SKRIPSI

KLASIFIKASI SPAM PADA EMAIL MENGGUNAKAN LONG SHORT-TERM MEMORY (LSTM) DAN SUPPORT VECTOR MACHINE (SVM)

Disusun dan diajukan oleh

AZZAHRA MUBARIKAH

H071171524

PROGRAM STUDI SISTEM INFORMASI

DEPARTEMENT MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

KLASIFIKASI SPAM PADA EMAIL MENGGUNAKAN LONG SHORT-TERM MEMORY (LSTM) DAN SUPPORT VECTOR MACHINE (SVM)

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

AZZAHRA MUBARIKAH

H071171524

PROGRAM STUDI SISTEM INFORMASI

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

LEMBAR PERNYATAAN KEOTENTIKAN

Yang bertanda tangan di bawah ini :

Nama	: Azzahra Mubarikah
Nim	: H071171524
Program Studi	: Sistem Informasi
Jenjang	: S1

Menyatakan dengan ini bahwa kasya tulisan saya berjudul:

KLASIFIKASI SPAM PADA EMAIL MENGGUNAKAN LONG SHORT-TERM MEMORY (LSTM) DAN SUPPORT VECTOR MACHINE (SVM)

Adalah benar hasil karya sendiri bukan merupakan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini merupakan hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 11 Juni 2021

AZZAHRA MUBARIKAH

NIM. H071171524

KLASIFIKASI SPAM PADA EMAIL MENGGUNAKAN LONG SHORT-TERM MEMORY (LSTM) DAN SUPPORT VECTOR MACHINE (SVM)

Disusun dan diajukan oleh:

AZZAHRA MUBARIKAH

H071171524

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui

Pembimbing Utama	Pembimbing Pertama
Dr. Hendra, S.Si., M.Kom.	**Supri Bin Hj Amir, S.Si., M.Eng.**
NIP. 19760102 200212 1 001	**NIP. 19880504 201903 1 012**

Ketua Program Studi

Dr. Muhammad Hasbi, M.Sc
NIP.196307201989031003

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama	: Azzahra Mubarikah
NIM	: H071171524
Program Studi	: Sistem Informasi
Judul Skripsi	: Klasifikasi spam pada email menggunakan LONG SHORT-TERM MEMORY (LSTM) dan SUPPORT VECTOR MACHINE (SVM)

Telah berhasil mempertahankan di hadapan dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

		Tanda Tangan
1. Ketua	: Dr. Hendra, S.Si., M.Kom.	(……)
2. Sekretaris	: Supri Bin Hj. Amir, S.Si., M.Eng.	(……)
3. Anggota	: Dr. Eng. Armin Lawi, M.Eng.	(……)
4. Anggota	: Nur Hilal A Syahrir, S.Si., M.Si.	(…Tidak Ada…)

Ditetapkan di : Makassar
Tanggal : 11 Juni 2021

KATA PENGANTAR

Alhamdulillahi Robbil ’alamin, segala puji syukur dipanjatkan atas kehadirat Allah SWT, atas segala karunia dan rahmat-Nya sehingga penulis dapat menyelesaikan penulisan Proposal Tugas Akhir ini dengan judul “**Klasifikasi Spam pada Email Mmenggunakan Metode Long Short-Term Memory dan Support Vector Machine**”. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Pada kesempatan kali ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua tercinta, Ayahanda **Muslimin** dan Ibuda **Hadirah** yang telah mendidik penulis dengan penuh kesabaran, cinta dan kasih sayang yang tak pernah putus, terima kasih atas segala dukungan , nasihat, serta doa yang tak henti-hentinya diberikan kepada penulis selama menjalani proses Pendidikan. Untuk kakak penulis **Akhmad Sudirman** serta keluarga yang senantiasa memberikan dukungan dan doa. Terima kasih atas segala perhatian yang diberikan kepada penulis.

Penghargaan yang tulus dan ucapan terimakasih dengan penuh keikhlasan juga penulis ucapkan kepada :

1. Rektor Universitas Hasanuddin, Ibu **Prof. Dr. Dwia Aries Tina Pulubuhu** beserta jajarannya.
2. Dekan Fakultas Matematika dan Ilmu Pengetahuna Alam (FMIPA), Bapak **Dr.Eng. Amiruddin,M.Si** beserta jajarannya.
3. **Dosen pengajar Jurusan Matematika** yang telah membekali ilmu kepada penulis selama menjadi mahasiswa di Jurusan Matematika. Serta Staf Jurusan Matematika yang telah membantu banyak dalam perkuliahan.

4. Bapak **Dr Hendra, S.Si M.Kom** dan Bapak **Supri bin Hj Amir, S.Si, M.Eng** atas kesediaan, kesabaran, dan kesetiaan untuk membimbing dan membagi ilmunya kepada penulis dalam penyusunan skripsi ini.
5. Bapak **Dr. Eng. Armin Lawi, M.Eng** dan Ibu **Nur Hilal A Syahrir, S.Si, M.Si** atas kesediaannya untuk memberikan saran dan arahan kepada penulis dalam penyususunan skripsi ini.
6. Kepada **Hamba Allah**, terima kasih atas bantuan dan kesabarannya selama pengerjaan skripsi ini, tanpa anda penulis tidak akan sehebat ini. Sekali lagi terima kasih.
7. Kepada **Rafly Ahmad Mubin** yang selama setahun belakangan ini selalu menemani, membantu dan memberikan dukungan kepada penulis dalam melaksanakan kuliah, tugas kuliah dan skripsi ini.
8. Seluruh **Angkatan 2017 SISTEM INFORMASI**, terkhusus kepada **Muhammad Fitrah, A. Amaliah Dwi Ayu, Edo Bayu Pamungkas, Kennedy, Muh Amirullah, Ahmad Ali Winandar, Muhaimin Anwar, Muhammad Arizki** terima kasih telah banyak membantu mengerjakan tugas dan mengajari materi selama perkuliahan**,** kepada **Gepeng** (**Geby Nionsi, Nurfadila Firdani Salam, Eka Fitriani, Siti Rabiatul, Mutiah Amanah Arum, Eka Kurnia, Nur Khairunisa, Fadillah Putri Taha, Mir Ataini Aprilia**) terima kasih sudah hadir sejak kita maba sampai sekarang, kalian banyak membantu selama kuliah penulis.
9. Saudara-saudari **MIPA 2017** tanpa terkecuali.
10. Kepada saudara tak sedarah penulis **Annisa Salsa Dwishany** terima kasih atas segala perhatiannya dan dukungannya kepada penulis selama sepuluh tahun.
11. Kepada **Miftha Huljannah, Karina Eka Pratiwi, Yuni Satria Putri, Nadia Desriyanti** atas segala dukungannya kepada penulis.
12. Kepada **Andi Nurwasiawasiawati K** terima kasih atas bantuan dan nasihat kepada penulis.

13. Kepada semua pihak yang tidak dapat disebutkan satu persatu, atas segala bentuk kontribusi, partisipasi, dan motivasi yang diberikan kepada penulis selama ini. Semoga yang telah kalian berikan dilipatgandakan oleh Allah SWT.

Makassar, 11 Juni 2021

Azzahra Mubarikah

H071171524

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama	: Azzahra Mubarikah
NIM	: H071171524
Programa Studi	: Sistem Informasi
Departemen	: Matematika
Fakultas	: Matematika dan Ilmu Pengetahuan Alam
Jenis Karya	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Prediktor Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas tugas akhir saya yang berjudul:

Klasifikasi Spam Pada Email Menggunakan Long Short-Term Memory (LSTM) Dan Support Vector Machine (SVM)

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal diatas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada 11 Juni 2021

Yang menyatakan

(Azzahra Mubarikah)

ABSTRAK

Teknologi, informasi dan komunikasi telah berkembang dengan pesat, khususnya dibidang internet. Dengan adanya internet setiap orang dapat mengirim dan menerima pesan dari orang satu ke orang lainnya yang lebih dikenal sebagai email. Eletronic mail (email) merupakan media komunikasi yang murah, cepat dan mudah penggunaannya, memiliki sifat data berupa teks yang semi terstruktur dan memiliki dimensi yang tinggi. Penggunaan email yang sangat intens menimbulkan penyalahgunaan email sehingga berpotensi untuk merugikan orang lain biasa kita kenal sebagai spam. Email spam adalah e-mail yang dikirimkan kepada ribuan penerima (recipient) yang pada umumnya berisi akan konten-konten merugikan dan berbahaya. Para pengguna email tidak perlu khawatir dengan adanya spam tersebut, karena akan diklasifikasi menggunakan metode algoritma long short-term memory dan support vector machine. Long Short-Term memory (LSTM) merupakan struktur lanjutan dari Recurrent Neural Network (RNN) yang dapat menangani masalah klasifikasi text. Support Vector Machine (SVM) merupakan salah satu metode yang memberikan hasil terbaik untuk klasifikasi pada data biner. Hasil penelitian klasifikasi spam menggunakan Long Short-Term memory (LSTM) dengan nilai Accuracy 99,1%, f1-score pada kelas spam sebesar 96,2% dan f1-score pada kelas ham sebesar 99,4%. dan Support Vector Machine (SVM) dengan nilai Accuracy 97,4%, f1-score pada kelas spam sebesar 88,6% dan f1-score pada kelas ham sebesar 98,5%. Menunjukkan bahwa Long Short-Term memory (LSTM) lebih baik dibandingkan dengan Support Vector Machine (SVM).

