BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Tahap tinjauan perusahaan ini merupakan peninjauan langsung terhadap tempat penelitian stadi kasus yang dilakukan di CV. Nusantara Technology Solution (Nusantech). Tinjauan perusahaan ini meliputi profil perusahaan, visi dan misi perusahaan, logo perusahaan, struktur organisasi, serta deskripsi setiap jabatan tempat penelitian.

2.1.1 Profil CV. Nusantara Technology Solution

CV. Nusantara Technology Solution (Nusantech) merupakan *startup* yang bergerak dibidang IT, melayani permintaan kebutuhan dengan pengembangan aplikasi berbasis *website* dan *mobile* (*android* & *iOS*). Nusantech beralamat di Jl. Sekeloa Timur No. 62, Kecamatan Coblong, Kota Bandung.

Berawal dari rasa kekhawatiran CEO CV. Nusantara Technology Solution (Nusantech) Nicolas Novian Ruslim terhadap perkembangan IT di Indonesia yang dinilai belum siap dalam menghadapi Masyarakat Ekonomi Asean (MEA). maka pada tahun 2014 beliau mulai mengumpulkan mahasiswa-mahasiswa dalam sebuah lingkungan perkuliahan, sehingga terbentuklah tim dengan beranggotakan 9 orang dengan kemampuan dibidang IT yang terdiri dari berbagai minat (*website*, *mobile*, robotika, jaringan, sistem kontrol, desain).

Untuk menunjang perkembangan CV. Nusantara Technology Solution (Nusantech) beliau memakai gajinya selama 1 tahun sebagai peneliti di lab *smart city* ITB untuk membiayai penyewaan tempat serta alat-alat penunjang lainnya seperti meja, kursi, internet dan fasilitas lainnya.

Kemudian barulah pada tanggal 19 mei 2015 secara resmi CV. Nusantara Technology Solution (Nusantech) berdiri, dengan Akta Notaris Irma Rahmawati, SH. No 33, Dengan S. K. Menteri Kehakiman dan Hak Asasi Manusia Republik Indonesia No C – 409 – HT.03.01. Th 2001 Tanggal 30 Oktober 2001 dan S. K. Kepala Badan Pertanahan Nasional Republik Indonesia No 4 – X – 2002 Tanggal 15 April 2002.

Visi dari CV. Nusantara Technology Solution (Nusantech) adalah "Penggerak utama dalam pengembangan SDM professional bidang IT di Indonesia". Sedangkan misi dari CV. Nusantara Technology Solution (Nusantech) adalah sebagai berikut:

1. Menciptakan ekosistem pertukaran informasi dan kolaborasi di bidang IT
2. Memberikan bimbingan untuk menghasilkan professional Indonesia di bidang IT
3. Mendorong pengembangan produk IT nasional secara berkelanjutan
4. Melakukan standarisasi dan control kualitas produk IT yang dihasilkan.

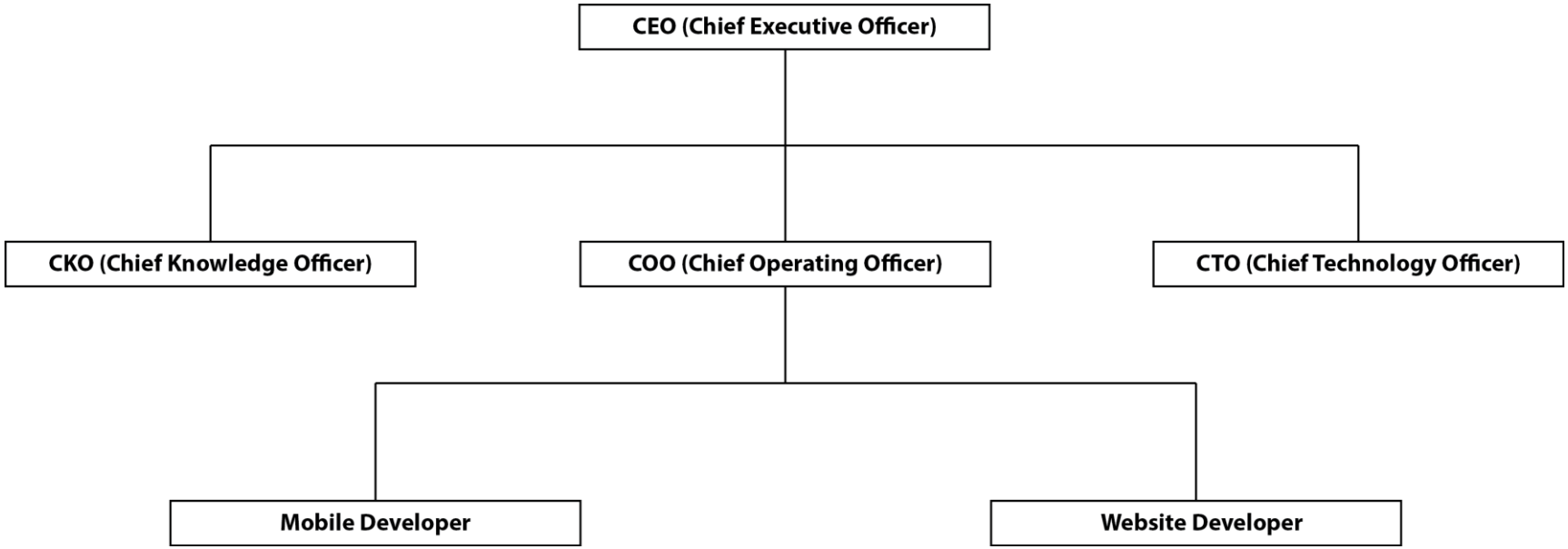
CV. Nusantara Technology Solution (Nusantech) memiliki sebuah logo. Logo adalah lambang atau simbol khusus yang mewakili suatu perusahaan atau organisasi. Sebuah logo bisa berupa nama, lambang atau elemen grafis lain yang ditampilkan secara visual. Gambar 2.1 merupakan logo dari CV. Nusantara Technology Solution (Nusantech)



Gambar 2. 1 Logo CV. Nusantara Technology Solution

2.1.2 Struktur Oganisasi CV. Nusantara Technology Solution

Struktur organisasi perusahaan merupakan sebuah garis hierarki yang mendeskripsikan kompenen-komponen yang menyusun perusahaan dimana setiap individu (sumber daya menusia) yang berada pada lingkup perusahaan tersebut memiliki posisi dan fungsi masing-masing.



Gambar 2. 2 Struktur Organisasi CV. Nusantara Technology Solution.

Berdasakan struktur organisasi pada gambar 2.2 diatas dapat dijelaskan deskipsi jabatan seagai berikut:

1. CEO (Chief Executive Officer)

CEO sebagai penanggung jawab perusahaan, yang mempunyai wewenang atas perusahaan, serta yang mengeluarkan kebijakan-kebijakan perusahaan.

2. COO (Chief Operating Officer)

COO bertanggung jawab atas lancarnya kegiatan produksi dan produktifitas karyawan-karyawa sehingga perusahaan dapat berjalan dengan baik dan lancar. Serta bertugas menggantikan tugas CEO jika CEO tidak ada di perusahaan, semua tanggung jawab dan wewenang atas perusahaan diambil alih oleh COO.

3. CTO (Chief Technology Officer)

CTO bertanggung jawab untuk penelitian dan pengembangan serta untuk perencanaan produk baru. Serta Mengembangkan strategi untuk meningkatkan penghasilan/pemasukan.

4. CKO (Chief Knowledge Officer)

CKO bertugas mengumplkan, serta menggunakan pengetahuan yang ada untuk selalu memperbaiki proses bisnis dalam perusahaan. Serta memanage semua pengetahuan dari setiap pegawai untuk digunakan lagi oleh pegawai lain.

5. Mobile Developer

Tugas dan tanggung jawab *Mobile Developer*:

a. Membangun aplikasi mobile (android, iOS) sesuai rancangan
b. Mencari dan menerapkan algoritma terbaik dalam membangun aplikasi
c. Melakukan perbaikan atas aplikasi atau pemeliharaan aplikasi
d. Membuat spesifikasi program secara detail melalui diskusi dengan client
e. Menjelaskan secara tepat apa tindakan (aksi) program yang diinginkan
f. Menguraikan spesifikasi program ke dalam elemen-elemen sederhana dan menerjemahkan logikanya ke dalam bahasa pemrograman

g. Memikirkan solusi yang mungkin untuk menprediksi masalah, mengevaluasi pilihan lain
h. Bekerja sebagai bagian dari tim, dimana mengadakan proyek khusus, untuk membuat bagian tertentu dari program
i. Mengkombinasikan semua elemen dari rancangan program dan mengujinya
j. Menguji sample data-set untuk memeriksa keluaran dari program sesuai dengan yang diinginkan.

6. Website Developer

Tugas dan tanggung jawabWebsite Devloper:

a. Membuat gambar-gambar yang akan disisipkan kedalam website dengan menggunakan Photoshop dan sebagainya.
b. Mengatur Konten Website sesuai dengan kategori & diintegrasikan dengan link navigasi.
c. Membangun Website mulai dari halaman depan sampai akhir sesuai dengan struktur web yang dipilih.
d. Membagi tugas pemrograman dengan memecah-mecah setiap unit pemrograman.
e. Melakukan pengkodean terhadap setiap unit pemrograman tersebut.
f. Memasukkan Data kedalam Database berikut dengan table-teblenya.
g. Membuat script untuk mengakses Database & menghubungkan tabel-tabel dalam Database
h. Melakukan uji coba kinerja program pada setiap unit pemrograman
i. Menata Layout untuk pemasangan Banner, Link, Animasi & sebagainya.
j. Melakukan uji coba setiap halaman & memeriksa kesalahan penulisan HTML, CSS, & sebagainya

2.2 Landasan Teori

Landasan teori pada penulisan skripsi ini akan menerangkan mengenai teori-teor yang berhubungan dengan *Knowledge Management System* di CV. Nusantara Technology Solution (Nusantech).

