**PROPOSAL**

## PENERAPAN METODE BM25 PADA SISTEM REKOMENDASI

**DOSEN PEMBIMBING DAN PENGUJI TUGAS AKHIR MAHASISWA BERBASIS WEB.**

**(Studi Kasus : Jurusan Teknik Informatika Universitas Halu Oleo)**

### Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Teknik



**SAFRIL**

**E1E1 16 031**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HALUOLEO**

### KENDARI

#### 2020

# DAFTAR ISI

### HALAMAN JUDUL ..................................................................................... i HALAMAN PENGESAHAN ....................................................................... ii HALAMAN PERNYATAAN ....................................................................... iv INTISARI ......................................................................................................... v

**ABSTRAK ..................................................................................................... vi**

[HALAMAN PERSEMBAHAN vii](#_Toc165418)

[KATA PENGANTAR x](#_Toc165419)

[DAFTAR ISI xii](#_Toc165420)

[DAFTAR GAMBAR xv](#_Toc165421)

**DAFTAR TABEL........................................................................................... xvii**

### BAB I PENDAHULUAN ................................................................................. 1

1.1 Latar Belakang ................................................................................. 1

1.2 Rumusan Masalah ............................................................................ 2

1.3 Batasan Masalah ............................................................................... 3

1.4 Tujuan Penelitian .............................................................................. 3

1.5 Manfaat Penelitian ............................................................................ 3

1.6 Sistematika Penulisan ....................................................................... 4

1.7 Tinjauan Pustaka .............................................................................. 4

### BAB II LANDASAN TEORI .......................................................................... 6

2.1 Tugas Akhir ...................................................................................... 6

2.1.1 Penetapan Pembimbing Skripsi *................................................* 6

2.1.2 Seminar Proposal *......................................................................* 7

2.1.3 Seminar Hasil *...........................................................................* 8

2.1.4 Seminar Skripsi *........................................................................* 9

2.2 Dosen Pembimbing .......................................................................... 10

2.3 Dosen Penguji .................................................................................. 10

2.4 Konsep Sistem Informasi ................................................................. 10

2.4.1 Defenisi Sistem *........................................................................* 10

2.4.2 Defenisi Informasi *....................................................................* 12

2.4.3 Defenisi Sistem Informasi *........................................................* 14

2.5 *Text Mining* ....................................................................................... 16

2.5.1 Tahapan *Text* *Mining ................................................................* 17

2.6 *BM25* ................................................................................................. 19

2.7 *Unfied Modeling Language* (UML) *..................................................* 23

2.7.1 Pengertian UML *.......................................................................* 23

2.7.2 Diagram-diagram UML *............................................................* 23

2.8 *Flowchart* ......................................................................................... 28

2.9 Basis Data (*Database)* ...................................................................... 29

2.10 *Codeigniter* ..................................................................................... 30

2.11 PHP (*Hypertext Preprocessor)* ........................................................ 31

2.12 XAMPP *...........................................................................................* 31

2.13 MySQL *............................................................................................* 32

2.14 *Black Box Testing ............................................................................* 33

### BAB III METODOLOGI PENELITIAN ...................................................... 34

3.1 Metode Pengumpulan Data .............................................................. 34

3.2 Metode Pengembangan Sistem ....................................................... 34

3.3 Waktu dan Tempat Penelitian ......................................................... 35

3.3.1 Waktu Penelitian .................................................................... 35

3.3.2 Tempat Penelitian ................................................................... 35

3.4 Analisis Sistem ................................................................................. 36

4.1.1 Analisis Kebutuhan Fungsional ............................................. 36

4.1.2 Analisis Kebutuhan NonFungsional........................................ 36

4.1.2.1 Kebutuhan Perangkat Keras........................................ 37

4.1.2.2 Kebutuhan Perangkat Lunak ....................................... 37

4.2 Analisis Perancangan Sistem ........................................................... 38

4.2.1 Analisis Perancangan *Flowchart* ............................................. 38

4.2.2 Analisis Perancangan UML .................................................... 39

4.2.2.1 *Usecase Diagram* ........................................................ 39

4.2.2.2 *Class Diagram* ............................................................ 40

4.2.2.3 *Activity Diagram* ......................................................... 41

4.2.2.3.1 *Activity Diagram* Untuk *User* ...................... 41

4.2.2.3.2 *Activity Diagram* Untuk Admin ................... 41

4.2.2.4 *Sequence Diagram* ...................................................... 49

4.2.3 Perancangan *User Interface* .................................................... 58

4.2.3.1 Antarmuka *User* .......................................................... 58

**DAFTAR PUSTAKA**

# DAFTAR GAMBAR

Gambar 2.1 Contoh Tahap *Tokenizing* ............................................................... 17

Gambar 2.2 Contoh Tahap Seleksi Fitur (*Feature Selection*) ............................. 18

Gambar 4.1 *Flowchart Diagram* Sistem ............................................................. 38

Gambar 4.2 *Usecase Diagram* Sistem ................................................................ 39

Gambar 4.3 *Class Diagram* Sistem ..................................................................... 40

Gambar 4.4 *Activity Diagram* Beranda *User* ...................................................... 41

Gambar 4.5 *Activity Diagram* *Login* Admin ....................................................... 42

Gambar 4.6 *Activity Diagram* Beranda Admin ................................................... 43

Gambar 4.7 *Activity Diagram* Kumpulan Skripsi ............................................... 44

Gambar 4.8 *Activity Diagram* Tambah Skripsi ................................................... 44

Gambar 4.9 *Activity Diagram* Pemilihan Dosen ................................................. 45