Kata Kunci : Email, Long Short-Term Memory, Support Vector Machine

ABSTRACT

Technology, information and communication have developed rapidly, especially in the internet sector. With the internet, everyone can send and receive messages from one person to another, which what we called an email. Electronic mail (e-mail) is a form of communication that is cheap, fast and easy to use, the characteristic of its data is in the form of semi-structured text and high dimensions. The misuse of email has the potential to harm other people, known as spam. Spam email is an e-mail sent to thousands of recipients which generally contains harmful and dangerous content. Email users do not need to worry about spam, because it will be classified using a long short-term memory algorithm and a support vector machine. Long Short-Term memory (LSTM) is an advanced structure of the Recurrent Neural Network (RNN) that can handle text classification problems. Support Vector Machine (SVM) is one of the methods that provides the best results for classification on binary data. The results of the research on the classification of spam using Long Short-Term memory (LSTM) with an Accuracy value of 99.1%, the f1-score in the spam class of 96.2% and the f1-score in the ham class of 99.4%. Meanwhile, Support Vector Machine (SVM) has an Accuracy value of 97.4%, f1-score in the spam class of 88.6% and f1-score in the ham class of 98.5% shows that Long Short-Term memory (LSTM) is better than Support Vector Machine (SVM).

KEYWORD : Email, Long Short-Term Memory, Support Vector Machine

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Teknologi, informasi dan komunikasi telah berkembang dengan pesat, khususnya dibidang internet. Dengan adanya internet, segala informasi dan berita dapat diterima dan diakses oleh setiap orang. Bahkan dengan internet, setiap orang dapat mengirim dan menerima pesan dari orang satu ke orang lainnya, yang lebih dikenal sebagai email.

Eletronic mail (email) merupakan salah satu layanan internet yang paling banyak digunakan. Email adalah media komunikasi yang murah, cepat dan mudah penggunaannya. Format email terdiri dari sebuah *envelope*, beberapa *field header*, sebuah *blank line* dan *body*. Menurut Rachli (2007), Email memiliki sifat data berupa teks yang semi terstruktur dan memiliki dimensi yang tinggi (Fitriyanto, 2019). Berdasarkan (Pratiwi & Ulama, 2016), penelitian yang dilakukan Radicati group jumlah akun email tahun 2012 diperkirakan sebanyak 3,3 miliar akun. Dengan rincian 75% pemilik akun adalah perseorangan atau pribadi, sisanya sebanyak 25% digunakan oleh perusahaan dan diprediksi pada tahun 2016 akan menjadi 4,3 miliar akun. Penggunaan email yang sangat intens ini menimbulkan dampak positif dan negatif karena pada kenyataannya tidak semua orang menggunakan email dengan baik dan bahkan ada banyak sekali penyalahgunaan email sehingga berpotensi untuk merugikan orang lain. Email yang disalah gunakan ini biasa kita kenal sebagai spam atau junkmail (email sampah) yang mana email tersebut berisikan iklan, penipuan dan bahkan virus.

Email spam dapat didefinisikan sebagai "*unsolicited bulk e-mail* "yaitu e-mail yang dikirimkan kepada ribuan penerima (*recipient*) yang pada umumnya berisi akan konten-konten merugikan dan berbahaya. Adanya hal tersebut

sangat mengganggu pengguna email, terlebih para perusahaan atau pekerja individual yang sehari-harinya menggunakan sarana email sebagai penunjang kerjanya. Pengklasifikasian email spam sangat diperlukan dalam menangani hal ini. Para pengguna email tidak perlu khawatir dengan adanya spam tersebut, karena akan diklasifikasi menggunakan metode algoritma *long short-term memory* dan *support vector machine*.

Menurut Kristiansen, dkk (2005), Spam adalah aksi pengiriman pesan bisnis atau iklan kepada sejumlah kantor besar atau email yang sebetulnya tidak ingin diterima oleh penerima tersebut (Fitriyanto, 2019). Spam merupakan sebuah tindakan yang dilakukan berulang-ulang dapat juga dikatakan pengirim informasi melakukan spam (spammer) secara sengaja untuk berbuat kejahatan atau pengirim spam tidak disengaja sehingga tidak mengetahui bahwa diri nya telah melakukan spam. Seiring dengan pertumbuhan internet dan Email, pertumbuhan spam terjadi dalam beberapa tahun terakhir. Spam dapat berasal dari setiap lokasi di seluruh dunia dimana akses internet tersedia. Jumlah pesan spam terus meningkat pesat. Pada saat ini, lebih dari 85% dari total Email yang masuk adalah spam (Aisyah, 2020).

Menurut Chen, dkk (2009) mengatakan bahwa SVM merupakan salah satu metode pengklasifikasian yang memberikan hasil terbaik (Pratiwi & Ulama, 2016). Model klasifikasi SVM berfungsi sebagai mesin inferensi dari sistem yang ahli, yang membangun *hyperplane* keputusan yang optimal dalam *feature space* ini dan memisahkan menjadi dua kelas, yaitu spam dan ham. SVM sudah dikenal sebagai algoritma pembelajaran terbaik untuk klasifikasi pada data biner. Keuntungan dari SVM adalah bahwa akurasinya tidak menurunkan bahkan ketika banyak fitur yang hadir. Oleh karena itu, pendekatan tersebut telah diadopsi untuk penyaringan email spam.

Long Short-Term Memory (LSTM), yang merupakan struktur lanjutan dari *Recurrent Neural Network* (RNN) yang memiliki mekanisme *gat*e termasuk *cell memory*. LSTM memiliki ide utama untuk mengingat nilai-nilai bobot

menggunakan *memory cell*. Pada pengaplikasiannya, LSTM memiliki masalah kompleksitas yang sangat tinggi.

Dari permasalahan tersebut peneliti kemudian tertarik untuk melakukan penelitian dengan mengangkat judul " Klasifikasi Spam Email Menggunakan *Long Short-Term Memory (LSTM) dan Support Vector Machine (SVM)* "

1.2 Rumusan Masalah

Rumusan masalah dalam penulisan Tugas Akhir ini adalah :

1. Bagaimana cara kerja *Long Short-Term Memory* dalam mengklasifikasi spam email?
2. Bagaimana cara kerja *Support Vector Machine* dalam mengklasifikasi spam email?
3. Bagaimana tingkat f1 - *score* dan *accuracy* yang diperoleh antara *Long Short-Term Memory* dan *Support Vector Machine* pada klasifikasi email spam?

1.3 Batasan Masalah

Batasan masalah dalam penulisan Tugas Akhir ini adalah :

1. Metode klasifikasi yang digunakan adalah *Long Short-Term Memory* dan *Support Vector Machine*
2. Klasifikasi spam yang dilakukan hanya pada Email
3. Data yang digunakan adalah dataset Spam

1.4 Tujuan Penelitian

Tujuan Penelitian dalam penulisan Tugas Akhir ini adalah:

1. Untuk mengetahui cara kerja *Long Short-Term Memory* dalam mengklasifikasi spam email
2. Untuk mengetahui cara kerja *Support Vector Machine* dalam mengklasifikasi spam email

3. Untuk mengetahui tingkat f1 - *score* dan *accuracy Long Short-Term Memory* dan *Support Vector Machine* dalam mengklasifikasi email spam

1.5 Manfaat Penelitian

Manfaat penelitian dalam penulisan Tugas Akhir ini adalah :

1. Dapat menerapkan metode *Long Short-Term Memory* dan *Support vector Machine* dalam mengklasifikasikan spam pada email
2. Sebagai referensi bagi penelitian lain yang hendak melakukan penelitian sejenis

BAB 2
TINJAUAN PUSTAKA

2.1 Data Mining

Menurut Utama, dkk (2014) *Data mining* adalah suatu proses menemukan hubungan dan kecenderungan dengan memeriksa sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika (Setiono & Pardede, 2019).

Menurut Han & Kamber (2011:6) menjelaskan bahwa "*Data mining* merupakan pemilihan atau "menggali" pengetahuan dari jumlah data yang banyak." Berbeda dengan Segall, dkk (2008:127) menjelaskan "*Data mining* disebut penemuan pengetahuan atau menemukan pola yang tersembunyi dalam data. *Data mining* adalah proses menganalisis data dari perspektif yang berbeda dan meringkas menjadi informasi yang berguna". Bisa disimpulkan *Data mining* adalah Proses menganalisis data yang banyak dan membuat suatu pola untuk menjadi informasi yang berguna (Fitriyanto, 2019).

