# BAB II
# TINJAUAN PUSTAKA

Tinjauan pustaka menjelaskan beberapa konsep dan dasar teori yang berkaitan dengan permasalahan yang akan dibahas sebagai dasar pemahaman dalam mengimplementasikan konsep-konsep tersebut ke dalam semua kegiatan pengembangan sistem, yaitu profil sekolah, *learning management system*, algoritma Rabin Karp, *synonym recognition* serta *stemming.*

## 2.1 Profil SMA Negeri 2 Tasikmalaya

Profil SMA Negeri 2 Tasikmalaya meliputi sejarah, visi, misi dan struktur organisasi sebagai berikut :

### 2.1.1 Sejarah

Tanggal 22 Maret 1966 dengan surat bernomor 350/C.4/Id.SMA/K.66, inspeksi daerah SMA Jawa Barat menganjurkan kepada SMA Negeri 1 Tasikmalaya untuk mendirikan SMA Negeri 2 Tasikmalaya. Hal ini dikarenakan kapasitas SMA Negeri 1 Tasikmalaya sudah tidak memadai lagi untuk menampung semua lulusan SMP yang berminat melanjutkan pendidikan di SMA. sebagai tindak lanjut dari hal tersebut, pemerintah Daerah Tasikmalaya dengan suratnya bernomor 334/6/66 tanggal 11 Oktober 1966 mengusulkan kepada Inspeksi Daerah SMA Provinsi Jawa Barat untuk mendirikan SMA Negeri 2 Tasikmalaya.

Surat ini juga dilampiri susunan panitia pendiri SMA Negeri 2 Tasikmalaya :

Ketua : Husen Wangsaatmaja (Bupati)

Wakil Ketua : R. Mustopa (Kepala SMA Negeri 1 Tasikmalaya)

Sekretaris : Engkus Subarman, BA. (Ipdap Tasikmalaya)

Anggota :

1. Kapten Suardi Kusuman (BPH Sie Tasikmalaya)

2. Rs. Kastama (Kepala SMP Negeri 1 Tasikmalaya)
3. Totong Rusmana (Wa. Kepala SMA Negeri 1 Tasikmalaya)
4. Drs. Tirio (Guru SMA Negeri 1 Tasikmalaya)
5. Dodo Sahoman (Guru SMA Negeri 1 Tasikmalaya)
6. O. Hardipraja (Koordinator PGRI Tasikmalaya)
7. Mastur (Camat Indihiang)
8. Sugaman (Wakil Kepala SMP Negeri 2 Tasikmalaya)

Setelah itu didirikanlah kelas jauh SMA Negeri 2 Tasikmalaya dengan harapan pada tahun 1968 dapat menjelma menjadi SMA Negeri 2 Tasikmalaya. Hal ini sesuai dengan balasan Kepala Ipdap SMA Provinsi Jawa Barat dengan suratnya bernomor 573/D.2a/K.66 tanggal 26 Oktober 1966 yang isinya mendukung usul Pemda Tasikmalaya serta berjanji meneruskan usul tersebut ke Direktorat Pendidikan Umum di Jakarta. Langkah selanjutnya yang diambil oleh Ipdap SMA Provinsi Jawa Barat yaitu menunjukan Bapak Totong Rusmana menjadi pimpinan kelas jauh tersebut sejak tanggal 1 Desember 1966 ditandai dengan keluarnya surat tugas sementara no. 521/b.3/K.66 tanggal 10 Januari 1966.

Pada bulan September 1975 diterima tembusan surat dari Kassubag penyusunan anggaran Direktorat Jendral Pendidikan Dasar dan Menengah di Jakarta dengan surat Nota Dinas tanggal 4 September 1975 kepada bagian Efesiensi Tata Laksana Dirjen Pendidikan Dasar dan Menengah di Jakarta bahwa pada prinsipnya menyetujui dengan sepenuhnya kelas jauh SMA Negeri 2 Tasikmalaya menjadi SMA Negeri 2 Tasikmalaya seutuhnya.

Sejak tanggal 1 Nopember 1975 sesuai dengan SK Mendikbud No. 018/01/1976 status kelas jauh SMAN 2 tasikmalaya berubah menjadi SMAN 2 Tasilmalaya yang tetap dipimpin oleh Bapak Totong Rusmana. Kemudian pada tanggal 16 September 1976 bertepatan pada hari Kamis diadakan peresmian SMA Negeri 2 Tasikmalaya sekaligus serah terima pimpinan dari Bapak Totong Rusmana kepada R. Siti Aisyah Wahyu. Sesuai dengan perubahan nama SMA menjadi SMU, maka nama SMAN 2 Tasikmalaya pun berubah menjadi SMU Negeri 2 Tasikmalaya.

Berdasarkan Surat Keputusan Walikota Tasikmalaya Nomor 420/Kep. 81-Disdik/2002, tanggal 1 Agustus tentang Perubahan Nama, Nomor Urut dan Nomor Statistik SD, SLTP, SMU dan SMK Negeri/Swasta di Kota Tasikmalaya, maka nama SMU Negeri 1 Indihiang berubah menjadi SMU Negeri 2 Tasikmalaya. Kemudian berdasarkan SK Walikota no. 421/Kep. 7A Disdik/2004 tanggal 21 Januari 2004 nama SMU Negeri 2 Tasikmalaya berubah lagi menjadi SMA Negeri 2 Tasikmalaya.

### 2.1.2 Visi

Visi SMA Negeri 2 Tasikmalaya adalah "mewujudkan sekolah yang unggul, kompetitif dengan dilandasi akhlak mulia".

### 2.1.3 Misi

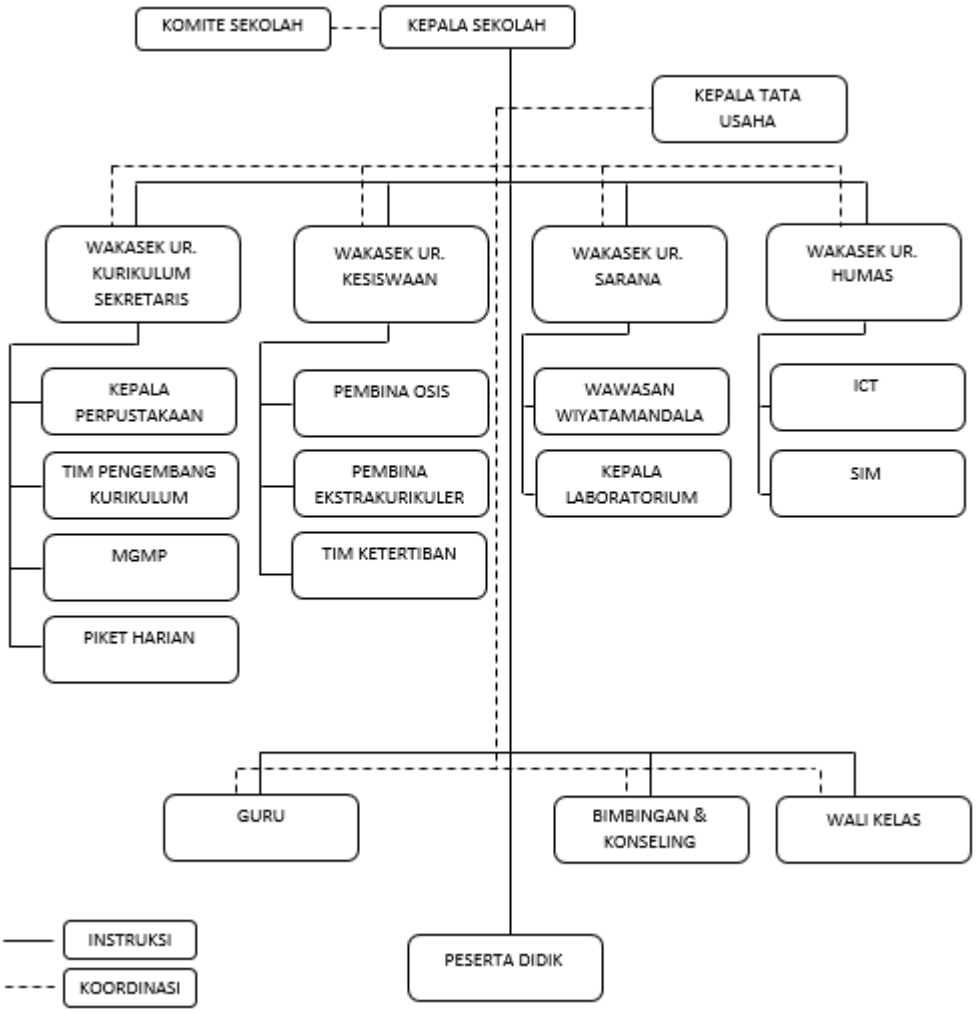
Misi SMA Negeri 2 Tasikmalaya adalah sebagai berikut :

1. Mewujudkan pengelolaan sekolah secara efektif, inovatif dan akuntabel
2. Mengembangkan potensi dan minat siswa secara optimal sehingga mampu hidup mandiri ,
3. unggul, terampil dan memiliki akhlak mulia
4. Meningkatkan mutu tenaga pendidik dan tenaga kependidikan secara berkelanjutan
5. Mengupayakan perluasan dan peningkatan mutu sarana prasarana.
6. Mengoptimalkan kemitraan dengan alumni, dunia usaha dan masyarakat.

