BAB 2
TINJAUAN PUSTAKA

Pada bab ini dijelaskan beberapa konsep dan dasar teori yang berkaitan dengan permasalahan yang akan dibahas sebagai dasar pemahaman dalam mengimplementasikan konsep-konsep tersebut kedalam semua kegiatan pengembangan sistem yaitu profil sekolah, *learning management system*, algoritma koreksi essay, konsep perancangan sistem, dan pengembangan perangkat lunak.

2.1. Profil SMP Negeri 1 Cimahi

2.1.1. Sejarah SMP Negeri 1 Cimahi

SMP Negeri 1 Cimahi berawal dari dari sebuah sekolah untuk golongan pribumi yang sudah berdiri sejak tahun 1926, dan dibangun oleh Pemerintah Hindia Belanda dengan nama HIS. Kemudian pada saat penjajahan Jepang tahun 1942–1945 HIS diubah menjadi Sekolah Rakyat (SR) No 3 Cimahi. Setelah Indonesia merdeka diubah lagi menjadi Sekolah Rakyat (SR) no 4 Cimahi. Pada Tahun 1949 sekolah ini menjadi sebuah sekolah swasta dengan nama SMP Parki di bawah pimpinan Bapak Samir.

Setelah melewati beberapa perubahan dan perkembangan, akhirnya pada tanggal 15 Agustus tahun 1950 dengan Surat Keputusan Menteri Pendidikan dan Kebudayaan No 6535/R berdirilah SMP Negeri 1 Cimahi dengan dipimpin pertama kali oleh Bapa Rd. Samad Bratawijaya. Pada saat itu sekolah ini memiliki 9 ruang, yang digunakan untuk kelas sebanyak 8 ruang dan satu untuk ruang Guru, TU dan Kepala Sekolah. Dan tahun itu pulalah dijadikan awal kelahiran SMP Negeri 1 Cimahi.

Setelah berakhir masa kepemimpinan Bapak Rd. Samad Bratawijaya selanjutnya sekolah ini berturut-turut dipimpin oleh kepala-kepala sekolah diantaranya :

1. Tahun 1955–1962 oleh Bapak Mochamad Kasim.
2. Tahun 1962–1966 oleh Bapak R. Embang Artawijaja (sekarang namanya diabadikan untuk nama jalan yang dulunya bernama Jalan

SMP).

3. Tahun 1966–1968 oleh Bapak M. Djudju Sadjid.
4. Tahun 1968–1974 oleh Bapak Drs. R. P. Hartoyo.
5. Tahun 1974–1986 oleh Bapak T. Darsah.
6. Tahun 1986–1988 oleh Bapak H. E Hasan Badjuri BA.
7. Tahun 1988–1992 oleh Bapak Saryama Lala Mugia BA.
8. Tahun 1992–1994 oleh Bapak Endang Dachlan.
9. Tahun 1994 oleh Bapak Ade Juhandi sebagai PYMT.
10. Tahun 1994–1996 oleh Bapak Drs. Ubes Sumantri.
11. Tahun 1996 oleh Bapak Drs. H. Odjo Suheman sebagai PYMT.
12. Tahun 1996–1999 oleh Bapak Drs. Sholeh Gunawan.
13. Tahun 1999–2002 oleh Bapak Drs. E. Saefudin Satyaputra.
14. Tahun 2002 oleh Bapak Drs. Hadi Supriatna, MM sebagai PLH.
15. Tahun 2002–sekarang oleh Ibu Dra. L.G. Arisuweni, M.Pd.

2.1.2. Visi dan Misi SMP Negeri 1 Cimahi

Dalam menjalankan institusinya, SMPN 1 Cimahi memiliki visi dan misi sebagai pedoman dalam mencapai tujuannya. Adapun visi dan misi yang dimiliki oleh SMPN 1 Cimahi adalah sebagai berikut:

1. Visi

   Terwujudnya sekolah yang memiliki siswa berkualitas, cerdas dan kompetitif dalam bidang IMTAQ dan IPTEK serta berwawasan lingkungan. Dengan motto IDEAL yaitu :

   1. Itikad yang baik.
   2. Dedikasi yang tinggi.
   3. Evaluasi diri.
   4. Akhlak yang mulia.
   5. Luwes dalam ucapan dan tindakan.

