BAB 2
LANDASAN TEORI

2.1 Text Mining

Text mining merupakan proses mendapatkan informasi penting yang berasal dari sekumpulan dokumen dengan melakukan pencarian kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

Proses yang dilakukan oleh text mining diawali dengan penerapan struktur terhadap sumber data teks dan dilanjutkan dengan ekstraksi informasi dan pengetahuan yang relevan dari data teks terstruktur ini. Ada beberapa hal yang perlu dilakukan pada tahap *pre-processing* ini yaitu *parsing,tokenizing,stopword* dan *stemming*[3].

2.2 Parsing

Parsing merupakan proses memilah isi dokumen menjadi unit-unit kecil yang akan menjadi penciri misalnya berupa kata,frase atau kalimat. Unit terkecil ini yang disebut token. Proses parsing merujuk pada proses pengidentifikasian token dalam rangkaian teks sehingga bagian dalam parsing dari dokumen teks disebut tokenizer. Proses ini memerlukan pengetahuan tentang bahasa untuk menangani karakter-karakter khusus dan menentukan batasan satuan unit dalam dokumen [12].

2.3 Tokenizing

Tahap *tokenizing / parsing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Karakter selain huruf dihilangkan dan dianggap *delimiter*. *Tokenizing* biasanya digunakan pada tahap *preprocessing* sehingga kata-kata pada sebuah dokumen dibagi menjadi beberapa kata sesuai dengan delimeter pembagi kata yang telah ditentukan. *Tokenizing* sangat berguna ketika sebuah program pengolah teks memerlukan data sebuah kata yang tersusun dan terbagi menjadi array.

2.4 Stopword Removal

Stopword adalah kata-kata umum yang sering muncul. Stopword removal adalah proses penghapusan kata-kata yang termasuk ke dalam stopword biasanya dilakukan agar stemming menjadi efektif dan efisien. Contoh stopword bahasa Indonesia antara lain “yang”,”di”,”ke”,dll.

2.5 Stemming

Stemming adalah suatu proses pencarian bentuk dasar dari tiap kata yang berada pada suatu dokumen teks, selain untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen, juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk atau form yang berbeda karena mendapatkan imbuhan yang berbeda dengan menerapkan aturan morfologi bahasa Indonesia yang baik dan benar [1,6,7,10].

Proses stemming dilakukan dengan menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari awalan (*preffixes*) sisipan (*infixes*) maupun akhiran (*suffixes*), stemming dilakukan atas dasar asumsi bahwa kata-kata yang memiliki stem yang sama memiliki makna dasar yang sama[9].

Teknik stemming dapat dikategorikan menjadi 3 yaitu berdasarkan aturan dalam bahasa tertentu,berdasarkan kamus, dan berdasarkan kemunculan bersama. Salah satu tujuan utama dilakukan proses stemming adalah meningkatkan efisiensi dengan cara memilah isi dokumen menjadi unit-unit kecil yang akan menjadi penciri misalnya berupa kata,frase atau kalimat[12].

2.6 Algoritma

Algoritma adalah suatu perintah yang berisi langkah-langkah untuk menyelesaikan masalah. Algoritma berasal dari nama tokoh ilmuan islam pada masa itu yaitu Abu Ja’far Muhammad Ibu Musa Al Khawārizmi yang hidup sekitar abad ke-9. Dengan karya bukunya yang terkenal yaitu Al Jabar Wal Muqabala yang berarti “Buku Pemugaran dan Pengurangan”.

Pada awalnya kata algoritma adalah istilah yang merujuk kepada aturan-aturan aritmetis untuk menyelesaikan persoalan dengan menggunakan bilangan numerik arab (sebenarnya dari India, sepertitertulis pada judul di atas). Pada abad ke-18, istilah ini berkembang menjadi algoritma, yang mencakup semua prosedur

atau urutan langkah yang jelas dan diperlukan untuk menyelesaikan suatu permasalahan.