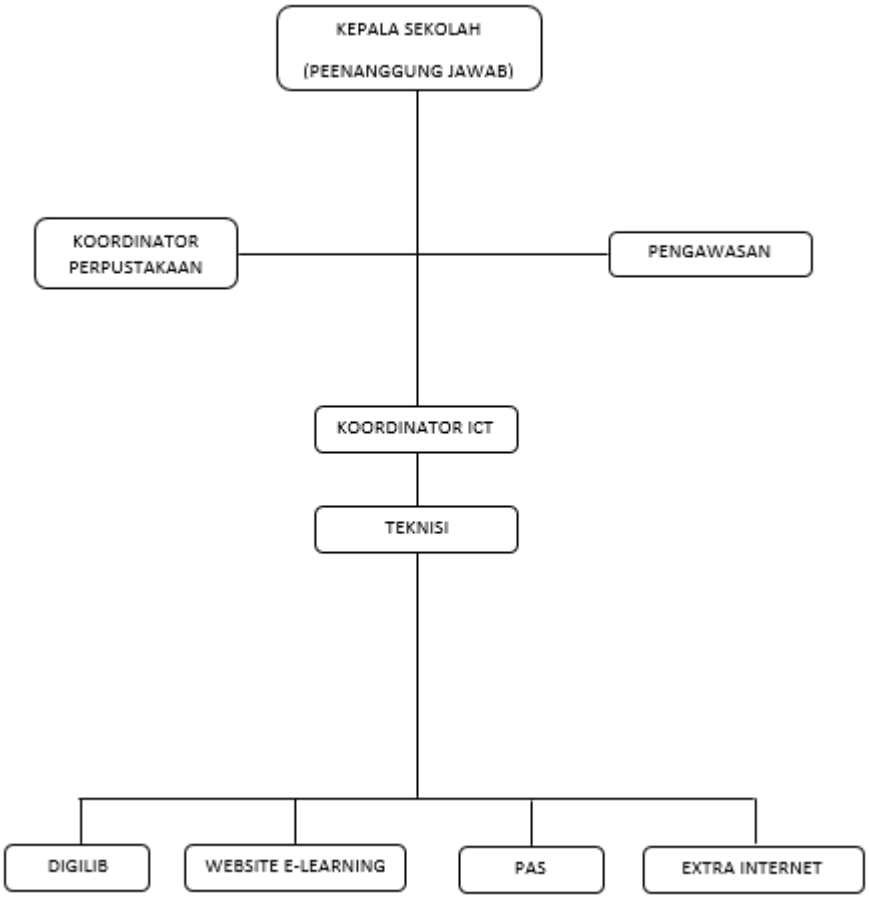
### 2.1.4 Struktur Organisasi

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

Bagan struktur organisasi SMA Negeri 2 Tasikmalaya dan struktur organisasi ICT SMA Negeri 2 Tasikmalaya ditunjukan pada gambar 2.1 dan gambar 2.2.




Gambar 2.1 Struktur Organisasi SMAN 2 Tasikmalaya




Gambar 2.2 Struktur Organisasi ICT SMAN 2 Tasikmalaya

Rincian tugas pengelolaan ICT SMA Negeri 2 Tasikmalaya berdasarkan gambar 2.2.

1. **Penanggung Jawab ( Kepala sekolah)**

   Penanggung jawab (kepala sekolah) memiliki tugas-tugas sebagai berikut:

   a) Bertanggung jawab atas seluruh kegiatan dalam pengelolaan ICT di SMA Negeri 2 Tasikmalaya, mulai dari tahap persiapan, pelaksanaan sampai pelaporan.
   b) Mengarahkan dan membimbing seluruh pengurus dan pengelola ICT dalam melaksanakan tugasnya.
   c) Mengevaluasi hasil kegiatan dalam mengelola ICT

**2. Pengawas ICT**

Pengawas ICT memiliki tugas-tugas sebagai berikut:

a) Ikut mengawasi jalannya pengelolaan ICT di SMA Negeri 2 Tasikmalaya
b) Memberikan motivasi dalam memanfaatkan potensi guru dan peserta didik untuk mengembangkan *website*, *e- learning* dan *digital library*.
c) Ikut mengawasi dan/atau memberi bantuan dalam pemeliharaan dan perawatan sarana / prasarana yang berkaitan dengan kepentingan ICT

**3. Petugas Keuangan Labkom**

Petugas keuangan labkom memiliki tugas-tugas sebagai berikut:

a) Membukukan biaya pengelolaan ICT dan laboratorium komputer yang bersumber dari iuran peserta didik
b) Pemegang keuangan ICT
c) Membuat laporan pertanggungjawaban keuangan pengelolaan ICT

**4. Koordinator ICT dan Teknisi**

**a. Koordinator ICT**

Koordinator ICT memiliki tugas-tugas sebagai berikut:

1) Membuat Program Kerja dan anggaran tahunan yang diusulkan kepada Kepala Sekolah
2) Mengusulkan, mengubah (berdasarkan masukan IT *Advisor*), dan mengevaluasi kebijakan-kebijakan dan rencana strategis TIK, setiap tahun
3) Menjalankan kebijakan dan rencana strategis TIK
4) Pengelolaan infrastruktur (*hardware*, *software*, *network*) dibantu oleh teknisi/guru TIK.
5) Koordinasi dengan wakasek dan koordinator perpustakaan guna memberikan pelayanan ICT setiap diperlukan
6) Membuat laporan atau evaluasi pengelolaan ICT pada akhir tahun bersama tim teknisi

**b. Teknisi-1**

Teknisi 1 memiliki tugas-tugas sebagai berikut:

1) Menjalankan, dan mengevaluasi kebijakan-kebijakan ICT yang berhubungan dengan pengelolaan infrastruktur jaringan internet.
2) Menjalankan dan mendokumentasikan prosedur-prosedur, setiap saat diperlukan.
3) Mengelola *network* yang efisien dan tepat guna, setiap hari.
4) Menjaga keamanan *network* (*firewall*, *network policy*, *security policy*, *etc*.), setiap hari.
5) Mengelola *server* yang efisien dan tepat guna.
6) Menjaga keamanan *server*.
7) Instalasi aplikasi di *operational environment* (*note*: *developer* sistem informasi tidak boleh memiliki akses di *operational environment*), setiap saat diperlukan
8) Mengelola *account* yang ada, setiap hari
9) Perawatan *software website*, *E-learning*, *digital library* dan *e-mail* sekolah (www.sman2-tsm.sch.id)

**c. Teknisi-2**

Teknisi 2 memiliki tugas-tugas sebagai berikut:

1) Menjalankan, dan mengevaluasi kebijakan-kebijakan ICT yang berhubungan dengan pengelolaan infrastruktur jaringan.di area sekolah
2) Menjalankan dan mendokumentasikan prosedur-prosedur, setiap saat diperlukan
3) Mengelola *network* yang efisien dan tepat guna, setiap hari
4) Menjaga keamanan *network* (*firewall*, *network policy*, *security policy*, *etc*.), setiap hari
5) Mengelola *server* yang efisien dan tepat guna
6) Menjaga keamanan *server*
7) Penanggung jawab Ekstra Pembelajaran Internet

8) Perawatan dan pemeliharaan jaringan internet ke ruangan kelas XII Ruang Laboratorium Komputer, Lab. Kimia. Lab. Fisika, Lab. Biologi , ruang perpustakaan, ruang OSIS

**d. Teknisi-3**

Teknisi 3 memiliki tugas-tugas sebagai berikut:

1) Menjalankan, dan mengevaluasi kebijakan-kebijakan ICT yang berhubungan dengan pengelolaan infrastruktur jaringan.di area sekolah
2) Menjalankan dan mendokumentasikan prosedur-prosedur, setiap saat diperlukan
3) Mengelola *network* yang efisien dan tepat guna, setiap hari
4) Menjaga keamanan *network* (*firewall*, *network policy*, *security policy*, *etc*.), setiap hari
5) Mengelola *server* yang efisien dan tepat guna
6) Menjaga keamanan *server*
7) Penanggung jawab Kegiatan ekstra Pembelajaran Internet
8) Perawatan dan pemeliharaan jaringan internet ke ruangan kelas X danm XI, ruang Wakasek. Ruang TU, ruang Guru dan ruang BP

