BAB 2
LANDASAN TEORI

2.1 Tinjauan Sekolah

Pada tahap ini merupakan tinjauan terhadap tempat penelitian yaitu SMK Negeri 1 Bulakamba, diantaranya adalah sejarah singkat sekolah, visi dan misi sekolah dan struktur organisasi serta deskripsi pekerjaan.

2.1.1 Profil SMK Negeri 1 Bulakamba

SMK Negeri 1 Bulakamba merupakan sekolah menengah kejuruan Negeri pertama di kabupaten Brebes yang berbasis kejuruan teknologi. SMK Negeri 1 Bulakamba beralamat di Jalan Raya Kluwut kec. Bulakamba kab.Brebes dengan akses yang mudah dicapai karena berada di jalur pantura kabupaten Brebes, sekolah ini dapat ditempuh 14 km sebelah barat ibukota kabupaten Brebes, atau 13 km dari perbatasan Jawa Tengah – Jawa Barat.

Sekolah ini mulai berdiri pada tahun ajaran 1999/2000. Awalnya sekolahan ini hanya terdiri dari empat program keahlian yaitu Teknik Mekanik Otomotif, Teknik Audio Video, Teknik Instalasi Tenaga Listrik, dan Agronomi Tanaman Pangan Hortikultura. Namun semenjak tahun 2004 sudah bertambah satu program keahlian yaitu nautika kapal penangkap ikan.

Sebagai salah satu subsistem pendidikan nasional, SMK Negeri 1 Bulakamba memiliki tanggung jawab yang besar dalam membangun bansa melalui pendidikan teknologi dan kejuruan, karena itu SMK Negeri 1 Bulakamba senantiasa harus berada pada posisi terdepan dalam setiap upaya pembaharuan pendidikan khususnya dalam bidang Pendidikan Teknologi dan Kejuruan. Untuk mewujudkannya SMK Negeri 1 Bulakamba mulai menerapkan sistem manajemen mutu sesuai standar ISO 9001 : 2008.

2.1.2 Visi dan Misi

Adapun visi yang diterapkan dan dilaksanakan sebagai pedoman wajib di SMK Negeri 1 Bulakamba adalah :

"SMK Negeri 1 Bulakamba penghasil tamatan yang disiplin, unggul, iman dan terampil"

Sedangkan misi dari SMK Negeri 1 Bulakamba adalah

1) Menyelenggarakan kegiatan belajar mengajar yang produktif sesuai tuntutan standar kompetensi dunia kerja
2) Menumbuhkan budaya hidup disiplin dan unggul di kalangan warga sekolah yang dilandasi nilai keimanan.

Selain Visi dan Misi SMK Negeri 1 Bulakamba pun memiliki kebijakan mutu yang ditujukan untuk memberikan layanan pendidikan yang tinggi dan mampu membentuk peserta didik dan mutu organisasi dengan slogan DUIT APIK :

D = Disiplin yang tinggi

U = Unggul dalam prestasi

I = Iman yang kuat

T = Terampil dalam berkarya

A = Akuntabel

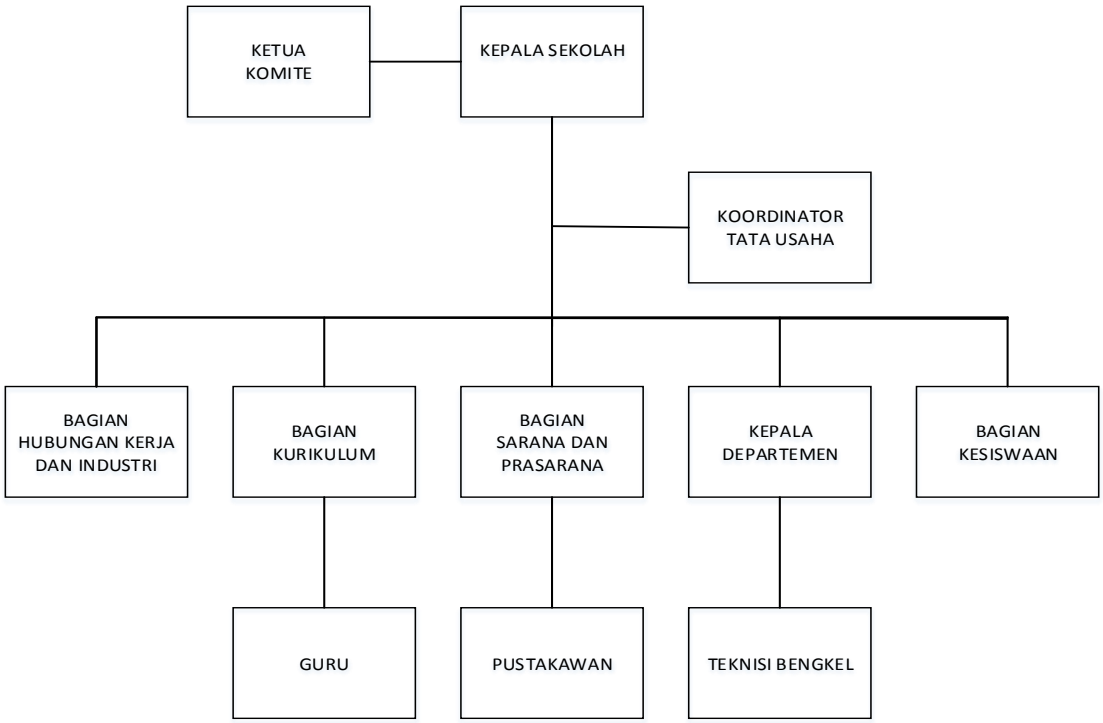
P = Partisipatif

I = Inovatif

K = Kompetitif

2.1.3 Struktur Organisasi Sekolah

Dibawah ini adalah bagan struktur organisasi SMK N 1 Bulakamba



Gambar 2. 1. Struktur Organisasi SMK N 1 Bulakamba

2.1.4 Deskripsi Pekerjaan

Dibagian ini akan menerangkan gambaran besar tentang tugas-tugas satuan organsiasai yang ada di SMK Negeri 1 Bulakamba.

1. Ketua Komite

   Komite sekolah adalah sebagai pengawas mutu/kualitas pendidikan di suatu sekolah. Ketua komite bertugas mengawasi transparansi perencanaan pengadaan dana dan penggunaannya. Selain itu komite juga merupakan mitra kepala sekolah untuk mewadahi peran serta masyarakat

dalam rangka meningkatkan mutu pendidikan, memberikan masukan, pertimbangan, rekomendasi mengenai kebijaksanaan dan program pendidikan, dukungan finansial dan lain lain yang berkaitan dengan pendidikan. Komite sekolah terdiri dari partisipasi masyarakat karena komite adalah mediator antara masyarakat dengan instansi sekolah

2. Kepala sekolah

   Kepala sekolah adalah pimpinan tertinggi dari suatu instansi pendidikan. dan mempunyai fungsi sebagai berikut :

   a. pendidik , manager, pengelola administrasi, supervisor leader, innovator, dan motivator.
   b. Selain itu kepala sekolah memiliki tugas dalam membimbing guru, karyawan, siswa dan staf dalam masing masing tugasnya serta penyusunan rencana dan program sekolah, administrasi sekolah , melakukan kerjasama / hubungan dengan masyarakat dan memantau jalannya proses belajar mengajar di sekolah.