Gambar 4.10 *Activity Diagram* Tambah Pemilihan Dosen ................................. 46

Gambar 4.11 *Activity Diagram* Daftar Dosen ..................................................... 47

Gambar 4.12 *Activity Diagram* Tambah Daftar Dosen ....................................... 47

Gambar 4.13 *Activity Diagram* Daftar Kata Dasar ............................................. 48

Gambar 4.14 *Activity Diagram* Tambah Daftar Kata Dasar ............................... 49

Gambar 4.15 *Sequence Diagram* Beranda *User* ................................................. 49

Gambar 4.16 *Sequence Diagram* *Login* .............................................................. 50

Gambar 4.17 *Sequence Diagram* Kumpulan Skripsi .......................................... 51

Gambar 4.18 *Sequence Diagram* Tambah Skripsi .............................................. 52

Gambar 4.19 *Sequence Diagram* Pemilihan Dosen ............................................ 53

Gambar 4.20 *Sequence Diagram* Tambah Pemilihan Dosen .............................. 54

Gambar 4.21 *Sequence Diagram* Daftar Dosen .................................................. 55

Gambar 4.22 *Sequence Diagram* Tambah Daftar Dosen .................................... 56

Gambar 4.23 Sequance Diagram Daftar Kata Dasar .......................................... 57

Gambar 4.24 *Sequence Diagram* Tambah Kata Dasar ....................................... 57 Gambar 4.25 Menu Beranda *User* ...................................................................... 58

Gambar 4.26 Menu *Login* ................................................................................... 59

Gambar 4.27 Menu Beranda Admin ................................................................... 59

Gambar 4.28 Menu Kumpulan Skripsi ............................................................... 60

Gambar 4.29 Menu Pemilihan Dosen ................................................................. 61

Gambar 4.30 Menu Tambah Pemilihan Dosen ................................................... 62

Gambar 4.31 Menu Daftar Dosen ....................................................................... 63

Gambar 4.32 Menu Tambah Daftar Dosen ......................................................... 63

Gambar 4.33 Menu Daftar Kata Dasar ............................................................... 64

Gambar 4.34 Menu Tambah Daftar Kata Dasar ................................................. 65

Gambar 5.1 Menu Beranda *User* ........................................................................ 67

Gambar 5.2 Menu Beranda Admin ..................................................................... 67

Gambar 5.3 Menu Kumpulan Skripsi ................................................................. 68

Gambar 5.4 Menu Pemilihan Dosen ................................................................... 68

Gambar 5.5 *Form* Tambah Judul ........................................................................ 69

Gambar 5.6 Tampilan Penggalan Kata ............................................................... 69

Gambar 5.7 Tampilan Rekomendasi Dosen........................................................ 70

Gambar 5.8 Menu Daftar Dosen ......................................................................... 70

Gambar 5.9 Menu Daftar Kata Dasar ................................................................. 71

Gambar 5.10 *source code* Algoritma BM25 ....................................................... 72

Gambar 5.11 *source code* Tahapan *Text Mining* ................................................ 74

Gambar 5.12 Tambah Pemilihan (Peminatan RPL) ............................................ 87

Gambar 5.13 Tampilan Penggalan Kata ............................................................. 88

Gambar 5.14 Inputan Admin (Peminatan RPL) .................................................. 88

Gambar 5.15 Daftar Data Latih ........................................................................... 89

Gambar 5.16 *Output* Dosen pembimbing dan penguji (Peminatan RPL) ........... 89

Gambar 5.17 Tambah Pemilihan (Peminatan RPL dan KCV) ........................... 90 Gambar 5.18 Tampilan Penggalan Kata ............................................................. 90

Gambar 5.19 Inputan Admin (Peminatan RPL dan KCV) ................................. 91

Gambar 5.20 Daftar Data Latih ........................................................................... 91

Gambar 5.21 *Output* Dosen pembimbing dan penguji (RPL dan KCV) ............ 92

Gambar 5.22 Tambah Pemilihan (Peminatan RPL, KBJ dan KCV) .................. 92

Gambar 5.23 Tampilan Penggalan Kata ............................................................. 93

Gambar 5.24 Inputan Admin (Peminatan RPL, KBJ dan KCV) ........................ 93

Gambar 5.25 Daftar Data Latih ........................................................................... 94

Gambar 5.26 *Output* Dosen pembimbing dan penguji (RPL, KBJ dan KCV) ... 94

## DAFTAR TABEL

Tabel 2.1 Tahap *Preprocessing* .......................................................................... 20

Tabel 2.2 Proses Pembobotan *Term* pada Dokumen .......................................... 21

Tabel 2.3 Hasil Akhir *BM25* .............................................................................. 22

Tabel 2.4 Simbol *Use Case Diagram* ................................................................ 24

Tabel 2.5 Simbol *Activity Diagram* .................................................................... 25

Tabel 2.6 Simbol *Class Diagram* ........................................................................ 26

Tabel 2.7 Simbol *Sequence Diagram*.................................................................. 27

Tabel 2.8 Simbol *Flowchart* ............................................................................... 28

Tabel 3.1 Waktu Penelitian ................................................................................. 35

Tabel 4.1 Spesifikasi Perangkat Keras ............................................................... 37

Tabel 4.2 Spesifikasi Perangkat Lunak ............................................................... 37

Tabel 5.1 Tahap *Preprocessing*........................................................................... 75

Tabel 5.2 Hasil Akhir BM25 ............................................................................... 78

Tabel 5.3 Tahap *Preprocessing*........................................................................... 79

Tabel 5.4 Hasil Akhir BM25 ............................................................................... 83