Salah satu rangkain proses, *data mining* dapat dibagi menjadi beberapa tahap proses. Tahap-tahap tersebut bersifat interaktif, pemakai terlibat langsung atau dengan perantaraan *knowledge base*. Tahap-tahap *data mining* menurut Sukardi dkk (2014) adalah sebagai berikut (Hayuningtyas, 2017):

1. Pembersihan Data (*Data Cleaning*)

   Pembersihan data merupakan proses menghilang-kan noise.
2. Integrasi Data (*Data Integration*)

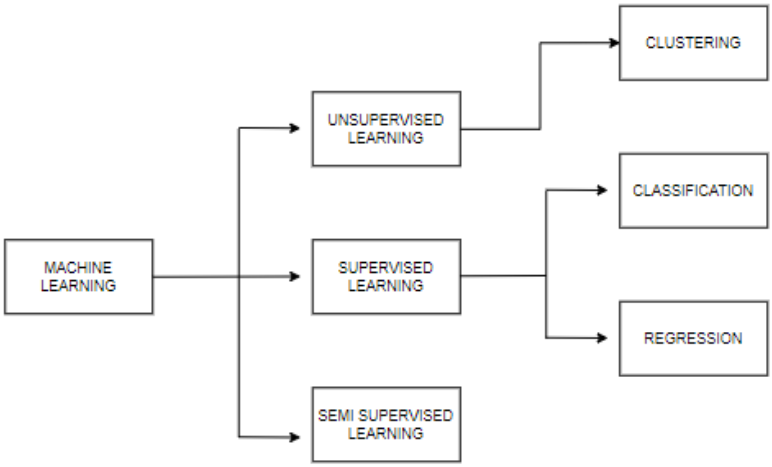
   Integrasi data merupakan penggabungan data dari berbagai database ke dalam satu database baru
3. Seleksi Data (*Data Selection*)

Data yang ada pada database sering kali tidak semuanya dipakai, oleh karena itu hanya data yang sesuai untuk dianalisis yang akan diambil dari database

4. Transformasi Data (*Data Transformation*)
   Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam *Data Mining*
5. Proses Mining
   Merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data. Beberapa metode yang dapat digunakan berdasarkan pengelompokkan *Data Mining*
6. Evaluasi Pola (*Pattern Evaluation*)
   Untuk mengidentifikasi pola-pola menarik ke dalam *knowledge base* yang ditemukan.
7. Presentasi Pengetahuan (*Knowledge Presentation*)
   Merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna.

2.2 Machine Learning

Machine learning seperti namanya menunjukkan bahwa menggunakan *machine* dalam arti dapat belajar dengan sendirinya seperti yang dilakukan manusia dari pengalaman. Lebih tepatnya, seluruh proses belajar dan memprediksi informasi dari pengalamannya (data) yang dikenal sebagai *machine learning*. Di sini, dalam contoh *machine learning* yang merupakan situasi yang sangat umum seperti, *online customer support*, layanan *media social*, *virtual personal assistant*, pemfilteran Email dan *Malware*, dsb. Mungkin sangat sedikit orang yang tidak mengetahui bahwa mereka diarahkan oleh gelombang *machine learning*.



Gambar 2.1 *Machine Learning*

Dari gambar 2.1 *Machine Learning* terbagi atas tiga bagian berdasarkan *input* dan *output* yaitu *unsupervised learning* yang meliputi *clustering, supervised* yang meliputi *classification* dan *regression*, dan *semi supervised learning* yang merupakan penggabungan *supervised learning* dan *unseupervised learning* (Raj, dkk, 2018).

Saat ini, *supervised learning*, *semi-supervised learning*, dan *unsupervised learning* adalah bagian dari *machine learning. Machine learning* adalah pemanfaatan informasi yang ada sebagai pengalaman untuk belajar agar dapat menggunakan informasi ini untuk membuat keputusan yang lebih baik di masa depan. Sebagai NLP (*natural language processing*), matematika dalam sains, dan teknologi komputer yang diatur oleh *machine learning*, menunjukkan eksplorasi lain dari penelitian dan teknik. Selain itu, NLP sangat popular pada sistem saraf dengan mencapai hasil yang baik di bidang penyematan kata. Pencapaian ini tidak dapat dipisahkan dari machine learning dan dampak dari dua bidang itu akan jadi masalah.

2.3 Text Mining

Text mining merupakan salah satu cabang ilmu *data mining* yang mengacu pada pencarian informasi, dengan menganalisis data berupa dokumen teks. *Text*

mining, mengacu pada proses mengambil informasi berkualitas tinggi dari teks dan tujuannya adalah mencari kata kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

Text mining adalah aplikasi *data mining* untuk *file* teks tidak terstruktur atau kurang terstruktur (Turban, 2005). *Data mining* memanfaatkan infrastruktur dari data yang disimpan untuk mengekstraksi informasi tambahan yang bermanfaat. *Text mining* beroperasi dengan informasi yang kurang terstruktur. *Text mining* berfungsi untuk :

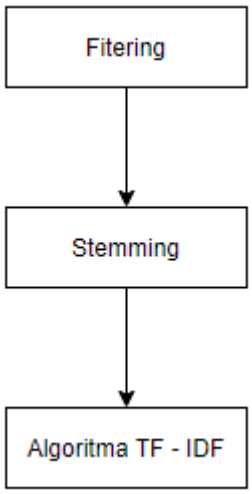
1. Menemukan isi dokumen yang tersembunyi termasuk hubungan yang bermanfaat.
2. Menghubungkan dokumen diluar divisi – divisi yang sebelumnya tidak diketahui.
3. Mengelompokkan dokumen – dokumen berdasarkan tema umum.

Dalam penerapannya, text mining memiliki beberapa permasalahan umum seperti jumlah data yang besar, *high dimensional*, struktur yang berubah – ubah, ambigu *dependency data* dan *data noise* (Even, dkk 2002). Ekstraksi merupakan bentuk paling dasar dari *text mining*. Proses ekstraksi akan memetakan informasi dari data tidak terstruktur kedalam suatu format yang terstruktur. Struktur data yang paling sederhana dalam *text mining* adalah vektor atau daftar kata – kata yang telah dibobot. Kata yang paling penting di dalam suatu teks didaftarkan bersama dengan suatu ukuran dari kepentingan relatifnya. Untuk melakukan hal tersebut, dalam *text mining* akan dilakukan beberapa tahap berikut (Turban, 2005):

1. Menghapus kata yang umum digunakan (*filtering*).
2. Mengganti kata – kata dengan kata dasarnya (*stemming*), misalnya learning menjadi learn.
3. Tahap yang terakhir dalam *text mining* adalah tahap *analyzing* yaitu tahap penentuan seberapa jauh keterhubungan antar kata – kata terhadap dokumen

yang ada. Untuk melakukan analisa pada tahap *analyzing* dapat digunakan algoritma TF/IDF (*Term Frequency – Inverse Document Frequency*).

Tahapan dari *text mining* ditunjukkan pada gambar 2.2 :



Gambar 2.2 Bagan Tahapan *Text Mining*

2.4 Klasifikasi

Menurut Widiasari dan Bayu (2013), Klasifikasi adalah proses dengan model yang menggambarkan dan membedakan kelas data atau konsep. Model merupakan analisis objek yang label kelasnya belum diketahui (Hayuningtyas, 2017).

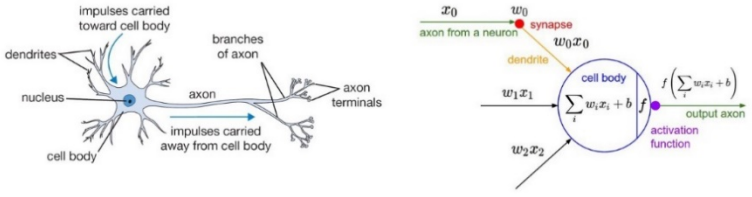
Klasifikasi memiliki dua proses yaitu membangun model klasifikasi dari sekumpulan kelas data yang sudah didefinisikan sebelumnya (*training data*) dan menggunakan model tersebut untuk klasifikasi data uji serta mengukur akurasi dari model. Model klasifikasi dapat disajikan dalam berbagai macam model klasifikasi seperti *decision trees, bayesian classification, k-nearest neighbourhood classifier, neural network, classification (IF-THEN) rule*, dan lain-lain. Klasifikasi dapat dimanfaatkan dalam berbagai aplikasi seperti diagnosa medis, *selective marketing*, pengajuan kredit perbankan dan Email.

Klasifikasi merupakan salah satu metode dalam *data mining* yang dapat mengklasifikasikan email sebagai spam atau non spam. Adapun pengklasifikasian ini berdasarkan karakteristik dari spam (Hayuningtyas, 2017):

1. Alamat pengirim yang tidak benar
2. Pemalsuan header mail untuk menyembunyikan email sesungguhnya
3. Identitas penerima tidak nyata
4. Kamus alamat penyerang. Alamat email yang berada dalam "To" memiliki variasi alamat email penerima.
5. Isi subject tidak berhubungan dengan isi email.
6. Isi email memiliki sifat keragu-raguan.
7. Unsubscribe tidak bekerja pada spam mail.
8. Mengandung script tersembunyi.

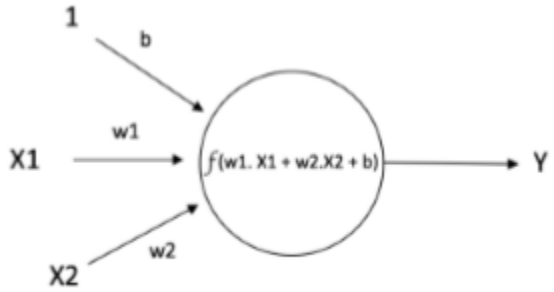
2.5 Neural Network

Neural network atau jaringan saraf merupakan sebuah model dari jaringan saraf otak manusia yang ditiru oleh banyak peniliti untuk diadopsi cara kerjanya di berbagai bidang kajian seperti biologi, fisika, ilmu komputer, dll (Setiawan, 2018). Di bidang ilmu komputer terdapat istilah *artifical neural networks* (ANN). Konsep ANN diadopsi dari jaringan saraf otak pada manusia sehingga ANN biasa disebut jaringan saraf tiruan. Model yang dibuat oleh ANN dapat untuk beradaptasi, belajar dan mengelompokkan berbasis pemrosesan parallel. Pada gambar 2.3 menunjukkan perbedaan neural network pada otak dan tiruan.