2.2.1 Knowledge

Menurut Van Der Spek dan Spijkervet "*knowledge* adalah sebagian besar dari ide, pengalaman dan prosedur yang dianggap benar, mengarahkan untuk berfikir, beritngkah laku dan berkomunikasi dengan orang lain".

Sedangkan menurut Turban "*Knowledge* adalah informasi yang telah diorganisir dan dianalisis agar dapat dipahami dan diaplikasikan untuk memecahkan masalah atau mengambil keputusan".

Berdasarkan pendapat diatas, maka penulis dapat simpulkan bahwa pengetahuan (*knowledge*) setidaknya memiliki tiga hal penting yaitu:

1. *Knowledge* merupakan kumpulan informasi mengenai intuisi, pengalaman (*experience*) dan urutan kegiatan (*procedure*).
2. *Knowledge* diorganisir dan dianalisis sehingga dapat dimengerti dan diaplikasikan.
3. *Knowledge* digunakan sebagai pedoman untuk berfikir, bertingkah laku, berkomunikasi, memecahkan masalah dan mengambil keputusan.

Pengetahuan tercipta dari interaksi komponen tipe pengetahuan. Menurut Nonaka, Grof & Jones dan Polanyi terdapat dua tipe pengetahuan dalama diri manusia yaitu pengetahuan tacit (*tacit knowledge*) dan pengetahuan eksplisit (*eksplicit knowledge*).

Menururt Polanyi pengetahuan tacit bersifat 1) tidak dapat dibagi 2) merupakan hal yang lebih banyak diketahui daripada disampaikan 3) seringkali terdiri dari kebiasaan-kebiasaan dan budaya yang tidak dapat ditentukan sendiri 4) tidak dapat dikodifikasi, tetapi hanya dapat dipindahkan atau diperoleh dari pengalaman 5) menggambarkan *know what* (fakta) dan *know why* (sains) 6) melibatkan pembelajaran skill 7) terbentuk dalam kelompok dan hubungan organisional, nilai inti, asumsi-asumsi dan keyakinan, sulit diidentifikasi,

disimpan, dihitung dan dipetakan. Sedangkan menurut Malhotra pengetahuan *tacit* adalah pengetahuan tentang cara yang ada dalam benak manusia. Hal yang berkaitan dengan pengetahuan tacit adalah pengetahuan mengenali, menghasilkan, membagi dan mengatur sesuatu.

Berdasarkan beberapa pendapat diatas maka penulis dapat menyimpulkan bahwa pengetahuan *tacit* adalah pengethuan yang bersumber dari pengalaman, keyakinan, asumsi, kebiasaan dan budaya atau proses pembelajaran yang menggambarkan tentang *know what* dan *know why*, yang terbentuk dalam pribadi maupun kelompok yang sifatnya sulit diidentifikasi, disimpan, dan sulit dibagikan.

Menurut Graff dan Jones pengetahuan *explicit* adalah pengetahuan tacit yang telah didokumentasikan, telah diartikulasikan dalam bahasa formal sehingga lebih mudah dipindahkan diantara orang-orang. Sedangkan menurut Nonaka dan Takeuchi pengetahuan *explicit* adalah pengetahuan yang siap diakses, telah didokumentasikan dalam sumber pengetahuan formal yang telah diorganisir dengan baik.

Berdasarkan pengertian diatas, maka penulis dapat simpulkan bahwa *explicit knowledge* adalah pengetahuan yang bersumber dari pengetahuan *tacit* (*tacit knowledge*) yang diartikulasi, didokumentasikan, dikodifikasi, diorganisir, dalam sebuah media tertentu, sehingga dapat mudah diakses dan di sebarkan ke pihak lain yang membutuhkan.

2.2.2 Knowledge management

Menurut Laudon dan Laudon (2002), "Knowledge Management adalah serangkaian proses yang dikembangkan dalam suatu organisasi untuk menciptakan, mengumpulkan, memelihara dan mendiseminasikan pengetahuan organisasi tersebut".

Sedangkan menurut Barclay dan Murray mendefinisikan knowledge management sebagai suatu aktivitas bisnis yang mempunyai dua aspek penting, yaitu:

1. Memperlakukan komponen pengetahuan dalam aktivitas-aktivitas bisnis yang direfleksikan dalam strategi, kebijakan, dan berbagai praktek perusahaan secara keseluruhan.
2. Membuat suatu hubungan langsung antara aset intelektual perusahaan baik yang explicit maupun tacit untuk mencapai tujuan perusahaan.

Adapun pengertian lain dari *knowledge management* dalam perspektif bisnis dan teknologi adalah :

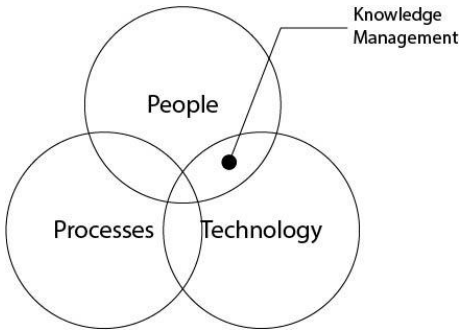
1. Pada perpektif bisnis, knowledge management merupakan sebuah aktivitas bisnis yang memiliki dua aspek utama, yaitu :
   a. Perlakuan terhadap komponen knowledge kegiatan organisasi yang merupakan fokus bisnis bisa tercermin dalam strategi, kebijakan, dan praktek dari semua tingkatan dalam organisasi dan membuat hubungan langsung antara aset intelektual baik tacit maupun explicit knowledge.
   b. *Knowledge management* merupakan pendekatan yang bersifat kolaboratif dan terpadu dalam hal penciptaan, penangkapan, dan penggunaan aset dalam suatu perusahaan.
2. Pada perspektif teknologi, knowledge management memiliki pengertian sebagai berikut :
   a. *Knowledge management* merupakan konsep dimana informasi diubah menjadi pengetahuan yang ditindaklanjuti dan tersedia dengan mudah dalam bentuk yang dapat digunakan oleh pengguna.
   b. *Knowledge management* merupakan sebuah virtual repository yang berisi informasi yang relevan dan penting untuk mendukung tugas yang dilakukan oleh pengguna dalam kehidupan sehari-hari dalam sebuah organisasi.

Knowledge Management memiliki komponen yang saling terkait satu sama lain, adapun komponennya :

1. *People* (Sumber Daya Manusia)
2. *Technology* (Teknologi yang dipakai)

3. *Process* (Proses terjadinya *knowledge*)

Yang mana ketiganya dapat menghasilkan suatu pembelajaran bagi organisasi.



Gambar 2. 3 Komponen Knowledge (Collison, 2004)

2.2.3 System

Menurut Turban dan Aronson, sistem adalah kumpulan objek seperti orang, sumber daya, konsep dan prosedur yang dimaksudkan untuk melakukan suatu fungsi yang dapat diidentifikasi atau untuk melayani suatu tujuan.

Menurut Prahasta, sistem adalah sekumpulan objek, ide, berikut saling keterhubungannya (inter-relasi) dalam mencapai tujuan atau sasaran bersama.

Menurut Ladjamudin, Suatu sistem mempunyai karakteristik atau sifat- sifat yang tertentu, yaitu:

a. Komponen Sistem

   Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

b. Batasan Sistem

   Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan dan menunjukkan ruang lingkup dari sistem tersebut.

c. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan demikian harus dijaga dan dipelihara. Sedangkan lingkungan yang merugikan harus ditahan dan dikendalikan, jika tidak akan mengganggu kelangsungan hidup dari sistem.

d. Penghubung Sistem

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini, sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya.

e. Masukan Sistem

Merupakan segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses.

f. Keluaran Sistem

Merupakan hasil dari pemrosesan sistem, yang bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainnya.

g. Pengolahan Sistem

Merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna.

h. Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.

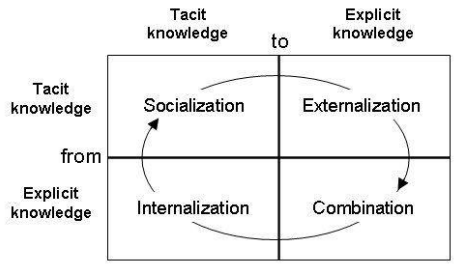
2.2.4 Knowledge Management System

Knowledge Management System (KMS) adalah suatu sistem yang dapat menyimpan pengetahuan untuk dapat dimanfaatkan oleh anggota KMS (*web*). Dapat diartikan bahwa dinamika interaksi tersebut adalah proses belajar yang dilakukankan anggota organisasi dalam rangka meningkatkan (basis)

pengetahuannya, jadi organisasi tersebut haruslah menjadi organisasi pembelajar. Jadi, pengetahuan yang menjadi kunci utama harus dapat diciptakan oleh anggota organisasi, yang kemudian dapat disimpan pada lokasi tertentu dan dapat diseminasi sehingga diketahui dan dipelajari oleh anggota yang lain, dan pengetahuan tersebut dapat diketahui keterhubungannya dengan pengetahuan yang sudah ada sebelumnya.