Tabel 5.5 Tahap *Preprocessing*........................................................................... 84

Tabel 5.6 Hasil Akhir BM25 ............................................................................... 87

Tabel 5.7 Perbandingan Rekomendasi Dosen pembimbing dan Penguji Satu ... 95

Tabel 5.8 Perbandingan Rekomendasi Dosen pembimbing dan Penguji Dua .... 98

Tabel 5.9 Pengujian *Black Box* ........................................................................... 101

xvi

### BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Salah satu penyebab keterlambatan mahasiswa di Jurusan Teknik Informatika menyelesaikan studinya tepat waktu adalah masalah ide skripsi. Proses menemukan ide skripsi sampai persetujuan judulnya oleh dosen, membutuhkan waktu yang cukup lama. Banyak faktor yang menyebabkan hal ini terjadi. Salah satunya adalah sering terjadi kesamaan ide atau topik skripsi yang diajukan oleh mahasiswa dengan skripsi yang sudah ada sebelumnya. Hal ini disebabkan karena kurangnya informasi yang didapat oleh mahasiswa mengenai topik-topik skripsi yang sudah pernah diangkat. informasi mengenai ide atau topik skripsi yang sudah ada sebelumnya sangat penting untuk diketahui oleh mahasiswa, sebelum mengajukan ide permasalahan kepada Ketua Jurusan di Jurusan. Meskipun sudah ada koleksi skripsi di Jurusan Teknik Informatika Universitas Halu Oleo, namun masih jarang mahasiswa yang memanfaatkannya, dikarenakan banyaknya tumpukan skripsi dalam lemari dan juga masih kurangnya kumpulan skripsi yang ada. Hal ini berdampak kepada mahasiswa tingkat akhir dalam mencari ide atau topik tugas akhir. Selain mahasiswa, dosen pembimbing juga membutuhkan waktu yang cukup lama untuk mengecek apakah judul yang didaftar sudah pernah dikerjakan oleh mahasiswa sebelumnya atau belum pernah sama sekali.

Tugas akhir merupakan salah satu mata kuliah yang harus ditempuh oleh seorang mahasiswa untuk memperoleh gelar sarjananya. Dalam menyelesaikan tugas akhir ini biasanya mahasiswa yang dalam proses penyusunan skripsi dibantu oleh 2 orang dosen pembimbing yang akan membimbing serta membantunya dalam penyusunan tugas akhir. Biasanya dalam menentukan calon dosen pembimbing terdapat beberapa dosen yang bisa menjadi dosen pembimbing, namun jumlah mahasiswa yang mengajukan tugas akhir lebih banyak dibanding dosen pembimbing menyebabkan proses penentuan dosen pembimbing menjadi kurang optimal. Selain dosen pembimbing, mahasiswa juga membutuhkan dosen penguji yang akan menguji tugas akhir mahasiswa apabila masih terdapat kekurangan, maka dapat disemputnakan lagi pada tahap revisi. Tugas penentuan pembimbing

1

dan penguji tugas akhir biasanya diberikan kepada ketua jurusan, yang dalam proses penentuanya masih menggunakan cara konvensional dengan mengandalkan pengetahuan pribadi tentang spesifikasi keahlian dosen yang sesuai dengan topik tugas akhir. Begitu pula yang terjadi di Jurusan Teknik Informatika Universitas Halu Oleo, hal ini sering menimbulkan keputusan yang kurang optimal, dimana dosen yang ditunjuk spesifikasinya masih kurang padahal masih ada dosen yang lebih kompeten dengan judul skripsi mahasiswa. Oleh karena itu dibutuhkan sistem yang dapat mendukung pengambilan keputusan dalam menentukan dosen pembimbing dan penguji yang sesuai dengan judul tugas akhir mahasiwa. BM25 merupakan formula terbaik dalam kelas *Best Match,* dikarenakan formula ini efektif dan memiliki ketepatan dalam mengurutkan dokumen berdasarkan *query* yang dicari (Bintana, et al., 2012). Media pengelola data-data tugas akhir yang memanfaatkan teknologi *web* dan penentuan dosen pembimbing dan penguji tugas akhir sangat dibutuhkan dalam hal ini.

Pada Peraturan Rektor Universitas Halu Oleo Nomor 1 Tahun 2019 Tentang Peraturan Akademik dilingkungan Universitas Halu Oleo BAB XVII Ujian Tugas Akhir Bagian Kesatu Tentang Pelaksanaan Ujian Tugas Akhir,Skripsi, Tesis dan Disetasi Pasal 83 No (1) Pembimbing/ Promotor/ Co- Promotor Untuk Tugas Akhir/ Skripsi/ Tesis/ Disertasi/ Ditentukan Oleh Ketua Jurusan/ Koordinator Program Studi Berdasarkan Keahlian dan Kompetensinya. Maka dari itu, dari latar belakang diatas penulis berkeinginan menuangkan kedalam bentuk skripsi dan memberi judul **“Penerapan Metode BM25** **Pada Sistem Rekomendasi Dosen**

#### Pembimbing Dan Penguji Tugas Akhir Mahasiswa Berbasis Web. Studi Kasus

**: Jurusan Teknik Informatika Universitas Halu Oleo ”.**

##### 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan diatas, maka penulis merumuskan masalah penelitian yaitu:

1. Bagaimana Merancang dan Membangun Sistem Informasi Tugas Akhir Mahasiswa di Jurusan Teknik Informatika Universitas Halu Oleo?
2. Bagaimana menerapkan Metode BM25 dalam Rekomendasi Penentuan Dosen Pembimbing dan Penguji Tugas Akhir ?