Gambar 2.3 Perbedaan Neural Network Pada Otak dan Tiruan

Adapun beberapa hal yang mendasar pada komputasi *neural network* adalah *neuron*, atau biasa disebut juga dengan *node*. *Node* dapat menerima input dari *node* lainnya atau dari sumber eksternal kemudian dihitung untuk mendapatkan sebuah *output*. Setiap *input* memiliki bobot (w) tersendiri, yang dimana bobot ini diberikan dengan dasar hubungannya dengan *input* lainnya. *Node* tersebut menggunakan fungsi ke dalam input yang telah diberi bobot. Contoh ilustrasi *neuron* dapat dilihat pada gambar 2.4.



Gambar 2.4 Ilustrasi *Neural Network*

Sebuah *neuron input* berupa x1 dan x2 dengan bobotnya yaitu w1 dan w2. Selain dari input dan bobot juga terdapat inputan lain berupa 1 dengan bobot b (bias). Fungsi utama dari bias adalah untuk menyediakan setiap *node* dengan sebuah nilai *constant* yang dapat dilatih (sebagai tambahan selain dari *input* normal yang diterima). *Output* pada *neuron* tersebut yaitu *Y* yang merupakan hasil perhitungan dari sebuah fungsi *non-linear* f yang disebut dengan fungsi aktivasi. Tujuan dari sebuah fungsi aktivasi adalah untuk menunjukkan *non-linear* terhadap *output* dari *neuron* tersebut (Setiawan, 2018).

2.5.1 *Batch Size* dan *Epoch*

Batch size adalah jumlah sampel data yang disebarkan ke *neural network*, sedangkan *epoch* adalah ketika seluruh dataset sudah melalui proses *training* pada *neural network* sampai dikembalikan ke awal untuk sekali putaran, karena satu *epoch* terlalu besar untuk dimasukkan (*feeding*)

kedalam komputer maka dari itu perlu membaginya kedalam satuan kecil (*batches*) (Imam, 2018).

2.5.2 Dropout

Dropout adalah metode regularisasi di mana input dan koneksi berulang ke unit LSTM secara probabilistik dikecualikan dari aktivasi dan pembaruan bobot saat melatih jaringan. Ini memiliki efek mengurangi *overfitting* dan meningkatkan performa model (Brownlee, 2017).

2.6 Backpropagation

Algoritma *backpropagation* digunakan oleh *neural network* untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma *Backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* tersebut, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu (Fausett, 1994).

Pada dasarnya, pelatihan dengan metode *backpropagtion* terdiri atas tiga langkah, yaitu sebagai berikut:

a. Data dimasukkan ke input jaringan (*feedforward*)
b. Perhitungan dan propagasi balik dari *error* yang bersangkutan
c. Pembaharuan (*adjustment*) bobot

Jika nilai *error* yang dihasilkan lebih besar dari batas *error* yang digunakan dalam sistem, maka akan dilakukan koreksi bobot. Koreksi bobot dapat dilakukan dengan menambah atau menurunkan nilai bobot.

Jika sinyal keluaran terlalu besar dari target yang ditentukan maka bobotnya diturunkan, sebaliknya jika sinyal keluaran terlalu kecil dari target yang ditentukan maka bobotnya dinaikkan. Koreksi bobot akan dilakukan sampai selisih target dan sinyal keluaran sekecil mungkin atau sama dengan batas *error*. Untuk melakukan koreksi bobot akan dilakukan penelusuran ke belakang seperti ditunjukkan dengan tanda panah mundur.

Adapun cara kerja dari *Backpropagation:*

1. Tiap-tiap *output* menerima target pola yang berhubungan dengan input pembelajaran, hitung informasi errornya.

$$\delta_K = \check{y} - y$$

2. Kemudian hitung koreksi bobot (yang akan digunakan untuk memperbaiki nilai bobot (w_{jk}) dengan laju pembelajaran α.

$$\Delta w_{jk} = \alpha \delta_k Z_j, \text{ (k = 1,2,…,m dan j = 0,1,2,…, p)}$$

3. Tiap-tiap unit tersembunyi (Z_j, j = 1, 2, 3 … . . p) menjumlahkan delta inputnya (dari unit-unit yang ada pada lapisan dibawahnya).

$$\delta_{\_in\ j} = \sum_{k=1}^{m} \delta_k w_{jk}$$

Hitung informasi errornya, dimana $\theta^{(l)^T}$ adalah matriks bobot yang ada di *hidden* ke *layer* yang ada dibelakangnya $\delta^{(l+1)}$ adalah *hidden* ke *layer* ke i+1 dari arah belakang kemudian dikalikan dengan fungsi aktivasi.

$$\delta_j = \theta^{(l)^T} \theta^{(l+1)} * (z\_in_j)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}dengan laju pembelajaran α.

$$\Delta v_{ij}\ \alpha \vartheta_j x_i, \text{ (j = 1, 2 … p;i = 0,1,2, … n)}$$

4. Hitung semua perubahan bobot

a. Perubahan bobot garis yang menuju ke unit keluaran.

$$w_{jk} \text{ (baru)} = w_{jk}\text{(lama)} + \Delta w_{jk}$$

b. Perubahan bobot garis yang menuju ke unit tersembunyi.

$$v_{ij} \text{ (baru)} = v_{ij}\text{(lama)} + \Delta w_{ij}$$

c. Proses iterasi selesai.

Keterangan :

ϑ_k = Informasi tentang kesalahan pada unit Y_k yang disebarkan kembali ke unit tesembunyi.

$\boldsymbol{\alpha}$ = Laju Pembelajaran (*Learning Rate*).

2.7 Fungsi Aktivasi

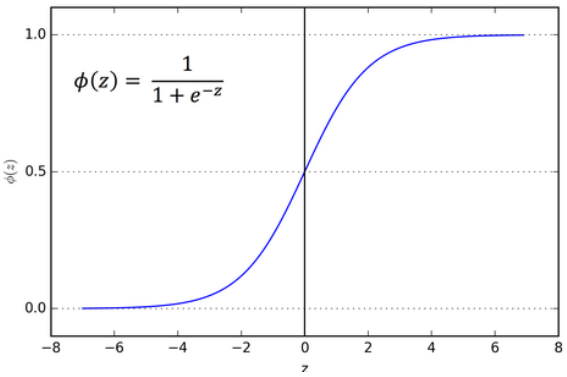
Fungsi aktivasi adalah untuk memberikan kemampuan *network* agar dapat melakukan tugas *non-linear*. Tanpa fungsi aktivasi, *neural network* hanyalah kombinasi operasi *linear* yang hanya dapat melakukan tugas-tugas yang *linear* pula. Padahal kebanyakan kasus nyata di lapangan merupakan kasus *non-linear* (Umam,2018).

Terminologi utama yang perlu dipahami untuk fungsi nonlinier adalah:

a. Turunan atau *Diferensial*: Perubahan dalam sumbu y perubahan pada sumbu x. Ini juga dikenal sebagai kemiringan.
b. Fungsi monotonik: Fungsi yang seluruhnya tidak meningkat atau tidak menurun.

Fungsi Aktivasi Nonlinier terutama dibagi berdasarkan rentang atau kurva

1. Fungsi Aktivasi *Sigmoid*



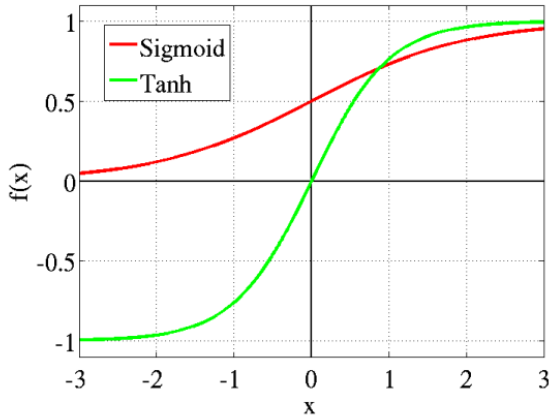
Gambar 2.5 Fungsi Aktivasi *Sigmoid*

Berdasarkan gambar 2.5 fungsi aktivasi *sigmoid* berada di antara 0 hingga 1 yang dapat dinyatakan pada persamaan berikut :

$$f(x) = \frac{1}{1 + e^{-z}}$$

Fungsi ini digunakan untuk model di mana kita harus memprediksi *probabilitas* sebagai *output*. Karena *probabilitas* apa pun hanya ada di antara rentang 0 dan 1.