2.7 Algoritma Arifin Setiono

Algoritma Arifin Setiono merupakan algoritma yang digunakan untuk pencarian kata dasar pada dokumen teks dengan teknik *stemming*[5]. Input dari algoritma ini adalah dokumen teks yang diproses sehingga menghasilkan output berupa kata dasar. Algoritma Arifin Setiono mengasumsikan bahwa setiap kata memiliki dua awalan dan tiga akhiran,yaitu:

[AW1] + [AW2] + KD + [AK3] + [AK2] + [AK1]

Dimana AW = awalan, KD = kata dasar dan AK = akhiran [4].

Langkah – langkah algoritma arifin setiono dalam proses stemming isi dokumen teks.

1. Lakukan pemeriksaan setiap kata, siapkan variabel p1,p2,s1,s2,
2. Pemotangan dilakukan secara berurut, yaitu:
   a. Awalan I, hasil disimpan pada p1
   b. Awalan II, hasil disimpan pada p2
   c. Akhiran I, hasil disimpan dalam s1
   d. Akhiran II, hasil disimpan dalam s2
   e. Akhiran III, hasil disimpan dalam s3

   Setiap tahap pemotongan hasil dicek dalam kamus, jika ada dalam kamus algoritma selesai, jika tidak ada proses dilanjutnya ke pemotongan berikutnya.Jika sampai pada langkah 2.e. belum ditemukan dalam kamus, maka dilakukan proses kombinasi. Kata dasar yang dihasilkan dikombinasikan dengan imbuhan-imbuhan dalam 12 kombinasi, yaitu:

   a. Kata Dasar
   b. Kata Dasar + AK III
   c. Kata Dasar + AK III + AK II

d. Kata Dasar + AK III + AK II + AK I
e. AW I + AW II + Kata Dasar
f. AW I + AW II + Kata Dasar + AK III
g. AW I + AW II + Kata Dasar + AK III + AK II
h. AW I + AW II + Kata Dasar + AK III + AKII + AK I
i. AW II + Kata Dasar
j. AW II + Kata Dasar + AK III
k. AW II + Kata Dasar + AK III + AK II
l. AW II + Kata Dasar + AK III + AK II + AK I

2.8 Algoritma Porter

Algoritma kedua yang digunakan adalah algoritma Porter. Adapun langkah – langkah algoritma ini adalah sebagai berikut :

1. Hapus Particle,
2. Hapus Possesive Pronoun.
3. Hapus awalan pertama. Jika tidak ada lanjutkan ke langkah 4a, jika ada cari maka lanjutkan ke langkah 4b.
4. a. Hapus awalan kedua, lanjutkan ke langkah 5a.
   b. Hapus akhiran, jika tidak ditemukan maka kata tersebut diasumsikan sebagai root word. Jika ditemukan maka lanjutkan ke langkah 5b.
5. a. Hapus akhiran. Kemudian kata akhir diasumsikan sebagai kata dasar
   b. Hapus awalan kedua. Kemudian kata akhir diasumsikan sebagai root word.

Terdapat 5 kelompok aturan pada Algoritma Porter untuk bahasa indonesia ini.

Tabel 2.1 Aturan Untuk Inflectional Particle[3]

Akhiran	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
kah	NULL	2	NULL	mungkinkah→mungkin
lah	NULL	2	NULL	pergilah → pergi
pun	NULL	2	NULL	bukupun → buku

Tabel 2.2 Aturan Untuk inflectional Possesive Pronoun[3]

Akhiran	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
ku	NULL	2	NULL	bukuku → buku
mu	NULL	2	NULL	bukumu → buku
nya	NULL	2	NULL	bukunya → buku

Tabel 2.3 Aturan Untuk First Order Derivational Prefix[3]

Awalan	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
Meng	NULL	2	NULL	Mengukur → ukur
Meny	S	2	V....*	Menyapu →sapu
Men	NULL	2	NULL	Menduga→ duga Menuduh→ uduh
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 Penuduh→ uduh
Pem	NULL	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