Pengetahuan dibagi menjadi dua yaitu *Explicit Knowledge* dan *Tacit Knowledge*. *Explicit Knowledge* yaitu pengetahuan yang tertulis, terarsip, tersebar (cetak maupun elektronik) dan bisa sebagai bahan pembelajaran (*reference*) untuk orang lain. *Tacit Knowledge* yaitu pengetahuan yang berbentuk know-how (pemahaman), pengalaman, dan *skill*. Konsep dari siklus pengetahuan digambarkan dalam nonaka's model yaitu:

1. *Socialization* merupakan transformasi dari tacit ke tacit. Proses sosialisasi disampaikan melalui bentuk seminar, rapat, dan berbagai bentuk sosialisasi lainnya yang memungkinkan terjadinya perpindahan pengetahuan seseorang ke orang lain. Contonya transfer informasi diantara orang-orang dengan percakapan.
2. *Externalization* merupakan transformasi dari *tacit* ke *explicit*. Proses eksternalisasi perpindahan pengetahuan yang diwujudkan dalam bentuk nyata seperti buku, presentasi, dan lain-lain. Contohnya menulis buku.
3. *Combination* merupakan transformasi dari *explicit* ke *explicit* yaitu terjadi penggabungan pengetahuan dari berbagai wujud. Contohnya mengambil beberapa referensi dari berbagai sumber kemudian menghasilkan informasi yang lengkap.
4. *Internalization* merupakan transformasi dari *explicit* ke *tacit* yaitu proses terakhir di mana seseorang menyerap pengetahuan yang ada kemudian menjadi pengetahuan baru oleh dirinya sendiri. Contonya keryawan menjadi narasumber di acara seminar.



Gambar 2. 4 Siklus Alur Pengetahuan (Nonaka, 1995)

2.2.5 Knowledge Taxonomy

Menurut Dalkir, dapat dianggap sebagai bangunan dari suatu pengetahuan dan keahlian. *Taxonomy* merupakan sistem aplikasi dasar yang berguna untuk memaparkan konsep-konsep dalam bentuk *Hierarchical Model*. Semakin tinggi suatu konsep diletakan, maka semakin umum dan dapat dirincikan. Begitu juga sebaliknya, semakin rendah suatu konsep, maka semakin spesifik nama dari satu *subclass*. Menurut Dalkir, Konsep penting yang menggaris bawahi taksonomi adalah gagasan tentang turunan. Setiap *node* merupakan suatu sub kelompok dari kelas atasnya, itu mengartikan bahwa simpul yang lebih tinggi akan di pindahkan dari kelas induk ke anak [4].

Menurut Dalkir, *Taxonomy* adalah suatu skema klasifikasi kelompok yang saling terkait bersama-sama, sering dinamai sebagai suatu jenis konsep hubungan kepada satu sama lain dan memberikan pengertian tentang kategori secara umum dibandingkan contoh atau kasus khusus. Skema klasifikasi dapat sangat bersifat pribadi, seperti contoh penamaan pada sebuah *folder* pribadi [4].

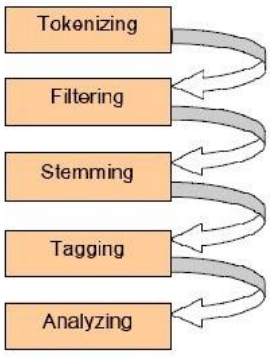
2.2.6 Text Mining

Text mining dapat diartikan sebagai penemuan informasi yang baru dan tidak diketahui sebelumnya oleh komputer, dengan secara otomatis mengekstrak informasi dari sumber-sumber yang berbeda. Kunci dari proses ini adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber. Sedangkan menurut Harlian, *text mining* memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, dan

tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

Text mining mengacu pada proses mengambil informasi berkualitas tinggi dari teks. Informasi berkualitas tinggi biasanya diperoleh melalui peramalan pola dan kecenderungan melalui sarana seperti pembelajaran pola statistik. *Text mining* biasanya melibatkan proses penataan teks input (biasanya parsing, bersama dengan penambahan beberapa fitur linguistik turunan dan penghilangan beberapa diantaranya, dan penyisipan subsequent ke dalam database), menentukan pola dalam data terstruktur, dan akhirnya mengevaluasi dan menginterpretasi output. Berkualitas tinggi di bidang text mining biasanya mengacu ke beberapa kombinasi relevansi, kebaruan, dan interestingness.

Proses *text mining* yang khas meliputi kategorisasi teks, *text clustering*, ekstraksi konsep/entitas, produksi taksonomi granular, sentiment analysis, penyimpulan dokumen, dan pemodelan relasi entitas (yaitu, pembelajaran hubungan antara entitas bernama). Tahapan-tahapan dalam *text mining* secara umum adalah *text preprocessing* dan *feature selection* (Feldman & Sanger 2007, Berry & Kogan 2010). Didalam proses *text mining* dilakukan beberapa tahapan umum diantaranya adalah *tokenizing, filtering, stemming, tagging, dan analyzing.*



Gambar 2. 5 Tahapan Text Mining

1. *Tokenizing*

   Tahap *tokenizing / parsing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya menjadi kumpulan kata yang menyusunnya. Hanya huruf “a” sampai dengan “z” yang diterima. Karakter selain huruf

dihilangkan dan dianggap delimiter. Tabel 2. 1 merupakan karakter yang akan dihilangkan pada proses ini.

Tabel 2. 1 Karakter yang Dihilangkan

Karakter			
!	~	+	\
@	&	=	/
#	*	{	“
$	(	}	‘
%	)	[	:
^	-	]	;
`	_	|	.
,	<	>	?
White space (tab,spasi, enter)			

2. Filtering

Filtering adalah tahap mengambil kata-kata penting dari hasil token. Bisa menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist*/*stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*.

Tabel 2. 2 Stopwords

Stopwords				
Ada	mengucapkan	kembali	dimisalkan	diingatkan
Adalah	mengucapkannya	kemudian	dimulai	diinginkan
Adanya	mengungkapkan	kemungkinan	dimulailah	dijawab
Adapun	menjadi	kemungkinannya	dimulainya	dijelaskan
Agak	menjawab	kenapa	dimungkinkan	dijelaskannya
Agaknya	menjelaskan	kepada	dini	dikarenakan
Agar	menuju	kepadanya	dipastikan	dikatakan
akan	menunjuk	kesampaian	diperbuat	dikatakannya
akankah	menunjuki	keseluruhan	diperbuatnya	dikerjakan
akhir	menunjukkan	keseluruhannya	dipergunakan	diketahui
akhiri	menunjuknya	keterlaluan	diperkirakan	diketahuinya
akhirnya	menurut	ketika	diperlihatkan	dikira
aku	menuturkan	khususnya	diperlukan	dilakukan
akulah	menyampaikan	kini	diperlukannya	dilalui
amat	menyangkut	kinilah	dipersoalkan	dilihat
amatlah	menyatakan	kira	dipertanyakan	dimaksud
anda	menyebutkan	kira-kira	dipunyai	dimaksudkan
andalah	menyeluruh	kiranya	diri	dimaksudkannya

antar	menyiapkan	kita	dirinya	dimaksudnya
antara	merasa	kitalah	disampaikan	diminta
antaranya	mereka	kok	disebut	dimintai
apa	merekalah	kurang	disebutkan	semampunya
apaan	merupakan	lagi	disebutkannya	semasa
apabila	meski	lagian	disini	semasih
apakah	meskipun	lah	disinilah	semata
apalagi	meyakini	lain	ditambahkan	semata-mata
apatah	meyakinkan	lainnya	ditandaskan	semaunya
artinya	minta	lalu	ditanya	sementara
asal	mirip	lama	ditanyai	semisal
asalkan	misal	lamanya	ditanyakan	semisalnya
atas	misalkan	lanjut	ditegaskan	sempat
atau	misalnya	lanjutnya	ditujukan	semua
ataukah	mula	lebih	ditunjuk	semuanya
ataupun	mulai	lewat	ditunjuki	semula
awal	mulailah	lima	ditunjukkan	sendiri
awalnya	mulanya	luar	ditunjukkannya	sendirian
bagai	mungkin	macam	ditunjuknya	sendirinya
bagaikan	mungkinkah	maka	dituturkan	seolah
bagaimana	nah	makanya	dituturkannya	seolah-olah
bagaimanakah	naik	makin	diucapkan	seorang
bagaimanapun	namun	malah	diucapkannya	sepanjang
bagi	nanti	malahan	diungkapkan	sepantasnya
bagian	nantinya	mampu	dong	tanyakan
bahkan	nyaris	mampukah	dua	tanyanya
bahwa	nyatanya	mana	dulu	tapi
bahwasanya	oleh	manakala	empat	tegas
baik	olehnya	manalagi	enggak	tegasnya
bakal	pada	masa	enggaknya	telah
bakalan	padahal	masalah	entah	tempat
balik	padanya	masalahnya	entahlah	tengah
banyak	pak	masih	guna	tentang
bapak	paling	masihkah	gunakan	tentu
baru	panjang	masing	hal	tentulah
bawah	pantas	masing-masing	hampir	tentunya
beberapa	para	mau	hanya	tepat
begini	pasti	maupun	hanyalah	terakhir
beginian	pastilah	melainkan	hari	terasa
beginikah	penting	melakukan	harus	terbanyak
beginilah	pentingnya	melalui	haruslah	terdahulu
begitu	per	melihat	harusnya	terdapat
begitukah	percuma	melihatnya	hendak	terdiri