**5. Petugas *Website*, *Digilib* dan PAS**

Petugas *website*, *diglib* dan PAS memiliki tugas-tugas sebagai berikut:

1) Petugas *website* bertanggung jawab dalam menampilkan berita-berita dari hasil liputan di *website* smandatas
2) Petugas *website* dapat memberikan bimbingan kepada guru mata pelajaran tentang cara *upload* bahan pembelajaran di situs *e-learning*
3) Petugas *digilib* mengoperasionalkan *web digital library* sesuai dengan aturan standar internasional
4) Petugas *digilib* bekerjasama dengan koordinator perpustakaan dan stafnya
5) Penanggung Jawab PAS melengkapi data sekolah, yang terdiri dari:
    a. Data Kepegawaian dan keuangan
    b. Bidang Kurikulum, seperti Nilai Ulangan dsb

c. Bidang Kesiswaan, seperti Absensi Sidik Jari Siswa, absensi Sidik Jari guru dan Pegawai
d. Bidang Humas, seperti: data Alumni,
e. Bidang Sarana, seperti: data sarana dan prasarana yang ada di sekolah

## 2.2 Landasan Teori

Landasan teori yang mendukung dalam mengembangkan *Learning Management System* SMA Negeri 2 Tasikmalaya, diantaranya adalah sebagai berikut:

### 2.2.1 *Learning Management System*

*Learning Management System* (LMS) adalah suatu perangkat lunak yang digunakan secara online untuk mendukung kegiatan pembelajaran dalam pengelolaan materi dan bahan ajar, dokumentasi, keperluan administrasi, pemberian dan pengumpulan tugas, serta pelatihan dan laporan kegiatan belajar mengajar.

LMS juga menyediakan fitur-fitur yang dapat memenuhi semua kebutuhan dari pengguna dalam hal proses pembelajaran. Saat ini ada banyak jenis LMS yang ditawarkan, setiap jenis LMS memiliki fitur-fiturnya masing-masing sesuai dengan fasilitas yang telah diberikan. LMS dapat berisi materi-materi yang dikemas dalam bentuk multimedia (teks, animasi, *video*, *sound*), yang diberikan sebagai *supplement* dan *enrichment* bagi pengembangan kompetensi pembelajar. LMS menawarkan sistem pembelajaran inovasi yang mencakup dalam bidang teknologi informasi, khususnya yang berbasis *virtual* melalui *web online learning*, *multimedia* dan *video conference*. LMS pembelajaran berbasis *web* dikembangkan secara dinamis (*dynamic e-learning*).

Fungsi secara umum yang harus dimiliki *Learning Management System* (LMS) antara lain:

a. *Uploading and sharing material*: LMS menyediakan layanan untuk mempermudah proses publikasi material proses pembalajaran. Instruktur

akan mengunggah materi ajar sesuai dengan silabus yang telah dibuat, bisa berupa catatan materi, artikel-artikel *quiz*, penilaian dan lainnya.

b. *Forum and chart*: forum dan *chatting online* merupakan komunikasi dua arah antara instruktur/dosen dengan mahasiswanya, baik dilakukan secara sinkron (*char*), maupun asinkron (forum, *email*). Dengan fasilitas yang ada ini memungkinkan peserta didik untuk menulis tanggapannya, dan mendiskusikannya dengan teman-temannya yang lain.
c. *Quizzes and surveys*: kuis dan *survey* secara online dapat memberikan *grade* secara instan bagi peserta didik. Hal ini merupakan *tool* yang sangat baik digunakan untuk mendapatkan respon (*feedback*) langsung dari peserta didik yang sesuai dengan kemampuan dan daya serap yang mereka miliki.
d. *Gathering and reviewing assignment*: hasil yang diperoleh dari evaluasi/ *monitoring* keberhasialan pembelajaran yaitu pemberian nilai atau skor kepada peserta didik dilakukan secara otomatis dan *online*.
e. *Recording grades*: untuk evaluasi peserta didik dalam LMS telah ada fasilitas untuk pemantauan dan perekaman data *grade* dari peserta didik secara otomatis. [6]

### 2.2.2 Algoritma Rabin Karp

Algoritma yang digunakan dalam pembangunan aplikasi pemeriksaan jawaban esai otomatis sebagai pengembangan LMS SMA Negeri 2 Tasikmalaya, adalah algoritma Rabin Karp yang ditambahkan teknik *stemming* dan *synonym recognition.*

Algoritma Rabin-Karp adalah algoritma pencocokan *string* yang menggunakan fungsi *hash* sebagai pembanding antara *string* yang dicari (m) dengan *substring* pada teks (n). Algoritma Rabin-Karp didasarkan pada fakta jika dua buah *string* sama maka harga *hash value*-nya pasti sama. Akan tetapi ada dua masalah yang timbul dari hal ini, masalah pertama yaitu ada begitu banyak *string* yang berbeda, permasalahan ini dapat dipecahkan dengan meng-*assign* beberapa *string* dengan *hash value* yang sama. Masalah yang kedua belum tentu *string* yang mempunyai *hash value* yang sama cocok untuk mengatasinya, maka untuk setiap

*string* yang di-*assign* dilakukan pencocokan *string* secara Brute-Force. Kunci agar algoritma Rabin Karp efisien, terdapat pada pemilihan *hash value*-nya. Salah satu cara yang terkenal dan efektif adalah memperlakukan setiap *substring* sebagai suatu bilangan dengan basis tertentu. [1]

Tahap-tahap yang dilakukan di dalam sistem untuk proses pemeriksaan dan penilaian jawaban esai otomatis yang akan dibangun adalah tahap *preprocessing* (*parsing, filtering*), *synonym recognition*, *stemming* dan algoritma Rabin Karp yang terdiri dari tahap menentukan k-gram, *hashing* dan *similarity*.

**1. *Parsing***

Tahap *parsing* adalah tahap mengubah semua huruf dalam dokumen menjadi huruf kecil dimana hanya huruf "a" sampai dengan "z" yang diterima serta karakter selain huruf dan spasi dihilangkan.

**Contoh *Parsing***

Kalimat awal : Mudah-mudahan hari ini libur dan semoga hari ini cerah.

Hasil proses *parsing* : mudah-mudahan hari ini libur dan semoga hari ini cerah

- Huruf 'M' kapital berubah menjadi huruf 'm' kecil
- Tanda titik '.' dihilangkan

**2. *Filtering***

*Filtering* adalah tahap mengambil kata-kata penting dari hasil *token*, dengan menggunakan *stoplist* / *stopword* (membuang kata yang kurang penting). *Stoplist* / *stopword* adalah kata-kata yang tidak deskriptif.

**Contoh *Filtering***

Beberapa kata yang termasuk dalam daftar *stoplist*:

saya, aku, kamu, kau, kita, mereka, dia, ke, ini, untuk, dimana, dan, yang

Hasil proses *parsing* : mudah-mudahan hari ini libur
dan semoga hari ini cerah
Hasil proses *filtering* : mudah-mudahan hari libur
semoga hari cerah

- Kata 'ini' dan kata 'dan' dihilangkan karena termasuk dalam daftar *stoplist*

Kedua kalimat dipisahkan untuk melihat perbedaan, sehingga menjadi '**mudah-mudahan hari libur'** dan '**semoga hari cerah**'

### 3. *Synonym Recognition*

*Synonym Reognition* atau pengenalan sinonim adalah cara yang digunakan untuk membandingkan suatu dokumen dengan dokumen lainnya dengan mencari kata-kata yang mengandung sinonim untuk mendapatkan tingkat kemiripan yang lebih akurat.

Proses pencocokan sinonim yang dilakukan merupakan keterkaitan antara isi dokumen dengan kamus sinonim yang terdapat di dalam basisdata, dimana proses *synonym recognition* tidak akan dilakukan jika kata tidak tersedia pada kamus sinonim. Seluruh isi dokumen akan discan dan dicocokkan dengan kata yang ada dalam kamus, jika kata tersedia maka akan diubah berdasarkan isi kamus sinonim tersebut. [7]

**Contoh *Synonym Recognition***

Data pada kamus sinonim:

mudah-mudahan = semoga

Hasil proses *filtering* :

Kalimat 1 : mudah-mudahan hari libur
Kalimat 2 : semoga hari cerah

Hasil proses *synonym recognition* :

Kalimat 1 : semoga hari libur
Kalimat 2 : semoga hari cerah

- Kata 'mudah-mudahan' diganti menjadi 'semoga' karena terdapat pada kamus sinonim

**4. *Stemming***

*Stemming* merupakan proses yang digunakan untuk menemukan kata dasar dari sebuah kata dengan menghilangkan semua imbuhan (*affixes*) baik yang merupakan awalan (*prefixes*), sisipan (*infixex*), akhiran (*suffixes*), dan kombinasi dari awalan serta akhiran (*confixes*) pada kata turunan.

Algoritma *Enhanced Confix Stripping* (ECS) merupakan algoritma untuk melakukan proses Stemming. Algoritma ini diperkenalkan oleh Jelita Asian sebagai pengembangan dari algoritma Nazief dan Adriani (1996).

**[DP+[DP + [DP+]]] Kata Dasar [[+DS][+PP][+P]]**

Keterangan :

| | |
|---|---|
| DP (*Derivation Prefixes*) | : kumpulan awalan yang diberikan langsung pada kata dasar |
| DS (*Derivation Suffixes*) | : kumpulan akhiran yang ditambahkan langsung pada kata dasar |
| PP (*Possessive Pronoun*) | : kata ganti kepunyaan |
| *Particle* | : partikel |

Algoritma *Enhanched Confix Stripping* ini mempunyai tahapan proses sebagai berikut :

1) Kata yang hendak di-stemming dicari terlebih dahulu pada kamus. Jika ditemukan, berarti kata tersebut adalah kata dasar, jika tidak maka langkah 2 dilakukan.
2) Cek *rule precedence*. Apabila suatu kata memiliki pasangan awalan-akhiran "be-lah", "be-an", "me-i", "di-i", "pe-i", atau "te-i" maka langkah stemming selanjutnya adalah (5, 3, 4, 6). Apabila kata tidak memiliki pasangan awalan-akhiran tersebut, langkah stemming berjalan normal (3, 4, 5, 6).
3) Hilangkan *inflectional particle* P ("-lah", "-kah", "-tah", "-pun") dan kata ganti kepunyaan atau possessive pronoun PP ("-ku", "-mu", "-nya").
4) Hilangkan *Derivation Suffixes* DS ("-i", "-kan", atau "-an").

5) Hilangkan *Derivational Prefixes* DP {“di-”,“ke-”,“se-”,“me-”,“be-”,“pe”, “te-”}.
   a) 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.
   b) Cari kata yang telah dihilangkan awalannya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 5 diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan.
6) Jika semua gagal, maka masukan kata yang diuji pada algoritma ini dianggap sebagai kata dasar. [8]

**Contoh *Stemming***

Hasil proses *filtering* :

Kalimat 1 : semoga hari libur
Kalimat 2 : semoga hari cerah

Hasil proses *stemming* :

Kalimat 1 : semoga hari libur
Kalimat 2 : semoga hari cerah

- Kedua kalimat tidak berubah karena tidak ditemukan kata bentukan yang akan diubah ke dalam bentuk kata dasarnya setelah proses *stemming*

## 5. K-gram

K-gram adalah rangkaian *terms* dengan panjang K. Kebanyakan yang digunakan sebagai terms adalah kata. K-gram merupakan sebuah metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode k-grams ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah k dari sebuah kata yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen. [9]

**Contoh K-gram**

Misalkan nilai k-gram adalah 4.

Hasil proses *stemming* :

Kalimat 1 : semoga hari libur

Kalimat 2 : semoga hari cerah

Hasil proses *synonym recognition* :

Kalimat 1 : $\underset{1}{\{\text{semo}\}}, \underset{2}{\{\text{emog}\}}, \underset{3}{\{\text{moga}\}}, \underset{4}{\{\text{oga }\}}, \underset{5}{\{\text{ga h}\}}, \underset{6}{\{\text{a ha}\}}, \underset{7}{\{\text{ har}\}}, \underset{8}{\{\text{hari}\}},$
$\underset{9}{\{\text{ari }\}}, \underset{10}{\{\text{ri l}\}}, \underset{11}{\{\text{i li}\}}, \underset{12}{\{\text{ lib}\}}, \underset{13}{\{\text{libu}\}}, \underset{14}{\{\text{ibur}\}}$

Terdapat 14 k-grams.

Kalimat 2 : $\underset{1}{\{\text{semo}\}}, \underset{2}{\{\text{emog}\}}, \underset{3}{\{\text{moga}\}}, \underset{4}{\{\text{oga }\}}, \underset{5}{\{\text{ga h}\}}, \underset{6}{\{\text{a ha}\}}, \underset{7}{\{\text{ har}\}}, \underset{8}{\{\text{hari}\}},$
$\underset{9}{\{\text{ari }\}}, \underset{10}{\{\text{ri c}\}}, \underset{11}{\{\text{i ce}\}}, \underset{12}{\{\text{ cer}\}}, \underset{13}{\{\text{cera}\}}, \underset{14}{\{\text{erah}\}}$

Terdapat 14 k-grams.

**6. *Hashing***

*Hashing* adalah suatu cara untuk mentransformasi sebuah string menjadi suatu nilai yang unik dengan panjang tertentu (*fixed-length*) yang berfungsi sebagai penanda *string* tersebut. Fungsi untuk menghasilkan nilai ini disebut fungsi *hash*, sedangkan nilai yang dihasilkan disebut nilai *hash*. Contoh sederhana *hashing* adalah:

Firdaus, Hari

Munir, Rinaldi

Rabin, Michael

Karp, Richard

menjadi :

7864 = Firdaus, Hari

9802 = Munir, Rinaldi

1990 = Rabin, Michael

8822 = Karp, Richard

Contoh di atas adalah pengunaan *hashing* dalam pencarian pada *database*. Apabila tidak di-*hash*, pencarian akan dilakukan karakter per karakter pada nama-nama yang panjangnya bervariasi dan ada 26 kemungkinan pada setiap karakter. Namun pencarian akan menjadi lebih efisien setelah di-*hash* karena hanya akan membandingkan empat digit angka dengan hanya 10 kemungkinan setiap angka. Nilai *hash* pada umumnya digambarkan sebagai *fingerprint*, yaitu suatu *string* pendek yang terdiri atas huruf dan angka yang terlihat acak (data *biner* yang ditulis dalam heksadesimal). [1]

*Rolling hash* merupakan fungsi yang digunakan untuk menghasilkan nilai *hash* dari rangkaian gram. Fungsi ini digunakan pada algoritma Rabin Karp untuk membandingkan nilai *hash* dari semua k-gram, dimana setiap nilai *hash* yang sama mempunyai isi *string* yang sama. Akan tetapi proses *hashing* pada setiap *string* sepanjang k akan menghabiskan waktu komputasi yang lama jika nilai k besar. Rabin Karp menggunakan *Rolling Hash* dimana $H(c_1...c_k)$ didefinisikan pada *Persamaan 2.1*.

$$H_{(c1)} = c_1 * b^{k-1} + c_2 * b^{k-2} * ... + c_{k-1} * b + c_k \qquad (2.1)$$

Keterangan:

c : nilai ascii karakter

b : basis (11)

k : banyak karakter

Untuk mendapatkan nilai *hash* gram berikutnya $H(c_2...c_{k+1})$ dapat dilakukan dengan menggunakan *Persamaan 2.2*.

$$H_{(c_2 \ldots c_{k+1})} = \left(H_{(c_2 \ldots c_k)} - c_1 * b^{(k-1)}\right) * b + c_{(k+1)} \quad (2.2)$$

Dengan begitu tidak perlu melakukan iterasi dari indeks pertama sampai terakhir untuk menghitung nilai *hash* untuk gram ke-2 sampai terakhir, sehingga dapat menghemat waktu komputasi saat menghitung nilai *hash* dari sebuah gram. [10]

Contoh penggunaan persamaan 2.1 dan persamaan 2.2 untuk menentukan gram pertama dan gram kedua dari kalimat pertama.

a. K-grams : {semo}, {emog}
b. Banyak karakter : 4 (sesuai nilai k-gram)
c. Basis yang digunakan : 11
d. Nilai ASCII : ‘s’ = 115
   ‘e’ = 101
   ‘m’ = 109
   ‘o’ = 111
   ‘g‘ = 103

Untuk menghitung nilai *hash* k-gram pertama, digunakan Persamaan 2.1.

K-gram : {kami}

c = 115, 101, 109, 111, 103

b = 11

k = 4

$$115 * 11^{4-1} + 101 * 11^{3-1} + 109 * 11^{2-1} + 111 = \underline{166596}$$

Untuk menghitung nilai *hash* k-gram kedua sampai k-gram ke-n, dapat digunakan Persamaan 2.2.

K-gram : {ami }

c = 115, 101, 109, 111, 103

b = 11

k = 4

$$(166596 - 115 * 11^{4-1}) * 11 + 103 = \underline{148944}$$

**Contoh *Hashing***

Hasil proses k-gram :

Kalimat 1 : $\frac{\{semo\}}{1}, \frac{\{emog\}}{2}, \frac{\{moga\}}{3}, \frac{\{oga\ \}}{4}, \frac{\{ga\ h\}}{5}, \frac{\{a\ ha\}}{6}, \frac{\{\ har\}}{7}, \frac{\{hari\}}{8},$
$\frac{\{ari\ \}}{9}, \frac{\{ri\ l\}}{10}, \frac{\{i\ li\}}{11}, \frac{\{\ lib\}}{12}, \frac{\{libu\}}{13}, \frac{\{ibur\}}{14}$

Kalimat 2 : $\frac{\{semo\}}{1}, \frac{\{emog\}}{2}, \frac{\{moga\}}{3}, \frac{\{oga\ \}}{4}, \frac{\{ga\ h\}}{5}, \frac{\{a\ ha\}}{6}, \frac{\{\ har\}}{7}, \frac{\{hari\}}{8},$
$\frac{\{ari\ \}}{9}, \frac{\{ri\ c\}}{10}, \frac{\{i\ ce\}}{11}, \frac{\{\ cer\}}{12}, \frac{\{cera\}}{13}, \frac{\{erah\}}{14}$

Hasil proses *hashing* :

Kalimat 1 : $\frac{166596}{1}, \frac{148944}{2}, \frac{159740}{3}, \frac{161303}{4}, \frac{149286}{5}, \frac{134220}{6},$
$\frac{56357}{7}, \frac{151520}{8}, \frac{144088}{9}, \frac{164899}{10}, \frac{144920}{11}, \frac{56913}{12},$
$\frac{157648}{13}, \frac{153015}{14}$

Terdapat 14 nilai *hash*.

Kalimat 2 : $\frac{166596}{1}, \frac{148944}{2}, \frac{159740}{3}, \frac{161303}{4}, \frac{149286}{5}, \frac{134220}{6},$
$\frac{56357}{7}, \frac{151520}{8}, \frac{144088}{9}, \frac{164890}{10}, \frac{144817}{11}, \frac{55796}{12},$
$\frac{145341}{13}, \frac{149396}{14}$

Terdapat 14 nilai *hash*.

**7. *Similarity***

*Similarity* adalah proses menghitung tingkat kemiripan dua objek dengan merepresentasikan dua set objek yang dibandingkan dalam bentuk *probability*. Pengukuran nilai *similarity* dilakukan menggunakan fungsi *Dice's Similarity Coefficients* yang menggunakan pendekatan k-gram yang ditunjukan pada persamaan 2.3. [11]

$$S = \frac{2*C}{(A+B)} * 100\% \quad (2.3)$$

Keterangan :

S : nilai *similarity*

C : banyak nilai *hash* yang sama dari teks yang dibandingkan

A : banyak nilai *hash* pada teks 1

B : banyak nilai *hash* pada teks 2

**Contoh *Similarity***

Berdasarkan nilai *hash* kedua kalimat dari hasil proses *hashing* sebelumnya, dapat dicari nilai *hash* yang sama.

1. Nilai *hash* kalimat pertama (14 nilai) :
   166596, 148944, 159740, 161303, 149286, 134220, 56357, 151520, 144088
   164899, 144920, 56913, 157648, 153014
2. Nilai *hash* kalimat kedua (14 nilai) :
   166596, 148944, 159740, 161303, 149286, 134220, 56357, 151520, 144088
   164890, 144817, 55796, 145341, 149396
3. Nilai *hash* yang sama (9 nilai) :
   $\frac{166596}{1}, \frac{1148944}{2}, \frac{159740}{3}, \frac{161303}{4}, \frac{149286}{5}, \frac{134220}{6}, \frac{56357}{7}, \frac{151520}{8}, \frac{144088}{9}$

4. Nilai kemiripan antara kalimat pertama dan kedua dapat dicari dengan menggunakan Persamaan 2.3.

   C = 9

   A = 14

   B = 14

$$S = \frac{2 * 9}{(14 + 14)} * 100\% = 64{,}29\%$$

### 2.2.3 Analisis dan Perancangan Terstruktur

Analisis dan perancangan terstruktur menurut Whitten dan Bentley [12] adalah suatu teknik analisis sistem yang menekankan pada proses yang terjadi dalam sistem, dan digunakan untuk memodelkan kebutuhan-kebutuhan sistem. Model analisis terstruktur menggambarkan proses, *input*, *output* dan dokumen-dokumen yang diperlukan pada suatu proses dalam sistem.

Pemodelan sistem menggunakan analisis terstruktur menurut Mynatt [13] terdiri dari tiga komponen yaitu, *data flow diagram* (DFD), kamus data dan spesifikasi proses.

#### 2.2.3.1 *Data Flow Diagram* (DFD)

*Data Flow Diagram* menurut Kendal [14] adalah grafik yang menggambarkan pandangan sejauh mungkin mengenai *input*, proses dan *output* sistem, yang berhubungan dengan *input*, proses dan *output* dari model sistem secara umum. Ketentuan-ketentuan dalam penggambaran *DFD* yaitu :

a. Di antara entitas-entitas luar tidak diperbolehkan ada aliran data secara Langsung
b. Tidak diperbolehkan adanya aliran data secara langsung antara entitas luar dengan penyimpanan data
c. Tidak diperbolehkan suatu proses hanya memiliki aliran data masuk atau aliran data keluar saja. Proses-proses harus memiliki sedikitnya satu aliran data masuk dan satu aliran data keluar
d. Setiap proses harus dapat mentransformasikan data. Pada setiap proses harus dapat menerima *input* dan mengeluarkan *output*
e. Di antara penyimpanan-penyimpanan data tidak diperbolehkan adanya aliran data secara langsung.

#### 2.2.3.2 Kamus Data

Kamus data menurut Kendall [14] adalah suatu aplikasi khusus dari jenis kamus-kamus yang digunakan sebagai referensi kehidupan setiap hari. Sebagai suatu dokumen, kamus data berguna untuk mengumpulkan dan mengkoordinasi

istilah-istilah tertentu, dan menjelaskan apa arti setiap istilah yang ada. *DFD* merupakan satu titik awal yang baik untuk mengumpulkan elemen-elemen data.

#### 2.2.3.3 Spesifikasi Proses

Spesifikasi proses menurut Kendall [14] berfungsi untuk menjelaskan logika pembuatan keputusan dan rumusan yang mentransformasikan proses data *input* menjadi data *output*.

Salah satu metode untuk menyusun spesifikasi proses menurut Mynatt [13] adalah menggunakan *pseudocode* terstruktur. *Pseudocode* memiliki format mirip dengan sintaks bahasa pemrograman, tetapi dengan aturan penulisan tidak formal. *Pseudocode* dibangun menggunakan proses-proses dasar dari bahasa terstruktur. Bahasa terstruktur yang digunakan adalah :

a. Urutan, menyatakan urutan kegiatan pada suatu proses. Setiap baris urutan menyatakan satu kegiatan.
b. Keputusan, dinyatakan dalam kalimat bersyarat.

### 2.2.4 Dreamweaver

Macromedia dreamweaver merupakan program *editor* HTML *professional* untuk mendesain, mengkode, dan mengembangkan situs, halaman situs, serta aplikasi situs. Dreamweaver juga dapat membantu dalam membangun *database* untuk aplikasi situs dengan menggunakan *server* seperti ASP, ASPNet, *ColdFusion Markup Language* (CFML), JSP dan PHP.

### 2.2.5 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (localhost), yang terdiri atas program *Apache HTTP Server, MySQL database,* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache, MySQL, PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan web server yang mudah

digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkanya dapat mendownload langsung dari web resminya.

### 2.2.6 Microsoft Visio 2013

Microsoft Visio (atau sering disebut Visio) merupakan sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir (*flowchart*), *brainstorm*, dan skema jaringan yang dirilis oleh Microsoft Corporation. Aplikasi ini menggunakan grafik vektor untuk membuat diagram-diagramnya.

Microsoft Visio 2013 terdiri dari 3 versi, yaitu Microsoft Visio Standar 2013, Microsoft Visio Professional 2013, Visio Premium 2013. Microsoft Visio Standar 2013 merupakan edisi visio yang standar dari semua versi visio yang ada. Fitur pada visio standar dan dimiliki oleh edisi Visio 2013 lainnya membawa tampilan yang baru dengan sudah menggunakan Microsoft Fluent UI dan *shape-shape* yang didesain lebih menarik. Selain itu fitur Visio standar 2013 yang juga dimiliki oleh edisi Visio lainnya antara lain *Quick Shapes*, *Page Auto Size*, *Dynamic Grid*, *Improved Cross Functional Flowchart*.

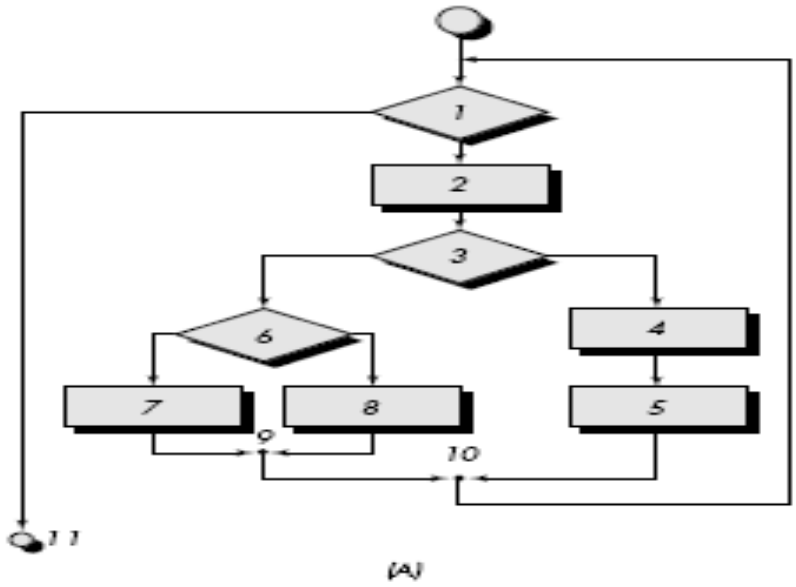
Microsoft Visio Professional 2013 merupakan edisi Visio yang dibangun dari Visio standar. Semua fitur yang ada di Visio standar dimiliki juga oleh Visio Professional ini. Penambahan edisi professional dari sisi *template*, data *feature* dan Visio *service*. Penambahan *template* antara lain adanya *detailed network diagram*, *engineering diagram*, *software and database diagram*, *wireframe diagram*, *ITIL diagram*, *website mapping and documentation diagram*, dan *building architecture* dan *service diagram*.

Microsoft Visio Premium 2013 merupakan edisi dari Visio yang sudah mencakup Visio versi standar dan Visio Professional dengan menambahkan fitur *advance proccess management* seperti *template* untuk *SharePoint Workflow*, *Business Process Modelling Notation* (BPMN) dan *Six Sigma*. *SharePoint Workflow* dapat diekspor ke *SharePoint Designer* yang dapat disesuaikan.fitur lainnya yaitu diagram *validation*, dengan *diagram validation* dapat memastikan bahwa diagram yang kita buat dapat di validasi.

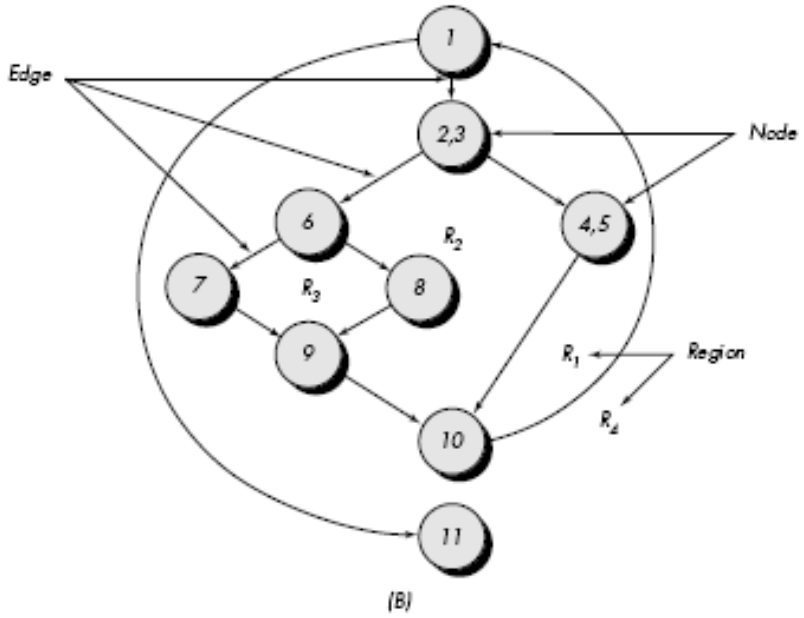
### 2.2.7 Pengujian *White-Box*

Menurut Rizky [15], *White Box Testing* merupakan jenis testing yang lebih berkonsentrasi terhadap "isi" dari perangkat lunak itu sendiri. Jenis ini lebih banyak berkonsentrasi kepada *source code* dari perangkat lunak yang dibuat. Pada *White Box Testing*, menurut Lewis [16] kondisi pengujian dirancang dengan menguji alur dari logika. Penguji menguji struktur internal dari program atau sistem. Data pengujian dilakukan berdasarkan pengujian logika dari program atau sistem, tanpa memperhatikan persyaratan program atau sistem. Dari definisi tersebut, *white box testing* adalah pengujian terhadap cara kerja sebuah sistem secara internal dan struktural untuk mengidentifikasikan kesalahan alur dari logika yang mungkin terhadi ketika sistem dijalankan.

*White Box Testing* dapat dilakukan dengan menggunakan pengujian *Basis Path*. *Basis Path* memungkinkan perancang kasus uji untuk mengukur kompleksitas logis dari rancangan prosedural dan menggunakannya sebagai pedoman untuk menetapkan sekumpulan jalur eksekusi dasar (basis set). *Basis Path* menggunakan *flowchart* atau notasi *flowgraph* yang menggambarkan aliran kontrol logika seperti yang ditunjukan pada gambar 2.3 dan gambar 2.4.




Gambar 2.3 Flowchat Pengujian Basis Path




Gambar 2.4 Flowgraph Pengujian Basis Path

Untuk menentukan banyaknya jalur independen pada *Basis Path* dapat digunakan "Kompleksitas Siklomatis". Jalur independen adalah jalur yang melalui program yang menghasilkan sedikitnya satu rangkaian statemen proses baru atau suatu kondisi baru. Contoh serangkaian jalur independen untuk *flowgraph* yang ditunjukan pada gambar 2.4.

Jalur 1 : 1-11
Jalur 2 : 1-2-3-4-5-10-1-11
Jalur 3 : 1-2-3-6-8-9-10-1-11
Jalur 4 : 1-2-3-6-7-9-10-1-11

Dapat dilihat bahwa masing-masing jalur baru memperkenalkan sebuah *edge* baru, misalkan jalur 5.

Jalur 5 : 1-2-3-4-5-10-1-2-3-6-8-9-10-1-11

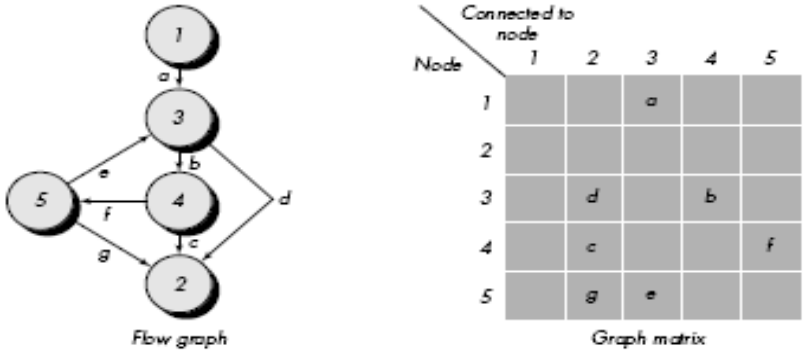
Jalur 5 tidak dianggap jalur independen karena merupakan gabungan dari jalur-jalur yang telah ditentukan dan tidak melewati *edge* baru.

Kompleksitas siklomatis dihitung menggunakan salah satu dari ketiga cara berikut ini:

1. Jumlah *region flowgraph*

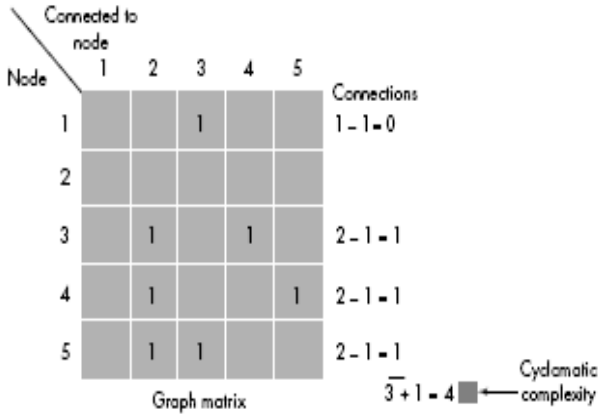
2. Kompleksitas siklomatis V(G) untuk *flowgraph* G dihitung sebagai "V(G) = E-N+2" di mana E adalah jumlah *edge* dan N adalah jumlah *node*.

Untuk membantu pengujian *Basis Path*, dapat digunakan struktur data berbentuk matriks yang disebut sebagai matriks *graph*. Matriks *graph* merupakan matriks bujur sangkar yang ukurannya sesuai dengan jumlah node pada flow graph. Masing-masing baris dan kolom sesuai dengan node masing-masing. Contoh gambaran dari matriks *graph* ditunjukan oleh gambar 2.5.




Gambar 2.5 Matriks Graph

Jika isi dari matriks diganti dengan *link weight*, dan direpresentasikan dengan angka 1 dan 0 atau kosong, dimana angka 1 jika ada *link* dan 0 atau kosong jika tidak ada (matriks dengan bentuk seperti ini sering disebut matriks koneksi), maka akan didapatkan metode lain untuk menghitung kompleksitas siklomatis. Contoh matriks koneksi ditunjukan gambar 2.6.




Gambar 2.6 Matriks Graph (Matriks Koneksi)

### 2.2.8 Pengujian *Black-Box*

Menurut Rizky [15], *Black Box Testing* adalah tipe pengujian yang memperlakukan perangkat lunak yang tidak diketahui kinerjanya pada bagian *internal*. Para *tester* memandang perangkat lunak sebagai "kotak hitam" yang tidak penting di lihat isinya, tapi cukup dikenali proses *testing* di bagian luar, dimana *testing* ini banyak berkonsentrasi pada bagaimana proses berjalannya perangkat lunak. Menurut Lewis [16], pada *black box testing* atau pengujian fungsional, kondisi pengujian dikembangkan berdasarkan fungsionalitas dari program atau sistem yang akan diuji, oleh karena itu, penguji membutuhkan informasi mengenai data input dan output yang diamati, tetapi tidak mengetahui bagaimana program atau sistem tersebut bekerja atau tidak perlu mengetahui bagaimana struktur internal dari program tersebut melakukan eksekusi. Penguji berfokus pada pengujian fungsionalitas dari program terhadap spesifikasi.

## 2.3 *State of The Art*

Pada *state of the art* ini, diambil contoh penelitian terdahulu sebagai panduan ataupun contoh untuk penelitian yang dilakukan yang nantinya akan menjadi acuan dan perbandingan dalam melakukan penelitian ini. Dalam *state of the art* ini terdapat 5 contoh penelitian sebelumnya yang ditunjukan tabel 2.1 hingga tabel 2.12.

Tabel 2.1 State of The Art (1)

| | |
|---|---|
| Judul | Sistem Koreksi Soal Essay Otomatis dengan Menggunakan Metode Rabin Karp |
| Penulis | Sahriar Hamza, M. Sarosa, Purnomo Budi Santoso |
| Hasil Pengujian | Uji coba yang dilakukan di dalam penelitian ini dilakukan sebanyak tiga kali untuk membandingkan hasil jawaban sistem dan jawaban guru dalam mengoreksi jawaban ujian siswa. Hasil yang didapat dari tiga uji coba ini, menunjukan perbedaan nilai rata-rata hasil ujian sistem dan guru masing-masing adalah 0,05%, 0,07% dan 0,01%. Berdasarkan hasil uji coba tersebut, dapat disimpulkan bahwa sistem mampu mengoreksi jawaban siswa dengan baik, dengan range perbedaan rata-rata nilai dan guru antara 0,01%-0,07 %. |
| Kesimpulan | Kesimpulan yang dapat diambil dari penelitian tersebut adalah metode Rabin Karp dapat melakukan proses pengoreksian jawaban dengan cukup baik karena dapat menemukan kecocokan jawaban antara jawaban siswa dan jawaban guru berdasarkan nilai k-gram dan hashing dari proses metode Rabin Karp tersebut, dimana range perbedaan rata-rata nilai dan giru adalah 0,01%-0,07%. Kekurang dari penelitian ini adalah tidak dijelaskannya proses stemming yang dilakukan, apakah menggunakan sebuah algoritma stemmer atau hanya menggunakan kamus kata stem pada database, |

Tabel 2.2 State of The Art (1) (Lanjutan)

| | |
|---|---|
| Judul | Sistem Koreksi Soal Essay Otomatis dengan Menggunakan Metode Rabin Karp |
| Kesimpulan | kemudian dalam proses pengoreksiannya tidak digunakan bobot nilai untuk setiap jawaban. Dalam penelitian ini juga tidak dilakukan proses pengenalan sinonim (synonym recognition) yang dapat digunakan untuk mencari persamaan kata dari kedua jawaban berdasarkan kamus sinonim kata pada database. Saran yang dapat diberikan adalah pada proses stemming-nya untuk digunakan suatu algoritma stemmer yang ada, kemudian diberikan bobot nilai untuk setiap jawaban siswa dan ditambahkan proses atau tahap synonym recognition untuk pengenalan kata dari jawaban siswa pada kamus sinonim. |

Tabel 2.3 State of The Art (2)

| | |
|---|---|
| Judul | Deteksi Duplikat Dokumen Menggunakan Algoritma Rabin Karp |
| Penulis | Hari Bagus Firdaus |
| Hasil Pengujian | Uji coba yang dilakukan di dalam penelitian ini dilakukan dengan membandingkan dua buah, yaitu dokumen mahasiswa dengan sebuah dokumen dari suatu situs untuk mengetahui apakah dokumen mahasiswa tersebut plagiat atau tidak. Proses yang dilakukan yaitu dengan melakukan proses filtering terhadap kedua dokumen, kemudian dilanjutkan dengan pencarian nilai hash dengan pemenggalan teks sebanyak 4 karakter. Hasil dari pengujian yang dilakukan menunjukan bahwa sebagian besar isi pada dokumen mahasiswa mengambil dari dokumen asli dari suatu situs tersebut. |

Tabel 2.4 State of The Art (2) (Lanjutan)

| | |
|---|---|
| Judul | Deteksi Duplikat Dokumen Menggunakan Algoritma Rabin Karp |
| Kesimpulan | Kesimpulan yang dapat diambil dari penelitian tersebut adalah Rabin Karp dapat digunakan untuk mencari kemiripan dari dua dokumen atau lebih, sehingga dapat dimanfaatkan untuk melakukan pendeteksian terhadap tindak plagiat dokumen. Kekurang dari penelitian ini adalah tidak ada tahap preprocessing untuk parsing yang berfungsi untuk mengubah semua huruf kapital menjadi huruf kecil dan menghilangkan karakter selain huruf, dan tidak adanya tahap stemming yang dapat mengubah kata bentukan ke dalam bentuk kata dasarnya, sehingga hasil yang diberikan tidak terlalu optimal. Dalam penelitian ini juga tidak dilakukan proses pengenalan sinonim (synonym recognition) yang dapat digunakan untuk mencari persamaan kata berdasarkan kamus sinonim kata pada database yang dapat memberikan hasil akurasi kemiripan yang lebih baik untuk membandingkan dua buah dokumen atau lebih. Saran yang dapat diberikan adalah dilakukannya proses parsing pada tahap preprocessing dan proses stemming agar hasil berupa tingkat kemiripan yang diberikan antara dua dokumen lebih optimal, dan dengan menambahkan proses atau tahap synonym recognition untuk pengenalan kata di dalam dokumen berdasarkan kamus sinonim. |

Tabel 2.5 State of The Art (3)

| | |
|---|---|
| Judul | Implementasi Algoritma Rabin Karp untuk Pendeteksian Plagiat Dokumen Teks Menggunakan Konsep Similarity |
| Penulis | Salmuasih, Andi Sunyoto |

Tabel 2.6 State of The Art (3) (Lanjutan)

| | |
|---|---|
| Judul | Implementasi Algoritma Rabin Karp untuk Pendeteksian Plagiat Dokumen Teks Menggunakan Konsep Similarity |
| Hasil Pengujian | Uji coba yang dilakukan di dalam penelitian ini dilakukan dengan membandingkan empat buah file uji, dengan melakukan empat empat kali jenis pengujian yang berbeda, yaitu untuk mendapatkan hasil uji berdasarkan banyaknya konten file, hasil uji tanpa menggunakan stemming, hasil uji modul pada algoritma Rabin Karp, dan hasil uji k-gram pada algoritma Rabin Karp . Hasil dari pengujian yang pertama adalah bahwa semakin banyak konten file, maka semakin lama waktu proses, hasil dari pengujian kedua adalah jika tanpa menggunakan proses stemming akan mempercepat waktu proses, tetapi akurasi yang diberikan rendah, hasil pengujian yang ketiga menunjukan bahwa modulo tidak berpengaruh pada prosentase similarity, tetapi berpengaruh pada waktu proses, dan hasil pengujian yang keempat menunjukan bahwa semakin kecil k-gram, akurasi similarity-nya semakin tinggi. |
| Kesimpulan | Kesimpulan yang dapat diambil dari penelitian tersebut adalah Rabin Karp dapat digunakan untuk membandingkan file dan memberikan hasil berupa prosentase similarity, dan terdapat beberapa faktor yang dapat mempengaruhi performansi algoritma Rabin Karp dalam membandingkan file, yaitu banyaknya konten file akan memperpanjang waktu prosesnya, stemming dan preprocessing membuat waktu proses yang lebih lama, tetapi akurasi yang diberikan lebih tinggi, modulo hanya berpengaruh pada waktu proses, dan semakin kecil nilai k-gram akan menghasilkan tingkat kemiripan (similarity) yang lebih baik. Kekurang dari penelitian ini adalah tidak dijelaskannya proses stemming yang dilakukan, |

Tabel 2.7 State of The Art (3) (Lanjutan)

| | |
|---|---|
| Judul | Implementasi Algoritma Rabin Karp untuk Pendeteksian Plagiat Dokumen Teks Menggunakan Konsep Similarity |
| Kesimpulan | apakah menggunakan sebuah algoritma stemmer atau hanya menggunakan kamus kata stem pada database. Dalam penelitian ini juga tidak dilakukan proses pengenalan sinonim (synonym recognition) yang dapat digunakan untuk mencari persamaan kata berdasarkan kamus sinonim kata pada database yang dapat memberikan hasil akurasi kemiripan yang lebih baik untuk membandingkan dua buah dokumen atau lebih. Saran yang dapat diberikan adalah pada proses stemming-nya untuk digunakan suatu algoritma stemmer yang ada, dan menambahkan proses atau tahap synonym recognition untuk pengenalan kata di dalam dokumen dengan menggunakan kamus sinonim. |

Tabel 2.8 State of The Art (4)

| | |
|---|---|
| Judul | Sistem Pengukuran Tingkat Similaritas Dokumen Menggunakan Algoritma Rabin Karp dan Enhanched Confix Stripping Stemmer |
| Hasil Pengujian | Uji coba yang dilakukan di dalam penelitian ini dilakukan dengan membandingkan beberapa dokumen dan teks, dengan melakukan empat empat kali jenis pengujian yang berbeda, yaitu untuk mendapatkan hasil uji menggunakan modulo, hasil uji k-gram, hasil uji banyaknya konten dengan k-gram 4, dan hasil uji akurasi dokumen. Hasil dari pengujian yang pertama adalah bahwa modulo tidak berpengaruh pada prosentase similarity, tetapi berpengaruh pada waktu proses, hasil dari pengujian kedua adalah jika semakin kecil k-gram, |

Tabel 2.9 State of The Art (4) (Lanjutan)

| | |
|---|---|
| Judul | Sistem Pengukuran Tingkat Similaritas Dokumen Menggunakan Algoritma Rabin Karp dan Enhanched Confix Stripping Stemmer |
| Hasil Pengujian | akurasi similarity semakin tinggi, dimana penggunaan k-gram yang dipakai antara k-gram 3 sampai k-gram 8. Pada uji coba hasil pengujian ini, k-gram 1 dan k-gam 2 memberikan hasil similarity melebihi 100%, sedangkan kedua dokumen yang dibandingkan berbeda, kemudian tidak digunakannya k-gram 9 dan seterusnya adalah karena hasil similarity yang diberikan terlalu kecil, sehingga k-gram yang digunakan pada sistem pengecekan similaritas ini adalah k-gram 4. Hasil pengujian yang ketiga menunjukan bahwa semakin banyak konten, maka semakin lama waktu proses yang dibutuhkan, dan jika tanpa stemming, waktu pemrosesan akan lebih cepat, tetapi hasil similarity kurang akurat. Hasil pengujian yang keempat menunjukan bahwa hasil similarity sudah menunjukan nilai yang sesuai dengan dokumen uji sebenarnya. |
| Kesimpulan | Kesimpulan yang dapat diambil dari penelitian tersebut adalah metode Rabin Karp dapat digunakan untuk membandingkan file dan memberikan hasil berupa prosentase similarity yang cukup akurat, performansi similaritas dokumen dengan k-gram 4 sudah cukup untuk memproses dan menjalankan fungsinya dengan baik, Rabin Karp mempunyai waktu proses similaritas yang lebih baik karena tidak melakukan pengulangan proses, dan stemming dengan menggunakan ECS Stemmer dapat menangani perbedaan kata dengan kata dasar yang sama sehingga dapat meningkatkan keakuratan prosentase similarity, meskipun waktu proses yang dibutuhkan sedikit lebih lama. |

Tabel 2.10 State of The Art (4) (Lanjutan)

| | |
|---|---|
| Judul | Sistem Pengukuran Tingkat Similaritas Dokumen Menggunakan Algoritma Rabin Karp dan Enhanched Confix Stripping Stemmer |
| Kesimpulan | Selain itu, terdapat beberapa faktor yang dapat mempengaruhi performansi algoritma Rabin Karp dalam membandingkan file, yaitu banyaknya konten file akan memperpanjang waktu prosesnya, penggunaan stemming akan berpengaruh pada waktu proses dan keakuratan nilai similarity, modulo hanya berpengaruh pada waktu proses. Kekurang dari penelitian ini adalah tidak adanya proses pengenalan sinonim (synonym recognition) yang dapat digunakan untuk mencari persamaan kata berdasarkan kamus sinonim kata pada database yang dapat memberikan hasil akurasi kemiripan yang lebih baik untuk membandingkan dua buah dokumen atau lebih. Saran yang dapat diberikan adalah dengan menambahkan proses atau tahap synonym recognition untuk pengenalan kata di dalam dokumen dengan menggunakan kamus sinonim. |

Tabel 2.11 State of The Art (5)

| | |
|---|---|
| Judul | Sistem Penilaian Otomatis Jawaban Esai Menggunakan Algoritma Levenshtein Distance |
| Penulis | Muhammad Zakiya Nafik, Indriati, S.T., M.Kom, Achmad Ridok M.Kom |
| Hasil Pengujian | Uji coba yang dilakukan di dalam penelitian ini dilakukan dengan mengujikan lima buah soal kepada dua puluh siswa dengan menggunakan preprocessing, dengan hasil rata-rata kesalahan proses sistem yang didapat mencapai 38,50% untuk penilaian di dalam sistem dengan penilaian manual. |

Tabel 2.12 State of The Art (5) (Lanjutan)

| | |
|---|---|
| Judul | Sistem Penilaian Otomatis Jawaban Esai Menggunakan Algoritma Levenshtein Distance |
| Kesimpulan | Kesimpulan yang dapat diambil dari penelitian tersebut adalah selisih yang jauh dari hasil yang diberikan dikarenakan beberapa faktor, yaitu sistem tidak dapat mendeteksi sinonim kata, sistem masih belum bias mendeteksi makna kalimat negatif dan positif dengan menggunakan antonim, selain itu dikarenakan algoritma Levenshtein Distance yang digunakan pada sistem pemeriksaan jawaban esai ini menggunakan konsep mencari keyword dari kunci jawaban dan jawaban siswa dan kemudian mencocokannya, maka kata yang ada pada kunci jawaban dan jawaban siswa tersebut harus mempunyai kemiripan kata 100%. Kekurang dari penelitian ini salah satunya adalah tidak adanya proses pengenalan sinonim (synonym recognition) yang dapat digunakan untuk mencari persamaan kata dari kedua jawaban berdasarkan kamus sinonim kata pada database yang dapat memberikan hasil akurasi kemiripan yang lebih baik untuk membandingkan dua buah jawaban. Saran yang dapat diberikan adalah dengan menambahkan proses atau tahap synonym recognition untuk pengenalan kata di dalam setiap jawaban dengan menggunakan kamus sinonim sehingga diharapkan dapat memberikan hasil kemiripan jawaban yang lebih baik. |

{HALAMAN INI DIKOSONGKAN}