2. Misi

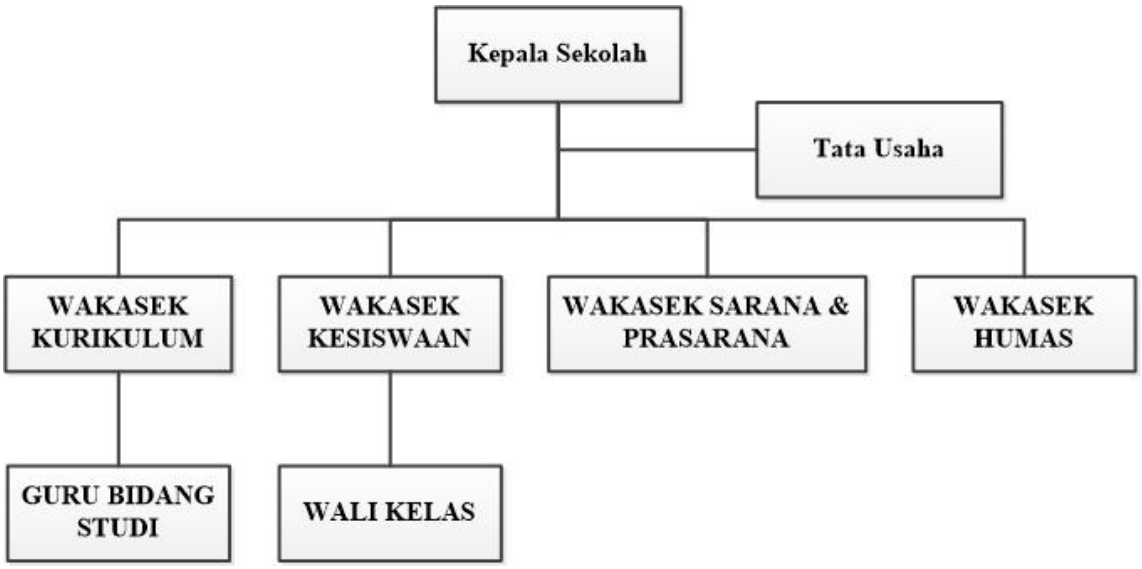
   1. Mewujudkan tercapainya mutu pendidikan nasional.

2. Membentuk pribadi yang luhur dan berakhlak mulia.
3. Meningkatkan etos kerja seluruh warga sekolah.
4. Mengupayakan kemandirian yang bertanggung jawab dalam proses pembelajaran dan pengelolaan pendidikan.
5. Mengembangkan minat dan bakat apresiasi seni-budaya.
6. Menumbuhkembangkan kearifan lokal dan kesolehan sosial

2.1.3. Struktur Organisasi Sekolah

Struktur organisasi menunjukan pola hubungan jalur komunikasi antara fungsi-fungsi, bagian-bagian atau orang-orang yang ada dalam suatu organisasi dapat diketahui bagaimana wewenang dan tanggung jawab mengalir diantara bagian tersebut.

Dibawah ini adalah Bagan Struktur Organisasi SMP Negeri 1 Cimahi.



Gambar 2.1. Struktur Organisasi SMPN 1 Cimahi

2.1.4. Deskripsi Tugas Struktur Organisasi SMPN 1 Cimahi

Melihat dari Gambar 2.1, setiap jabatan memiliki tugas masing-masing, berikut penjelasan mengenai tugas dari masing-masing jabatan yang ada di SMP Negeri 1 Cimahi :

a) Kepala sekolah bertugas :
    1. Merencanakan pengembangan penyelenggaraan pendidikan.
    2. Mengorganisasikan seluruh proses pendidikan di sekolah yang meliputi aspek edukatif dan admnistratif.
    3. Me-monitoring perkembangan siswa dan guru bidang studi.

b) Tata usaha bertugas :
    1. Membantu pimpinan dalam menyiapkan rencana pendidikan.
    2. Mengelola data siswa, guru, materi dan data-data yang bersangkutan dengan kegiatan sekolah.

c) Wakil kepala sekolah bidang kurikulum bertugas:
    1. Menyusun jadwal pelajaran.
    2. Mengatur pelaksanaan penilaiaan.
    3. Menyusun jadwal mengajar guru bidang studi.
    4. Menyusun program semester.

d) Wakil kepala sekolah bidang kesiswaan bertugas:
    1. Mengelola kesiswaan dari ekstrakulikuler dan kegiatan siswa lainnya baik intern atau ekstern sekolah.
    2. Bekerjasama dengan pihak OSIS (Organisasi Siswa Intern Sekolah).

e) Wakil kepala sekolah bidang sarana dan prasarana bertugas mengelola penyediaan dan pengadaan barang-barang yang dibutuhkan sekolah (barang inventaris sekolah).

f) Wakil kepala sekolah bidang humas bertugas mengelola informasi sekolah.

g) Guru bidang studi bertugas dalam proses belajar mengajar di sekolah berdasarkan ketentuan yang ditetapkan oleh pihak sekolah.

h) Wali Kelas bertanggung jawab atas seluruh siswa dalam satu kelas

2.2. Landasan Teori

Dalam membangun *Learning Management System* untuk SMPN 1 Cimahi, terdapat landasan teori yang mendukung diantaranya :

2.2.1. *Learning Management System*

Learning Management System (LMS) adalah system yang membantu administrasi dan berfungsi sebagai platform *e-learning content* [2]. Apabila memiliki banyak materi pelajaran *e-learning*, kita tidak hanya meletakkannya pada layar *desktop* komputer dalam bentuk *icon*. Bagaimana bila kita memiliki lebih dari 100 judul mata pelajaran *e-learning*? Layar computer kita tentu akan penuh. Oleh karena itu, kita perlu memiliki LMS sebagai system yang mengatur *e-learning content* atau mata pelajaran *e-learning*.