3. Kepala departemen

   Tugas dari kepala departemen adalah membantu dan bertanggungjawab kepada sekolah dalam beberapa hal sebagai berikut :

   a. menyusun perencanaan,
   b. membuat program kegiatan dan program pelaksanaan,
   c. pengorganisasian, pengarahan, ketenagaan, pengawasan, penilaian, identifikasi dan pengumpulan data,
   d. mewakili kepala sekolah untuk menghadiri rapat khususnya yang berkaitan dengan masalah pendidikan serta pembuatan laporan secara periodik.

4. Bagian kurikulum

   Tugas bagian kurikulum adalah membantu dan bertanggungjawab kepada kepala sekolah dalam

   a. penyusunan program pengajaran,
   b. penjabaran kalender pendidikan,
   c. pembagian tugas guru dan jadwal pelajaran,

d. menyusun jadwal evaluasi belajar dan pelaksanaan ujian akhir,
e. penerapan kriteria persyaratan kenaikan kelas dan ketamatan,
f. mengatur jadwal penerimaan rapor dan STTB,
g. mengkoordinasikan, menyusun dan mengarahkan kelengkapan mengajar,
h. mengatur pelaksanaan program perbaikan dan pengayaan,
i. mengatur pengembangan MGMP/MGBP dan koordinator mata pelajaran,
j. melakukan supervisi administrasi akademis,
k. melakukan pengarsipan program kurikulum,
l. serta penyusunan laporan secara berkala

5. Bagian kesiswaan

Tugas bagian kesiswaan adalah membantu dan bertanggungjawab kepada kepala sekolah dalam hal

a. Menyusun program pembinaan kesiswaan OSIS meliputi kepramukaan, PMR, KIR, UKS, PKS, Paskibraka, pesantren kilat
b. Melaksanakan bimbingan pengarahan dan pengendalian kegiatan kesiswaan dalam rangka menegakan disiplin dan tata tertibsekolah serta pemilihan pengurus OSIS
c. Membina pengurus OSIS dalam berorganisasi
d. Menyusun jadwal dan pembinaan serta secara berkala dan incidental
e. Membina dan melaksanakan koordinasi 9 K
f. Melaksanakan pemilihan calon siswa berprestasi dan penerima bea siswa
g. Mengadakan pemilihan siswa untuk mewakili sekolah dalam kegiatan di luar sekolah
h. Mengatur mutasi siswa
i. Menyusun dan membuat kepanitiaan penerimaan siswa baru dan pelaksanaan MOS
j. Menyusun dan membuat jadwal kegiatan akhir tahun sekolah
k. Menyelenggarakan cerdas cermat dan olah raga prestasi

l. Membuat laporan kegiatan kesiswaan secara berkala

6. Bagian sarana dan prasarana

Membantu dan bertanggungjawab kepada kepala sekolah dalam :

a. Menyusun program pengadaan sarana dan prasarana
b. Mengkoordinasikan penggunaan sarana dan prasarana
c. Pengelolaan pembiayaan alat-alat pengajaran
d. Mengelola perawatan dan perbaikan sarana prasarana
e. Bertanggung jawab terhadap kelengkapan data sekolah secara menyeluruh
f. Melaksanakan pembukuan sarana dan prasarana secara rutin
g. Menyusun laporan secara berkala

7. Bagian hubungan kerja dan industry

Membantu dan bertanggung jawab kepada kepala sekolah dalam :

a. Mengatur dan menyelenggarakan hubungan sekolah dengan dewan komite
b. Membina hubungan antara sekolah dengan wali murid
c. Membina pengembangan antar sekolah dengan lembaga pemerintah, dunia usaha dan lembaga social lainnya
d. Membuat dan menyusun program semua kebutuhan sekolah

8. Koordinator tata usaha

Bertanggung jawab kepada kepala sekolah dalam kegiatan :

a. Penyusunan program kerja tata usaha sekolah
b. Pengelolaan dan pengarsipan surat-surat masuk dan keluar
c. Pengurusan administrasi sekolah
d. Pembinaan dan pengembangan karir pengawai tata usaha sekolah
e. Penyusunan administrasi sekolah meliputi kesiswaan dan ketenagaan
f. Penyusunan dan penyajian data/statistic sekolah secara keseluruhan
g. Mengkoordinasikan dan melaksanakan 9K
h. Penyusunan laporan pelaksanaan secara berkala

9. Pustakawan sekolah

Membantu kepala sekolah dalam kegiatan

a. Perencanaan pengadaan buku/bahan pustaka/media elektronik
b. Pelayanan perpustakaan
c. Perencanaan pengembangan perpustakaan
d. Pemeliharaan dan perbaikan buku-buku/bahan pustaka/media elektronika
e. Inventarisasi dan pengadministrasian
f. Penyimpanan buku/bahan pustaka dan media elektronika
g. Menyusun tata tertib perpustakaan
h. Menyusun laporan pelaksanaan kegiatan perpustakaan secara berkala

10. Teknisi bengkel

Membantu kepala sekolah dalam kegiatan :

a. Perencanaan pengadaan alat dan bahan laboratorium/bengkel
b. Menyusun jadwal dan tata tertib penggunaan bengkel
c. Mengatur penyimpanan, pemeliharaan, dan perbaikan alat alat
d. Membuat dan menyusun daftar alat alat laboratorium/bengkel
e. Menyusun laporan pelaksanaan kegiatan laboratorium secara berkala

11. Guru
    a. Guru bertugas mengelola proses pembelajaran
    b. Bertanggungjawab kepada sekolah dan mempunyai tugas melaksanakan tugas belajar dan mengajar secara efektif dan efisien
    c. Menyusun program pembelajaran pendidikan siswa untuk mencapai target kurikulum yang sudah ditentukan

2.2 E-Learning

Perkembangan teknologi informasi yang semakin maju telah mampu membantu manusia dalam segala bidang. Salah satu bidang yang tak luput pemanfaatan teknologi informasi dalam prosesnya adalah dalam bidang pendidikan. Perkembangan yang semakin maju membuat pergeseran paradigma cara pembelajaran dan melahirkan metode-metode baru dalam proses pembelajaran tersebut seperti *e-learning* [4]. *E-learning* merupakan suatu jenis proses belajar

mengajar yang memungkinkan tersampaikannya bahan ajar ke siswa dengan menggunakan media internet, intranet atau media jaringan komputer lain [5].

Pada dasarnya konsep *e-learning* dapat dikelompokan berdasarkan basis teknologi yang digunakannya menjadi dua jenis, yaitu *computer based training* (CBT) yang menggunakan program komputer sebagai media utama yang digunakan oleh pelajar. Dan jenis satunya adalah *web based training* (WBT) yang merupakan perkembangan lanjutan dari sistem CBT dengan menggunakan tambahan teknologi berbasis internet. Pada jenis WBT penyampaian dan akses materi pelajaran dilakukan melalui internet.

