism*

Polymorphism yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. *Polymorphism* mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda.

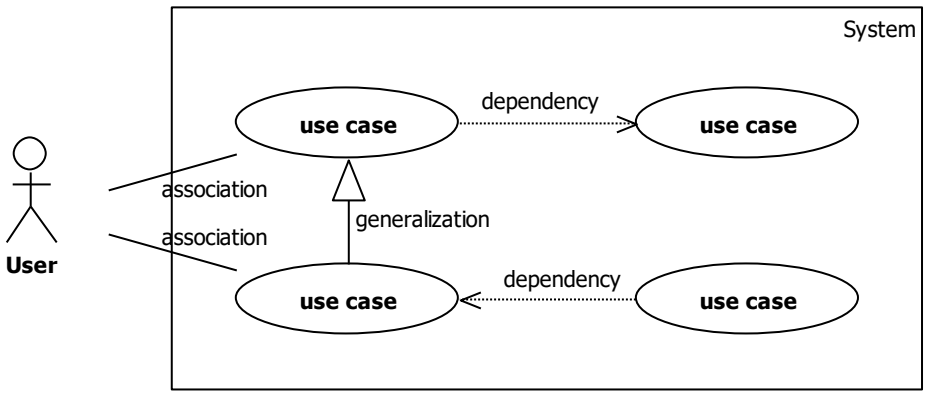
2.2.8. *Unified Modeling Language* (UML)

Unified Modeling Language (UML) merupakan salah satu bentuk bahasa, menurut pencetusnya UML di definisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem (Nugroho, 2005:16). Sementara menurut Whitten L. Jeffery, UML merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Whitten, 2004). Pada intinya, pemodelan digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sehingga lebih mudah dipelajari dan dipahami. UML digunakan untuk melakukan analisis dan perancangan sistem/perangkat lunak. UML merupakan bahasa standar untuk merancang dan mendokumentasikan perangkat lunak yang berorientasi objek. Berikut ini merupakan beberapa diagram yang ada pada UML, antara lain:

1. *Use Case Diagram*

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Terdiri atas diagram untuk *use case* dan aktor. Aktor

merepresentasikan orang yang akan mengoprasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh aktor. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan didalamnya. Aktor yang melakukan operasi dihubungkan dengan garis lurus ke *use case.* Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa-siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa dan Shalahuddin, 2015:155). Untuk lebih jelasnya, berikut ini merupakan contoh bentuk *use case diagram* yang dapat dilihat pada Gambar 2.13.



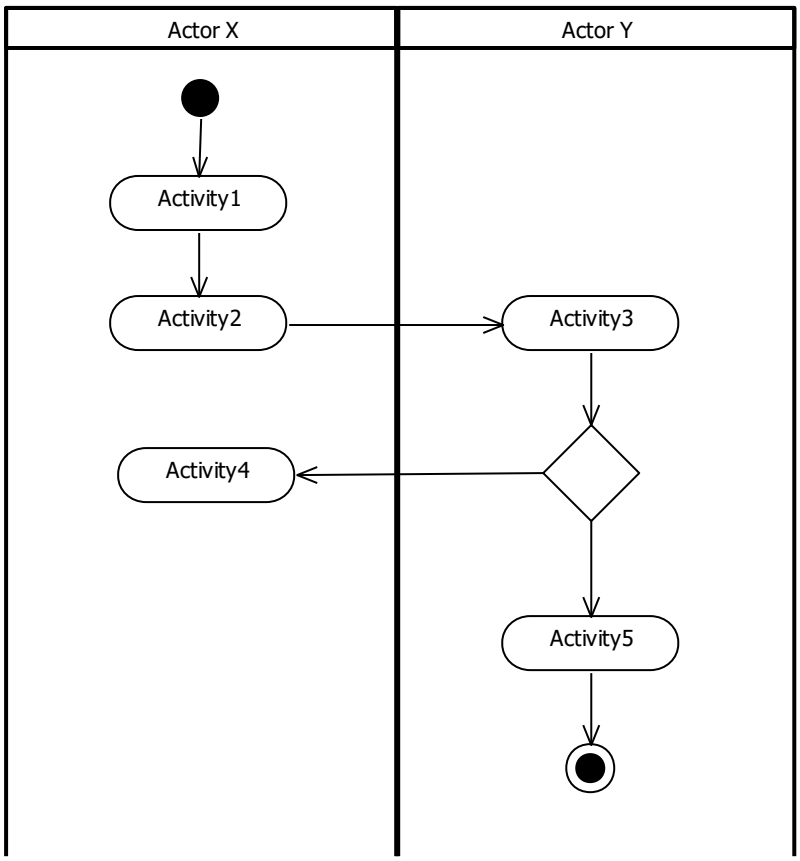
Gambar 2.13 Contoh Bentuk *Use Case Diagram*