##### 1.3 Batasan Masalah

Adapun batasan masalah yang akan dibahas dalam sistem ini adalah :

1. Penelitian ini hanya dilakukan dalam lingkup Jurusan Teknik Informatika Fakultas Teknik Universitas Halu Oleo.
2. Mahasiswa hanya dapat melihat informasi mengenai kumpulan judul-judul skripsi.
3. Rumus perhitungan BM25 digunakan untuk menentukan rekomendasi dosen pembimbing dan penguji berdasarkan judul tugas akhir mahasiswa.

##### 1.4 Tujuan Penelitian

Adapun tujuan penelitian ini adalah :

1. Dapat merancang dan membangun sistem informasi tugas akhir mahasiswa di Jurusan Teknik Informatika Universitas Halu Oleo.
2. Menganalisis Penerapan Metode BM25 dalam Sistem Rekomendasi Penentuan Dosen Pembimbing dan Penguji Tugas Akhir.

##### 1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Diharapkan Metode BM25 dapat diterapkan dalam penentuan dosen pembimbing dan penguji tugas akhir di Jurusan Teknik Informatika.
2. Diharapkan sistem ini dapat membantu dan mempermudah Ketua Jurusan Teknik Informatika dalam menentukan dosen pembimbing dan penguji tugas akhir mahasiswa.

**1.6 Sistematika Penulisan**

Adapun sistematika penulisan proposal penelitian ini yaitu sebagai berikut :

### BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

### BAB II LANDASAN TEORI

Teori-teori penunjang yang berhubungan dengan penelitian berupa teori dari metode yang digunakan.

### BAB III METODOLOGI PENELITIAN

Bab ini berisi tahapan tentang proses dan metode yang akan digunakan dalam menentukan dosen pembimbing dan penguji tugas akhir mahasiswa.

### BAB IV ANALISIS DAN PERANCANGAN

Pada bagian ini diuraikan analisis sistem yang akan dibuat dan kebutuhan sistem yang meliputi kebutuhan fungsional, kebutuhan non fungsional sistem. Rancangan sistem meliputi rancangan arsitektur sistem, rancangan proses, rancangan prosedural, rancangan data dan rancangan *user interface*

### BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM

Bagian ini menyajikan secara lengkap setiap langkah yang dilakukan dalam implementasi sistem, dilakukan pembahasan skenario dan pengujian pada BAB V.

### BAB VI PENUTUP

Bab ini menguraikan tentang kesimpulan dari keseluruhan bab serta memberikan saran-saran yang mungkin berguna untuk mengatasi masalah yang dihadapi.

#### 1.7 Tinjauan Pustaka

Penelitian ini dibuat berdasarkan penelitian yang pernah dilakukan oleh :

Firayati, 2016. Sistem pendukung Keputusan Penentuan Dosen Pembimbing tugas akhir menggunakan metode *Weighted Product (WP)*. Metode ini dapat diterapkan dalam pengambilan keputusan penentuan pembimbing tugas akhir dengan kriteria yang telah ditetapkan sehingga mendapatkan hasil untuk alternatif dosen pembimbing tugas akhir. Hasil keputusan yang dihasilkan oleh sistem ini adalah berupa nama dosen pembimbing I dan pembimbing II yang dapat dijadikan rekomendasi pilihan.

Lukman, 2012. Sistem Pengambil Keputusan Penentuan Pembimbing dan Penguji Skripsi Berdasarkan Spesifikasi Keahlian Dosen Menggunakan Logika Fuzzy. Penelitian tersebut menerapkan logika *fuzzy* untuk menentukan dosen pembimbing dan penguji skripsi berdasarkan bobot dan kriteria yang telah di tentukan. Output yang dihasilkan dari penelitian tersebut adalah pembimbing dan penguji yang mempunyai spesifikasi keahlian sesuai dengan topik skripsi mahasiswa.

Ian Septiana, 2016. SPK penentu dosen penguji dan pembimbing tugas akhir dengan *Fuzzy Multiple attribute decision making* dengan SAW menghasilkan sebuah sistem yang mampu memberikan informasi mengenai rekomendasi dosen penguji dan pembimbing tugas akhir.

Rizky Riansyah, dengan judul Implementasi dan Perbandingan Metode BM25 dan PLSA Pada Aplikasi *Information Retrieval.* Pada penelitian ini membandingkan dua metode yaitu metode BM25 dan *Probabilistic Latent Semantic Analysis* (PLSA*).* Metode BM25 lebih efektif dalam melakukan perangkingan dokumen, karena metode PLSA lebih mengutamakan nilai dokumen dibandingkan dengan kata yang memiliki *relevansi* terhadap *query.*

Rizqa Raaiqa Bintana 2012. Penerapan Model BM25 Pada Sistem Temu Kembali Informasi. Sistem IR (*information retrieval*) menyediakan sejumlah kemungkinan respon atau jawaban (berupa dokumen) yang dianggap relevan dan membiarkan pengguna memilih mana yang benar atau tepat menurut kebutuhannya (*query*). Kemampuan sistem IR dengan model BM25 dalam me-*retrieve* dokumen yang relevan dengan *query* pengguna dinilai baik berdasarkan nilai *precision* yang cukup tinggi, dan berada pada peringkat teratas pelacakan. Sistem IR dengan model BM25 ini dapat digunakan untuk situs-situs pencarian lokal organisasi, seperti perpustakaan digital (*digital library*), arsip dokumen hukum, teknik, dan sebagainya.