2.2.1 Learning Management System

Dalam *e-learning* ada dua bagian utama yang menopang sistemnya, yaitu *Learning Management Sistem* dan e-*learning content* atau materi pelajaran yang akan dipelajari oleh pemakai. *Learning Management Sistem* (LMS) adalah *software* yang mengotomisasi administrasi pelatihan dan juga menyediakan manajemen laporan, mengelola kegiatan pembelajaran serta hasil hasilnya. Secara umum bisa digunakan sebagai salah satu sarana pembelajaran di perusahan atau instansi pendidikan. Contoh dari LMS yang berbasis *Open Source* adalah *moodle*, *claroline*, *manhattan atutor* dll.

2.2.2 Penilaian Jawaban Otomatis

Setiap pembelajaran membutuhkan sebuah evaluasi untuk mengetahui sejauh mana daya serap informasi yang dilakukan oleh peserta didik. Secara umum terdapat dua macam bentuk evaluasi yang dapat dilakukan dalam proses belajar mengajar yaitu secara lisan dan tertulis. Ujian secara lisan memiliki kelemahan yaitu penguji dan pelajar yang diuji harus berada pada satu ruang dan satu waktu yang sama dan membutuhkan waktu yang cukup banyak terutama jika peserta ujian tersebut berjumlah banyak. Sedangkan ujian tulisan jauh lebih efisien karna ujian dapat dilakukan dalam waktu bersamaan dengan jumlah peserta ujian yang lebih banyak.

Jenis ujian tulis dapat dibagi menjadi tiga macam, yaitu pilihan ganda, isian singkat, dan esai. Dari ketiga jenis ujian yang dapat dilakukan secara tertulis tersebut ujian yang paling mudah untuk diperiksa secara otomatis adalah pilihan ganda. Pada ujian jenis pilihan ganda pelajar cukup memilih jawaban yang dianggapnya paling benar dari beberapa pilihan jawaban yang ada. Namun kelemahan dari model ujian pilihan ganda adalah kurang dapatnya penguji mengetahui sejauh mana pengetahuan pelajar yang diuji dan dimungkinkan adanya peserta jawaban yang mendapatkan nilai tinggi dari faktor keberuntungan dengan menebak jawaban yang tersedia. Tingkat akurasi yang dapat dilakukan oleh komputer untuk ujian jenis pilihan ganda dapat mencapai 100% karna bentuk uijan seperti ini sangat mudah diolah komputer cukup dengan membandingkan jawaban pelajar dengan kunci jawaban yang ada dalam *database*.

Jenis ujian tertulis berikutnya adalah isian singkat. Pada jenis ujian isian singkat pelajar cukup mengisi jawaban atas pertanyaan yang diajukan dalam satu atau beberapa kata. Ujian jenis ini sedikit lebih baik dibandingkan pilihan ganda karna menuntut pelajar untuk mempunyai sebuah jawaban sendiri. Dan jenis ujian tertulis lainnya adalah dengan soal esai. Esai adalah suatu tulisan yang menggambarkan opini penulis tentang subyek tertentu yang coba dinilainya atau juga dapat diartikan sebagai karangan prosa yang mengupas secara sepintas namun akurat, padat dan berisi mengenai masalah kesusastraan, seni dan budaya dari sudut pandang penulis secara subjektif. Menurut KBBI esai adalah karangan prosa yg membahas suatu masalah secara sepintas lalu dr sudut pandang pribadi penulisnya. Bentuk soal esai merupakan bentuk pengujian yang secara akurat mampu menggambarkan pemahaman pelajar secara mendalam mengenai topik yang diujikan. Selain itu ujian esai juga mampu merangsang kemampuan mengemukakan pendapat dalam bentuk tulisan.

Ujian dalam bentuk esai memiliki kekurangan dari segi sumber daya yang dibutuhkan yaitu manusia dan waktu. Ketika sebuah ujian esai diikuti oleh banyak peserta waktu yang dibutuhkan untuk memeriksa jawaban akan bertambah pula.

Untuk mengatasi masalah dalam penilaian ujian esai, sistem penilaian esai secara otomatis menggunakan komputer dapat menjadi sebuah solusi yang cukup baik.

2.2.3 Text Preprocessing

Text preprocessing adalah langkah awal sebelum dilakukan sebuah pencocokan *string*. Proses ini melalui beberapa tahap tindakan yang dilakukan pada sebuah *string* jawaban dan kunci yang diisi oleh guru dan siswa. Tahap-tahap yang dilakukan dalam proses ini adalah:

1. *Case folding / toLowerCase*

   Case folding adalah tahapan dimana *string* yang telah diinput oleh user yang awalnya merupakan campuran dari huruf kapital dan huruf kecil disama ratakan dengan dijadikan menjadi huruf kecil semua. Hal ini dilakukan karena sistem komputer membedakan antara "A" dan "a" sehingga antara "saya" dan "Saya" pun akan dinilai berbeda pada sistem. Tentunya hal ini akan berpengaruh pada pencocokan *string* antar kunci dan jawaban dari siswa.

2. Penghilangan tanda baca

   Tahapan ini adalah penghapusan tanda baca sehingga string yang tersisa hanya terdiri dari huruf dan angka saja. Tahapan ini dilakukan karena sistem yang akan dibangun hanya mencocokan *string* dan angka belum melibarkan operasi matematika.

3. Penghapusan kata umum (*stopWord)*

   Tahapan *stopWord* adalah penghapusan kata kata yang dinilai umum sehingga kata kata tersebut tidak dimasukan kedalam *array string* yang akan dicocokan.

4. *Tokenizing*

   Tokenizing adalah tahapan pemotongan *string* dalam teks berdasarkan kata yang menyusunya dan disimpan dalam sebuah *variable array*.

5. *Stemming* (penghapusan awalan dan akhiran)

Stemming adalah salah satu cara yang digunakan untuk meningkatkan performa pencarian dan pencocokan kata dengan cara mentransformasikan kata kata dalam sebuah dokumen teks ke bentuk dasarnya. Proses *stemming* dilakukan dengan menghilangkan semua imbuhan baik yang terdiri dari awalan, sisipan maupun akhiran dan kombinasi dari awalan dan akhiran. Ada banyak algoritma yang digunakan untuk melakukan proses *stemming*. Khusus untuk stemming bahasa indonesia algoritma yang sering digunakan adalah algoritma nazrief dan andriani, arifin dan setiono, algoritma veda, dll.

2.2.4 Latent Semantyc Analysis

Latent Semantic Analysis (LSA) adalah suatu metode perhitungan matematika teknik yang sepenuhnya otomatis dalam penggalian dan penyimpulan hubungan antara kata dan kalimat yang ada dalam paragraf [6]. Algoritma LSA merupakan salah satu algoritma pengembangan dalam bidang ilmu *Information Retrival* yang mampu menghimpun sejumlah besar dokumen dalam basis data dan mengghubungkan relasi antar dokumen dengan mencocokan *query* yang diberikan. Fungsi utama dari LSA ini adalah untuk menghitung kemiripan (*similarity*) dokumen dengan membandingkan representasi vektor tiap dokumen. Penilaian dengan metode LSA lebih kepada kata-kata yang ada dalam tulisan tanpa memperhatikan urutan kata dan tata bahasa dalam tulisan tersebut, sehingga suatu kalimat yang dinilai adalah berdasarkan kata-kata kunci yang ada pada kalimat tersebut.

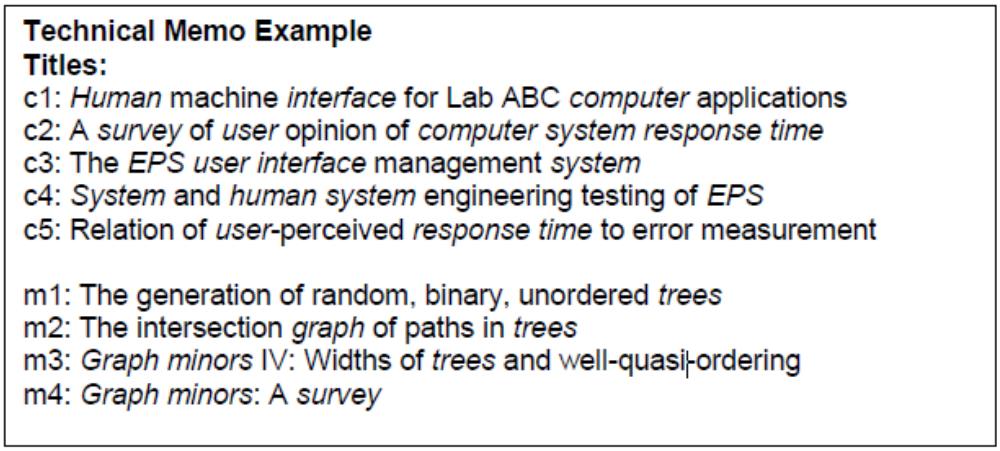
Untuk menghitung nilai jawaban esai menggunakan metode LSA, caranya adalah dengan cara membuat sebuah dokumen esai yang menjadi acuan (*query*) atau kunci jawaban dan kemudian esai yang akan dinilai dibandingkan dengan esai acuan jawaban, semakin banyak kemiripannya maka semakin besar nilai jawabannya.

Langkah-langkah pemeriksaan esai dengan metode LSA adalah pertama tulisan direpresentasikan ke dalam matriks dimana baris matriks menunjukkan kata-kata kunci (*queries*) pada dokumen tersebut dan setiap kolom menunjukkan kolom dokumen. Setiap sel menunjukkan banyaknya kemunculan kata kunci yang berada pada matriks pada dokumen yang ada di kolom matriks. Kemudian isi dari sel tersebut terlebih dahulu ditransformasikan dimana setiap frekuensi kata dibobotkan dengan sebuah fungsi yang menunjukkan pentingnya sebuah kalimat dalam *paragraph* dan juga derajat yang menunjukkan seberapa pentingnya tipe kata didalam suatu kalimat.

Langkah selanjutnya adalah dilakukan dekomposisi *Singular Value Decomposition* (SVD) pada matriks kata-dokumen. Pada SVD matriks memuat frekuensi pemunculan kata kunci di dekomposisi menjadi tiga buah matriks yang jika tiga buah matriks tersebut dikalikan maka akan muncul kembali matriks asalnya. Matriks pertamanya mendeskripsikan entitas kolom sebagai nilai vektor *orthogonal* matriks. Matriks keduanya berupa matriks diagonal yang memuat nilai skalar matriks. Secara matematis, faktor yang paling baik adalah menggunakan dimensi terkecil dari matriks awalnya, sehingga rekonstruksi matriks terbaik dihasilkan pada saat nilai faktor lebih kecil dari jumlah faktor yang digunakan.

Dimensi dari matriks yang telah disederhanakan dengan menghapus koefisien pada matriks diagonal sebanyak yang diinginkan sampai tersisa koefisien sebanyak dimensi yang terpilih. Tujuan penyederhanaan ini adalah agar terbentuk matriks yang memuat nilai korelasi yang diinginkan ketika tiga buah matriks direkonstruksi. Kemudian penilaian akan dilakukan dengan membandingkan matriks korelasi dari *query* dengan matriks korelasi dari setiap dokumen dengan menggunakan perhitungan *cosinus* α.

Untuk lebih memahami proses ini lebih jelas, berikut adalah contoh yang biasa digunakan. Pada gambar berikut dapat dilihat beberapa kalimat yang terdiri dari dua tema.



Gambar 2. 2 Contoh Kalimat dan Kata Kunci

Dari contoh pada gambar diatas ada sembilan kalimat dengan lima kalimat pertama bertema *human computer interaction* (kalimat c1 sampai c5), dan empat kalimat selanjutnya bertema *mathematical graph theory* (kalimat m1 sampai m4). Pentransformasian kata kata diatas menjadi sebuah matriks A ditunjukan pada tabel berikut dengan mengambil kata kata yang bertemakan komputer dan matematik (yang dituliskan dalam huruf miring).

Tabel 2. 1 Transformasi Matriks Kemunculan Kata

	C1	C2	C3	C4	C5	M1	M2	M3	M4
Human	1	0	0	1	0	0	0	0	0
Interface	1	0	1	0	0	0	0	0	0
Computer	1	1	0	0	0	0	0	0	0
User	0	1	1	0	1	0	0	0	0
System	0	1	1	1	0	0	0	0	0
Response	0	1	0	0	1	0	0	0	0
Time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
Survey	0	1	0	0	0	0	0	0	0
Trees	0	0	0	0	0	1	1	1	0

Graph	0	0	0	0	0	0	1	1	1
Minors	0	0	0	0	0	0	0	1	1

Dekomposisi matriks pada tabel diatas SVD merepresentasikan *semantic space* ke dalam bentuk matriks yang memiliki orde lebih kecil dibandingkan orde matriks aslinya, namun perhitungan matriks tetap menghasilkan matriks yang bernilai hampir sama. SVD merupakan teorema aljabar linear yang dikatakan mampu memecah blok suatu matriks A menjadi tiga matriks baru, yaitu sebuah matriks *orthogonal* U, matriks diagonal S, dan *transpose* matriks *orthogonal* V. teorema SVD adalah sebagai berikut :

$$Amn = Umm \times Smn \times V^{T}nn \quad \text{…. } \textbf{Persamaan (2.1)}$$

Dimana $U^{T}U = I$, $V^{T}V = I$. Kolom matriks U merupakan *Eigenvektor Orthonormal* dari AA^{T}, sedangkan kolom matriks V merupakan *Eigenvektor Orthonormal* dari $A^{T}A$, dan S merupakan matriks diagonal akar dari nilai *eigen* dari matriks U atau V dalam urutan dari yang terbesar (*Descending order*). Matriks U dari matriks kemunculan kata dapat dilihat pada tabel 2.2. Perhitungan manual untuk mendapatkan matriks U dapat dilihat pada contoh soal di persamaan (2.2) dengan perhitungan matriks yang lebih sederhana.

Tabel 2. 2 Matriks U Sebagai Komponen Matriks A

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.2	-0.07	0.14	-0.55	0.28	0.5	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.3	0.06	0.49
0.4	0.06	-0.34	0.1	0.33	0.38	0	0	0.01
0.64	-0,17	0,36	0,33	-0,16	-0,21	-0,17	0,03	0,27
0,27	0,11	-0,43	0,07	0,08	-0,17	0,28	-0,02	-0,05
0,27	0,11	-0,43	0,07	0,08	-0,17	0,28	-0,02	-0,05
0,3	-0,14	0,33	0,19	0,11	0,27	0,03	-0,02	-0,17
0,21	0,27	-0,18	-0,03	-0,54	0,08	-0,47	-0,04	-0,58
0,01	0,49	0,23	0,03	0,59	-0,39	-0,29	0,25	-0,23

0,04	0,62	0,22	0	-0,07	0,11	0,16	-0,68	0,23
0,03	0,45	0,14	-0,01	-0,3	0,28	0,34	0,68	0,18

Untuk matriks yang berikutnya adalah matriks V hasil perkalian dari matriks AA^T yang kemudian dicari nilai eigennya serta dinormalisasi dengan metode *gram schmidt* sama seperti pada pencarian matriks U. Proses pencarian matriks ini pada dasarnya sama dengan cara mencari matriks U, yang membedakan adalah langkah awalnya yaitu mengalikan antara matriks A dengan A^T. Untuk lebih memudahkan dalam pemahaman perhitungan manual dapat dilihat pada persamaan (2.5). hasil dari perhitungan matriks pada tabel 2.1 dapat dilihat pada tabel 2.3.

Tabel 2. 3 Matriks V sebagai komponen vektor orthogonal matriks A

0,2	-0,06	0,11	-0,95	0,05	-0,08	0,18	-0,01	-0,06
0,61	0,17	-0,5	-0,03	-0,21	-0,26	-0,43	0,05	0,24
0,46	-0,13	0,21	0,04	0,38	0,72	-0,24	0,01	0,02
0,54	-0,23	0,57	0,29	-0,21	-0,37	0,26	-0,02	-0,08
0,28	0,11	-0,51	0,15	0,33	0,03	0,67	-0,06	-0,26
0	0,19	0,1	0,02	0,39	-0,3	-0,34	0,45	-0,62
0,01	0,44	0,19	0,01	0,35	-0,21	-0,15	-0,76	0,02
0,02	0,62	0,25	0,01	0,15	0	0,25	0,48	0,52
0,08	0,53	0,08	-0,03	-0,16	0,36	0,04	-0,07	-0,45

Sedangkan matriks yang terakhir dicari adalah matriks S yang merupakan hasil dari akar kuadrat nilai eigen bukan nol matriks U dan matriks V dengan merubahnya menjadi matriks diagonal dan diurutkan secara *descending* dari nilai terbesar terlebih dahulu. Hasil dari perhitungan matriks pada tabel 2.1 dapat dilihat pada tabel 2.4.

Tabel 2. 4 Matriks S sebagai komponen matriks A

3,34	0	0	0	0	0	0	0	0
0	2,54	0	0	0	0	0	0	0
0	0	2,35	0	0	0	0	0	0
0	0	0	1,64	0	0	0	0	0
0	0	0	0	1,5	0	0	0	0
0	0	0	0	0	1,31	0	0	0
0	0	0	0	0	0	0,85	0	0
0	0	0	0	0	0	0	0,56	0
0	0	0	0	0	0	0	0	0,36

2.2.5 Matriks Singular Value Decomposition

SVD merupakan teorema aljabar linear yang disebut mampu untuk memecah blok suatu matriks A menjadi tiga matriks baru yaitu sebuah matriks *orthogonal* U, matriks diagonal S, dan *transpose* matriks *orthogonal* V. Teorema SVD adalah sebagai berikut :

$$Amn = Umm \times Smn \times V^{T}nn$$

Dimana $U^{T}U$= matriks identitas (I), $V^{T}V$ = matriks identitas (I). Kolom matriks U merupakan eigenvektor *orthonormal* dari AA^{T}. Sedangkan kolom matriks V merupakan *eigenvektor orthonormal* dari $A^{T}A$. dan S merupakan matriks diagonal akar dari nilai eigen dari matriks U atau V dalam urutan dari yang terbesar.

Untuk lebih memudahkan dalam pemahaman matriks SVD berikut adalah contoh sebuah matriks ordo kecil A yang akan digunakan untuk menghitung nilai SVD. Berikut adalah contoh perhitungannya.

1. Diketahui sebuah matriks A dengan ordo 2 x 3

$$A = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

2. Menghitung nilai matriks U yang diawali dengan menghitung nilai AA^T.

$$A^T = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix}$$

$$AA^T = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$$

…. **Persamaan (2.2)**

Selanjutnya setelah didapatkan matriks AA^T dilanjutkan menghitung nilai *eigen* dan vektor *eigen* dari AA^T. rumus yang digunakan untuk menghitung vector *eigen* adalah $Av = \lambda v$ yang diimplementasikan dalam AA^T sebagai berikut :

$$\begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Kemudian pisahkan tiap set rumus tersebut menjadi sebuah persamaan sebagai berikut :

$$11x_1 + x_2 = \lambda x_1$$
$$x_1 + 11x_2 = \lambda x_2$$

……. **Persamaan (2.3)**

Dan untuk mempermudah dalam perhitungan maka dirubah dalam bentuk sebagai berikut :

$$(11 - \lambda)x_1 + x_2 = 0$$
$$x_1 + (11 - \lambda)x_2 = 0$$

Ditentukan nilai λ dengan menghitung nilai determinan dari rumus diatas terhadap nilai nol.

$$\begin{vmatrix} (11-\lambda) & 1 \\ 1 & (11-\lambda) \end{vmatrix} = 0$$

Dan hasilnya adalah sebagai berikut :

$$(11-\lambda)(11-\lambda) - 1 \cdot 1 = 0$$
$$(\lambda - 10)(\lambda - 12) = 0$$
$$\lambda = 10, \lambda = 12$$

Nilai *eigen* terlah diperoleh yaitu λ=10 dan λ=12. Selanjutnya mengganti nilai λ pada rumus perhitungan sebenlumnya untuk memperoleh vector *eigen*. Untuk λ=10 diperoleh hasil sebagai berikut :

$$(11-10)x_1 + x_2 = 0$$
$$x_1 = -x_2$$

Sehingga diperoleh vector *eigen* [1, -1]. Kemudian untuk λ=12 diperoleh hasil sebagai berikut :

$$(11-12)x_1 + x_2 = 0$$
$$x_1 = x_2$$

Dan diperoleh vector *eigen* [1,1]. Vector eigen tersebut menjadi kolom vector pada matriks diurut berdasarkan nilai *eigen*. Dengan kata lain vector *eigen* dari nilai *eigen* yang terbesar menjadi kolom pertama sedangkan kolom berikutnya ditempati oleh vector *eigen* yang memiliki nilai eigen yang lebih kecil.

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Kemudian matriks tersebut dikonversi kedalam bentuk matriks *orthogonal* dengan menggunakan normalisasi *gramschmidt* terhadap kolom vector. Perhitungannya adalah sebagai berikut:

$$\vec{u_1} = \frac{\vec{v_1}}{|\vec{v_1}|} = \frac{[1,1]}{\sqrt{1^2+1^2}} = \frac{[1,1]}{\sqrt{2}} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$$

$$\vec{w_2} = \vec{v_2} - \vec{u_1} \cdot \vec{v_2} * \vec{u_1} =$$
$$[1,-1] - [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] \cdot [1,-1] * [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] =$$

…… **persamaan (2.4)**

$$[1,-1] - 0 * [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] = [1,-1] - [0,0] = [1,-1]$$

$$\vec{u_2} = \frac{\vec{w_2}}{|\vec{w_2}|} = [\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}]$$

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

3. Menghitung nilai matriks V. cara untuk mendapaktan nilainya sama seperti perhitungan untuk memperoleh matriks U dari A^TA

$$A^TA = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix}$$

… Persamaan (2.5)

Rumus nilai *eigen* adalah sebagai berikut :

$$\begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$10x_1 + 2x_3 = \lambda x_1$$

$$10x_2 + 4x_3 = \lambda x_2$$
$$2x_1 + 4x_2 + 2x_3 = \lambda x_2$$

$$(10 - \lambda)x_1 + 2x_3 = 0$$
$$(10 - \lambda)x_2 + 4x_3 = 0$$
$$2x_1 + 4x_2 + (2 - \lambda)x_3 = 0$$

Perhitungan determinan terhadap nol adalah sebagai berikut:

$$\begin{vmatrix} (10-\lambda) & 0 & 2 \\ 0 & (10-\lambda) & 4 \\ 2 & 4 & (2-\lambda) \end{vmatrix} = 0$$

$$(10 - \lambda)[(10 - \lambda)(2 - \lambda) - 16] + 2[0 - (20 - 2\lambda)] =$$
$$\lambda(\lambda - 10)(\lambda - 12) = 0,$$

Jadi nilai *eigen* yang diperoleh adalah λ= 0, λ=10, dan λ=12. Kemudian ganti nilai pada rumus sebelumnya untuk menentukan vector *eigen*.

Untuk λ=12

$$(10 - 12)x_1 + 2x_3 = -2x_1 + 2x_3 = 0$$

$$x_1 = 1, x_3 = 1$$
$$(10 - 12)x_2 + 4x_3 = -2x_2 + 4x_3 = 0$$
$$x_2 = 2x_3$$
$$x_2 = 2$$

Diperoleh V1 = [1,2,1]

Untuk λ = 10

$$(10-10)x_1 + 2x_3 = 2x_3 = 0$$

$$x_3 = 0$$

$$2x_1 + 4x_2 = 0$$

$$x_1 = -2x_2$$

$$x_1 = 2, x_2 = -1$$

Diperoleh V2 = [2,-1,0]

Untuk λ=0 :

$$10x_1 + 2x_3 = 0$$

$$x_3 = -5$$

$$10x_1 - 20 = 0$$

$$x_2 = 2$$

$$2x_1 + 8 - 10 = 0$$

$$x_1 = 1$$

Diperoleh V3 = [1,2,-5]. Kemudian urutan V1, V2, V3 sebagai vector kolom pada sebuah matriks sesuai nilai eigen terbesar adalah sebagai berikut :

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & -1 & 2 \\ 1 & 0 & 5 \end{bmatrix}$$

Normalisasi matriks tersebut adalah sebagai berikut :

$$\vec{u_1} = \frac{\vec{v_1}}{|\vec{v_1}|} = [\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}]$$
$$\vec{w_2} = \vec{v_2} - \vec{u_1} \cdot \vec{v_2} * \vec{u_1} = [2, -1, 0]$$
$$\vec{u_2} = \frac{\vec{w_2}}{|\vec{w_2}|} = [\frac{2}{\sqrt{5}}, \frac{-1}{\sqrt{5}}, 0]$$
$$\vec{w_3} = \vec{v_3} - \vec{u_1} \cdot \vec{v_3} * \vec{u_1} - \vec{u_2} \cdot \vec{v_3} * \vec{u_2} = [\frac{-2}{3}, \frac{-4}{3}, \frac{10}{3}]$$
$$\vec{u_3} = \frac{\vec{w_3}}{|\vec{w_3}|} = [\frac{1}{\sqrt{30}}, \frac{2}{\sqrt{30}}, \frac{-5}{\sqrt{30}}]$$

Matriks V nya adalah :

$$V = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{30}} \\ \frac{2}{\sqrt{6}} & \frac{-1}{\sqrt{5}} & \frac{2}{\sqrt{30}} \\ \frac{1}{\sqrt{6}} & 0 & \frac{-5}{\sqrt{30}} \end{bmatrix}$$

Dan setelah dilakukan transform maka matriks V^T adalah :

$$V^T = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix}$$

4. Menghitung matriks S berordo *m x n* dari akar nilai *eigen* bukan nol dan posisikan tiap nilai tersebut membentuk matriks *diagonal*. Nilai *eigen* terbesar akan ditempatkan dibaris matriks S yang lebih kecil. Jadi akar nilai *eigen* terbesar akan ditempatkan dibaris pertama matriks S kemudian diikuti dengan akar nilai *eigen* terbesar berikutnya.

$$S = \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix}$$

Sesuai dengan rumus SVD akan diperoleh kembali nilai yang sama ataupun hampir sama dengan matriks A jika dilakukanperhitungan dengan rumus tersebut.

$$A_{mn} = U_{mm} S_{mn} V_{nn}^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{\sqrt{12}}{\sqrt{2}} & \frac{\sqrt{10}}{\sqrt{2}} & 0 \\ \frac{\sqrt{12}}{\sqrt{2}} & \frac{-\sqrt{10}}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix} = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

2.2.6 Cosine Similarity

Cosine similarity adalah salah satu metode perhitungan *similarity* yang paling popular untuk diterapkan pada dokumen teks. Kelebihan utama dari metode *cosine similarity* adalah tidak terpengaruh pada panjang pendeknya suatu dokumen. Karena yang hanya diperhitungkan nilai *term* dari masing masing dokumen. Berikut adalah rumus untuk *cosine similarity*

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad \textbf{..persamaan (2.6)}$$

2.3 Teori Pembangunan Perangkat Lunak

2.3.1 *Object Oriented Programming* (OOP)

Objek adalah kesatuan entitas yang memiliki sifat dan tingkah laku. Dalam kehidupan sehari-hari, objek adalah benda, baik benda berwujud nyata seperti manusia, hewan, mobil, komputer, *handphone*, pena, ataupun benda yang tidak nyata atau konsep, seperti halnya tabungan bank, sistem antrian, sistem internet banking, dan sebagainya. Jadi pengertian OOP adalah konsep yang membagi program menjadi objek-objek yang saling berinteraksi satu sama lain. Objek adalah benda, baik benda yang berwujud nyata maupun benda yang tidak nyata (konsep) [7]. Jika kita menggunakan OOP maka akan ada enam keuntungan yang dapat diperoleh, yaitu :