Tabel 2.4 Aturan untuk Second order Derivational Prefix[3]

Awalan	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
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

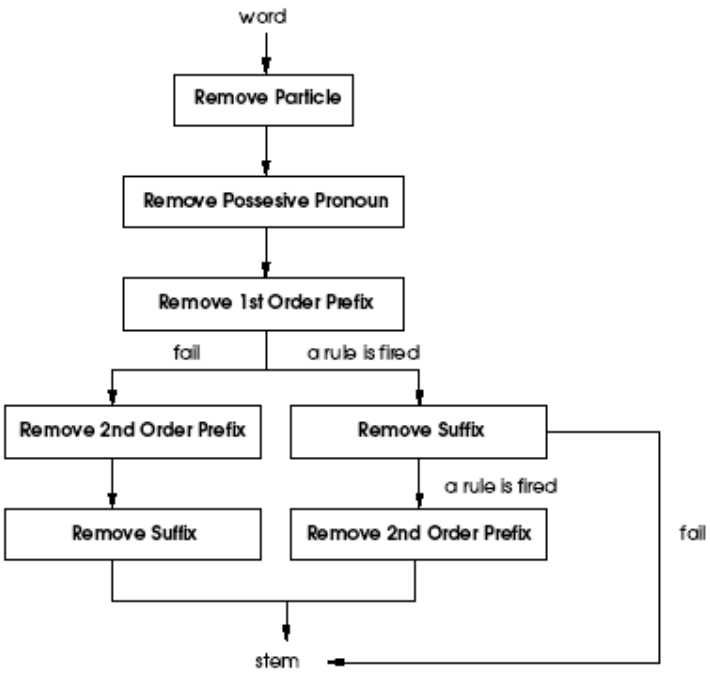
Tabel 2.5. Aturan untuk Derivational Suffix[3]

Akhiran	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
kan	NULL	2	Awalan € {ke,peng}	tarikan → tarik (meng)ambilkan ambil
an	NULL	2	Awalan € {di,meng,ter}	makanan →makan (per)janjian →janji
i	NULL	2	V|K..c1c1,c1 ≠ s,c2≠i dan Awalan € {ber,ke,peng}	tandai →tanda (men)dapati →dapat Pantai →panta

Pada Porter Stemmer untuk Indonesia perlu ditambahkan beberapa aturan dalam algoritma agar memberikan hasil yang lebih maksimal dan untuk mempermudah proses stem maka dibuatlah beberapa kamus kecil,antara lain sebagai berikut :

1. Kamus kata dasar yang dilekati partikel, untuk menyimpan kata dasar yang memiliki suku kata terakhir (partikel infleksional) serta kata tersebut tidak mendapat imbuhan apapun.Seperti : masalah
2. Kamus kata dasar yang dilekati partikel berprefiks untuk menyimpan kata dasar yang memiliki suku kata terakhir (partikel infleksional) dan mempunyai prefiks. Seperti : menikah.

3. Kamus kata dasar yang dilekati kata ganti milik, untuk menyimpan kata dasar yang memiliki suku kata terakhir (kata ganti infleksional) serta kata dasar tersebut tidak mendapatkan imbuhan apapun. Seperti : bangku.
4. Kamus kata dasar yang dilekati kata ganti milik berprefiks, untuk menyimpan kata dasar yang memiliki suku kata terakhir (kata ganti infleksional) dan mempunyai prefiks. Seperti : bersuku.
5. Kamus kata dasar yang dilekati prefiks pertama, untuk menyimpan kata dasar yang memiliki suku kata pertama (prefiks derivasional pertama) serta kata dasar tersebut tidak mendapatkan imbuhan apapun.. Seperti : median.
6. Kamus kata dasar yang dilekati prefiks pertama bersufiks, untuk menyimpan kata dasar yang memiliki suku kata pertama (prefiks derivasional pertama) dan mempunyai sufiks derivasional.Seperti : terapan.
7. Kamus kata dasar yang dilekati prefiks kedua, untuk menyimpan kata dasar yang memiliki suku kata pertama (prefiks derivasional kedua) serta kata dasar tersebut tidak mendapatkan imbuhan apapun. Seperti : percaya.
8. Kamus kata dasar yang dilekati prefiks kedua bersufiks, untuk menyimpan kata dasar yang memiliki suku kata pertama (prefiks derivasional) dan mempunyai sufiks derivasional. Seperti :perasaan.
9. Kamus kata dasar yang dilekati sufiks, untuk menyimpan kata dasar yang memiliki suku kata terakhir (sufiks derivasional). Seperti : pantaiProses stemming menggunakan Algoritma Porter dapat dilihat pada Gambar 2.1