2. Fungsi Aktivasi Tanh atau Hyperbolic Tangent



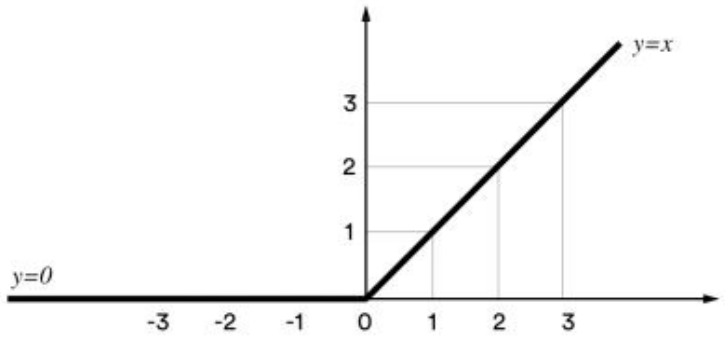
Gambar 2.6 Fungsi Aktivasi Tanh Dan Fungsi Aktivasi *Sigmoid*

Berdasarkan gambar 2.6 fungsi aktivasi tanh atau hyperbolic tangent berada di antara -1 hingga 1 yang dapat dinyatakan pada persamaan berikut :

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

Keuntungan dari fungsi ini yaitu inputan negative akan dipetakan negative dan inputan nol akan dipetakan mendekati nol dalam grafik tanh.

3. Fungsi Aktivasi ReLU atau *Rectified Linear Unit*



Gambar 2.7 Fungsi Aktivasi ReLU

Berdasarkan gambar 2.7 fungsi aktivasi ReLU atau *Rectified Linear Unit* berada di rentang 0 hingga tak terbatas yang dapat dinyatakan pada bentuk ReLU sebagai berikut :

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Pada fungsi ReLU jika nilai negatif segera menjadi nol yang menurunkan kemampuan model untuk menyesuaikan atau melatih dari data dengan benar. Itu berarti setiap masukan negatif yang diberikan ke fungsi aktivasi ReLU mengubah nilai menjadi nol segera di grafik, yang pada gilirannya memengaruhi grafik yang dihasilkan dengan tidak memetakan nilai negatif dengan tepat. Tetapi jika nilainya di atas nol, maka hasilnya tetap nilai tersebut.

2.8 *Electronic Mail* (E-Mail)

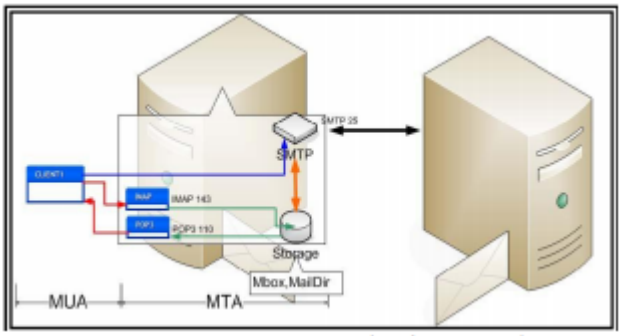
Menurut Widiasari & Bayu (2013), *Email* adalah cara yang efektif untuk berkomunikasi satu dengan lainnya. *E-mail* (*Electronic Mail*) atau surat elektronik sudah mulai dipakai pada tahun 1960y. Menurut Widiasari dan Bayu (2013) *Email* terdiri dari 3 komponen (Hayuningtyas, 2017) :

1. *Envelope* : Proses ini digunakan oleh *Mail Transport Agent* (MTA) untuk melihat rute atau jalur pesan.
2. *Header* : Digunakan sebagai informasi mengenai *e-mail* tersebut, mulai dari alamat pengirim, penerima, subjek dan lain-lain.
3. *Body* : Merupakan isi pesan dari pengirim ke penerima. Dalam mail *body* juga terdapat *file attachment* yang digunakan untuk mengirimkan *e-mail* berupa *file* (*mail attachment*).

Menurut Muhammad Rachli (2007), *Eletronic mail* atau lebih sering kita kenal dengan singkatan *Email* merupakan salah satu layanan internet yang paling banyak digunakan. *Email* adalah media komunikasi yang murah, cepat dan mudah penggunaannya. *Email* memiliki sifat data berupa teks yang semi terstruktur dan memiliki dimensi yang tinggi (Fitriyanto, 2019).

Berdasarkan pendapat dari para ahli diatas maka dapat ditarik kesimpulan bahwa *email* merupakan media elektronik yang dapat dgunakan untuk berkomunikasi satu dengan yang lain dengan cepat dan mudah penggunaanya.

Electronic-Mail (E-Mail) merupakan sebuah metode untuk mengirimkan pesan dalam bentuk digital. Pesan ini biasanya dikirimkan melalui *medium* internet. Sebuah pesan elektronis terdiri dari isi, alamat pengirim, dan alamat-alamat yang dituju. 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 . Jaringan komputer ini bisa berupa jaringan komputer intranet maupun jaringan komputer internet. Sistem e-mail yang beroperasi di atas jaringan berbasis pada *model store and forward*. Sistem ini mengaplikasikan sebuah sistem *server* e-mail yang menerima, meneruskan, mengirimkan, serta menyimpan pesan-pesan *user,* dimana *user* hanya perlu untuk mengkoneksikan komputer mereka ke dalam jaringan. E-mail dapat dianalogikan dengan kotak surat yang ada di kantor POS sedangkan server e-mail dapat diibaratkan sebagai kantor POS. Dengan analogi ini sebuah *mail server* dapat memiliki banyak account e-mail yang ada didalamnya. Penulisan e-mail dan *e-mail* sama saja. Namun lebih direkomendasikan untuk menuliskannya sebagai e-mail. Pada RFC, *spelling* e-mail yang digunakan adalah mail, dan sebuah e- mail dinamakan sebagai sebuah *message*. RFC yang baru dan grup IETF membutuhkan penulisan e-mail yang konsisten dari segi kapitalisasinya, penggunan *underscorenya,* serta ejaannya.



Gambar 2.8 Cara Kerja Email

Cara kerja *e-mail* yang dapat dilihat pada Gambar 2.8 menunjukkan bahwa e-mail yang dikirim belum tentu akan diteruskan ke komputer penerima (*end user*), tapi disimpan/dikumpulkan dahulu dalam sebuah komputer server (*host*) yang akan online secara terus menerus (*continue*) dengan media penyimpanan (*storage*) yang relatif lebih besar dibanding komputer biasa. Hal ini bisa diibaratkan dengan sebuah kantor pos, jika sesorang mempunyai alamat (*mailbox*), maka dia dapat memeriksa secara berkala jika dia mendapatkan surat. Komputer yang melayani penerimaan *e-mail* secara terus-menerus tersebut biasa disebut dengan *mailserver* atau *mailhost*.

2.9 Spam *Electronic Mail* (Email)

Menurut Paul Graham mendefinisikan *Spam* sampah adalah *Email* yang tidak diinginkan yang dikirimkan secara otomatis. Menurut Sofi Defiyanti (2008), Spam didefinisikan sebagai *Email* yang dikirimkan kepada ribuan penerima (Fitriyanto, 2019).

Menurut Ananda (2011) *Spam* merupakan akronim dari *Stupid Pointless Annoying Message*. *Spam* juga dapat berupa pengiriman pesan secara berulang-ulang ke berbagai *newsgroup* atau server milis dengan pokok bahasan yang tidak berkaitan dan tidak diinginkan atau diminta oleh penerimanya. *Spam* muncul pertama kali pada bulan mei tahun 1978. Menurut Widiasari & Bayu (2013), *Spam* tersebut bersifat iklan yang dikirimkan oleh *Digital Equipment Corporation (DEC)* (Hayuningtyas, 2017).

Berdasarkan pendapat dari para ahli diatas maka dapat ditarik kesimpulan bahwa *email spam* merupakan *email* yang dikirim secara berulang-ulang kepada ribuan penerima yang tidak menginginkan pesan tersebut.

Undang-undang CAN-SPAM memberikan definisi utama spam dengan menjelaskan apa yang (dan apa yang tidak) diperbolehkan bila mengirim *e-mail* komersial pemasaran. Undang-undang tersebut disahkan pada tahun 2004 oleh *Federal Trade Commission,* yang diperbarui tahun 2008. Selain FTC terdapat badan-badan lain yang mengklasifikasikan spam, yaitu *Internet Service Provider* (ISP). *Internet Service Provider* juga memiliki bagian besar dalam menentukan apa yang dianggap spam. Adapun contoh spam pada email yaitu " Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's"

Menurut Supri (2010), ada beberapa tipe-tipe email spam:

1. Iklan : Spam dapat digunakan untuk mempromosikan suatu produk ataupun layanan, mulai dari produk *software*, perumahan *real estate* hingga produk kesehatan dan produk vitamin.
2. Mengirimkan *Malware* : *Spam* adalah salah satu cara utama untuk mendistribusikan virus dan *malware*. Dengan target yang bersifat individual, akan memperdaya korban untuk mempercayai bahwa mereka menerima dokumen penting atau file tertentu, yang sebenarnya mengandung *malware.*
3. *Phising* : Bersembunyi dibalik nama- nama besar perusahaan besar, lembaga keuangan, lembaga pemerintah, lembaga amal, para phisher mencoba memikat korban untuk mengunjungi website palsu, dimana melalui *website* tersebut mereka dapat mencuri data keuangan pribadi atau informasi dengan mengenai identitas korbannya.
4. Scam: Berita elektronik dalam internet yang bersifat menipu sehingga pengirimnya dapat mendapatkan manfaat atau keuntungan.

5. Pesan yang tak berarti : Sebuah potongan pesan sampah seperti ini dapat memenuhi *inbox mail* kita. Bahkan beberapa pesan seperti ini dapat mengkelabui teknologi *spam filter*, banyak pesan tak berarti ini dikirimkan tanpa tujuan yang jelas.