### 

### BAB II

### LANDASAN TEORI

#### 2.1. Tugas Akhir

Tugas akhir merupakan suatu karya ilmiah yang disusun mahasiswa untuk menyelesaikan studinya melalui proses berpikir ilmiah, kreatif, integratif, dan sesuai dengan disiplin ilmunya yang disusun untuk memenuhi persyaratan kebulatan studi dalam program dan jenjang pendidikan yang ada di lingkungan tempat studi. Tugas akhir disusun dengan tujuan memberi kesempatan kepada mahasiswa agar dapat memformulasikan ide, konsep, pola berpikir, dan kreativitasnya yang dikemas secara terpadu dan komprehensif, dan dapat mengkomunikasikan dalam format yang lazim digunakan di kalangan masyarakat ilmiah (Yudanto. 2009). Di Universits Halu Oleo, proses pelaksanaan tugas akhir memiliki Standar Operasional Prosedur (SOP) dalam proses pelakasaan Ujian Tugas Akhir diantaranya yaitu:

##### 2.1.1 Penetapan Pembimbing Skripsi

1. Syarat Mahasiswa yang Memperoleh Hak Pembimbingan Skripsi :
   * 1. Mahasiswa terdaftar aktif pada semester berjalan.
     2. Skripsi telah diprogramkan dalam KRS.
     3. Pengajuan judul skripsi dapat dilakukan oleh mahasiswa setelah mengumpulkan SKS minimal sebanyak 120 SKS.
2. Proses Penetapan Pembimbing

a. Mahasiswa yang telah memenuhi syarat untuk melakukan penelitian skripsi dapat mengajukan syarat berikut kepada jurusan/program studi diawal semester berjalan dengan memperlihatkan:

* + - * Bukti slip pemayaran SPP semester berjalan.
      * Transkrip nilai yang dikeluarkan oleh Jurusan.
      * KRS semester berjalan yang telah ditandatangani oleh penasihat akademik (PA).
  1. Jurusan memeriksa kelengkapan administrasi mahasiswa yang akan melakukan penelitian skripsi.

6

* 1. Mahasiswa mengajukan rencana judul skripsi (mengisi formulir usulan yang disediakan jurusan) kepada ketua Jurusan atas persetujuan Penasehat Akademik (PA).
  2. Jurusan menunjuk dosen pengarah untuk memberikan masukan terkait rencana judul yang diusulkan. Dosen pengarah akan menjadi salah satu calon dosen pembimbing skripsi.
  3. Mahasiswa mengembalikan rencana judul skripsi yang telah dikonsultasikan dengan dosen pengarah kepada jurusan.
  4. Jurusan membuat formulir kesediaan kepada calon dosen pembimbing
  5. Apabila dosen pengarah/calon dosen pembimbing tidak bersedia sebagai dosen pembimbing maka jurusan akan memproses ulang.
  6. Setelah calon dosen pembimbing menyatakan bersedia maka Ketua Jurusan menerbitkan dan menandatangani Surat Penunjukan Dosen Pembimbing Skripsi.

##### 2.1.2 Seminar Proposal

Adapun persyaratan untuk Seminar Proposal yaitu :

1. Seminar proposal penelitian dapat dilaksanakan apabila mahasiswa telah menyelesaikan minimal 120 SKS (Sistem Kredit Semester), memprogram Mata Kuliah Skripsi (6 SKS) dan memenuhi semua persyaratan administrasi lainnya yang berlaku di setiap Universitas Halu Oleo.
2. Tim penguji pada seminar proposal penelitian terdiri dari tenaga pendidik/dosen yang ditetapkan oleh ketua jurusan dengan mempertimbangkan bidang keilmuan/kompetensi dosen yang bersangkutan (3 orang) ditambah dengan dosen pembimbing (2 orang).
3. Jurusan mempersiapkan dokumen-dokumen dan menetapkan waktu pelaksanaan seminar proposal penelitian.
4. Pelaksanaan seminar proposal penelitian dipimpin oleh ketua tim penguji yang berlangsung selama/maksimal 2 jam yang terdiri dari sesi presentasi oleh mahasiswa dan sesi tanya-jawab antara tim penguji dan mahasiswa.
5. Tim penguji menetapkan dan mengumumkan hasil seminar proposal penelitian kepada mahasiswa. Bagi mahasiswa yang dinyatakan lulus, dapat melanjutkan penelitian sedangkan mahasiswa yang tidak lulus, diberikan kesempatan maksimal dua kali untuk melakukan seminar proposal penelitian.
6. Berita acara seminar proposal penelitian dan dokumen-dokumen lainnya diserahkan kembali oleh tim penguji ke bagian administrasi jurusan.

##### 2.1.3 Seminar Hasil

Adapun persyaratan untuk Seminar Hasil yaitu :

1. Seminar hasil penelitian dapat dilaksanakan apabila mahasiswa telah menyelesaikan minimal 120 SKS (Sistem Kredit Semester), memprogram Mata Kuliah Skripsi (6 SKS) dan memenuhi semua persyaratan administrasi lainnya yang berlaku di setiap jurusan lingkup Universitas Halu Oleo.
2. Tim penguji pada seminar hasil penelitian terdiri dari tenaga pendidik/dosen yang ditetapkan oleh oleh ketua jurusan dengan mempertimbangkan bidang keilmuan/kompetensi dosen yang bersangkutan (3 orang) ditambah dengan dosen pembimbing (2 orang).
3. Jurusan mempersiapkan dokumen-dokumen dan menetapkan waktu pelaksanaan seminar hasil penelitian.
4. Pelaksanaan seminar hasil penelitian dipimpin oleh ketua tim penguji yang berlangsung selama/maksimal 2 jam yang terdiri dari sesi presentasi oleh mahasiswa dan sesi tanyajawab antara tim penguji dan mahasiswa.
5. Tim penguji menetapkan dan mengumumkan hasil seminar hasil penelitian kepada mahasiswa. Bagi mahasiswa yang dinyatakan lulus, dapat melanjutkan penelitian sedangkan mahasiswa yang tidak lulus, diberikan kesempatan maksimal dua kali untuk melakukan seminar hasil penelitian.
6. Berita acara seminar hasil penelitian dan dokumen-dokumen lainnya diserahkan kembali oleh tim penguji ke bagian administrasi jurusan.