2. *Activity Diagram*

Activity diagram secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses maupun *use case. Activity diagram* menggambarkan bagaimana masing-masing aliran berawal, *decision* yang mungkin terjadi, serta bagaimana alur berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Menurut Rosa dan Shalahuddin yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas yang dapat dilakukan sistem bukan apa yang dilakukan aktor (Rosa dan Shalahuddin, 2015:161).

Perbedaan mendasar dari *use case diagram* dengan *activity diagram* adalah bahwa sebuah *activity* dapat direalisasikan oleh satu *use case* atau lebih. *Activity diagram* menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

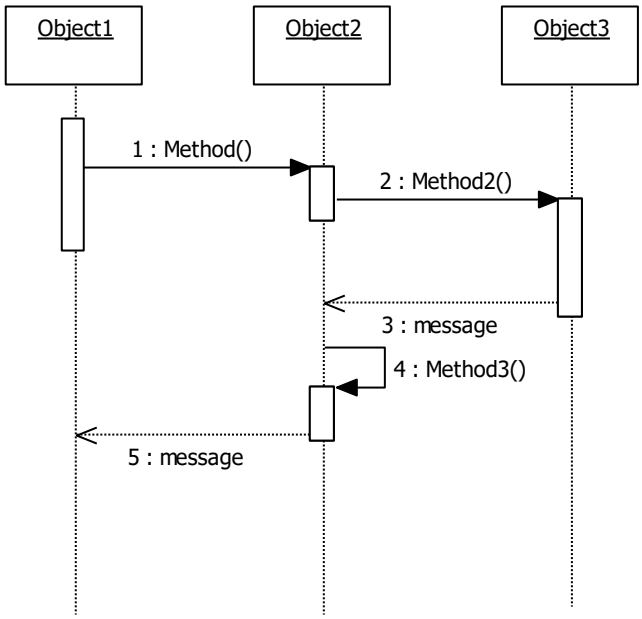
Untuk lebih jelasnya, berikut ini merupakan contoh bentuk *activity diagram* yang dapat dilihat pada Gambar 2.14.



Gambar 2.14 Contoh Bentuk *Activity Diagram*

3. *Sequence Diagram*

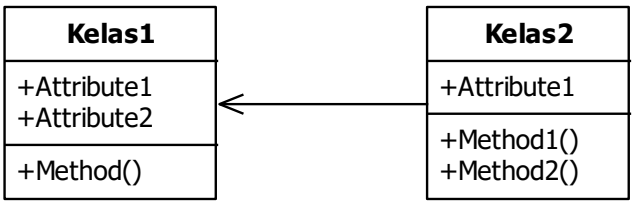
Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa dan Shalahuddin, 2015:165). Banyaknya diagram *sequence* yang harus dibuat adalah sebanyak pendefinisian *use case*. Untuk lebih jelasnya, berikut ini merupakan contoh bentuk *sequence diagram* yang dapat dilihat pada Gambar 2.15.



Gambar 2.15 Contoh Bentuk *Sequence Diagram*

4. *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi (Rosa dan Shalahuddin, 2015:141). Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Untuk lebih jelasnya, berikut ini merupakan contoh bentuk *class diagram* yang dapat dilihat pada Gambar 2.16.



Gambar 2.16 Contoh Bentuk *Class Diagram*