Gambar 2.1 Algoritma Porter [3]

2.9 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan

MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.Performance tuning

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

4. Ragam tipe data

   MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

5. Perintah dan Fungsi

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).

6. Keamanan

   MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

7. Skalabilitas dan Pembatasan

   MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

8. Konektivitas

   MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

9. Lokalisasi

   MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

10. Antar Muka

    MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

11. Klien dan Peralatan

MySQL dilengkapi dengan berbagai peralatan (tool)yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

12. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle

2.10 Pengenalan UML

Dalam merancang tugas akhir ini penyusun akan menggunakan UML sebagai alat bantu (tools) dalam perancangan dan pendokumentasian sistem yang akan dibuat.

2.10.1 Pengertian UML

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering). UML Menggabungkan konsep berarah objek seperti Booch, OMT, OOSE, dsb sehingga menjadi bahasa pemodelan tunggal. UML menekankan pada apa yang dapat dikerjakan dengan metode-metode tersebut.

UML terdiri dari diagram-diagram, dimana setiap diagram didalam UML memperlihatkan sistem dari berbagai sudut pandang yang berbeda.

a. Class Diagram

Class diagram adalah sebuah grafik presentasi dari gambaran statis yang menunjukan sekumpulan model elemen yang terdeklarasi (statis), seperti kelas, tipe dan isinya serta hubungannya.

Sebuah kelas diagram terdiri dari sejumlah kelas yang dihubungkan dengan garis yang menunjukan hubunga antar kelas yang disebut *Associations*, contoh class diagram dapat dilihat pada Gambar 2.2.

Jenis-jenis *Associations* [13] yaitu :

1. *Aggregation*

   Associations yang menggambarkan hubungan antar kelas dimana kelas yang satu merupakan bagian dari kelas yang lainnya.

2. *Composition*

   Associations yang menggambarkan hubungan erat antar kelas dimana kelas *composite* mempunyai segala tanggung jawab untuk mengatur kelas lainnya dan kedua kelas mempunyai *lifetime* yang sama.

3. *Bidirectionality*

   Associations yang menghubungkan antara dua kelas atau lebih tapi tidak bergantung satu sama lainnya, sehingga apabila salah satu kelas dihilangkan, kelas yang lain dapat tetap digunakan.

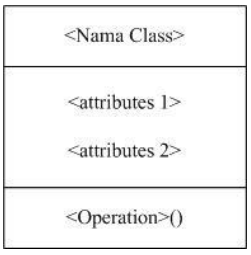
4. *Generalization*

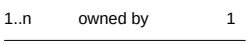
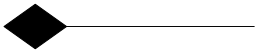
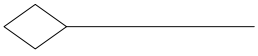
   Associations yang menghubungkan dua kelas atau lebih untuk membedakan antara kelas yang umum dengan kelas yang khusus.

5. *Inheritance*

   Associations yang meghubungkan duakelas atau lebih yang dapat menurunkan properties seperti attributes, operations antara kelas induk dengan kelas anak.