##### 2.1.4 Seminar Skripsi

Adapun persyaratan untuk Seminar Skripsi yaitu :

1. Mengajukan permohonan ujian pada Jurusan/Program Studi yang diketahui oleh PA dan disetujui oleh Dosen Pembimbing/Promotor.
2. Melampirkan bukti pembayaran SPP/UKT semester berjalan.
3. Telah menyelesaikan semua mata kuliah yang dipersyaratkan.
4. Telampirkan skripsi, tesis, disertasi, atau tugas akhir lainnya yang telah disetujui oleh Dosen Pembimbing/Promotor.
5. Melampirkan Surat Keterangan Bebas Peminjaman buku dari Perpustakaan Program Studi/Fakultas/Universitas, Keterangan Bebas Peminjaman alat dan/atau bahan dari Laboratorium yang relevan dengan Program Studi.
6. Melampirkan sertifikat Nilai *Test of English Foreign Language* (TOEFL) atau TOEFL*-like* dan atau yang disetarakan dengan nilai minimal 400 untuk S1 dan diploma, 450 untuk S2 dan 500 untuk S3 yang berlaku paling lama 2 tahun terakhir.
7. Ketentuan pelaksanaan untuk mendapatkan sertifikat TOEFL atau TOEFL*-like* ditetapkan dengan Keputusan Rektor.
8. Pelaksanaan Ujian:
   1. Jurusan menyiapkan kelengkapan dan dokumen yang diperlukan untuk pelaksanaan sidang ujian.
   2. Pelaksanaan sidang ujian skripsi dilaksanakan di dalam ruangan ujian.
   3. Seluruh panitia ujian wajib hadir pada waktu dan tempat ujian yang telah ditetapkan.
   4. Ketua Penitia Penguji membuka dan memimpin sidang ujian.
   5. Presentasi hasil penelitian oleh mahasiswa, tanya-jawab mengenai materi skripsi dan materi lain yang relevan. (komprehensif) sebagai indikator kompetensi, dan penetapan keputusan hasil ujian.
   6. Penguji memberikan penilaian yang meliputi aspek kemampuan presentasi, sikap, penulisan skripsi dan penguasaan materi.

#### 2.2. Dosen Pembimbing

Pembimbing adalah dosen yang membantu mahasiswa dalam menyelesaikan tugas akhir, memeriksa konsep, kerangka dan memberikan arahan kepada mahasiswanya. Dosen pembimbing harus sesuai dengan bidang ilmu dalam judul tugas akhir yang diajukan mahasiswa (Ian Septiana, dkk.,2016).

#### 2.3. Dosen Penguji

Penguji adalah dosen yang bertugas menguji mahasiswa untuk mendeskripsikan secara jelas dan mempertanggung jawabkan tugas akhir sebelum memperoleh gelar sarjana. Penguji akan mengajukan beberapa pertanyaan terkait dengan tugas akhir yang diajukan mahasiswa (Ian Septiana, dkk.,2016).

#### 2.4. Konsep Sistem Informasi

##### 2.4.1. Definisi Sistem

Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisir, saling berinteraksi, saling ketergantungan satu sama lain dan terpadu. Berikut ini adalah beberapa definisi sistem menurut para ahli :

1. Sistem adalah kumpulan/grup dari sub sistem/bagian/komponen apapun, baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu (Yulia Djahir dan Pratita.

2015).

1. Sistem bisa diartikan sebagai sekumpulan sub sistem, komponen yang saling bekerja sama dengan tujuan yang sama untuk menghasilkan output yang sudah ditentukan sebelumnya (Mulyani. 2016).
2. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. (Hutahaean. 2015)

Berdasarkan pendapat dari para ahli diatas, maka dapat disimpulkan bahwa sistem merupakan suatu kumpulan komponen dari sub sistem yang saling bekerja sama dari prosedur-prosedur yang saling berhubungan untuk menghasilkan output dalam mencapai tujuan tertentu.

Suatu sistem mempunyai ciri-ciri karakteristik yang terdapat pada sekumpulan elemen yang harus dipahami dalam mengidentifikasi pembuatan sistem. Adapun karakteristik sistem (Hutahaean. 2015) yang dimaksud adalah sebagai berikut :

1. Komponen

Sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama untuk membentuk satu kesatuan. Komponen sistem dapat berupa sub sistem atau bagian-bagian dari sistem.

1. Batas Sistem (*Boundary*)

Daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luar dinamakan dengan batasan sistem. Batasan sistem ini memungkinkan sistem dipandang sebagai satu kesatuan dan juga menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

1. Lingkungan Luar Sistem (*Environments*)

Apapun yang berada di luar batas dari sistem dan mempengaruhi sistem tersebut dinamakan dengan lingkungan luar sistem. Lingkungan luar yang bersifat menguntungkan wajib dipelihara dan yang merugikan harus dikendalikan agar tidak mengganggu kelangsungan sistem.

1. Penghubung sistem (*Interface*)

Media penghubung diperlukan untuk mengalirkan sumber-sumber daya dari sub sistem ke sub sistem lainnya dinamakan dengan penghubung sistem.