2.2.1.1. Fungsi LMS

Berikut adalah beberapa fungsi LMS :

1. Katalog

   LMS yang baik harus dapat menunjukkan materi pelajaran yang dimiliki. Materi-materi dapat berupa pelajaran *e-learning*, artikel, tesis, hasil diskusi dan lain sebagainya. Katalog yang baik pun harus dapat membedakan materi berdasarkan jenis materi, departemen yang memerlukan (misalnya, pelajaran marketing mi untuk department marketing dan penjualan) maupun kurikulum (misalnya, pelajaran anggaran untuk kurikulum keuangan, pelajaran *coaching* untuk kurikulum kepemimpinan).

   Katalog yang baik harus dapat menampilkan informasi tentang suatu pelajaran dengan lengkap, meliputi judul, tujuan, cakupan atau outline, durasi, target pelajar, tangggal tersedia, materi pendahuluan, tes yang harus diikuti dan lain sebagainya.

2. Registrasi dan persetujuan

   Fungsi ini memungkinkan calon peserta pelatihan mendaftarkan diri secara *online*, baik untuk pelajaran *online* maupun di kelas. Apabila calon peserta pelatihan selesai mendaftar dan pelajaran membutuhkan persetujuan atasan atau orang lain, maka LMS harus menginformasikan kepada atasan

(biasanya melalui *e-mail*) bahwa terdapat satu pendaftaran yang harus ditanggapi. Atasan dapat mengakses sistem dan melihat pendaftaran serta informasi tentang pelajar dan pelajaran. Kemudian, dia dapat mengambil tindakan, apakah disetujui atau ditolak. LMS kembali menginformasikan hal tersebut kepada calon peserta pelatihan.

LMS yang baik dapat pula menyimpan data pendaftaran dan persetujuan untuk membantu departemen pelatihan dalam memonitoring kegiatan *e-learning* di kemudian hari.

3. Menjalankan dan memonitoring e-learning

   LMS harus menjalankan materi pelajaran *e-learning* dengan baik. Apabila komputer pelajar tidak memiliki konfigurasi yang sesuai, maka LMS harus mengesannya dan memberikan peringatan kepada pelajar.

   Setelah materi pelajaran *e-learning* dijalankan, LMS harus mempunyai kemampuan merekam kegiatan agar dapat dibuat laporannya. LMS harus dapat merekam tentang berapa lama siswa mengakses materi pelajaran, berapa kali, tanggal dan jamnya dan lain sebagainya.

4. Evaluasi

   LMS yang baik pun harus dapat melakukan bermacam evaluasi yang dapat mengukur keahlian peserta pelatihan sebelum dan sesudah mengikuti pelatihan. Evaluasi harus dapat mengukur seberapa jauh peserta pelatihan menyerap materi. Berdasarkan hasil evaluasi, LMS secara otomatis menyarankan untuk mengulang kembali pelajaran, membaca beberapa artikel tambahan, mengikuti pelajaran yang lain atau tindakan lainnya.

5. Komunikasi

   LMS berguna pula sebagai sarana komunikasi bagi departemen pelatihan dan anggota organisasi. LMS dapat menyajikan atau memberikan pengumuman kepada pelajar tertentu. Pengumuman dapat dari pengajar atau administrator pelatihan.

   Komunikasi di sini dapat berarti pengajar memberikan materi bacaan tambahan kepada peserta pelatihan melalui system. Pengajar dapat pula memasukkan atau meng-*upload* sebuah artikel yang ditujukan pada

beberapa pelajar tertentu dan LMS dapat menginformasikannya kepada mereka agar dapat mengakses dan men-*download* artikel melalui LMS.

6. Laporan

   Melalui LMS, para administrator pelatihan dapat memperoleh laporan berisi data pelatihan. Atasan dan manajemen harus dapat mengakses system dan mencetak laporan secara langsung, tanpa meminta bantuan administrator. Misalnya, seorang manajer ingin mengetahui berapa banyak anak buahnya yang telah memperoleh pelatihan MS Powerpoint. Ia mudah masuk atau login ke dalam LMS dan mencari data pelatihan anak buahnya.

7. Rencana pembelajaran

   Seorang manajer dapat membuat rencana pembelajaran untuk seberapa pelajar mengenai analisa kebutuhan belajar. Alangkah baiknya, bila rencana pembelajaran atau training plan dapat diikutsertakan dalam LMS.