Tabel 2.6 komponen Class Diagram[13]

Nama Komponen	Keterangan	Simbol
Class	Class adalah blok-blok pembangun pada pemrograman berorientasi objek.sebuah class digambarkan sebagai sebuah kotak yang terdiri atas 3 bagian. Bagian atas adalah bagian nama dari class. Bagian tengah mendefinisikan property/atribut	

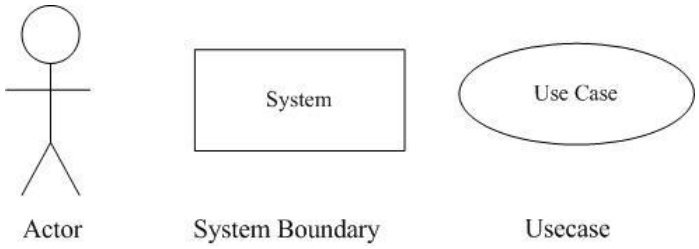
	class. Bagian akhir mendefinisikan method-method dari sebuah class	
Association	Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 class dan di lambangkan olwh sebuah garis yang menghubungkan antara 2 class. Garis ini bisa melambangkan tipe-tipe relationship dan jiga dapat menampilkan hukum-hukum multiplisitas pada sebuah relationship.(contoh : One-to one, one-to-many, many-to-many)	
Composition	Jika sebuah class tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi Composition terhadap class tempat dia bergantung tersebut. Sebuah relationship composition digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid	
Depedency	Kadangkala sebuah class menggunakan class yang lain. Hal ini disebut dependency. Umumnya penggunaan dependency digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain. Sebuah dependency dilambangkan sebagai sebuah panah bertitik-titik	
Aggregation	Aggregation mengindikasikan keseluruhan bagian relationship dan biasanya disebut sebagai relasi	

b. *UseCase Diagram*

Usecase diagram adalah sebuah gambaran dari fungsi sistem yang dipandang dari sudut pandang pemakai yang memperkenalkan suatu sistem yang akan dibangun.

1. *Actor* adalah segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi.
2. *System boundary* menunjukan cakupan dari sistem yang dibuat dan fungsi dari sistem tersebut.
3. *Usecase* adalah gambaran fungsionalitas dari suatu sistem sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

   Berikut ini dari gambar 2.2 tiga komponen sistem dalam use case diagram :



Gambar 2.2 Komponen-komponen *Use case* [13]

Jenis-jenis Use Case Relationships [13] antara lain :

1. *Association*

   Garis yang menghubungkan antara actor dengan use case.
2. *Extend*

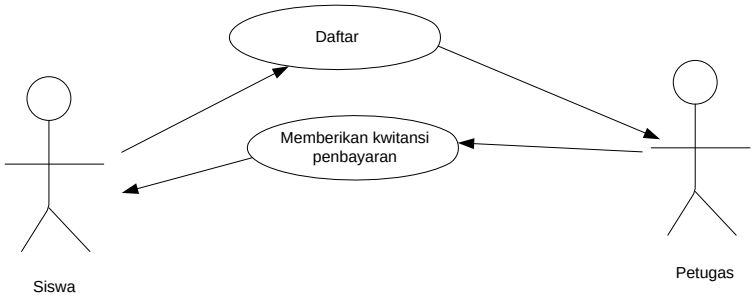
   Menghubungkan antara dua atau lebih use case yang merupakan tambahan dari base use case yang biasanya untuk mengatasi kasus pengecualian.
3. *Generalization*

   Hubungan antara use case umum dengan use case yang lebig khusus.
4. *Include*

   Menghubungkan antara 2 atau lebih use case untuk menunjukan use case tersebut merupakan bagian dari base use case.

Untuk lebih jelasnya diberikan contoh diagram dari gambar 2.3. pada gambar 2.3 dijelakan tentang pendaftaran.

siswa mendaftar kemudian petugas memberikan kwitansi pembayaran.