2.10 Cross Entropy

Metode *Cross Entropy* pada awalnya diterapkan untuk simulasi kejadian langka (*rare event*), lalu dikembangkan untuk beberapa kasus seperti optimasi kombinatorial, optimasi kontinyu, *machine learning*, penjadwalan dan beberapa masalah lain. Dalam hal estimasi, CE memberikan cara yang adaptif untuk menemukan distribusi sampling yang optimal untuk beberapa *problem* yang cukup luas cakupannya. *Cross entropy* didefinisikan seperti pada persamaan (2.1) :

$$CE = -\sum_{i}^{C} t_i \log (s_i) \qquad (2.1)$$

Dimana t_i dan s_i adalah *groundtruth* dan skor untuk setiap kelas i di C. karena biasanya fungsi aktivasi *sigmoid* diterapkan ke skor sebelum perhitungan *cross entropy loss* maka, f (s_i) merujuk ke aktivasi.

Binary cross entropy loss disebut juga *sigmoid cross entropy loss* yang merupakan aktivasi *sigmoid* ditabah *cross entropy loss*. *Sigmoid loss* tidak bergantung pada setiap komponen *vector* (kelas), yang berarti bahwa loss dihitung untuk setiap komponen *vector output*, tidak dipengaruhi oleh nilai komponen lainnya.

Dalam masalah klasifikasi biner, di mana C' =2, *cross entropy loss* dapat didefinisikan pada persamaan (2.2) :

$$CE = -\sum_{i=1}^{C'=2} t_i \log (s_i) = -t_1 \log (s_1) - (1 - t_1) \log (1 - s_1) \qquad (2.2)$$

Dimana diasumsikan ada dua kelas yaitu C_1 dan C_2. t_1 [0,1] dan s_1 adalah label *groundtruth* dan skor untuk C_1,yang merupakan C_i di C. dan $t_2 = 1 - t_1$ dan $s_2 = 1 - s_1$ adalah label *groundtruth* dan skor untuk C_2, yang bukan

merupakan “kelas” dalam kelas C, tetapi kelas yang disiapkan untuk masalah biner $C_1 = C_i$.

Loss dinyatakan seperti pada persamaan (2.3) :

$$CE = \begin{cases} -\log\big(f(s_1)\big) & if\ \ t_1 = 1 \\ -\log\big(1 - f(s_1)\big) & if\ \ t_1 = 0 \end{cases} \tag{2.3}$$

Dimana $t_1 = 1$ artinya kelas $C_1 = C_i$ positif untuk *sample* ini.

Dalam kasus ini, fungsi aktivasi tidak bergantung pada skor kelas lain di C lebih dari $C_1 = C_i$. jadi gradien menghormati setiap skor s_i di s hanya akan bergantung pada loss yang diberikan oleh masalah binernya.

Gradien sehubungan dengan skor $s_i = s_1$ dapat ditulis sebagai berikut, seperti pada persamaan (2.4) :

$$\frac{\partial}{\partial_{s_i}} \big(CE\big(f(s_i)\big) = \ t_1(f(s_1) - 1) + (1 - \ t_1)f(s_1)\big) \tag{2.4}$$

Dimana $f()$ adalah fungsi sigmoid. Dapat juga dituliskan sebagai berikut, seperti pada persamaan (2.5) :

$$\frac{\partial}{\partial_{s_i}} (CE\big(f(s_i)\big) = \begin{cases} f(s_i) - 1, & if\ t_i = 1 \\ f(s_i), & if\ t_i = 0 \end{cases} \tag{2.5}$$

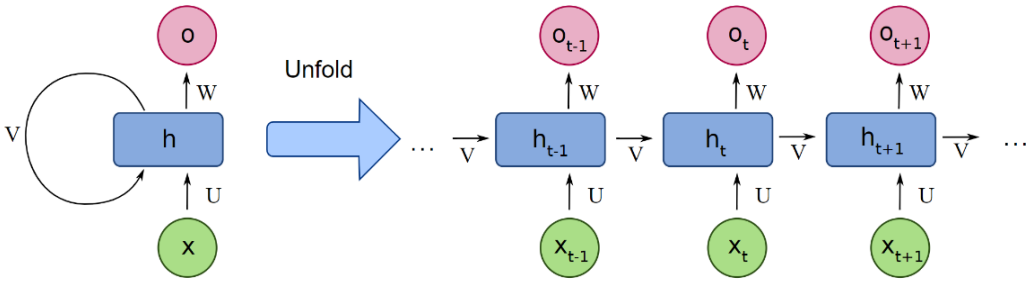
2.11 Recurrent Neural Network

Recurrent Neural Network (RNN) merupakan jaringan saraf berulang. Dikatakan jaringan saraf berulang karena nilai *neuron* pada *hidden layer* sebelumnya akan digunakan kembali sebagai data input (Mikolov dkk, 2015). Penggunaan *neuron* pada *hidden layer* akan disimpan ke dalam sebuah layer yang dinamakan *context layer*. Nilai *neuron* pada *context layer* akan terus update hingga kondisi RNN terpenuhi.

RNN adalah model pada *neural network* yang mengakomodasi *output* jaringan untuk menjadi *input* jaringan kembali, bisa disebut sebagai jaringan umpan balik. Algoritma ini dikembangkan dari algoritma *feedforward neural network*. RNN merupakan metode yang kompleks dan dinamis dikarenakan

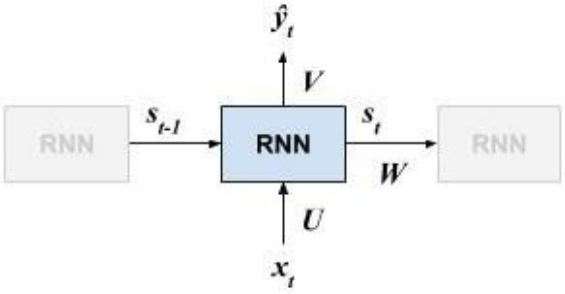
hasil yang dihasilkan tidak haya terpengaruh oleh input saja, melaikan juga dari hasil output sebelumya.

RNN masuk dalam kategori *deep learning* karena data diproses secara otomatis dan tanpa pendefinisian fitur. RNN dapat menggunakan status internal (memori) untuk memproses urutan masukan. Ini membuatnya dapat diterapkan untuk tugas-tugas seperti pemrosesan bahasa alami (PBA), pengenalan suara, sintesa musik, pemrosesan data finansial seri waktu (Choi dkk, 2017).



Gambar 2.9 Arsitektur RNN

Gambar 2.9 menunjukkan satu aspek arsitektur RNN yang sangat penting untuk diperhatikan adalah bahwa, meskipun versi yang terbuka menunjukkan beberapa blok h, blok h yang digunakan selalu sama. Blok h mengirimkan outputnya kembali ke dirinya sendiri. Proses ini terus berulang sampai disuruh berhenti.



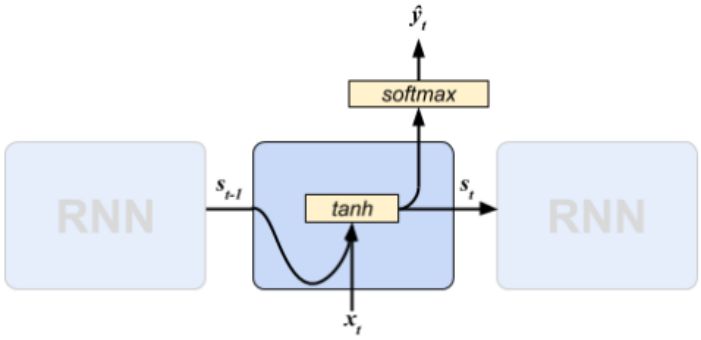
Gambar 2.10 Persamaan Dasar RNN

Persamaan dasar RNN gambar 2.10 menyatakan untuk setiap langkah waktu t, kalkulasi *state* s_t dari input (x_t) dan *state* sebelumnya (s_{t-1}), masing-

maisng dikalikan dengan parameter U dan W lalu diproses dengan fungsi aktivasi *tanh* dengan menggunakan persamaan berikut :

$$s_t = \tanh(\mathrm{U} \cdot x_t + \mathrm{W} \cdot s_{t-1})$$

Dari s_t kemudian dikalkulasi output $\hat{y}_t$ dengan cara mengalikan dengan parameter ***V*** dan melewatkan pada fungsi aktivasi :



Gambar 2.11Visualisasi Dasar RNN dengan Fungsi Aktivasi

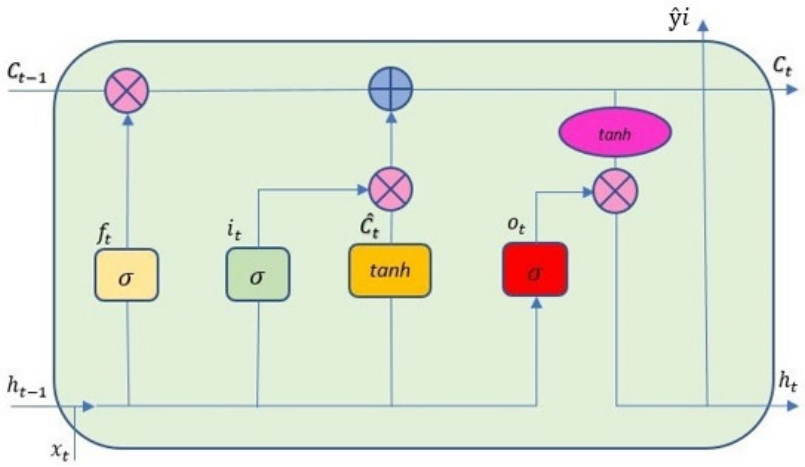
Pada gambar 2.11 s_t merupakan status internal yang diberikan satu langkah waktu ke langkah waktu berikutnya dinamakan memori dari RNN. $\hat{y}_t$ adalah keluaran kata demi kata yang diberikan oleh RNN. adalah proses masukan kata demi kata. U, V, dan W merupakan parameter yang dimiliki RNN. Sedangkan *t* adalah waktu. *Tanh* adalah fungsi aktivasi di lapisan tersembunyi.