begitulah	perlu	memang	hendaklah	terhadap
begitupun	perlukah	memastikan	hendaknya	terhadapnya
bekerja	perlunya	memberi	hingga	Ucapnya
belakang	pernah	memberikan	ia	ujar
belakangan	persoalan	membuat	ialah	ujarnya
belum	pertama	memerlukan	ibarat	umum
belumlah	pertama-tama	memihak	ibaratkan	umumnya
benar	pertanyaan	meminta	ibaratnya	ungkap
benarkah	pertanyakan	memintakan	ibu	ungkapnya
benarlah	pihak	memisalkan	ikut	untuk
berada	pihaknya	memperbuat	ingat	usah
berakhir	pukul	mempergunakan	ingat-ingat	usai
berakhirlah	pula	memperkirakan	ingin	waduh
berakhirnya	pun	memperlihatkan	inginkah	wah
berapa	punya	mempersiapkan	inginkan	wahai
berapakah	rasa	mempersoalkan	ini	waktu
berapalah	rasanya	mempertanyakan	inikah	waktunya
berapapun	rata	mempunyai	inilah	walau
berarti	rupanya	memulai	itu	walaupun
berawal	saat	memungkinkan	itukah	wong
berbagai	saatnya	menaiki	itulah	yaitu
berdatangan	saja	menambahkan	jadi	yakin
beri	sajalah	menandaskan	jadilah	yakni
berikan	saling	menanti	jadinya	Yang
berikut	sama	menanti-nanti	jangan	diberikannya
berikutnya	sama-sama	menantikan	jangankan	dibuat
berjumlah	sambil	menanya	janganlah	dibuatnya
berkali-kali	sampai	menanyai	jauh	didapat
berkata	sampai-sampai	menanyakan	jawab	didatangkan
berkehendak	sampaikan	mendapat	jawaban	digunakan
berkeinginan	sana	mendapatkan	jawabnya	diibaratkan
berkenaan	sangat	mendatang	jelas	diibaratkannya
berlainan	sangatlah	mendatangi	jelaskan	Diingat
berlalu	satu	mendatangkan	jelaslah	selama
berlangsung	saya	menegaskan	jelasnya	selama-lamanya
berlebihan	sayalah	mengakhiri	jika	selamanya
bermacam	se	mengapa	jikakalau	selanjutnya
bermacam-macam	sebab	mengatakan	juga	seluruh
bermaksud	sebabnya	mengatakannya	jumlah	seluruhnya
bermula	sebagai	mengenai	jumlahnya	semacam
bersama	sebagaimana	mengerjakan	justru	semakin
bersama-sama	sebagainya	mengetahui	kala	semampu
bersiap	sebagian	menggunakan	kalau	tak

bersiap-siap	sebaik	menghendaki	kalaulah	tambah
bertanya	sebaik-baiknya	mengibaratkan	kalaupun	tambahnya
bertanya-tanya	sebaiknya	mengibaratkannya	kalian	tampak
berturut	sebaliknya	mengingat	kami	tampaknya
berturut-turut	sebanyak	mengingatkan	kamilah	tandas
bertutur	sebegini	menginginkan	kamu	tandasnya
berujar	sebegitu	Mengira	kamulah	tanpa
berupa	sebelum	sepantasnyalah	kan	Tanya
besar	sebelumnya	seperlunya	kapan	tidaklah
betul	sebenarnya	seperti	kapankah	tiga
betulkah	seberapa	sepertinya	kapanpun	tinggi
biasa	sebesar	sepihak	karena	toh
biasanya	sebetulnya	sering	karenanya	tunjuk
bila	sebisanya	seringnya	kasus	turut
bilakah	sebuah	serta	kata	tutur
bisa	sebut	serupa	katakan	tuturnya
bisakah	sebutlah	sesaat	katakanlah	Ucap
boleh	sebutnya	sesama	katanya	ternyata
bolehkah	secara	sesampai	ke	tersampaikan
bolehlah	secukupnya	sesegera	keadaan	tersebut
buat	sedang	sesekali	kebetulan	tersebutlah
bukan	sedangkan	seseorang	kecil	tertentu
bukankah	sedemikian	sesuatu	kedua	tertuju
bukanlah	sedikit	sesuatunya	keduanya	terus
bukannya	sedikitnya	sesudah	keinginan	terutama
bulan	seenaknya	sesudahnya	kelamaan	tetap
bung	segala	setelah	kelihatan	tetapi
cara	segalanya	setempat	kelihatannya	tiap
caranya	segera	setengah	kelima	tiba
cukup	seharusnya	seterusnya	Keluar	tiba-tiba
cukupkah	sehingga	setiap	teringat	tidak
cukuplah	seingat	setiba	teringat-ingat	Tidakkah
cuma	sejak	setibanya	terjadi	sekiranya
dahulu	sejauh	setidak-tidaknya	terjadilah	sekitar
dalam	sejenak	setidaknya	terjadinya	sekitarnya
dan	sejumlah	setinggi	terkira	sekurang-kurangnya
dapat	sekadar	seusai	terlalu	sekurangnya
dari	sekadarnya	sewaktu	terlebih	sela
daripada	sekali	siap	terlihat	selain
datang	sekali-kali	siapa	termasuk	selaku
dekat	sekalian	siapakah	di	selalu
demi	sekaligus	siapapun	dia	suatu

demikian	sekalipun	sini	diakhiri	sudah
demikianlah	sekarang	sinilah	diakhirinya	sudahkah
dengan	sekecil	soal	dialah	sudahlah
depan	seketika	soalnya	diantara	supaya
diberi	tadinya	tahu	diantaranya	tadi
diberikan	tahun			

3. *Stemming*

   Tahap *stemming* adalah tahap mencari *root* kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Tahap ini kebanyakan dipakai untuk teks berbahasa nggris dan lebih sulit diterapkan pada teks berbahasa Indonesia. Hal ini dikarenakan bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen (Eko, 2011). *Stemming* merupakan suatu proses yang terdapat dalam sistem *IR* yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*root word*) dengan menggunakan aturan-aturan tertentu. Sebagai contoh, kata bersama, kebersamaan, menyamai, akan *distem* ke *root word*nya yaitu "sama". Proses *stemming* pada teks berbahasa Indonesia berbeda dengan *stemming* pada teks berbahasa Inggris. Pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan sufiks. Sedangkan pada teks berbahasa Indonesia, selain sufiks, prefiks, dan konfiks juga dihilangkan (Ledy, 2009).

4. *Tagging*

   Tahap berikutnya adalah Tahap tagging yang merupakan tahap mencari bentuk awal/root dari tiap kata lampau atau kata hasil stemming. Tahap ini tidak dipakai untuk teks bahasa Indonesia dikarenakan bahasa Indonesia tidak memiliki bentuk lampau.

5. *Analyzing*

   Tahap yang terakhir dalam *text mining* adalah tahap *analyzing* yaitu tahap penentuan seberapa jauh keterhubungan antar kata-kata antar dokumen yang ada. Dalam penelitian ini untuk melakukan analisa pada tahap *analyzing* dapat digunakan algoritma TF/IDF (*Term Frequency – Inversed Document Frequency*) dan Algoritma *Vector Space Model.*

2.2.7 Stemming dengan Algoritma Nazief dan Adriani

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

Algoritma yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut:

1. Kata yang hendak di-*stemming* dicari terlebih dahulu pada kamus. Jika kata ditemukan dalam kamus, berarti kata tersebut sudah berbentuk kata dasar (*root word*). Algoritma berhenti, jika tidak maka tahap selanjutnya dilakukan.
2. *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Hilangkan *inflection suffiexes* P (“-lah”, “-kah”, “-tah”, “-pun”) maka langkah ini diulangi lagi untuk menghapus atau *possessive pronoun* PP (“-ku”, “-mu”, “-nya”). Jika ada.
3. Hilangkan *derivation suffixes* DS (“-i”, “-kan”, atau “-an”). Jika kata ditemukan kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a
    a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
    b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hilangkan *derivation prefixes* DP {“di-”,“ke-”,“se-”,“me-”,“be-”,“pe”, “te-”}. Jika pada langkah 3 ada *sufiks* yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.
    a. Langkah 4 berhenti jika:
        1. Terjadi kombinasi awalan dan akhiran yang terlarang seperti pada Tabel

2. Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.
3. Tiga awalan telah dihilangkan.

b. Identifikasikan tipe awalan dan hilangkan. Awalan ada dua tipe:
   1. Standar: “di-”, “ke-”, “se-” yang dapat langsung dihilangkan dari kata.
   2. Kompleks: “me-”, “be-”, “pe”, “te-” adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Oleh karena itu, gunakan aturan pada Tabel untuk mendapatkan pemenggalan yang tepat.

c. Cari kata yang telah dihilangkan awalannya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 4 diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan.

5. Apabila setelah langkah 4 kata dasar masih belum ditemukan, maka proses *recoding* dilakukan dengan mengacu pada aturan pada Tabel *Recoding* dilakukan dengan menambahkan karakter recoding di awal kata yang dipenggal. Pada Tabel, karakter *recoding* adalah huruf kecil setelah tanda hubung (‘-‘) dan terkadang berada sebelum tanda kurung.

Jika semua langkah gagal, maka input kata yang diuji pada algoritma ini dianggap sebagai kata dasar.