   Jadi, berdasarkan rencana pembelajaran, LMS secara otomatis merekomendasikan program pembelajaran yang sesuai dan mengatur jadwalnya. Jadi, pelajar dapat melihat materi pembelajaran yang dia butuhkan melalui LMS, kapan dia bisa mengikuti dan harus menyelesaikan.

8. Integrasi

   Dalam suatu organisasi, ada beberapa system komputer. LMS yang baik dapat berkomunikasi dan berintegrasi dengan system – system yang ada.

   Integrasi dengan system SDM adalah hal yang vital. Dengan integrasi yang baik, LMS akan mendapatkan daftar informasi karyawan terbaru dari system SDM yang sudah ada. Jadi, apabila pada system SDM terdapat perubahan data karyawan, penambahan karyawan baru atau karyawan yang keluar, maka informasi dapat digunakan untuk memperbaharui data pelatihan terkait.

Selain kedelapan fungsi dasar, kita dapat menambahkan fungsi-fungsi penunjang lain, misalnya forum diskusi atau chatroom agar pelajar dan pengajar dapat berdiskusi mengenai pembelajaran, dan lain-lain.

2.2.2. Metode / Algoritma

2.2.2.1. *Text Preprocessing*

Text Preprocessing adalah langkah awal sebelum dilakukan sebuah pencocokan pada sting. Tindakan yang dilakukan pada tahap ini adalah:

1. toLowerCase / case folding, yaitu mengubah semua karakter huruf menjadi huruf kecil,
2. Menghilangkan delimiter-delimiter seperti tanda titik(.), koma(,), spasi dan karakter angka yang ada pada kata tersebut
3. Hapus kata umum (*stopword*).
4. Tokenizing yaitu pemotongan string dalam teks berdasarkan kata yang menyusunnya dan disimpan dalam sebuah variable *array* [3].

2.2.2.2. Algoritma Levenshtein Distance

Algoritma Levenshtein distance dibuat oleh Vladimir Levenshtein pada tahun 1965. Algoritma levenshtein distance merupakan metrik yang digunakan untuk mengukur perbedaan jarak antara dua sekuens. Perhitungan edit distance didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string. Perhitungan jarak antara dua string ini ditentukan dari jumlah minimum operasi perubahan untuk membuat string A menjadi string B. Levenshtein distance antara dua string ditentukan berdasarkan jumlah minimum perubahan/pengeditan yang dibutuhkan untuk melakukan transformasi dari satu bentuk string ke string yang lain. Ada 3 macam operasi utama yang dapat dilakukan oleh algoritma levenshtein distance yaitu:

a. Operasi pengubahan karakter

Operasi pengubahan karakter merupakan operasi menukar sebuah karakter dengan karakter lain. Contohnya penulisan *string* “informatica” menjadi “informatika”. Dalam kasus ini karakter ‘c’ diganti dengan huruf ‘k’.

b. Operasi penambahan karakter

Operasi penambahan karakter berarti mengambahakan karakter ke dalam suatu *string*. Contohnya *string* “infrmatika” dilakukan penambahan

karakter ‘o’ setalah karakter ‘r’ untuk membentuk *string* “informatika”. Penambahan karakter tidak hanya dilakukan dengan menyisipkan ditengah-tengah *string*, tapi juga bisa ditambahkan diawal atau diakhir *string*.

c. Operasi penghapusan karakter

Operasi penghapusan karakter dilakukan untuk menghilangkan karakter dari suatu *string.* Contohnya string “informatikar”, karakter terkhir dihilangkan sehingga menjadi string “informatika”. Pada operasi ini dilakukan operasi penghapusan karakter ‘r’. [4]

Contoh :

1. Jika a adalah “informatika” dan b adalah “informatika”, maka LD(a,b) = 0, karena tidak ada transformasi yang dibutuhkan. Kedua string adalah identik.
2. Jika a adalah “informatika” dan b adalah “infrmatika”, maka LD(a,b) = 1, karena dibutuhkan satu insert (menambahkan ‘o’) dicukupkan untuk menstransformasikan string a menjadi string b.

Perhitungan harga pengeditan pada setiap operasi yang dilakukan adalah sesuai dengan aturan berikut:

1. d(a,ε) = 1 harga untuk menghapus substring a
2. d(ε,a) = 1 harga untuk penyisipan substring a
3. d(a,a) = 1 harga untuk substitusi substring a ke substring b
4. d(a,a) = 0

Semakin besar nilai yang dihasilkan oleh operasi levenshtein distance maka semakin besar pula perbedaan di antara kedua string tersebut. Penggunaan levenshtein distance terbatas pada penentuan kemiripan dua buah string dilihat dari posisi huruf-huruf yang ada di kedua kata.