2.12 *Long Short-Term Memory* (LSTM)

Untuk peningkatan klasifikasi spam berupa akurasi dan rekayasa fitur, pada penilitian ini menggunakan *Long Short-Term Memory* (LSTM), penggunaan LSTM sangat penting diterapkan untuk keberhasilan penelitian ini. LSTM adalah versi perbaikan dari *recurrent neural network* dimana terdapat banyak tugas serta bekerja lebih baik dibandingkan dengan jaringan biasa. untuk itu metode LSTM pada kumpulan data email untuk klasifikasi spam tepat untuk diterapkan.

LSTM dikembangkan oleh Hochreiter dan Schmidhuber pada tahun 1997. LSTM peningkatan dari algoritma RNN dasar yang memecahkan masalah yang hilang dengan menambahkan *cell states* untuk mengingat atau melupakan data.

Cell states mengandung struktur yang disebut *cell gates*. *Cell gates* terdiri dari empat bagian yaitu *input gate, forget gate, memory-cell state gate, dan output gate*. *Input* merupakan *gate* yang digunakan untuk mengontrol input data yang layak disimpan atau tidak. *Forget gate* digunakan untuk mengontrol keadaan tersembunyi sebelumnya yang akan disimpan di *cell memory* keadaan tersembunyi saat ini. *Memory-cell state gate* digunakan untuk perbarui data berdasarkan informasi dari *input gate* dan *forget gate*. *Output gate* digunakan untuk menghitung *output* data dari jaringan berdasarkan *memory-cell state*. Rincian LSTM di ilustrasikan pada gambar 2.12.



Gambar 2.12 Rincian LSTM

LSTM *cell* diklasifikasi dengan persamaan di bawah ini (Hans, dkk, 2018): Pada Langkah pertama kita memutuskan informasi baru apa yang akan kita gunakan di Ct. Proses ini memiliki dua bagian. Pertama, gerbang *sigmoid* yang disebut "*input gate*" yang memutuskan nilai mana yang akan kita perbarui. Lalu sebuah lapis tanh "c_int" yang menghasilkan kandidat vektor konteks baru. Lalu kita gabungkan keduanya untuk membuat pembaruan ke konteks nanti.

$$i_t = \sigma\,(W_i\,.\,[h_{t-1}, x_t] + b_i)$$

$$\tilde{C} = \tanh(W_c\,.\,[h_{t-1}, x_t] + b_c)$$

Pada Langkah kedua, , memutuskan informasi apa yang akan dibuang dari konteks Ct-1 , dengan menggunakan gerbang *sigmoid* yang kita sebut "*forget gate*". Gerbang ini membaca nilai ht-1 dan xt , dan menghasilkan angka antara 0 dan 1 untuk setiap elemen dalam Ct-1. Nilai 1 artinya "mempertahankan elemen" sementara 0 artinya "melupakan elemen".

Selanjutnya memperbarui konteks lama Ct-1 ke konteks baru Ct. Kita kalikan konteks lama dengan ft untuk melupakan hal-hal yang kita putuskan untuk dilupakan. Kita kalikan kandidat konteks baru kita c_int dengan it untuk memutuskan seberapa banyak kita akan menyertakan kandidat konteks baru. Lalu kita tambahkan keduanya.

$$f_t = f_t = \sigma\left(W_f\ [h_{t-1}, x_t] +\ b_f\right)$$

$$C_t = \left(f_t\ * C_{t-1} + i_t\ *\ \tilde{\mathrm{C}}_t\right)$$

Akhirnya memutuskan apa yang akan kita hasilkan. Pertama, kita jalankan gerbang *sigmoid* yang kita namakan "*output gate*" untuk memutuskan bagian-bagian apa dari konteks yang akan kita hasilkan. Kemudian, kita lewatkan konteks melalui tanh (ct) untuk membuat nilainya menjadi antara −1 dan 1, dan kita kalikan dengan *output gate sigmoid* tadi sehingga kita hanya menghasilkan bagian yang kita putuskan.

$$O_t = \sigma\ (W_o\ .\ [h_{t-1}, x_t] +\ b_o)$$

$$h_t =\ o_t * tanh(\ C_t)$$

Untuk parameter *hyper* model LSTM, digunakan algortima optimasi pencarian grid untuk memilih nilai optimal untuk lima parameter *hyper*, yaitu *learning rate, batch size, epochs, dropout rate*, dan algoritma pengoptimalan. Rentang dan nilai optimal dari parameter *hyper* yang dipilih oleh model LSTM.

2.13 Support Vector Machine

Support Vector Machine pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang pattern recognition. SVM adalah metode learning machine yang bekerja atas

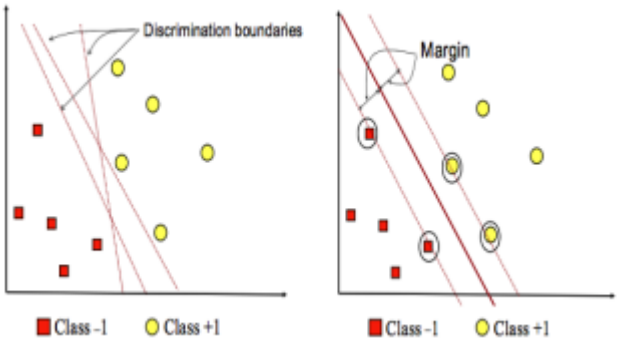
prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan hyperplane terbaik yang memisahkan dua buah class pada input space. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM.

Menurut Christianini & Shawe-Taylor (2000), *Support Vector Machine* (SVM) adalah sistem pembelajaran yang menggunakan hipotesis fungsi linear dalam ruang berdimensi tinggi dan dilatih dengan algoritma berdasarkan teori optimasi dengan menerapkan learning bias yang berasal dari teori statistic. Menurut Krisantus (2007) *Support Vector Machine* merupakan salah satu teknik yang relatif baru dibandingkan dengan teknik lain yang memiliki performansi yang lebih baik di berbagai bidang seperti *bioinformatics*, pengenalan tulisan tangan, klasifikasi teks dan lain sebagainya (Vynaima S, 2018).

Menurut Lidya, dkk (2015), *Support Vector Machine* adalah metode yang menganalisis data dan mengenali pola yang digunakan untuk klasifikasi data. Sistem kerja *Support Vector Machine* dengan cara memasukan data tertentu kedalam sebuah sistem kemudian memprediksikannya. Dari data tersebut akan di kelompokan kedalam dua kelas yang berbeda. Data baru tersebut akan diklasifikasikan kedalam kelas yang satu atau kelas yang lain. Dari hal tersebut maka akan jelas bahwa data tersebut akan dikelompokkan ke dalam kelas yang sesuai. Menurut Pratama, dkk (2018), *Support Vector Machine* adalah metode terbaik yang bisa dipakai dalam permasalahan klasifikasi. Konsep *Support Vector Machine* bermula dari masalah klasifikasi dua kelas sehingga membutuhkan training set positif dan negatif. *Support Vector Machine* berusaha menemukan pemisah terbaik untuk memisahkan ke dalam dua kelas dan memaksimalkan margin antara dua kelas. Pada banyak kasus, data tidak bisa diklasifikasi menggunakan metode linier *Support Vector Machine*, sehingga dikembangkanlah fungsi kernel untuk mengklasifikasikan data dalam bentuk non-linier (Setiono & Pardede, 2019).

Berdasarkan menurut para ahli diatas makaa dapat dsimpulkan bahwa *Support Vector Machine* adalah teknik pembelajaran menggunakan hipotetsis fungsi linear dalam ruangan berdimensi tinggi yang memiliki performasi terbaik dalam hal kalsifikasi.

Support vector machine (SVM) merupakan metode pembelajaran machine yang menerapkan prinsip *structural risk minimization* (SRM) untuk menentukan *hyperplane* terbaik seperti pada Gambar 2.13.