Tabel 2. 3 Daftar Prefiks yang Meluluh

Jenis Prefiks	Huruf	Hasil Peluluhan
pe-/me-	K	-ng-
pe-/me-	P	-m-
pe-/me-	S	-ny-
pe-/me-	T	-n-

Tabel 2. 4 Daftar Kemungkinan Perubahan Prefiks

Prefiks	Perubahan
se-	tidak berubah
ke-	tidak berubah
di-	tidak berubah

Prefiks	Perubahan
be-	ber-
te-	ter-
pe-	per-, pen-, pem-, peng-
me-	men-, mem-, meng-

Tabel 2. 5 Daftar Kombinasi Prefiks dan Sufiks yang Tidak Diperbolehkan

Prefiks	Sufiks yang tidak diperbolehkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an
pe-	-kan

2.2.8 TF-IDF (Term Frequency – Inversed Document Frequency)

Metode TF-IDF merupakan metode untuk menghitung bobot setiap kata yang paling umum digunakan pada information retrieval. Metode ini juga terkenal efisien, simpel dan memiliki hasil yang akurat. Metode ini akan menghitung nilai Term Frequency (TF) dan Inverse Document Frequency (IDF) pada setiap token (kata) di setiap dokumen dalam korpus dengan rumus yaitu :

$$\text{IDF} = \log (D/DF) \tag{2.1}$$

Dimana :

D = total dokumen

df = banyaknya dokumen yang mengandung kata yang dicari

$$W_{d.t} = tf_{d.t} * IDF_t \tag{2.2}$$

Dimana :

d = dokumen ke-d

t = kata ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

tf = bantaknya kata yang dicari pada sebuah dokumen

Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses pengurutan dimana semakin besar nilai W, semakin besar tingkat similaritas dokumen tersebut terhadap kata kunci, demikian juga sebaliknya.

2.2.9 Vector Space Model (VSM)

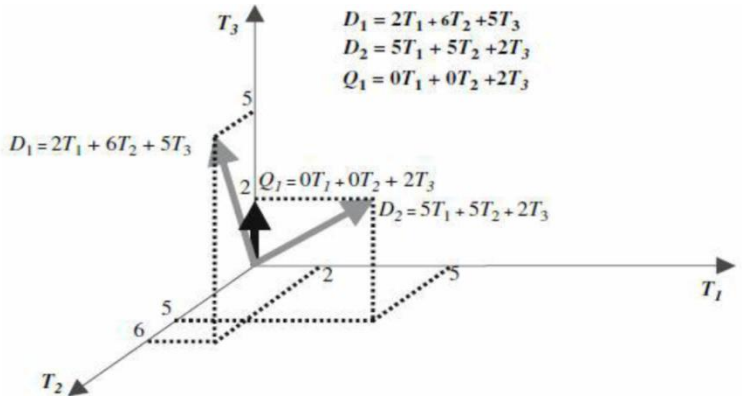
Vector Space Model (VSM) adalah metode untuk melihat tingkat kedekatan atau kesamaan (*similarity*) *term* dengan cara pembobotan *term*. Dokumen dipandang sebagi sebuah vektor yang memiliki *magnitude* (jarak) dan *direction* (arah). Pada *Vector Space Model*, sebuah istilah direpresentasikan dengan sebuah dimensi dari ruang vektor. Relevansi sebuah dokumen ke sebuah *query* didasarkan pada similaritas diantara vektor dokumen dan vektor *query* [13].

Secara umum prosedur VSM dibagi menjadi tiga tahap. Tahap pertama adalah *document indexing*, kumpulan *term* yang telah diekstrak dari teks dokumen. Tahap kedua yaitu pembobotan dari *term* yang sudah di-indeks untuk meningkatkan kualitas pencarian. Tahap terkahir, rangking dokumen berdasarkan tingkat kemiripan antara *query* dengan dokumen[9].

Pada *Vector Space Model* ini :

1. *Vocabulary* merupakan kumpulan semua *term* berbeda yang tersisa dari dokumen setelah preprocessing dan mengandung *t term index*. *Term-term* ini membentuk suatu ruang vektor.
2. Setiap *term i* di dalam dokumen atau query *j*, diberikan suatu bobot (weight) bernilai *real wij*.
3. Dokumen dan *query* diekspresikan sebagai vektor *t* dimensi *dj* = (*w1, w2, ..., wtj*) dan terdapat n dokumen di dalam koleksi, yaitu *j* = 1, 2, ..., n.

Contoh dari model ruang vektor tiga dimensi untuk dua dokumen D1 dan D2, satu query pengguna Q1, dan tiga term T1, T2 dan T3 diperlihatkan pada Gambar 2. 6



Gambar 2. 6 Vector Space Model [11]

Dalam model ruang vektor, koleksi dokumen direpresentasikan oleh matriks *term* document (atau matriks *term-frequency*). Setiap sel dalam matriks bersesuaian dengan bobot yang diberikan dari suatu *term* dalam dokumen yang ditentukan. Nilai nol berarti bahwa *term* tersebut tidak hadir di dalam dokumen[11].

$$\begin{pmatrix} & T_1 & T_2 & \dots & T_t \\ D_1 & W_{11} & W_{21} & \dots & W_{t1} \\ D_2 & W_{12} & W_{22} & \dots & W_{t2} \\ \dots & \dots & \dots & \dots & \dots \\ D_n & W_{1n} & W_{2n} & \dots & W_{tn} \end{pmatrix}$$

Gambar 2. 7 Matriks Term Document

2.2.9.1 Indexing

Pengindeksan / *indexing* dilakukan untuk mendapatkan bobot dari setiap term dalam dokumen. Penghitungan bobot tersebut dilakukan dengan melakukan perhitungan terhadap *Term Frequency* (Tf) dan *Document Frequency* (Df) dari tiap term yang terdapat di koleksi dokumen, nilai Df selanjutnya akan diproses menjadi nilai *Inverse Document Frequency* (Idf) yang akan digunakan dalam perhitungan bobot term[12].

Keberhasilan dari model VSM ini ditentukan oleh skema pembobotan terhadap suatu term baik untuk cakupan lokal maupun global, dan faktor normalisasi [13]. Pembobotan lokal hanya berpedoman pada frekuensi munculnya

term dalam suatu dokumen dan tidak melihat frekuensi kemunculan term tersebut di dalam dokumen lainnya. Pendekatan dalam pembobotan lokal yang paling banyak diterapkan adalah term frequency (tf) meskipun terdapat skema lain seperti pembobotan biner, augmented normalized tf, logaritmik tf dan logaritmik alternatif.

Idf dapat ditentukan dengan menggunakan rumus:

$$idf = \log\left[\frac{d}{df}\right] \quad (2.3)$$

Dengan ketentuan :

Idf = *inverse document frequency*

d = *total dokumen*

df = jumlah dokumen yang mengandung term

Bobot dari setiap *term* dapat dihitung dengan rumus *w = tf * idf*

2.2.9.2 Ukuran Kemiripan

Model ruang vektor dan pembobotan *tf-idf* digunakan untuk merepresentasikan nilai numerik dokumen sehingga kemudian dapat dihitung kedekatan antar dokumen. Semakin dekat dua vektor di dalam suatu VSM maka semakin mirip dua dokumen yang diwakili oleh vektor tersebut. Kemiripan antar dokumen dihitung menggunakan suatu fungsi ukuran kemiripan (*similarity measure*). Ukuran ini memungkinkan perankingan dokumen sesuai dengan kemiripan relevansinya terhadap *query*. Setelah dokumen diranking, sejumlah tetap dokumen *top-scoring* dikembalikan kepada pengguna.

Perhitungan jarak dokumen menggunakan persamaan:

$$|q| = \sqrt{\sum_{j=1}^{t}(W_{i,q})^2} \quad (2.4)$$

Dengan ketentuan:

|q| = jarak *query*

W_{iq} = bobot query dokumen ke-i

Maka jarak *query* (|q|) dihitung untuk didapatkan jarak *query* dari bobot *query* dokumen (W_{iq}) yang terambil oleh sistem. Jarak *query* bisa dihitung dengan persamaan akar jumlah kuadrat dari *query*.

$$|d_j| = \sqrt{\sum_{i=1}^{t}(W_{ij})^2} \quad (2.5)$$

Dengan ketentuan:

|dj | = jarak dokumen

W_{ij} = bobot dokumen ke-i

Maka Jarak dokumen (|dj |) dihitung untuk didapatkan jarak dokumen dari bobot dokumen dokumen (W_{ij}) yang terambil oleh sistem. Jarak dokumen bisa dihitung dengan persamaan akar jumlah kuadrat dari dokumen.

Perhitungan pengukuran Similaritas query document (*inner product*), menggunakan persamaan:

$$sim(q, d_j) = \sum_{i=1}^{t} W_{iq} \, . W_{ij} \quad (2.6)$$

Dengan Ketentuan:

W_{ij} = bobot *term* dalam dokumen

W_{iq} = bobot *query*

Sim (q, dj) = similaritas antara *query* dan dokumen.

Similaritas antara query dan dokumen atau *inner product*/Sim (q, dj) digunakan untuk mendapatkan bobot dengan didasarkan pada bobot *term* dalam dokumen (W_{ij}) dan bobot *query* (W_{iq}) atau dengan cara menjumlah bobot q dikalikan dengan bobot dokumen.