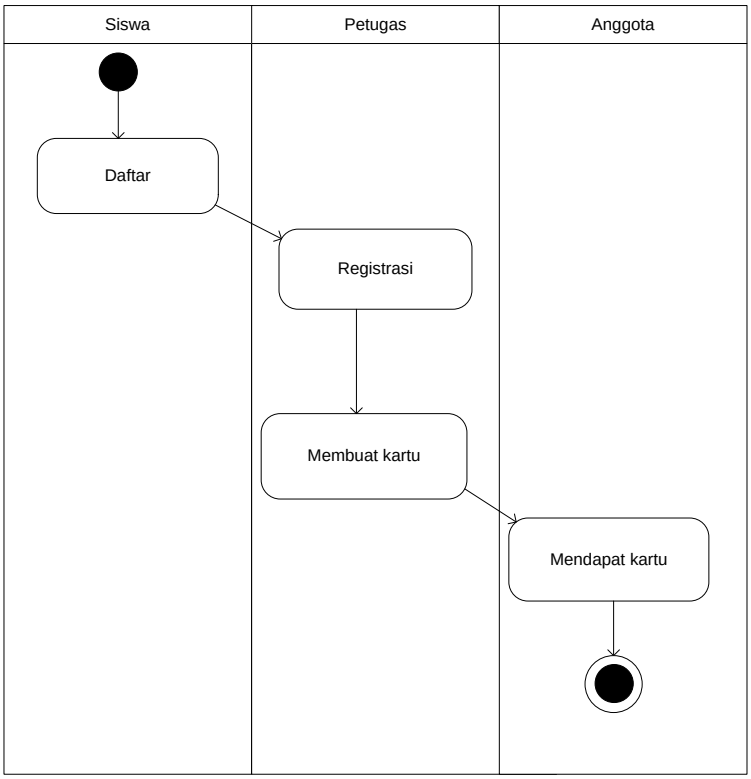


Gambar 2.3 *Usecase Diagram*[13]

c. *Activity diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *usecase*. Contoh *activity diagram* dapat dilihat pada Gambar 2.4. Untuk lebih jelasnya diberikan contoh diagram dari gambar 2.4. pada gambar 2.4 terdapat diagram pendaftaran.

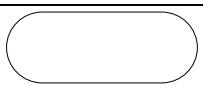
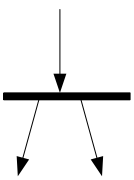
table pendaftaran yaitu Siswa harus mendaftar terlebih dahulu kepada petugas, kemudian petugas registrasi dan membuatkan kartu untuk siswa tersebut, setelah kartu tersebut di buat maka siswa tersebut mendapatkan kartu dan sah menjadi anggota.

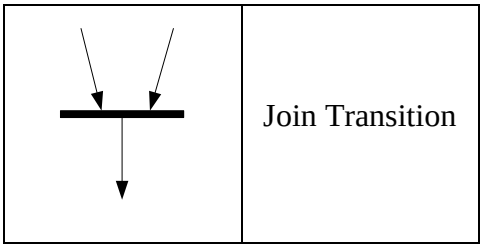


Gambar 2.4 *Activity diagram*[13]

Berikut ini merupakan komponen-komponen yang digunakan dalam *Activity Diagram* :

Tabel 2.7 komponen-komponen *Activity Diagram*[13]