Gambar 2.13 Menentukan Hyperlane Terbaik dari Dua Class

Gambar 2.13 memperlihatkan beberapa pattern dari dua class: +1 dan -1. Class -1 digambarkan dengan kotak berwarna merah dan class +1 digambarkan dengan lingkaran berwarna kuning. Teknik SVM berusaha untuk mencari hyperplane yang memisahkan dua class tersebut. Magin merupakan jarak garis dengan *pattern* terdekat, *pattern* tersebut adalah *support vector*, garis merah tersebut adalah *hyperplane* terbaik yang memisahkan dua buah class. Persamaan untuk mencari *hyperplane* memenuhi persamaan berikut :

$$f(x) = \sum_{i=1}^{SV} \alpha_i . K(SV_i, x) + b$$

Keterangan :

α_i = vector koefisien untuk lagrange multiplier

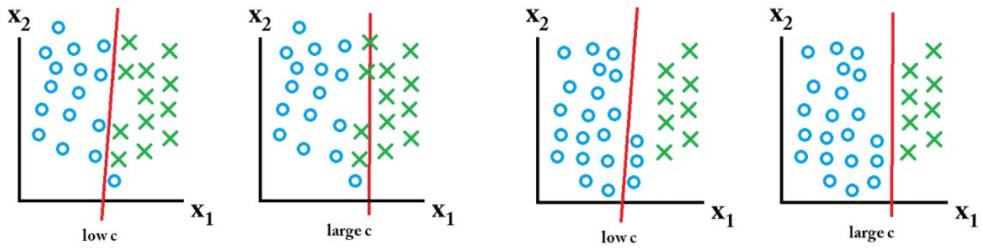
$K(SV_i, x)$ = fungsi kernel yang digunakan

b = error atau bias

Kernel *Radial Basis Function* (RBF) yang merupakan sebuah kernel yang memiliki performa yang baik dengan parameter tertentu yang hasil pelatihannya tidak menghasilkan kesalahan yang besar.

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

Akurasi model yang akan dihasikan dari proses pelatihan dengan SVM sangat bergantung pada fungsi kernel serta parameter yang digunakan. Oleh karena itu performansinya dapat dioptimasi dengan mencari atau mengestimasi parameter terbaik. Parameter C disebut juga dengan error penalty, karena berhubungan trade-off antara batas maksimum dan kesalahan klasifikasi selama proses training. Pemilihan nilai error penalty yang tepat akan membuat training SVM dapat mencegah terjadinya kesalahan klasifikasi. Dengan nilai error penalty yang tinggi dapat menghasilkan batas yang mengklasifikasikan semua data training secara tepat (Wanto, dkk 2019). Adapun contoh perbandingan nilai C tinggi dan C rendah ditunjukkan pada gambar 2.14.



Gambar 2.14 Perbandingan Nilai C tinggi dan C Rendah

2.14 Confusion Matrix

Menurut Gorunescu (2011), *Confusion Matrix* adalah alat visualisasi yang biasa digunakan pada supervised learning. Tiap kolom pada matriks adalah contoh kelas prediksi, sedangkan tiap baris mewakili kejadian dikelas yang sebenarnya (Hayuningtyas, 2017)

Menurut Han & Kamber (2011:365), *Confusion matrix* adalah alat yang berguna untuk menganalisis seberapa baik *classifier* mengenali *tuple* dari kelas

yang berbeda. TP dan TN memberikan informasi ketika classifier benar, sedangkan FP dan FN memberitahu ketika *classifier salah* (Fitriyanto, 2019)

Tabel 2.1 Confusion Matrix

Prediksi	Aktual	
	Positif	Negatif
TRUE	TP	FN
FALSE	FP	TN

Keterangan :

TP = *True Positif*

TN = *True Negatif*

FP = *False Positif*

FN = *False Negatif*

Berdasarkan tabel 2.1 *confusion matrix* menurut Adriani (2013) *True Positif* adalah jumlah record *positif* yang diklasifikasikan sebagai *positif, false positif* adalah jumlah record *negatif* yang diklasifikasikan sebagai *positif, false negatif* adalah jumlah record *positif* yang diklasifikasikan sebagai *negatif, true negatif* adalah jumlah record *negatif* yang diklasifikasikan sebagai *negative* (Hayuningtyas, 2017).

Evaluasi yang akan dilakukan menggunakan parameter *F-Measure* yang terdiri dari perhitungan *precision,* dan *recall. Recall, precision* dan *F-measure* merupakan metode pengukuran yang efektifitas dilakukan pada proses klasifikasi. Menurut Supri (2010), *Recall* dan *precision* adalah dua kriteria yang digunakan untuk mengevaluasi tingkat efektivitas kinerja sistem temu kembali informasi.

a. Precision

   Precision atau *Confidence* menunjukkan bahwa proporsi prediksi kasus positif yang benar-benar *real* positif*,* namun secara analogis dapat disebut *True Positive Accuracy* (tpa), menjadi ukuran akurasi *predicted positives*

berbeda dengan tingkat penemuan *real* positif (tpr) (Powers, 2020). Presisi didefinisikan sebagai berikut :

$$Precision\ (positif) = \frac{TP}{TP + FP}$$

Inverse precision adalah perbandingan dari kasus prediksi negatif yang benar – benar real negatif, dan juga dapat disebut *True Negative Accuracy* (tna) (Powers, 2020):

$$Inverse\ Precision\ (negatif) = \frac{TN}{TN + FN}$$

b. Recall

Recall atau *Sensitivitas* menunjukkan bahwa proporsi kasus *real* positif yang benar diprediksi positif. Fitur ini digunakan untuk menggambarkan berapa banyak kasus relevan yang diambil dari prediksi positif. Dalam konteks ini disebut *True Positive Rate (tpr)* (Powers, 2020). *Recall* didefinisikan sebagai berikut :

$$Recall\ (positif) = \frac{TP}{TP + FN}$$

Inverse Recall demikian perbandingan dari kasus negatif nyata yang benar diprediksi negatif, dan juga dikenal sebagai *True Negative Rate* (tnr) (Powers, 2020).

$$Recall\ (negatif) = \frac{TN}{FP + TN}$$

c. Accuracy

Accuracy adalah persentase dari total e-mail yang benar diidentifikasi

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

d. F1 *Score*

F1 *score* adalah rata – rata dari *precision* dan *recall* yang dibobotkan (Sokolova & Lapalme, 2009)

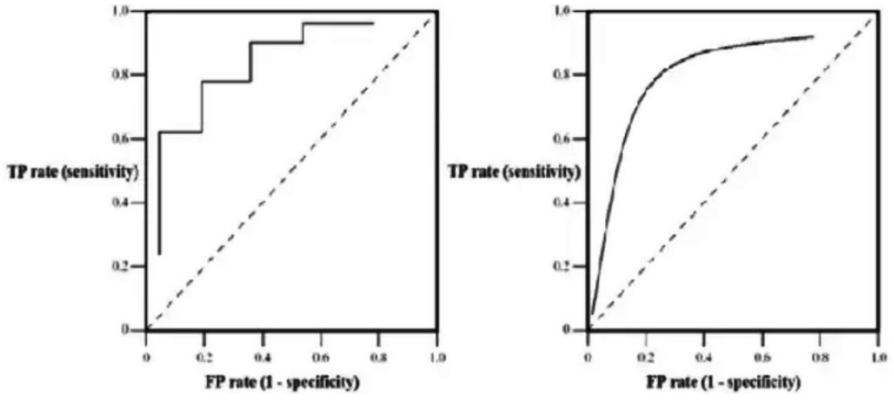
$$F1(positive) = 2 * \frac{precision(positive) * recall(positive)}{precision(positive) + recall(positive)}$$

$$F1(negative) = 2 * \frac{precision(negative) * recall(negative)}{precision(negative) + recall(negative)}$$

2.15 Kurva ROC (*Receiver Operating Caharacteristic*)

Grafik kurva ROC (*Receiver Operating Characteristic*) digunakan untuk mengevaluasi akurasi classifier dan untuk membandingkan klasifikasi yang berbeda model. Sebuah grafik ROC adalah grafik dua dimensi dengan *False Positive Rate* pada sumbu horisontal dan *True Positive Rate* yang benar di sumbu vertical (Pudjiarti, 2016).

Dalam masalah klasifikasi menggunakan kelas keputusan dua (klasifikasi biner), masing-masing objek dikelompokkan dalam (P, N), yaitu positif atau negatif. Sementara beberapa model klasifikasi (misalnya, pohon keputusan) menghasilkan label kelas diskrit (menunjukkan hanya kelas diprediksi objek), pengklasifikasi lainnya (misalnya, naive bayes, jaringan saraf) menghasilkan output yang berkesinambungan, yang ambang batas yang berbeda mungkin diterapkan untuk memprediksi keanggotaan kelas, secara teknis, ROC kurva, juga dikenal sebagai grafik ROC, dua-dimensi grafik dimana TP rate pada sumbu Y dan FP rate pada sumbu X (Pudjiarti, 2016).



Gambar 2.15 Grafik ROC

Pada gambar 2.12 garis diagonal membagi ruang ROC, yaitu : poin diatas garis diagonal merupakan hasil klasifikasi yang baik dan point dibawah garis diagonal merupakan hasil klasifikasi yang buruk dapat disimpulkan bahwa, satu point pada kurva ROC adalah lebih baik dari pada yang lainnya jika arah garis melintang dari kiri bawah ke kanan atas didalam grafik.

Untuk keakurasian nilai AUC dalam klasifikasi data mining dibagi menjadi lima kelompok (Gorunescu, 2011), yaitu : 0.90 – 1.00 = klasifikasi sangat baik (excellent classification), 0.80 – 0.90 = klasifikasi baik (good classification) , 0.70 – 0.80 = klasifikasi cukup (fair classification), 0.60 – 0.70 = klasifikasi buruk (poor classification), dan 0.50 – 0.60 = klasifikasi salah (failure)