Cosine Similarity tidak hanya digunakan untuk menghitung normalisasi panjang dokumen tapi juga menjadi salah satu ukuran kemiripan yang popular. Ukuran ini menghitung nilai kosinus sudut antara dua vektor. Jika terdapat dua vektor dokumen *d* dan *query q*, serta *t term* diekstrak dari koleksi dokumen maka nilai kosinus antara *d* dan *q* didefinisikan sebagai [11] :

$$Sim(q, d_j) = \frac{q \cdot d_j}{|q| * |d_j|} = \frac{\sum_{i=1}^{t} W_{iq} \, . W_{ij}}{\sqrt{\sum_{j=1}^{t}(W_{iq})^2} * \sqrt{\sum_{i=1}^{t}(W_{ij})^2}} \quad (2.7)$$

Similaritas antara *query* dan dokumen atau Sim(q,dj) berbanding lurus terhadap jumlah bobot *query* (q) dikali bobot dokumen (dj) dan berbanding

terbalik terhadap akar jumlah kuadrat q (|q|) dikali akar jumlah kuadrat dokumen (|dj|). Perhitungan similaritas menghasilkan bobot dokumen yang mendekati nilai 1 atau menghasilkan bobot dokumen yang lebih besar dibandingkan dengan nilai yang dihasilkan dari perhitungan inner product.

2.2.10 Analisis SWOT

Analisis SWOT adalah identifikasi berbagai faktor secara sistematis untuk merumuskan strategi perusahaan. Analisis ini didasarkan pada logika yang dapat memaksimalkan kekuatan (*Strenghts*) dan peluang (*Oppurtunities*), namun secara bersamaan dapat meminimalkan kelemahan (*Weaknesess*) dan ancaman (*Threats*). Proses pengambilan keputusan strategis selalu berkaitan dengan pengembangan misi, tujuan, strategi, dan kebijakan perusahaan. Dengan demikian perencanaan strategis perusahaan (kekuatan, kelemahan, peluang, dan ancaman) dalam kondisi yang ada saat ini. Hal ini disebut analisis situasi yaitu model yang paling popular untuk analisis situasi adalah Analisis SWOT.

Penelitian menunjukkan bahwa kinerja perusahaan dapat ditentukan oleh faktor internal dan eksternal. Kedua faktor tersebut harus dipertimbangkan dalam analisis SWOT. SWOT adalah singkatan dari lingkungan *Internal Strengths* dan *Weaknesses* serta lingkungan *External Opportunities* dan *Threats* yang dihadapi dunia bisnis. Analisis SWOT membandingkan antara faktor eksternal peluang (*Opportunities*) dan ancaman (*Threats*) dengan faktor internal kekuatan (*Strengths*) dan kelemahan (*Weakness*). Analisis SWOT hanya menggambarkan situasi yang terjadi bukan sebagai pemecah masalah. Analisis SWOT terdiri dari empat faktor, yaitu:

1. ***Strengths* (*kekuatan*)**

   Merupakan kondisi kekuatan yang terdapat dalam organisasi, proyek atau konsep bisnis yang ada. Kekuatan yang dianalisis merupakan faktor yang terdapat dalam tubuh organisasi, proyek atau konsep bisnis itu sendiri.

2. ***Weakness*** **(kelemahan)**

   Merupakan kondisi kelemahan yang terdapat dalam organisasi, proyek atau konsep bisnis yang ada.Kelemahan yang dianalisis merupakan faktor yang terdapat dalam tubuh organisasi, proyek atau konsep bisnis itu sendiri.

3. ***Opportunities*** **(peluang)**

   Merupakan kondisi peluang berkembang di masa datang yang terjadi. Kondisi yang terjadi merupakan peluang dari luar organisasi, proyek atau konsep bisnis itu sendiri. misalnya kompetitor, kebijakan pemerintah, kondisi lingkungan sekitar.

4. ***Threats*** **(ancaman)**

   Merupakan kondisi yang mengancam dari luar. Ancaman ini dapat mengganggu organisasi, proyek atau konsep bisnis. David tidak memakai singkatan SWOT seperti yang lazim didengar, tetapi lebih senang menggunakan TWOS yang tampaknya ingin mendahulukan analisis Ancaman dan Peluang untuk kemudian melihat sejauh mana kapabilitas internal sesuai dan cocok dengan faktor-faktor eksternal tersebut. Ada empat strategi yang tampil dari hasil analisis TWOS.

Strategi SO dipakai untuk menarik keuntungan dari peluang yang tersedia dalam lingkungan eksternal.

Strategi WO bertujuan untuk memperbaiki kelemahan internal dengan memanfaatkan peluang dari lingkungan luar.

Strategi ST akan digunakan organisasi untuk menghindari, paling tidak memperkecil dampak dari ancaman yang datang dari luar.

Strategi WT adalah taktik pertahanan yang diarahkan pada usaha memperkecil kelemahan internal dan menghindari ancaman eksternal (Salusu, 1996).

Analisis seluruh faktor internal dan eksternal yang ada. Dapat dihasilkan empat macam strategi organisasi dengan karakteristiknya masing-masing, yakni sebagai berikut:

Internal / Eksternal	Strengths (S) 1. 2. 3.	Weaknesses (W) 1. 2. 3.
Threat (T) 1. 2. 3.	Strategi ST Gunakan S untuk menghindari T	Strategi WT Minimalkan W dan hindari T
Opportunity (O) 1. 2. 3.	Strategi SO Gunakan S untuk memanfaatkan O	Strategi WO Atasi W dengan memanfaatkan O

Gambar 2. 8 Matrik SWOT

Analisis seluruh faktor internal dan eksternal yang ada. Dari matriks tiga dapat dihasilkan empat macam strategi organisasi dengan karakteristiknya masing-masing, yakni sebagai berikut:

1. Strategi SO adalah strategi yang harus dapat menggunakan kekuatan sekaligus memanfaatkan peluang yang ada.
2. Strategi WO adalah strategi yang harus ditunjukkan untuk mengurangi kelemahan yang dihadapi dan pada saat yang bersamaan memanfaatkan peluang yang ada.
3. Strategi ST adalah strategi yang harus mampu menonjolkan kekuatan guna mengatasi ancaman yang mungkin timbul.
4. Strategi WT adalah strategi yang bertujuan mengatasi hambatan serta meminimalkan dampak dari ancaman yang ada.

2.2.11 Business Process Modelling Notation (BPMN)

Sebuah model proses bisnis terdiri dari serangkaian model kegiatan dan *constraint* antara model-model kegiatan (Weske, 2007). BPMN merupakan singkatan dari *Business Process Modelling Notation*, yaitu suatu metodologi yang dikembangkan *Business Process Modelling Initiative* (BPMI) dalam memodelkan proses bisnis (Object Management Group, 2008). Tujuan dari BPMN adalah menyediakan notasi yang mudah dipahami oleh semua pengguna bisnis dan yang tidak kalah pentingnya adalah untuk memastikan bahwa bahasa XML yang dirancang untuk pelaksanaan proses bisnis dapat dinyatakan secara visual dengan notasi yang umum (Owen and Raj, 2003). Tidak seperti jenis diagram proses

bisnis yang sebelumnya, BPMN telah ditambahkan notasi khusus untuk menggambarkan peristiwa berbasis pesan dan pesan lewat diantara organisasi. Adapun notasi dasar yang ada pada BPMN dapat dilihat pada Tabel 2.1.

Adapun notasi yang digunakan pada proses yang lebih kompleks, seperti pada saat proses B2B (*business-to-business*). Notasi yang dimaksud seperti notasi pesan (*message*), waktu (*timer*), tautan (*link*), dan kondisi kesalahan (*error condition*).

Tabel 2. 6 Tabel Notasi Dasar BPMN

Start Event		Intermediate Event		End Event	
	Digunakan untuk memulai sebuah proses		Terjadi selama proses		Digunakan untuk mengakhiri sebuah proses

2.2.12 Analisis Sistem Terstruktur

Analisa sistem terstruktur adalah pendekatan dalam analisa sistem yang bermula dari deskripsi sangat umum mengenai sistem tertentu dan kemudian diproses dengan rangkaian langkah – langkah logis yang semakin rinci. Analisa sistem adalah proses memahami sistem yang sedang berjalan beserta masalahnya, menguraikan kebutuhan - kebutuhan informasi dan menetapkan prioritas untuk pekerjaan sistem selanjutnya.

2.2.12.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram adalah suatu penyajian data dengan mengunakan Entity dan Relationship. Komponen utama ERD adalah : (Jogiyanto, hal 810-820, 1990).

1. **Atribut**

   Secara umum adalah sifat atau karakteristik dari suatu Entitas.



Gambar 2. 9 Simbol Atribut

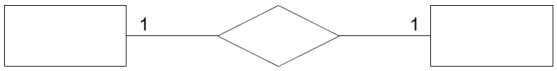
2. Cardinality

Dalam *Entity Relationship Diagram* terdapat tingkat hubungan antara hubungan yang satu dengan yang lain dilihat dari segi bahasa jumlah ketergantungan dalam suatu *Entity* dengan *Entity* lain inilah yang dinamakan *Cardinality*.

Ada tiga kemungkinan yang terjadi yaitu :

a. One to One (1:1)

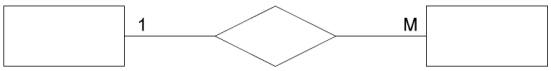
Terjadi suatu hubungan yang hanya memiliki sebuah hubungan yang satu dengan yang lain.