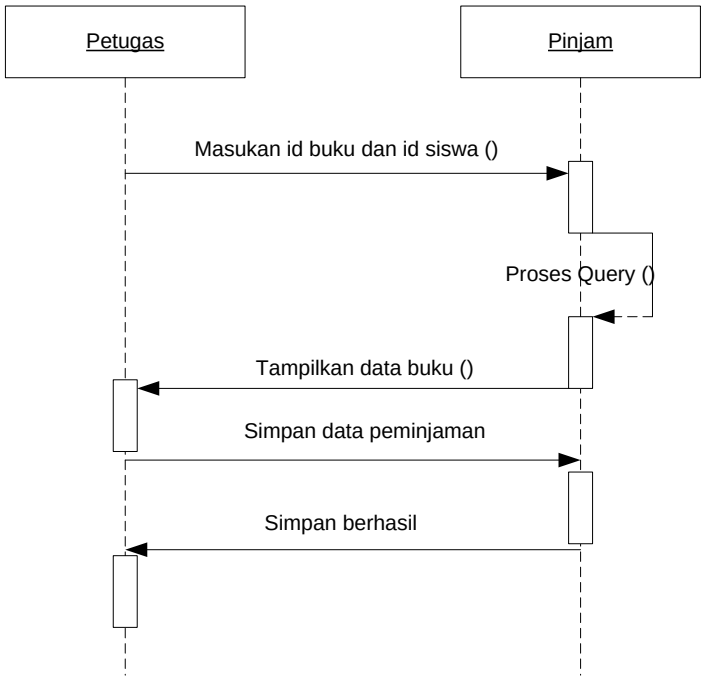
Simbol	keterangan
	Initial State
	Final State
	State
	Fork Transition



d. *Sequence Diagram*

Sebuah sequence diagram [13], menunjukan urutan pertukaran pesan yang dilakukan oleh sekumpulan objek atau aktor yang mengerjakan pekerjaan. Contoh sequence dapat dilihat pada Gambar 2.5.

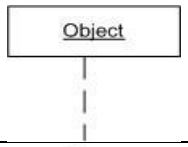
Untuk lebih jelasnya diberikan contoh diagram dari gambar 2.5. pada gambar 2.5 terdapat diagram pendaftaran. Pada diagram pendaftaran petugas dan daftar anggota sebagai objek, petugas memasukan data siswa pada taftar anggota, setelah itu disimpan data tersebut kedalam database daftar anggota kemudian sistem memberikan pesan kepada petugas bahwa pesan tersebut sudah berhasil disimpan.



Gambar 2.5 *Sequence diagram*[13]

Berikut ini merupakan komponon-komponen yang digunakan dalam *Sequence Diagram* :

Tabel 2.8 komponen-komponen *Sequence Diagram*[13]

Simbol	Keterangan
	Object Lifeline
	Activation
	Message
	Message Call
	Message return

2.11 Pengujian Software

Pengujian dilakukan dengan mengeksekusi data uji dan mengecek apakah fungsional perangkat lunak bekerja dengan baik.

2.11.1 Black box Testing

Pengujian *Black Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dieksekusi pada perangkat lunak dan kemudian keluar dari perangkat lunak dicek apakah telah sesuai yang diharapkan.

Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori :

1. Fungsi-fungsi yang tidak benar atau hilang

2. Kesalahan interface Kesalahan dalam struktur data atau akses database eksternal Kesalahan kinerja
3. Inisialisasi dan kesalahan terminasi

2.12 Konsep Perangkat Lunak pendukung

perangkat lunak pendukung merupakan perangkat yang digunakan untuk mendukung dalam perencanaan perancangan sistem yang akan digunakan.

2.12.1 Microsoft Visual Studio

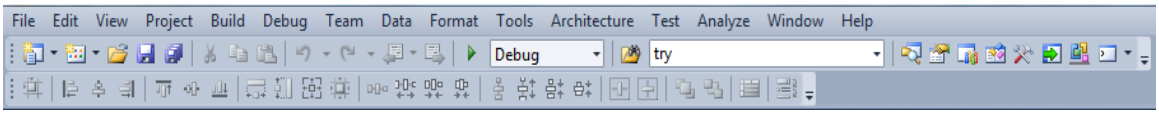
Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun managed code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

Untuk membuat suatu program aplikasi pada visual studio, maka diperlukan suatu struktur aplikasi atau komponen yang digunakan, berikut beberapa aplikasi yang penulis gunakan pada visual studio antara lain :

a. Toolbar

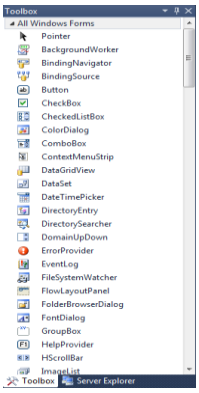
Terdiri atas beberapa komponen yang digunakan untuk membuat objek pada form dan dapat mempercepat pengaksesan perintah – perintah yang ada dalam pemrograman, dapat dilihat pada gambar 2.6.



Gambar 2.6 ToolBar Visual Studio

b. Toolbox

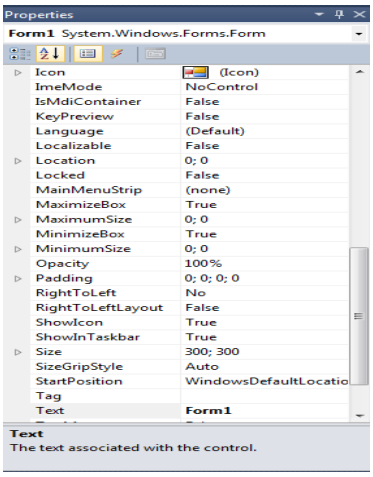
Toolbox Adalah sebuah jendela yang berisi tombol kontrol yang akan user gunakan untuk mendesain atau membangun sebuah form atau report. Toolbox terdiri atas beberapa tombol untuk mengendalikan tampilan, berikut gambar toolbox pada visual studio dapat dilihat pada gambar 2.7:



Gambar 2.7 Tampilan Toolbox Visual Studio

c. Jendela Propertis

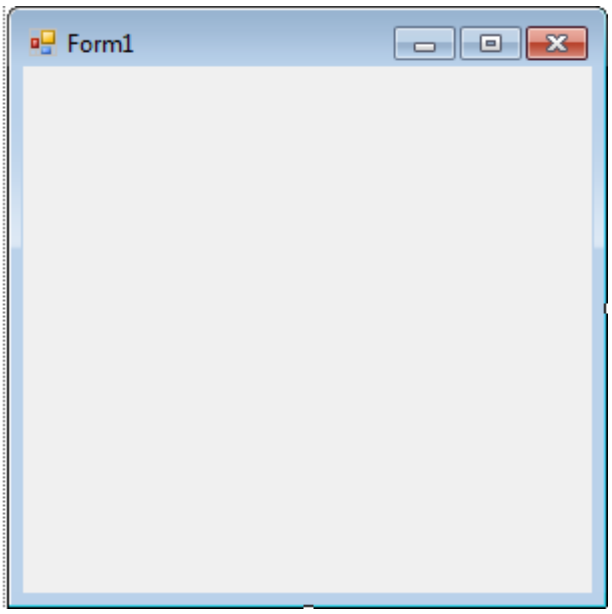
Jendela propertis adalah jendela yang berfungsi sebagai tempat mengedit properti pada suatu objek yang terpilih pada saat mendesain atau membangun aplikasi bagi user. Berikut gambar jendela propertis dapat dilihat pada gambar 2.8



Gambar 2.8 Tampilan Jendela Propertis

d. Form

Form merupakan tempat untuk mendesain atau membangun sebuah program, didalamnya bisa ditanamkan sebuah coding agar form tersebut dapat menjalankan program yang telah didesain oleh user. Berikut gambar form dapat dilihat pada gambar 2.9:



Gambar 2.9 Tampilan Form

2.13 Teori pengujian metode

Pengujian dengan menggunakan data ujin untuk menguji semua eleme program (datainternal,loop,logika,keputusan dan jalur). Data uji dibangkitkan dengan mengetahui struktur internal. Pada aplikasi stemming akan dilakukan pengukuran akurasi dilakukan dengan membandingkan term hasil stemming dengan kamus sebagai pembandingnya.

Perhitungan akurasi dilakukan dengan rumus sebagai berikut :

$$\text{Akurasi} = \frac{Jumlah\ term\ yang\ benar}{jumlah\ seluruh\ term} \text{ X } 100\% \quad [16]$$