Misalnya, jika pada kunci jawaban guru “kumpulan makhluk hidup” dengan jawaban siswa “kumpulan makluk hidup”. Pada jawaban siswa dan kunci

jawaban terlihat perbedaan pada string “makhluk” dengan “makluk”, maka salah salah satu string jawaban siswa tersebut berjarak 1 dari string kunci jawaban. Jarak tersebut secara persentase bernilai 14% dari ukuran string “makluk” sehingga kemiripan relatif dari string tersebut dengan salah satu string kunci jawaban adalah 86%.

2.2.2.3. Latent Semantic Analysis

Latent Semantic Analysis (LSA) adalah suatu metode untuk mengekstrak sebuah tulisan dalam suatu dokumen dan kemudian mengaplikasikannya dalam perhitungan matematis. 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.[5]

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

Langkah-langkah pemeriksaan *essay* dengan metode LSA adalah pertama-tama 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 ini adalah contoh yang biasa digunakan. Pada gambar 2.2 dapat dilihat beberapa kalimat yang terdiri dari dua tema.

Technical Memo Example
Titles:
c1: *Human* machine *interface* for Lab ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user*-perceived *response time* to error measurement

m1: The generation of random, binary, unordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

Gambar 2.2 Contoh Kalimat dan Kata Kunci

Dari contoh pada gambar 2.2 ada sembilan kalimat dengan lima kalimat pertama bertema human computer interaction (kalimat c1 sampi c5), dan empat kalimat selanjutnya bertema mathematical graph theory (kalimat m1 sampai m4). Syarat sebuah kata menjadi kata kunci pada paragraf ini adalah setidaknya sebuah

kata harus muncul minimal 2 kali dalam *queries*. Pentransformasian kata-kata diatas menjadi sebuah matriks A ditunjukkan pada tabel 2.1.

Tabel 2.1. Transformasi Matriks Kemunculan Kata Kunci Pada Susunan Kalimat dalam Gambar 2.2

	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	2	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 2.1. 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 :

$$A_{mn} = U_{mm} \text{ x } S_{mn} \text{ x } V^{T}_{nn} \quad \textbf{...(2-3)}$$

Dimana $U^TU=I$, $V^TV=I$. Kolom matriks U merupakan Eigenvektor Orthonormal dari AA^T, sedangkan kolom matriks V merupakan Eigenvektor Orthonormal dari A^TA, dan S merupakan matriks diagonal akar dari nilai eigen dari matriks U atau V dalam urutan dari yang terbesar (Descending order).

Matriks U dari hasil perhitungan untuk contoh pada gambar 2.2

ditunjukkan pada tabel 2-2.

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

Sedangkan matriks kedua berupa matriks diagonal yang memuat nilai skalar matriks didapat dari akar kuadrat dari nilai eigen matriks A^TA dan disebut sebagai matriks S. Matriks S dari hasil perhitungan untuk contoh pada gambar 2.2 ditunjukkan pada tabel 2.3. Untuk matriks terakhir cara mencarinya sama dengan matriks pertama, yang menjadi perbedaan adalah yang dicari matriks AA^T bukan A^TA, dan matriks ini disebut sebagai matriks V Matriks V dari hasil perhitungan untuk contoh pada gambar 2.2ditunjukkan pada tabel 2-4.

Tabel 2.3. Matriks S sebagai komponen dari 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

Tabel 2.4. Matriks V sebagai komponen vektor orthogonal transpose 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

Reduce matriks dipilih sebesar *k* dimensi dalam contoh ini k = 2, sehingga matriks komponen S hanya menyisakan nilai seperti pada Tabel 2.5, kemudian komponen U, S yang disederhanakan, dan V dikalikan kembali untuk mendapat matriks A_k yang merupakan matriks rekonstruksi A. Hasilnya adalah seperti tabel yang ditunjukkan oleh tabel 2.5.

Tabel 2.5. Matriks A_k sebagai matriks rekonstruksi A

0,16	0,4	0,38	0,47	0,18	-0,05	-0,12	-0,16	-0,09
0,14	0,37	0,33	0,4	0,16	-0,03	-0,07	-0,1	-0,04
0,15	0,51	0,36	0,41	0,24	0,02	0,06	0,09	0,12
0,26	0,84	0,51	0,7	0,39	0,03	0,08	0,12	0,19
0,45	1,23	1,05	1,27	0,56	-0,07	-0,15	-0,21	-0,05
0,16	0,58	0,38	0,42	0,28	0,06	0,13	0,19	0,22
0,16	0,58	0,38	0,42	0,28	0,06	0,13	0,19	0,22
0,22	0,55	0,51	0,63	0,24	-0,07	0,14	-0,2	-0,11
0,1	0,53	0,23	0,21	0,27	0,14	0,31	0,44	0,42
-0,06	0,23	-0,14	-0,27	0,14	0,24	0,55	0,77	0,66
-0,06	0,34	-0,15	-0,3	0,2	0,31	0,69	0,98	0,85
-0,04	0,25	-0,1	-0,21	0,15	0,22	0,5	0,71	0,62