Gambar 2. 10 Cardinality One To One

b. One to Many / Many to One (1:M / M:1)

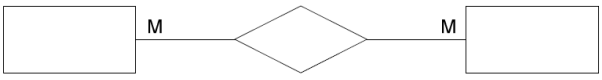
Terjadi apabila sebuah hubungan memiliki banyak hubungan atau banyak hubungan memiliki sebuah hubungan lain.



Gambar 2. 11 Cardinality One To Many / Many To One

c. Many to Many (M:M)

Terjadi apabila banyak hubungan memiliki banyak hubungan.



Gambar 2. 12 Cardinality Many To Many

3. Entitas Objek (Objek Data)

Entitas adalah suatu objek yang dapat dibedakan dari yang lain.

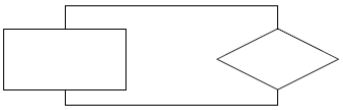


Gambar 2. 13 Simbol Entitas

Participation Containt menjelaskan apakah keberadaan suatu Entity tergantung pada hubungan dengan Entity lain. Terdapat dua Perticipation Containt yaitu :

a. Total Participation

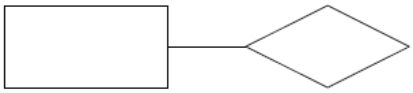
Yaitu keberadaan suatu Entity tergantung pada Entity yang lain. Didalam Entity Relationship digambarkan dengan dua garis penghubung antar Entity dengan Relationship.



Gambar 2. 14 Total Participation.

b. Partial Participation

Yaitu keberadaan suatu Entity tidak tergantung pada hubungannya dengan Entity lainnya. Didalam Entity Relationship digambarkan dengan satu garis penghubung antar Entity dengan Relationship.



Gambar 2. 15 Partial Participation

2.2.12.2 Normalisasi

Dengan pendekatan database, ingin dicapai agar program tidak tergantung pada struktur dan teknik akses data (data independence) serta data yang rangkap atau berulangulang dapat diminimalkan sehingga memudahkan pemeliharaan data. Suatu data atau file yang terdiri dari beberapa grup elemen yang berulang-ulang perlu diorganisaikan kembali. Proses untuk mengorganisasikan file untuk menghilangkan grup elemen yang berulang-ulang ini disebut normalisasi *(normalization).* (Jogiyanto, hal403, 1999).

Normalisasi dibagi atas beberapa tingkatan, antara lain :

1. Bentuk tidak normal (unnormalized form), merupakan kumpulan data yang direkam, tidak ada keharusan mengikuti format tertentu dan dapat saja tidak lengkap. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

2. Bentuk normal pertama (first normal form/1NF), dicapai bila tiap nilai atribut adalah tunggal. Kondisi ini dapat diperoleh dengan menghapus semua elemen yang berulang dalam suatu entitas.
3. Bentuk normal kedua (second normal form/2NF) adalah bentuk normal pertama dengan menghilangkan atribut data yang tidak tergantung fungsional penuh pada kandidat kunci.
4. Bentuk normal ketiga (third normal form/3NF) adalah bentuk normal kedua dengan menghilangkan semua atribut data yang tergantung transitif pada kandidat kunci (tidak adanya ketergantungan transitif).
5. Bentuk normal Boyce-Codd (Boyce-Codd normal form/BCNF) adalah bentuk normal di mana setiap atribut penentu merupakan calon kunci.

2.2.12.3 Diagram Alir Data (*Data Flow Diagram*)

Data Flow Diagram adalah diagram untuk mengambarkan arus data didalam sistem terstruktur dan jelas yang dikembangkan secara logika. Data Flow Diagram sangat membantu sekali dalam berkomonikasi dengan pemakai sistem untuk memahami secara logika. Simbol dalam Data Flow Diagram adalah: (Jogiyanto, hal 800 - 805, 1990)

1. **Terminator**

   Terminator merupakan sumber atau tujuan data dan dilambangkan dengan persegi panjang.



Gambar 2. 16 Simbol Terminator

2. **Data Flow (Arus Data)**

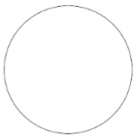
   Data Flow atau Arus Data menunjukan keluar atau masuk kesuatu proses. Jadi sebagai alur data atau informasi dari suatu bagian kebagian lainnya dalam suatu sistem.



Gambar 2. 17 Simbol Arus Data

3. **Process (Proses)**

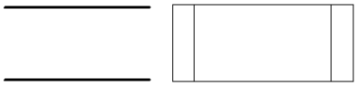
   Proses merupakan kegiatan yang dilakukan oleh orang, mesin, atau komputer dari hasil suatu arus data yang masuk kedalam proses, untuk dihasilkan arus data yang akan keluar dari proses.



Gambar 2. 18 Simbol Proses

4. **Data Store (Simpan Data)**

   Data Store digunakan sebagai sarana untuk pengumpulan data. Data Store juga diberi nama.



Gambar 2. 19 Simbol Store

2.2.13 Black Box Testing

Pengujian black box berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan analisis sistem memperoleh kumpulan kondisi input yang akan mengerjakan seluruh keprluan funsional program [14].

Tujuan dari metode ini mencari kesalahan pada :

1. Fungsi yang salah atau hilang
2. Kesalahan pada interface
3. Kesalahan pada struktur data atau akses database
4. Kesalahan performasi
5. Kesalahan inisialisasi dan tujuan akhir

Metode ini tidak fokus pada struktur kontrol seperti pada pengujian *white-box* tetapi pada domain informasi.

Pengujian dirancang untuk menjawab pertanyaan sebagai berikut :

1. Bagaimana validasi fungsional diuji?
2. Apa kelas input yang terbaik untuk uji coba yang baik?
3. Apakah sistem sangat peka terhadap nilai nilai input tertentu?
4. Bagaimana jika kelas data yang terbatas dipisahkan?
5. Bagaimana volume data yang dapat ditolerasi oleh sistem?
6. Bagaimana pengaruh kombinasi data terhadap pengoprasian sistem?

2.2.13.1 Tipe dari Pengujian Black Box

1. *Equivalence Class Testing*

   Tipe pengujian ini memecah atau membagi domain input dari program kedalam kelas-kelas data sehingga test case dapat diperoleh. Perancangan *equivalence test* berdasarkan evaluasi kelas equivalence untuk kondisi input yang menggambarkan keadaan yang valid atau tidak. Kondisi input dapat berupa nilai numeric, range nilai, kumpulan nilai yang berhubungan atau kondisi Boolean.

2. *Sample testing*

   Melibatkan dari sejumlah nilai yang dipilih dari data masukan kelas ekivalensi kemudian mengintegrasikan nilai tersebut kedalam kasus uji. Nilai yang dipilih dapat berupa konstanta atau variable.

3. *Limit testing*

   Limit testing adalah pengujian dengan kasus uji yang memperolah nilai batas (atau titik singular). Nilai batas disimpulkan dari batas ekivalensi dengan mengmbil nilai yang sama atau mendekati nilai yang membatasi kelas ekivalensi tersebut. Limit juga melibatkan data keluaran dari ekivalensi kelas pada kasus uji segitiga misalnya limit testing mencoba untuk menditeksi apakah a+b >= c dan bukan a+b>c. bila kondisi input menentukan suatu range, maka kasus ujinya harus mencakup pengujian nilai batas dari range dan nilai invalid yang dekat dengan nilai batas. Bila kondisi inputnya berupa harga khusus kasus ujinya harus mengcakup nilai minimum dan maksimum. Misalnya suatu file dapat terdiri dari 1 sampai 255 record,

maka kasus ujinya harus mencakup untuk nilai 0,1,225 dan 256, atau uji saat keadaan record kosong dan record penuh.

4. *Robustness testing*

   Jenis tipe pengujian ini data dipilih dari luar range yang di definisikan. Tujuan dari pengujian ini adalah membuktikan tidak adanya kejadian yang katastropik yang dihasilkan akibat adanya keabnormalan.

5. *Behavior testing*

   Suatu pengujian yang hasil hasnya dapat dievaluasi per sub program, tidak bisa dilakukan permodul (CSU : *Computer Software Unit*)

6. *Requitment Testing*

   Menyusun kasus uji untuk setiap kebutuhan yang berkorelasi dengan modul. Tiap kasus uji harus dapat dirunut dengan kebutuhan perangkat lunaknya melalui matriks keterunutan.

2.2.14 Web

Menurut Yuhefizar "World Wide Web atau WWW atau juga dikenal dengan WEB adalah salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet. Web ini menyediakan informasi bagi pemakai komputer yang terhubung ke internet dari sekedar informasi kecil atau informasi yang tidak berguna sama sekali sampai informasi yang serius, dari informasi yang gratisan sampai informasi yang komersial."(Yuhefizar, 2008)

Menurut Wahana Komputer "*Web* adalah formulir komunikasi interaktif yang digunakan pada suatu jaringan komputer." (Wahana, 2003) Adapun cara kerja *web* adalah sebagai berikut:

a. Informasi *web* disimpan dalam dokumen dalam bentuk halaman-halaman *web* atau *web page*.
b. Halaman *web* tersebut disimpan dalam *server web* komputer.
c. Sementara dipihak pemakai ada komputer yang bertindak sebagai komputer klien dimana ditempatkan program untuk membaca halaman web yang ada di *server web* (*browser*).
d. Browser membaca halaman *web* yang ada di *server web*.

2.2.15 Website

Menurut YM Kusuma Ardhana "*Website* adalah sejumlah halaman *web* yang memiliki topik saling terkait, terkadang disertai pula dengan berkas-berkas gambar, video, atau berkas lainnya."(Ardhana, 2012).