1. Alami (*Natural*)
2. Dapat diandalkan (*reliable*)
3. Dapat digunakan kembali (*Reusable*)
4. Mudah untuk perawatan (*Maintainable*)
5. Dapat diperluas (*extendable*)
6. Efisiensi waktu

2.3.2 Pemodelan dengan UML

UML (*Unified Modeling Language*) adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek [6]. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikaskan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh *Booch, Object Modeling Technique (OMT)* dan *Object Oriented Software Engineering (OOSE)*. Metode *Booch* dari *Grady Booch* sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan iterative, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantic dari hubungan obyek dan kelas tersebut, perincian *interface* dan implementasi. Keunggulan metode *Booch* adalah pada detail dan kayanya dengan notasi dan elemen. Pemodelan *Object Modeling Technique* yang dikembangkan Rumbaugh didasarkan pada analisis terstruktur dan permodelan *entity-relationship*. Tahapan utama dalam metodologi ini adalah analisis, design sistem, design obyek dan implementasi. Keungulan metode ini adalah dalam penotasian yang mendukung semua konsep *Object Oriented Software Engineering*. Metode *Object Oriented Software Engineering Jacobson* lebih memberikan penekanan pada *use case*. *Object Oriented Software Engineering* memiliki tiga tahapan yaitu membuat model *requirement* dan analisis, *design* dan implementasi, dan model pengujian. Keungulan metode ini adalah mudah dipelajari karena

memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

2.3.3 Diagram-Diagram UML

Didalam UML terdapat beberapa macam diagram yang dapat menggambarkan suatu sistem. Berikut adalah beberapa diagram yang terdapat dalam pemodelan UML.

1. *Use Case* dan *Use Case Diagram*

   Use case merupakan pemodelan untuk kelakuan (*behaviour*) sistem yang akan dibuat [7]. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan, *use case* adalah serangkaian *scenario* yang digabungkan bersama-sama oleh tujuan umum pengguna. *Use case Diagram* digunakan untuk menunjukan fungsionalitas dari suatu entitas seperti *system*, *sub-sistem* atau *class* dengan menggunakan *actor*, *use case* dan hubungannya.

2. *Activity Diagram*

   Activity Diagram adalah teknik untuk mediskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku parallel sedangkan *flowchart* tidak bisa. *Activity diagram* tidak menunjukan apa yang terjadi, tetap tidak menunjukan siapa yang melakukan apa. *Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir, *Activity diagram*

juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

3. *Class Diagram*

Diagram *class* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat dalam pembangunan sistem [5]. *Class* dalam notasi UML digambarkan dengan kotak. *Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. *Class* menggambarkan keadaan (*atribut* atau *properti*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). *Class* mempunyai beberapa bagian yang menjelaskan isi dari class itu sendiri seperti :

a. *Attribute* adalah *property* dari sebuah *class*. *Attribute* ini melukiskan batas nilai yang mungkin ada pada obyek sebuah *class*. Sebuah *class* mungkin mempunyai nol atau lebih dari satu *attribute*.
b. *Operation* adalah sesuatu yang bisa dilakukan oleh sebuah *class* atau *class* yang lain dapat lakukan untuk sebuah *class*
c. *Responsibility* adalah keterangan tentang apa yang akan dilakukan *class* yaitu apa yang aakan dicapai oleh *attribute* dan *operation*.

4. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukan sejumlah contoh obyek dan *message* pesan yang diletakan diantara obyek-obyek ini dalam *use case*. *Sequence diagram* menggambarkan interaksi antar obyek di dalam dan di sekitar sistem berupa *message* yang digambarkan terhadap waktu . *Sequence diagram* terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (obyek-obyek yang terkait). *Sequence* diagram biasa digunakan untuk menggambarkan *scenario* atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang

dihasilkan. Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segi empat. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukan dengan *progress vertical*. Penjelasan dari komponen utama *sequence diagram* yaitu :

a. Obyek atau participant

Obyek atau *participant* diletakan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Mereka diatur dalam urutan guna menyederhanakan diagram. Setiap *participant* terhubung dengan garis titik-titik yang disebut *lifeline*. Sepanjang *lifeline* ada kotak yang disebut *activation*. *Activation* mewakili sebuah eksekusi *operation* dari *participiant*. Panjang kotak ini berbanding lurus dengan durasi *activation*.

b. Message

Sebuah *message* bergerak dari satu *participant* ke *participant* yang lain dan dari satu *lifeline* ke *lifeline* yang lain. Sebuah *participant* bisa mengirim sebuah *message* kepada dirinya sendiri.

c. Time

Time adalah diagram yang mewakili waktu pada arah *vertical*. Waktu dimulai dari atas kebawah. *Message* yang lebih dekat dari atas akan dijalankan terlebih dahulu dibandingkan *message* yang lebih dekat ke bawah.

2.4 Program Pendukung

2.4.1 Internet

Internet merupakan tempat terhubungnya berbagai mesin komputer yang mengolah informasi secara global, baik berupa *server*, komputer pribadi, *handphone*, komputer genggam, PDA, dan lain sebagainya. Masing-masing mesin ini bekerja sesuai dengan fugsinya, baik sebagai *server* yaitu penyedia layanan maupun sebagai *client* yaitu pengguna layanan. Berbagai komputer dari seluruh dunia terhubung melalui jaringan yang disebut *internet*.

Berbagai mesin komputer terhubung baik melalui kabel, saluran telepon, saluran *handphone*, satelit, *fiber* optik, gelombang, listrik, cahaya, serta media apa saja yang mungkin dialiri oleh data. Siapapun yang memiliki komputer, dapat menjadi bagian dari jaringan ini, sehingga dalam dunia maya inilah bisa dikatakan tidak ada istilah jauh atau dekat, tidak ada batasan negara dan lain-lain. Jika kita ingin mengetahui informasi dari suatu negara dimana saja, membaca berita lewat internet justru lebih cepat dibanding koran yang sama sekali belum dicetak dan diterbitkan.

Kelebihan saran internet yang tidak mengenal batas geografis juga menjadikan internet sebagai sarana yang ideal untuk melakukan kegiatan belajar jarak jauh, baik melalui kursus tertulis, sekolah maupun perkuliahan. Bermacam referensi, materi pembelajaran, jurnal, maupun hasil penelitian yang dipublikasikan melalui internet tersedia dalam jumlah yang berlimpah. Cukup memanfaatkan *search engine*, materi-materi yang relevan dapat segera ditemukan

2.4.2 Website

Website atau juga disebut situs adalah sebuah tempat dalam *World Wide Web* dimana *homepage* (halaman) sebuah organisasi atau individual berada. Sebuah halaman *web* adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website-website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

Situs juga dapat diartikan sebagai kumpulan halaman yang memiliki atau menampilkan gambar diam, gambar bergerak, data, teks, animasi, suara, video dan atau gabungan keseluruhan elemen yang bersifat statis atau dinamis dalam sebuah kerangka dimana halaman-halaman tersebut saling berhubungan satu sama lain

2.4.3 Moodle

Moodle adalah paket perangkat lunak untuk pembuatan kursus berbasis internet yang tersedia secara gratis sebagai perangkat lunak *Open Source* dibawah GNU *Public License*. Moodle dapat di instal di setiap komputer yang mampu menjalankan PHP dan mendukung basis data tipe SQL. Moodle dapat dijalankan di sistem operasi *windows*, mac, maupun linux.

Istilah moodle merupakan akronim dari *Modular Object Oriented Dynamic Learning Environment*. Kata moodle juga merupakan kata kerja yang mendeskripsikan proses pengerjaan sesuatu yang mendatangkan wawasan dan kreativitas.

2.4.4 HTML

Hypertext Markup Language (HTML) adalah bahasa dari *world wide web* (www) yang dipergunakan untuk menyusun dan membentuk dokumen agar dapat ditampilkan pada program browser. HTML merupakan protokol yang digunakan untuk mentransfer data atau document dari *web server* ke browser. HTML lebih menekankan pada penggambaran komponen-komponen struktur didalam halaman web dari pada menentukan penampilannya. Sedangkan web browser digunakan untuk menjalankan perintah-perintah HTML. Prinsipnya adalah bahawa teks markup berkonsentrasi pada struktur dari pada penampilan, membuat file lebih nyata dan meninggalkan rincian visual untuk perangkat lunak yang digunakan untuk menampilkan sebuah halaman web.

2.4.5 CSS

Cascading syle sheet, disingkat sebagai CSS adalah suatu teknologi yang digunakan untuk emperindah tampilan halaman *website* (situs) sehingga tampilannya akan menjadi lebih menarik dan mudah dikelola. CSS muncul sebagai jawaban atas sulitnya mengatur *layout* tampilan dokumen web yang dibuat dengan HTML murni meskipun telah menggunakan berbagai kombinasi format. Dengan

kata lain CSS adalah Bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen HTML sehingga memungkinkan untuk menampilkan halaman web yang sama dengan format berbeda. CSS mulai direkomendasikan oleh *World Wide Web Consortium* (W3C) sejak tahun 1996.

Di dalam CSS terdapat banyak *style*. *Style* adalah suatu definisi untuk *font* (huruf), *color* (warna), dan lain-lain. Setiap *style* memiliki nama unik yang disebut *selector*. *Selector* dan *style*-nya didefinisikan dalam satu tempat. CSS dapat ditulis pada bagian *body, head* suatu dokumen HTML atau diletakkan di sebuah file terpisah. Perintah-perintah CSS dibatasi oleh tag <style> dan </style>. Dalam *content* dokumen HTML anda dapat menunjuk *selector* yang telah didefinisikan kapan saja anda ingin mengaktifkan *style* tertentu.

Secara spesifik, berikut merupakan hal-hal yang dapat dilakukan oleh CSS :

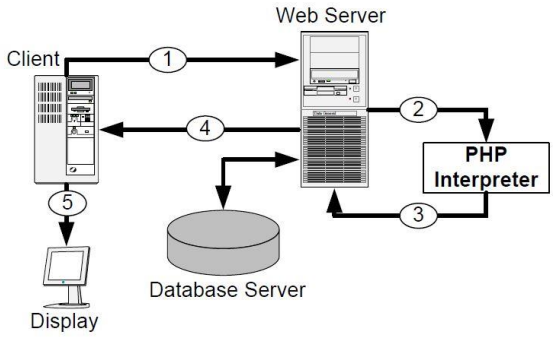
1. Mendefinisikan tampilan halaman *web* yang dibuat dalam satu tempat khusus. Ada baiknya *code* css yang dibuat tidak dimasukan didalam satu *file* halaman HTMLnya, karena dengan membuat *file* css terpisah kita tidak perlu repot-repot membuat *code* css disemua halaman yang dibuat.
2. Mengubah tampilan halaman *web* bahkan setelah halaman selesai dibuat. Ini fungsi dari memisahkan *file* css, sehingga hanya perlu pemanggilan *code-style* untuk membuat tampilan halaman *web* menjadi menarik bahkan sebelum isi dari halaman *web* itu dibuat.
3. Mendefinisikan ukuran huruf dan atribut-atribut.
4. Mendefinisikan *style* sesuai kebutuhan.
5. Mendefinisikan *layer* yang dapat diletakkan di atas elemen lain (misal membuat menu *pop-up*).

2.4.6 PHP

PHP adalah bahasa *server-side scripting* yang bersatu dengan HTML untuk membuat halaman *web* yang dinamis. Maksud dari *server-side scripting* adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan di server

tetapi disertakan pada dokumen HTML. Pembuatan *web* ini merupakan kombinasi antara PHP sendiri sebagai bahasa pemrograman dan HTML sebagai pembangun halaman *web*.

PHP dibuat pertama kali oleh Rasmus Lerdof yang awalnya dikhususkan untuk menerima input melalui form yang ditampilkan dalam browser web. Pada awalnya PHP adalah singkatan dari *Personal Home Page*, namun kini PHP adalah kependekan dari *HyperText Preprocessor* merupakan Bahasa utama *script server-side* yang disisipkan pada HTML dan dijalankan di server. Cara kerja PHP adalah sebagai berikut :



Gambar 2. 3 Cara Kerja PHP

Proses-proses yang terjadi adalah

1. *Client* me-*request* halaman *website* yang berisi *script* PHP
2. Jika *file* yang di *request* ditemukan maka *server* akan meneruskannya ke PHP *interpreter* (penerjemah PHP) yang akan bekerja menghasilkan dokumen/halaman HTML berdasarkan *script* PHP. Jika dalam *script* tersebut terdapat permintaan terhadap *database* maka akan terjadi proses *query* data ke database server
3. Dokumen HTML hasil interprestasi oleh PHP interpreter dikembalikan ke web *server*
4. Web *server* mengirim dokumen HTML ke client
5. Web *browser* akan menterjemahkan dokumen HTML ke *display*

2.4.7 MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

MySQL merupakan sebuah program database *server* yang mampu menerima dan mengirimkan datanya dengan cepat, multi *user* serta menggunakan perintah standar SQL. SQL adalah bahasa standar yang digunakan mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industri. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan menggunakan *dBASE* atau *Clipper* yang masih menggunakan perintah-perintah pemrograman.

2.4.8 JAMA PHP

JAMA adalah singkatan dari *Java Matrix*, sebuah aplikasi matematis *web based* yang dapat digunakan untuk melakukan perhitungan matriks. JAMA merupakan paket linear aljabar dasar untuk JAVA yang menyediakan *class user-level* untuk membangun, dan memanipulasi matriks yang cukup padat. Hal ini dimaksudkan untuk menyediakan fungsionalitas dalam pemecahan matriks yang cukup rumit dan dikemas dengan cara yang dapat dimengerti oleh *non*-ahli. Saat ini JAMA sudah dapat digunakan dalam *script* PHP.

Paket JAMA memiliki beberapa *source listing* yang dapat digunakan untuk memecahkan beberapa permasalahan matriks aljabar dasar seperti :

1. *Cholesky decomposition*
2. *Eigenvalue decomposition*
3. *LU decomposition*
4. Matriks
5. *QR decomposition*
6. *Singular Value Decomposition*.