Matriks A_k tidak sama dengan matriks A yang asli. Matriks A_k hanyalah sebuah pendekatan atau aproksimasi A pada faktor *k*. SVD yang dilakukan setelah penyederhanaan mengambil sebagian besar struktur penting yang terdapat pada korelasi kata kunci dan dokumen. Pada saat yang sama juga menghilangkan variabilitas penggunaan kata yang menjadi gangguan utama. Selama nilai dari *k* jauh lebih kecil dari banyaknya kata kunci maka perbedaan minor dalam terminologi dapat diabaikan. Kata-kata kunci yang terdapat dalam kalimat yang sama, akan berdekatan satu sama lain dalam ruang *k* walaupun kata kunci itu tidak pernah muncul bersamaan lagi pada kalimat yang sama. Hal ini berarti beberapa kalimat yang tidak memiliki satupun kata kunci yang sama, maka dia tidak akan mendekati saling berhubungan dalam ruang-*k*.

2.2.2.4. Penilaian Essay

Query yang merupakan kata kunci dari kunci jawaban dapat direpresentasikan sebagai vektor dalam ruang-k. Vektor inilah yang kemudian dibandingkan dengan vektor-vektor jawaban essay siswa (dokumen) untuk selanjutkan dinilai kemiripannya. Sebuah query seperti halnya dokumen, merupakan kumpulan dari kata-kata. Query dapat representasikan sebagai

$$\overline{q} = q^T U_k \Sigma_k^{-1} \qquad \textbf{...(2-4)}$$

Seperti vektor *query*, vektor dokumen direpresentasikan sebagai

$$\overline{d} = d^T U_k \Sigma_k^{-1} \qquad \textbf{...(2-5)}$$

Matriks q adalah matriks satu kolom yang elemennya berisi jumlah kata kunci dalam *query*. Sementara matriks d adalah matriks satu kolom. Elemennya berisi nilai kemunculan kata kunci dalam dokumen. Matriks d sama dengan kolom matriks A. $\overline{q}$ adalah vektor query dan $\overline{d}$ adalah vektor dokumen. Vektor query

dikorelasikan dengan semua vektor dokumen yang ada. Teknik korelasi yang umum digunakan adalah dengan mencari nilai kemiripan kosinus yang dibentuk antara vektor query dan vektor dokumen. Korelasi kosinus antara vektor query dan vektor dokumen diberikan oleh persamaan

$$\cos\alpha = \frac{\overline{q} \cdot \overline{d}}{\|\overline{q}\|\|\overline{d}\|} \qquad \textbf{...(2-6)}$$

α adalah sudut diantara kedua vektor tersebut. Jadi, nilai korelasi adalah perhitungan sudut berdasarkan kosinus antara $\overline{q}$ dan $\overline{d}$. Jika dilakukan penilaian dari jawaban siswa yang paling besar ke paling kecil nilainya, maka jawaban yang paling besar nilainya adalah yang memiliki sudut α dengan yang paling kecil.

2.2.3. Konsep Perancangan Sistem

Dalam perancangan system yang akan dibangun, perlu adanya perencanaan pengolahan data dimana semua data disimpan ke dalam sebuah *database* agar seluruh data saling berelasi.

2.2.3.1. Basis Data

Basis data (database) adalah kumpulan dari berbagai data yang saling berhubungan satu dengan yang lainnya dan dapat diakses dengan mudah dan cepat. Basis data tersimpan di perangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para user atau user. Data diartikan sebagai representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, konsep, dan lain-lain. Terdapat sejumlah sudut pandang pengertian basis data, yaitu:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan

kembali dengan cepat dan mudah.

2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengquiz (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Penyusunan basis data meliputi proses memasukkan data kedalam media penyimpanan data dan diatur dengan menggunakan perangkat Sistem Manajemen Basis Data (Database Management System DBMS). Manipulasi basis data meliputi pembuatan pernyataan (query) untuk mendapatkan informasi tertentu, melakukan pembaharuan atau penggantian (update) data, serta pembuatan report data. Dalam sebuah media penyimpanan, basis data dapat diciptakan maupun dihilangkan. Dalam sebuah basis data terdiri atas dua atau lebih tabel yang saling berhubungan. Dalam operasi basis data, tabel-tabel tersebut dapat diciptakan dan dapat dihilangkan juga.

Elemen Basis Data terdiri dari :

1. Entitas adalah sekumpulan objek yang terdefinisikan yang mempunyai karakteristik sama dan bisa dibedakan satu dengan lainnya. Objek dapat berupa barang, orang, tempat atau suatu kejadian.
2. Atribut adalah deskripsi data yang bisa mengidentifikasi entitas yang membedakan entitas tersebut dengan entitas yang lain. Seluruh atribut harus cukup untuk menyatakan identitas obyek, atau dengan kata lain kumpulan atribut dari setiap entitas dapat mengidentifikasi keunikan suatu individu.
3. Data value (nilai data) adalah data aktual atau informasi yang disimpan pada tiap data, elemen atau atribut. Atribut nama pegawai menunjukan tempat dimana informasi nama karyawan disimpan.
4. File/Tabel merupakan kumpulan record sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda nilai datanya.

5. Record/Tuple merupakan kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu record mewakili satu data atau informasi.

2.2.3.2. Flowmap

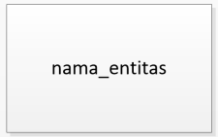
Flowmap adalah campuran peta dan flowchart, yang menunjukkan pergerakan benda dari satu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, jumlah barang yang diperdagangkan, atau jumlah paket dalam jaringan. Flowmap menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian, yang akan dijelaskan dalam tabel.

2.2.3.3. *Entity Relationship Diagram (ERD)*

ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi chen.

Pada model *Entity-Relationship Diagram*, semesta data yang ada di dunia nyata diterjemahkan/ditranformasikan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data, yang umum disebut sebagai Diagram Entity-Relationship [6]. Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi chen:

Tabel 2.6. Simbol-simbol pada ERD

Simbol	Deskripsi
Entitas / *Entity* 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel
Atribut 	Field atau kolom data yang butuh disimpan dalam suatu entitas
Atribut Kunci Primer 	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
Atribut Multinilai / *Multivalue* 	Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi / *Association* 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.

ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah entitas) atau N-*ary* (satu relasi menghubungkan banyak entitas), tetapi banyak metode perancangan ERD yang tidak mengijinkan hubungan *ternary* atau N-*ary*.

2.2.3.4. *Data Flow Diagram (DFD)*

Data Flow Diagram (DFD) awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured System Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson [6].

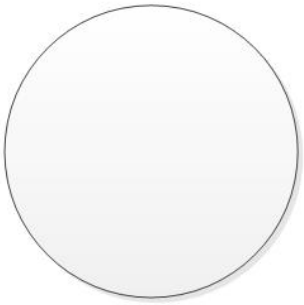
Pada tahun 1980-an Edward Yourdon dan Tom Demarco memperkenalkan metode yang berbeda, dimana dengan mengubah persegi dengan sudut lengkung (pada DFD Chris Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. DFD Edward Yourdon dan Tom Demarco popular digunakan sebagai model analisis sistem perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur.

Data Flow Diagram atau pemodelan aliran data adalah aktivitas pemodelan inti pada analisis terstruktur. DFD memperlihatkan gambaran tentang masukan-proses-keluaran dari suatu perangkat lunak. Yaitu, objek-objek data mengalir ke dalam perangkat lunak, kemudian objek-objek data itu akan ditransformasi oleh elemen-elemen pemrosesan, dan objek-objek data hasilnya akan mengalir keluar dari sistem/perangkat lunak.

DFD yang pertama (DFD peringkat 0 atau diagram konteks) menggambarkan sistem secara keseluruhan. DFD-DFD berikutnya sesungguhnya merupakan penghalusan dari diagram konteks, memberikan gambaran yang semakin rinci dari diagram konteks, dan hal ini akan berlanjut ke peringkat-peringkat selanjutnya [7].

DFD tidak sesuai untuk memodelkan sistem perangkat lunak yang akan dibangun menggunakan pemrograman berorientasi objek. Paradigma pemrograman terstruktur dan pemrograman berorientasi objek merupakan hal yang berbeda. Notasi-notasi pada DFD (Edward Yourdon dan Tom Demarco) adalah sebagai berikut: [6]

Tabel 2.7. Notasi-notasi pada DFD

Simbol	Deskripsi
	Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.
	File atau basisdata atau penyimpanan (storage); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (*Entity Relationship Diagram* (ERD))
	Entitas luar (*external entity*) atau masukan (*input*) atau keluaran (*output*) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.
	Aliran data; marupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (*input*) atau keluaran (*output*).

Pada satu diagram DFD sebaiknya jumlah modul tidak boleh melebihi 20 buah. Jika lebih dari 20 buah modul, diagram akan terlihat rumit atau susah untuk dibaca sehingga menyebabkan sistem yang dikembangkan juga menjadi rumit [6].

2.2.4. Pengembangan Perangkat Lunak

2.2.4.1. *Personal Home Page (PHP)*

PHP adalah kombinasi antara bahasa pemograman dan aplikasi *server* [8]. PHP merupakan *server-side* scripting yang digunakan untuk membuat dan menjalankan aplikasi web *server* yang dinamis, interaktif dan mempunyai performasi tinggi. Aplikasi *server* adalah program yang terdiri atas teknik-teknik dalam satu paket yang meliputi :

1. Ketangguhan bahasa pemograman
2. Pengaksesan basis data ke media penyimpanan yang permanent.
3. Mendukung internet protocol, khususnya HTTP dan e-mail.

PHP mampu berhubungan dengan basis data dan dapat diintegrasikan dengan HTML. PHP dapat berjalan dalam web server yang berbeda dalam sistem operasi yang berbeda pula. PHP ditulis dalam bahasa C, sehingga sebagian besar sintak PHP mirip dengan C dan *perl*. Beberapa keuntungan menggunakan PHP adalah sebagai berikut :

1. *Open source,* semua *source* PHP tersedia.
2. PHP diterbitkan secara gratis, tidak ada biaya dalam mengimplementasikan.
3. *Cross-platform,* PHP dapat berjalan dalam sistem operasi Windows 98, Windows NT dan Macintosh.
4. PHP juga dapat berjalan pada server Microsoft Personal Web Server, IIS dan Aphace.
5. Embedded-html, karena itu PHP mudah dipelajari.

2.2.4.2. *Database Management System* (DBMS)

DBMS (*Database Management System*) adalah suatu system aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. suatu system aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut:

a. Menyediakan fasilitas untuk mengelola akses data
b. Mampu menangani integritas data
c. Mampu menangani akses data secara bersamaan
d. Mampu menangani *backup* data

DBMS sudah mulai berkembang sejak tahun 1960an. Kemudian sekitar tahun 1970an mulai berkembang teknologi *Relational DBMS* yaitu DBMS berbasis relasional model. Relasional model pertama kali dikembangkan oleh Edgar J. Codd pada tahun 1970. Secara sederhana relasional dapat dipahami sebagai suatu model yang memandang data sebagai sekumpulan tabel yang saling terkait. Hampir semua DBMS komersial dan *open source* saat ini berbasis *Relastional DBMS*.

Pada tahun 1980an mulai berkembang *Object Oriented DBMS* (OODBMS). OODBMS berkembang seiring dengan perkembangan teknologi pemrograman berorientasi objek. Saat ini OODBMS juga cukup berkembang namun belum menggeser kepopuleran RDBMS.

MySQL adalah perangkat lunak DBMS atau server basis data yang dapat melayani banyak pengguna, dan dapat melakukan proses dalam waktu yang bersamaan *(multi threaded).* MySQL juga sangat cepat dan fleksibel dan dapat menyimpan data yang berkapasitas besar. Dalam perancangannya, MySQL mengutamakan kecepatan proses, ketangguhan dan kemudahan bagi penggunanya.

Pada saat ini MySQL adalah server basis data yang paling banyak diminati oleh banyak perusahaan di dunia. MySQL menyediakan lingkungan yang mudah dan efisien untuk menyimpan data dan mengakses basis data berkapasitas besar lebih cepat dari pada server basis data lainnya. Keuntungan lainnya adalah bahwa MySQL server tidak membutuhkan spesifikasi hardware yang mahal.

MySQL server memungkinkan membuat aplikasi client untuk mengakses dan memanipulasi data di server basis data melalui basis data provider. Keuntungan utama menggunakan MySQL server adalah mudah digunakan, berkecepatan tinggi dan membutuhkan sedikit memori serta tidak membutuhkan spesifikasi hardware yang mahal [7].

2.2.4.3. Dreamweaver

Macromedia Dreamweaver adalah program editor HTML professional untuk mendesain, mengkode dan mengembangkan situs, halaman web, serta aplikasi web. Tampilan fitur editing dalam Dreamweaver akan membantu dalam membuat sebuah halaman web tanpa harus menuliskan kode java script maupun HTML. Dreamweaver juga dapat membantu dalam membangun dinamik database untuk aplikasi *web* dengan menggunakan server seperti ASP, ASPNet, ColdFusion Markup Language (CFML), JSP dan PHP.

2.2.4.4. Web Browser

Browser web adalah software yang digunakan untuk menampilkan informasi dari server web. Software ini kini telah dikembangkan dengan menggunakan user interface grafis, sehingga pemakai dapat melakukan point dan click untuk pindah dokumen. Dapat dikatakan saat ini hanya ada 2 browser GUI yang popular, yaitu Intenet Explorer dan Netscape Navigator [9].

2.2.4.5. Jama PHP

Selain MATLAB, aplikasi matematis web based yang bisa digunakan untuk penghitungan SVD adalah JAMA. JAMA adalah singkatan dari Java Matrix. JAMA merupakan script php untuk perhitungan matriks kompleks. Class-class dari package JAMA akan sering digunakan dalam operasi matriks seperti perkalian matriks, transpose, dan inverse. Karena JAMA hanya merupakan script PHP bukan merupakan aplikasi maka kinerja server-pun tidak akan terlalu terbebani.