Penemu situs *web* adalah Sir Timothy Kohn Berners-Lee, sedangkan situs *web* yang tersambung dengan jaringan pertama kali muncul pada tahun 1991.

Website mempunyai fungsi yang bermacam-macam, tergantung dari tujuan dan jenis *website* yang dibangun, tetapi secara garis besar fungsi *website* menurut YM Kusuma Ardhana adalah sebagai berikut (Ardhana, 2012):

a. Media Promosi
b. Media Pemasaran
c. Media Informasi
d. Media Pendidikan

Perkembangan *web* berdasarkan teknologi dan cara penggunaannya menurut pada praktisi informatika adalah sebagai berikut:

a. *Web* 1.0

Web 1.0 merupakan teknologi awal dari sebuah *website*, teknologi ini masih statis dimana antara pembuat *website* dan penikmat *website* hanya tejadi komunikasi 1 arah dimana pembuat sebagai pemberi informasi dan penikmat hanya sebagai pembaca. *Web* 1.0 dikembangkan untuk pengaksesan informasi dan memiliki sifat yang sedikit interaktif. Secara garis besar, sifat *Web* 1.0 adalah *Read-Only* dan bahasa yang digunakan pada *Web* 1.0 menggunakan bahasa HTML.

b. Web 2.0

Sifat Web 2.0 adalah *Read-Write*. *Web* 2.0 lebih menekankan pada perubahan cara berpikir dalam menyajikan konten dan tampilan di dalam sebuah *website*. Di era *Web* 2.0 sekarang, *web* digunakan untuk berbagi, pertemanan, kolaborasi menjadi sesuatu yang penting. *Web* 2.0 memungkinkan pengguna untuk melakukan lebih dari sekedar mengambil informasi. Mereka dapat membangun pada fasilitas interaktif. Kunci perbedaan dalam *Web* 2.0 dan *Web* 1.0 adalah keterbatasan pada *Web* 1.0

yang mengharuskan pengguna internet untuk datang ke dalam *website* tersebut dan melihat satu persatu konten di dalamnya. Sedangkan *Web* 2.0 memungkinkan pengguna internet dapat melihat konten suatu website tanpa harus berkunjung ke alamat situs yang bersangkutan.Kemampuan *web* 2.0 dalam melakukan aktivitas *drag and drop, auto complete, chat, voice* dapat dilakukan layaknya aplikasi *desktop*.

c. *Web* 3.0

Web 3.0 adalah generasi ketiga dari layanan internet berbasis *web*. *Web* 3.0 merupakan pengembangan dari *website* di mana konten *web* tidak hanya dalam format bahasa manusia yang umum , tetapi juga dalam format yang dapat dibaca oleh bahasa mesin. *Web* ini masih dalam tahap pengembangan. Akan tetapi, banyak yang memperkirakan *web* ini akan berkembang pesat pada tahun 2010-2020. Konsep *Web* 3.0 itu sendiri pertama kali diperkenalkan pada tahun 2001, saat Tim Berners-Lee, penemu *World Wide Web,* menulis sebuah artikel ilmiah yang menggambarkan *Web* 3.0 sebagai sebuah sarana bagi mesin untuk membaca halaman-halaman *Web*. Fungsi *web* menjadi wadah untuk pertukaran data, informasi, dan pengetahuan yang dapat menghasilkan kecerdasan buatan yang dapat mengerti keinginan penggunanya, dengan menggunakan *web* 3.0 kita juga diberikan keleluasaan untuk dapat melakukan modifikasi pada *website* itu sendiri.

2.2.16 Web Server

Web server adalah suatu *server* yang memberikan layanan kepada konsumen yang meminta informasi yang berkaitan dengan *web*. Fungsi utama dari *web server* adalah untuk mentransfer data dan seluruh berkasnya dalam sebuah *web* termasuk di dalamnya teks, gambar, video dan lainnya atas permintaan klien melalui protokol komunikasi yang telah ditentukan.

2.2.17 Web Based Application

Web based application adalah suatu aplikasi yang dapat berjalan dengan menggunakan basis teknologi *web* atau *browser*. Aplikasi ini dibuat dengan menggunakan program *server-side.*

Server-side adalah *script* yang diolah di *server*. *Script* adalah program yang ditulis dalam bahasa pemrograman khusus dan biasanya terdiri dari serangkaian perintah. *Script* biasanya bekerja bila suatu program aplikasi dijalankan. Contoh-contoh *server-side*: ASP, JSP, PHP, dan bahasa lainnya.

a. ASP *(Active Server Side)* merupakan turunan dari MS Visual Basic yang dibuat oleh Microsoft.
b. JSP *(Java Server Page)* merupakan bahasa berbasis Java yang dikembangkan oleh Sun Microsistem
c. PHP *(PHP Hypertext PreProsessor atau PHP Hypertext Preprocessing)* merupakan bahasa pemrograman yang paling banyak komunitasnya saat ini dan dikembangkan secara *open source*.

2.2.18 MYSQL

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public Licence*). Dimana setiap orang bebas untuk menggunakannya, tapi tidak boleh dijadikan produk turunan yang bersifat Closed Source atau komersial.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structur Query Language*). SQL adalah sebuah konsep pengoperasian database terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

MySQL merupakan database yang paling digemari dikalangan *Programmer Web*, dengan alasan bahwa program ini merupakan database yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data. Sebagai sebuah database server yang mampu untuk memanajemen database dengan baik, MySQL terhitung merupakan *database* yang paling digemari dan paling banyak digunakan dibandingkan database lainnya.

Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya sebagai data base server lainnya dalam

query data. Hal ini terbukti untuk *query* yang dilakukan oleh single user, kecepatan *query* MySQL bisa sepuluh kali lipat lebih cepat.

2.2.19 PHP

PHP berawal dari skrip *Perl/CGI* yang dibuat oleh seorang pengembang perangkat lunak bernama Rasmus Lerdorf untuk menghitung jumlah pengunjung *homepage*-nya. Karena banyaknya pengunjung yang meminta skrip tersebut, Lerdorf akhirnya membagi-bagikan skrip buatannya yang diberi nama *Personal Home Page (PHP).*

Banyaknya permintaan membuat Lerdorf terus mengembangkan skripnya. Beberapa orang akhirnya bergabung membentuk tim untuk mengembangkan PHP. Sejak itu PHP berkembang pesat dengan banyak fungsi baru yang ditambahkan. Kepanjangan dari PHP kini berubah menjadi PHP: Hypertext Preprocessor.

Ada tiga macam penggunaan PHP:

1. *Server-side scripting*. Ini merupakan jenis penggunaan yang paling banyak dilakukan pengguna PHP. Untuk menggunakannya, dibutuhkan tiga hal: PHP *parser*, aplikasi *web server* yang terkoneksi dengan instalasi PHP, dan aplikasi *web browser*.
2. *Command line scripting*. Pada penggunaan PHP jenis ini hanya dibutuhkan PHP parser.
3. Pembuatan aplikasi berbasis *desktop*. Pada penggunaan PHP jenis ini, dibutuhkan ekstensi tambahan *PHP-GTK*.

PHP memiliki empat kelebihan utama yang menarik minat banyak pengguna. Kelebihan utama PHP tersebut diringkas dalam 4P berikut:

1. *Practicality*. PHP dibuat dengan menitikberatkan pada kepraktisan. Hasilnya, PHP adalah bahasa pemrograman minimalis, dilihat dari segi kebutuhan pengguna dan kebutuhan sintaks.
2. *Power*. PHP memiliki banyak kemampuan, mulai dari kemampuan untuk terhubung dengan basis data, membuat halaman web dinamis, membuat dan memanipulasi berkas gambar, Flash dan PDF, berkomunikasi dengan bermacam protokol seperti IMAP dan POP3, dan masih banyak lagi.

3. *Possibility*. PHP dapat menyediakan lebih dari satu solusi untuk suatu masalah
4. *Price*. PHP selalu dirilis kepada publik tanpa ada batasan untuk penggunaan, modifikasi, atau redistribusi.

2.2.20 Cascading Style Sheets (CSS)

CSS adalah sebuah bahasa *style sheet* (lembar gaya) yang digunakan untuk mengatur tampilan dokumen yang ditulis dalam bahasa *markup*. *CSS Level 1* (CSS1) diperkenalkan pada tahun 1995 oleh *World Wide Web Consortium* (W3C) dan dimaksudkan untuk mengatasi keterbatasan HTML dalam hal keleluasaan pengaturan desain dan tampilan pada sebuah dokumen HTML. Setahun kemudian CSS1 diberi status rekomendasi penuh oleh W3C yang juga mengatur spesifikasi CSS. Saat ini ada tiga level CSS, yaitu CSS *Level* 1 (*Recommendation*), CSS *Level* 2 (*Recommendation*), dan CSS *Level* 2 *Revision* 1 (*Candidate Recommendation*). Penggunaan CSS paling banyak untuk memformat halaman web yang ditulis dengan HTML dan XHTML. Walau demikian, CSS dapat dipergunakan untuk bahasa markup lain seperti SVG dan XUL.

2.2.21 XAMPP

XAMPP merupakan singkatan dari X (empat *operating system* apapun) Apache, MySQL, PHP, Perl. XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dalam paketnya sudah terdapat Apache (web server), MySQL (database), PHP (server side scripting), Perl, FTP server, phpMyAdmin dan berbagai pustaka bantu lainnya. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis.