BAB 2
LANDASAN TEORI

2.1. *Email Spam* dan *Spam Filtering*

Pengertian *email* atau *e-mail* dapat diketahui dari kepanjangan *email* itu sendiri. Kata *Email* merupakan singkatan dari *Electronic Mail*, yang dalam bahasa indonesia memiliki arti 'surat elektronik'. *Email* merupakan sistem yang memungkinkan pesan berbasis teks untuk dikirim dan diterima secara elektronik melalui komputer atau telepon genggam atau HP. Pengertian *email* jika ingin dibuat lebih spesifik menjadi sebagai berikut; *email* adalah cara pengiriman data, file teks, foto digital, atau file-file audio dan video dari satu komputer ke komputer lainnya, dalam suatu jaringan komputer [5]. Jaringan komputer ini bisa berupa jaringan komputer intranet maupun jaringan komputer internet.

Email spam adalah *email* yang tidak diminta oleh pengguna (*unsolicitied email*) yang di kirim ke banyak orang. Contoh *email* yang berisi *spam* adalah: iklan, undian, informasi palsu, phishing, penipuan, dan lain sebagainya [6].

Ciri-ciri dari *email epam* bisa dikatakan tidak pasti dengan perkembangan zaman, namun biasanya memiliki karakteristik sebagai berikut:

a. *Spam* berisi penawaran barang, *software*, video, musik atau berbau pornografi.
b. Isi *spam* terkadang berisi sesuatu hal yang tidak kita lakukan, misalnya anda adalah pemenang undian (lotere), padahal anda belum pernah mengikutinya sama sekali.
c. *Spam* biasanya berisi himbauan kepada anda untuk mengisi identitas pribadi rahasia yang anda miliki.
d. *Spam* ada juga yang isinya menawarkan produk obat-obatan atau kesehatan. Misalnya saja, cara meninggikan badan secara cepat atau menurunkan berat badan secara cepat.
e. *Spam* sifatnya tidak merusak, tetapi membuat anda tidak nyaman.

f. *Spam* biasanya dijadikan sebagai media penghubung virus, misalnya ketika anda mengklik *link* yang diberikan oleh *email spam* sebut, padahal itu hanya tipuan agar virus bisa tereksekusi.

Spam juga menyebabkan pemborosan uang bagi *user* yang menggunakan koneksi *dial-up*. Selain itu spam juga membuang bandwith dan dapat menyebabkan penerima di bawah umur mengakses situs-situs yang tidak seharusnya. Banyaknya spam memunculkan kerugian dalam hal memakan banyak sumber daya dan memerlukan waktu untuk menghapusnya. Menyaring *spam* secara manual sulit dilakukan untuk ukuran dokumen *email* yang sangat besar. Oleh karena itu diperlukan suatu metode yang dapat mengklasifikasikan *spam* dan bukan *spam* (ham) secara otomatis. Metode penyaringan secara otomatis ini biasanya disebut dengan *spam filtering* [2].

Spam filtering adalah suatu proses yang memisahkan *email* berdasarkan kategorinya baik *spam* maupun bukan *spam* atau biasa disebut ham. Dalam pengklasifikasian *email spam*, dibutuhkan suatu sistem cerdas yang dapat memilih atau mengklasifikasikan *email spam* dan bukan *spam* (ham) secara baik dan benar. Pada proses pengklasifikasian *spam email* tersebut dapat diterapkan beberapa metode/algoritma yang dapat membantu sistem dalam melakukan pengklasifikasian [7].

2.2. Metode Naïve Bayes

Teori Probabilitas Bayesian digunakan untuk menghitung probabilitas terjadinya suatu peristiwa berdasarkan pengaruh yang didapat dari pengujian. Di dalam teori probabilitas dan statistika, teorema Bayes juga dikenal sebagai sebuah teori dengan dua penafsiran berbeda. Dalam penafsiran Bayes, teori ini menyatakan seberapa jauh derajat kepercayaan subjektif harus berubah secara rasional ketika ada petunjuk baru. Teori ini merupakan dasar dari statistika Bayes dan memiliki penerapan dalam sains, rekayasa, ilmu ekonomi, teori *games*, kedokteran dan hukum [8].

Naïve Bayes *Classifier* merupakan sebuah metoda klasifikasi yang berakar pada teorema Bayes. Ciri utama dari Naïve Bayes *Classifier* ini adalah asumsi yang

sangat kuat (naif) akan independensi dari masing-masing kondisi kejadian. Sebelum menjelaskan Naïve Bayes *Classifier* ini, akan dijelaskan terlebih dahulu Teorema Bayes yang menjadi dasar dari metoda tersebut. Pada teorema Bayes, bila terdapat dua kejadian yang terpisah (misalkan A dan B), maka teorema Bayes dirumuskan pada Rumus 2.1:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{2.1}$$

Dengan ketentuan:

h	:	Hipotesis
D	:	Data
P(h)	:	Probabilitas independen dari h
P(D)	:	Probabilitas inependen dari D
P(D|h)	:	Probabilitas kondisional D jika diberikan h
P(h|D)	:	Probabilitas kondisional h jika diberikan D

Naïve Bayes *Classifier* adalah metode klasifikasi yang didasarkan pada nilai probabilitas. Klasifikasi adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu obyek. Oleh karena itu kelas yang ada tentulah lebih dari satu. Penentuan kelas dari suatu dokumen dilakukan dengan cara membandingkan nilai probabilitas suatu sampel berada di kelas yang satu dengan nilai probabilitas suatu sampel berada di kelas yang lain.

Sebagai contoh, ahli statistik ingin mengklasifikasikan sampel kucing ke dalam jenis kelaminnya. Oleh karena itu, terdapat dua kelas yaitu jantan dan betina. Suatu sampel kucing akan diklasifikasikan ke dalam satu kelas saja, entah itu jantan atau betina, dengan melihat petunjuk-petunjuk yang ada (misalnya berat badan, panjang ekor). Penentuan kelas yang cocok bagi suatu sampel dilakukan dengan cara membandingkan nilai probabilitas untuk masing-masing kelas, dan mengambil kelas dengan nilai probabilitas yang tinggi. Secara matematis klasifikasi dirumuskan pada Rumus 2.2:

$$C_{NB} = argmax_{C \in C} P(c) \prod_{i=1}^{n} P(f_i|c) \quad (2.2)$$

2.3. Metode *Smoothing Dirichlet*

Pada proses penghitungan Naïve Bayes terdapat sedikit keraguan apabila ada peluang yang bernilai 1. Oleh karena itu digunakan *Laplace Smoothing* yaitu penambahan dengan angka 1 sehingga tidak ada peluang yang akan bernilai 1. Selain *laplace smoothing* ada beberapa teknik *smoothing* lainnya.

Pada penelitian Quan Yuan, Gao Cong, dan Nadia M. Thalmann (2012) dilakukan 4 jenis teknik *smoothing* dari Naïve Bayes untuk proses klasifikasi teks. Teknik tersebut adalah *Jelinek-Mercer (JM) Smoothing*, *Dirichlet (Dir) Smoothing*, *Absolute Discounting (AD) Smoothing*, dan *Two-Stage (TS) Smoothing* [8]. Dari hasil penelitian tersebut diperlihatkan bahwa metode *smoothing* mampu meningkatkan tingkat akurasi dari Naïve Bayes untuk proses klasifikasi teks.

Perhitungan peluang setiap dokumen menggunakan empat metode *smoothing* ini masih menggunakan kaidah Naïve Bayes namun berbeda pada Rumus pendugaan parameter P($tk|c$), sesuai Rumus 2.3 :

$$P\mu(t|c) = \frac{T_{ct} + \mu \, . P(t|C)}{\sum_{t \epsilon V} T_{ct'} + \mu} \quad (2.3)$$

Dimana :

T_{ct}	:	Banyaknya t dalam data latih dari kelas c
μ	:	Koefisien control
$T_{ct'}$	:	Banyaknya t yang cocok dengan data latih
c	:	Kombinasi yang memiliki status *spam*/ham
t	:	Kombinasi yang terbentuk dari data uji

Dan rumus untuk $P(t|C)$ adalah Rumus 2.4 :

$$P(t|C) = \frac{count(t,c)}{count(c)} \quad (2.4)$$

Dimana:

t	:	kombinasi yang terbentuk dari Proses Kombinasi
c	:	kombinasi yang memiliki status *spam*/ham
$count(t, c)$	:	Jumlah kombinasi t berstatus c yang sesuai dengan data latih
$count(c)$	:	Jumlah seluruh kombinasi berstatus c yang pada data latih

2.4. Teori Kombinasi

Teori Kombinasi atau Kombinatorial adalah cabang matematika yang mempelajari pengaturan objek-objek. Solusi yang ingin diperoleh dengan kombinatorial ini adalah jumlah cara pengaturan objek-objek tertentu di dalam himpunannya [9].

Kombinatorial didasarkan pada hasil yang diperoleh dari suati percobaan (*experiment*). Percobaan adalah proses fisik yang hasilnya diamati. Contoh-contoh percobaan dan hasilnya adalah:

a. Melempar dadu: Enam hasil percobaan yang mungkin untuk pelemparan dadu adalah muka dadu 1, 2, 3, 4, 5 atau 6.
b. Melempar koin uang Rp100: Hasil percobaan melempar koin 100 ada dua kemungkinan, muka koin yang bergambar rumah gadang atau muka koin yang bergambar wayang.
c. Memilih lima orang wakil dari 100 orang mahasiswa: Hasil yang diperoleh adalah perwakilan yang beranggotakan lima orang mahasiswa. Kemungkinan perwakilan yang dapat dibentuk banyak sekali.
d. Menyusun jumlah kata yang panjangnya 5 huruf yang dapat dibentuk dari huruf-huruf a, b, c, d, e, tidak boleh ada huruf yang berulang di dalam kata. Hasil yang diperoleh addalah string yang disusun oleh huruf-huruf tersebut, misalnya abcde, abced, acdeb, dan seterusnya.

Bentuk khusus permutasi adalah kombinasi. Jika pada permutasi ururtan kemunculan diperhitungkan, maka pada kombinasi, urutan kemunculan diabaikan. Urutan acb, abc, dan bca dianggap sama dan dihitung sekali.

Interpretasi Kombinasi

a. Persoalan kombinasi, *C(n,r)*, sama dengan menghitung banyaknya himpunan bagian yang terdiri dari r elemen yang dpat dibentuk dari himpunan dengan n elemen. Dua atau lebih himpunan bagian dengan elemen-elemen yang sama dianggap sebagai himpunan yang sama, meskipun urutan elemen-elemennya berbeda.

   Jumlah kombinasi dapat dihitung dengan rumus

$$C(n,r) = \frac{n!}{r!\,(n-r)!} \tag{2.5}$$

   Dimana:

   n : Jumlah objek dalam himpuanan

   r : Objek yang diambil dari *n* buah objek

b. Persoalan Kombinasi , $C(n,r)$, dapat dipandang sebagai cara memilih r buah elemen dari n buah elemen yang ada, tetapi urutan elemen dalam susunan hasil pemilihan tidak penting.

2.5. *Confusion Matrix*

Confusion Matrix (Kohavi dan Provost, 1998) 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 [10].

Tabel 2. 1 Tabel *Confusion Matrix*

		Predicted Class	
		Positif	**Negatif**
Actual Class	**Positif**	True Positives	False Positive
	Negatif	False Negative	True Negative

True positives adalah jumlah *record* positif yang diklasifikasikan sebagai positif, *false positives* adalah jumlah *record* positif yang diklasifikasikan sebagai negatif, *false negatives* adalah jumlah *record* negatif yang diklasifikasikan sebagai positif, *true negatives* adalah jumlah *record* negatif yang diklasifikasikan sebagai negatif. Setiap kolom dari *Confusion Matrix* merupakan contoh di kelas yang telah diprediksi, sedangkan setiap baris mewakili contoh di kelas yang sebenarnya. Setelah didapat *true positives, false positives, true negatives* dan *false negatives*, selanjutnya hitung nilai *precision* dan akurasinya. *Precision* adalah ukuran terhadap suatu kelas yang telah diprediksi. Berikut Rumus dari *precision* dan akurasi.

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.6}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.7}$$

Dimana:

TP : Jumlah *True Positive*

FP : Jumlah *False Positive*

TN : Jumlah *True Negative*

FN : Jumlah *False Negative*

2.6. Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap 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 19 Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe [11].

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).

2.7. MySQL

MySQL adalah *multiuser database* yang menggunakan bahasa *structured Query Language* (SQL). MySQL dalam operasi *client server* melibatkan server daemon MySQL di sisi *server* dan berbagai macam program serta *library* yang berjalan di sisi *client.*MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku bahwa MySQL yang mampu menyimpan data lebih dari 40 database, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data [12].

SQL adalah bahasa standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai bahasa yang standar industri. Dengan menggunakan SQL, proses akses database menjadi lebih *user-friendly* dibandingkan dengan menggunakan DBASE atau *Clipper* yang masih menggunakan perintah-perintah pemrograman.Sebagai *database server* yang memiliki konsep *database modern*, MySQL memiliki banyak keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL [13].

1. *Portability*

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X *Server*, dan lain-lain.

2. *Open source*

   MySQL didistribusikan secara *open source*, sehingga dapat digunakan secara bebas.

3. *Multi-user*

MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.Hal ini memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan.

4. *Performance tuning*

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

5. *Column types*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integet, float, double,char, varchar, text, blob, date, time, timestamp, year, set,* dan *enum.*

6. *Command* dan *Function*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam *query.*

7. *Security*

   MySQL memiliki lapisan sekuritas seperti *level subnetmask*, nama*host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.

8. *Scalability* dan *Limits*

   MySQL mampu menangani *database* dengan skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 juta *table* serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tablenya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan *client* menggunakan *protocol* TCP/IP, *Unix Socet* (UNIX), atau *Named Pipes* (NT).

10. *Localization*

    Deteksi pesan kesalahan pada *client* dengan menggunakan lebih dari 20 bahasa.

11. *Interface*

    Interface terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*application Programming Interface*).

12. *Client* dan *Tools*

    Dilengkapi dengan berbagai *Tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.

13. Struktur tabel

    Struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibanding database lainnya semacam PostgresSQL ataupun Oracle

2.8. *Object Oriented Analysis and Design* (OOAD)

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memerika *requirement* (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem [14].

2.9. *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah termasuk ke dalam rumpun jenis pemodelan notasi grafis yang didukung oleh meta-model tunggal. Pemodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang dibangun dengan *object-oriented* (OO). UML merupakan standar terbuka yang dikelola oleh *Open Management Group* (OMG) yang berada dibawah naungan perusahaan-perusahaan konsorsium terbuka [14]. UML merupakan bahasa pemodelan yang terdiri banyak model diantaranya adalah :

1. *Use Case Diagram*

   Use case adalah teknik untuk menggambarkan fungsional sebuah sistem. *Use case* diagram memperlihatkan hubungan diantara *actor* dan *use case*. Aktor mempresentasikan seorang *user* atau subsistem lain yang akan berinteraksi dengan sistem. Sedangkan *use case* merupakan urutan kejadian yang menggambarkan interaksi antara user dengan sistem. Fungsionalitas sistem

didefinisikan ke dalam *use case* dari sudut eksternal sistem yang berguna untuk diuji kelayakan sistem.

2. *Sequence Diagram*

   Sequence diagram menggambarkan interaksi diagram yang menunjukan bagaimana kelompok-kelompok objek saling berkolaborasi dalam beberapa behavior, Sebuah *Sequence diagram* secara khusus menjabarkan behavior sebuah skenario tunggal. Diagram tersebut menunjukan sejumlah objek contoh dan pesan-pesan yang melewati objek ini dalam *use case*. *Sequence diagram* adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan *output* tertentu.

3. *Class Diagram*

   Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendifinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

2.10. Apache

Server HTTP *Apache* atau Server Web/WWW *Apache* adalah server web yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta *platform* lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini mengunakan HTTP. *Apache* memiliki fitur-fitur canggih seperti

pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain. *Apache* juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI, *Grafik User Interface*) yang memungkinkan penanganan server menjadi mudah. *Apache* merupakan perangkat lunak kode sumber program terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*. *Apache* adalah komponen server web dari paket perangkat lunak LAMP (Linux, *Apache*, MySQL, PHP/Perl/bahasa pemrograman Python). Tetapi pada pengembangannya sekarang ini *apache* dapat di install dan dijalankan di berbagai macam platform, seperti Unix, BSD, Microsoft Windows dan Novell Netware serta platform lainnya. Pada awal mulanya, *Apache* merupakan perangkat lunak sumber terbuka yang menjadi alternatif dari server web Netscape (sekarang dikenal sebagai Sun Java System Web Server) [13].

Apache mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunanya. Beberapa dukungan *Apache* :

1. Kontrol Akses. Kontrol ini dapat dijalankan berdasarkan nama *host* atau nomor *IP*.
2. *Common Gateway Interface (CGI)* , yang paling terkenal untuk digunakan adalah *Practical Extraction and Report Language (Perl)* didukung oleh *Apache* dengan menempatkannya sebagai modul (*mod_perl*).
3. *Personal Home Page (PHP Hypertext Processor)* : program dengan metode semacam *CGI*, yang memproses teks dan bekerja di *server*. *Apache* mendukung *PHP* dengan menempatkannya sebagai salah satu modulnya (*mod_php*). Hal ini membuat kinerja *PHP* menjadi lebih baik.
4. *Secure Socket Layer (SSL)* : adalah mekanisme pembungkus data yang berlalu-lalang antara *client* dan *server* dengan suatu enkripsi data. Apache mendukung *SSL* dengan menempatkannya sebagai salah satu modulnya (*mod_ssl*).