1. Masukan sistem (*Input*)

Energi yang dimasukkan ke dalam sistem dinamakan dengan masukan sistem (*input*) dapat berupa perawatan dan masukan sinyal. Perawatan ini berfungsi agar sistem dapat beroperasi dan masukan sinyal adalah energi yang diproses untuk menghasilkan keluaran (*output*).

1. Keluaran (*Output*)

Hasil dari energi yang telah diolah dan diklasifikasikan menjadi keluaran yang berguna dinamakan dengan keluaran sistem (*output*). Informasi merupakan contoh keluaran sistem.

1. Pengolah (*Process*)

Untuk mengolah masukan menjadi keluaran diperlukan suatu pengolah yang dinamakan dengan pengolah sistem.

1. Sasaran atau tujuan

Sistem pasti memiliki tujuan atau sasaran yang sangat menentukan *input* yang dibutuhkan oleh sistem dan keluaran yang dihasilkan.

##### 2.4.2. Definisi Informasi

Beberapa pendapat para ahli tentang definisi dari informasi adalah sebagai berikut :

1. Informasi merupakan hasil dari pengolahan data, akan tetapi tidak semua hasil dari pengolahan data yang tidak dari pengolahan tersebut bisa disebut informasi, hasil pengolahan data yang tidak memberikan makna, arti atau penjelasan serta tidak bermanfaat bagi seseorang atau penerima informasi bukanlah merupakan informasi bagi orang tersebut (Dr. Deni Darmawan. 2013).
2. Informasi adalah data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan (Romney dan Steinbart. 2014).
3. Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berguna dan nyata atau berupa nilai yang dapat dipahami oleh penerima informasi sehingga dapat digunakan menjadi sebuah keputusan sekarang maupun yang akan datang (Gordon B. Davis dalam bukunya Bambang Hatono. 2013).

Berdasarkan pendapat beberapa ahli tersebut, maka dapat disimpulkan bahwa informasi merupakan data yang telah diproses sehingga menjadi data yang berguna dalam mengambil sebuah keputusan.

1. Siklus Informasi

Data merupakan bentuk yang masih mentah, belum banyak bercerita, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk menghasilkan informasi.

Data dapat berbentuk simbol-simbol semacam huruf-huruf atau *alphabet*, angkaangka, bentuk-bentuk suara, sinyal-sinyal, gambar-gambar dan sebagainya.

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus informasi (*information* *cycle*) atau ada yang menyebutnya dengan istilah siklus pengolahan data (*data* *processing* *cycle*).

1. Kualitas Informasi

Kualitas dari suatu informasi tergantung dari 3 hal yaitu akurat, tepat pada waktunya dan relevan.

1. Akurat

Informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.

1. Tepat pada waktunya

Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan didalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi.

1. Relevan

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

3. Nilai Informasi

Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Kegunaan informasi adalah untuk mengurangi hal ketidakpastian didalam proses pengambilan keputusan tentang suatu keadaan. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah tertentu dengan biaya untuk memperolehnya, karena sebagian informasi dinikmati oleh satu pihak di dalam perusahaan.

Sebagian besar informasi tidak ditaksir keuntungannya dengan suatu nilai usang, tetapi dapat ditaksir nilai efektifitasnya. Pengukuran nilai investasi biasanya dihubungkan dengan analisis *cost* *effectiveness* atau *cost* *benefit*.

##### 2.4.3. Definisi Sistem Informasi

Sistem informasi dapat didefinisikan sebagai suatu sistem dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdik. Berikut adalah beberapa definisi sistem informasi menurut para ahli :

1. Sistem informasi adalah seperangkat fungsi operasional manajemen kepada yang mampu menghasilkan suatu keputusan yang tepat, cepat dan jelas sehingga menjadi suatu susunan yang disusun secara sistematik dan teratur (Aswati dkk. dalam Jurnal Teknologi dan Sistem Informasi, 2015).
2. Sistem informasi adalah sistem buatan manusia yang terdiri dari komponen baik manual maupun yang berbasis komputer dan berintegrasi untuk mengumpulkan, menyimpan dan mengelola data serta menyediakan informasi untuk pihak-pihak yang bersangkutan sebagai pemakai sistem tersebut (Anastasia Diana dan Lilis Setiawati yang dikutip oleh Al Husain dkk dalam jurnal “Perancangan *Database* Relational pada Toko Buku *Online* Tangerang”.

2016).

Dari berbagai definisi para ahli tersebut, dapat disimpulkan bahwa sistem informasi merupakan seperangkat komponen dengan cara mengumpulkan, menyimpan dan mengelola data tersebut dengan tujuan menjadi susunan yang sistematik dan teratur sehingga menghasilkan sebuah komunikasi atau informasi.

Adapun tujuan dari sistem informasi adalah sebagai berikut :

1. Kegunaan (*usefullness*)

Sistem harus mampu menyajikan informasi guna mendukung pengambilan keputusan manajerial dan operasional.

1. Ekonomi (*economy*)

Semua komponen sistem termasuk kontrol, mesin dan sebagainya diharapkan dapat meningkatkan keuntungan paling sedikit sebesar biaya yang telah dikeluarkan.

1. Reabilitas (*reability*)

Hasil dari sistem harus memiliki tingkat akurasi yang tinggi dan sistem harus dapat beroperasi secara efektif, bahkan ketika komponen manusianya tidak ada ataupun mesin jika ada tidak beroperasi untuk sementara waktu.

1. Pelayanan pelanggan

Sistem dapat menyediakan pelayanan yang baik dan tepat guna.

1. Kapasitas (*capacity*)

Sistem harus mempunyai kapasitas yang cukup untuk mengatasi operasioperasi pada titik maksimum sebaik kegiatan normal.

1. Sederhana (*simplicity*)

Sistem harus sederhana, struktur maupun operasinya dapat dengan mudah dimengerti dan prosedur-prosedur mudah disempurnakan.

1. Fleksibel (*flexibility*)

Sistem harus cukup fleksibel dalam menyesuaikan diri terhadap perubahanperubahan dan permintaa perbaikan yang dikehendaki oleh organisasi.

#### *2.5.Text Mining*

*Text Mining* merupakan salah satu aplikasi dari bidang data mining, yang khusus mengolah data dalam bentuk teks. Tujuan dari *text mining* adalah mencari informasi implisit dari data teks sehingga bisa digunakan oleh pengguna untuk mengambil keputusan. *Text mining* merupakan teknik yang digunakan untuk menangani masalah klasifikasi (*classification*), *clustering*, *information extraction*, dan *information retrieval*. *Text mining* merupakan variasi dari *Data mining* yang berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar. *Text mining* dapat diartikan sebagai penemuan informasi yang baru dan tidak diketahui sebelumnya oleh komputer, secara otomatis mengekstrak informasi dari sumber-sumber yang berbeda. Kunci dari proses ini adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber (Hearst,2003). (Menurut Harlian,2006), *text mining* didefinisikan sebagai data yang berupa teks yang biasanya sumber data didapatkan dari dokumen, dengan tujuan adalah mencari kata-kata yang dapat mewakili isi dari dokumen tersebut yang nantinya dapat dilakukan analisa hubungan antar dokumen.

##### 2.5.1 Tahapan *Text Mining*

Tahapan *text mining* secara umum dibagi menjadi beberapa tahapan umum (Triawati, 2009).

#### 1. *Text Preprocessing*

*Text Preprocessing* merupakan tahapan awal dari *text mining* yang bertujuan mempersiapakan teks menjadi data yang akan mengalami pengolahan pada tahap selanjutnya. Pada *text mining*, data mentah yang berisi informasi memiliki struktur yang sembarang, sehingga diperlukan proses pengubahan bentuk menjadi data yang terstruktur sesuai kebutuhan, yaitu biasanya akan mejadi nilai-nilai numerik. Proses ini disebut *Text Preprocessing*. Pada tahap ini, tindakan yang dilakukan adalah *toLowerCase*, dengan mengubah semua karakter huruf menjadi huruf kecil, dan *tokenizing* yaitu proses penguraian deskripsi yang semula berupa kalimat mejadi kata-kata kemudian menghilangkan delimiter-delimiter seperti tanda koma (,), tanda titik (.), spasi, dan karakter angka yang terdapat pada kata tersebut (Triawati, 2009).

Contoh dari tahap ini seperti pada Gambar 2.1 berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | Tugas akhir adalah sebuah mata kuliah yang harus  ditempuh oleh seorang mahasiswa menjelang wisuda | |  | |  | | --- | | tugas akhir adalah sebuah mata kuliah yang harus di  tempuh oleh seorang mahasiswa menjelang wisuda | |

#### Gambar 2.1 Contoh Tahap *Tokenizing*

##### 2. Seleksi Fitur (*Feature Selection*)

Pada tahap ini akan dilakukan seleksi dengan mengurangi jumlah kata-kata yang dianggap tidak penting dalam dokumen tersebut untuk menghasilkan proses pengklasifikasian yang lebih efektif dan akurat. Tahapan ini adalah dengan melakukan penghilangan *stopword* dan juga mengubah kata-kata kedalam bentuk dasar terhadap kata yang berimbuhan.

*Stopword* merupakan kosakata yang bukan merupakan ciri atau kata unik dari suatu dokumen seperti kata sambung. Yang termasuk *stopword* yaitu “ di”, “pada”, ”sebuah”, ”karena”, ”oleh” dan sebagainya. Sebelum memasuki tahapan penghilang *stopword*, daftar *stopword* harus dibuat terlebih dahulu. Jika kata-kata yang termasuk s*topword* masuk dalam *stoplist*, maka kata tersebut akan dihapus dari deskripsi sehingga sisanya dianggap sebagai kata-kata yang mencirikan isi dokumen atau *keywords* (Triawati, 2009).

Contoh dari tahap ini seperti pada Gambar 2.2 berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | Tugas akhir adalah sebuah mata kuliah yang harus di  tempuh oleh seorang mahasiswa menjelang  wisuda | |  | |  | | --- | | Tugas akhir mata kuliah harus tempuh mahasiswa menjelang wisuda | |

#### Gambar 2.2 Contoh Tahap Seleksi Fitur (*Feature Selection*)

##### 3. *Stemming*

*Stemming* adalah proses pemetaan dan penguraian berbagai bentuk dari suatu kata menjadi kata dasarnya. Tujuan dilakukannya proses *stemming* adalah menghilangkan imbuhan-imbuhan berupa *prefix, suffix*, maupun konfiks yang terdapat pada setiap kata. Apabila imbuhan tadi tidak dihilangkan maka setiap kata akan disimpan di dalam *database*, sehingga nantinya akan menjadi beban di dalam *database*. Bahasa Indonesia memiliki aturan morfologi maka proses *stemming* harus berdasarkan aturan morfologi Bahasa Indonesia (Triawati, 2009).

###### 2.6 BM25

BM25 adalah model pembobotan hasil pencarian yang dikembangkan oleh City University London berdasarkan pada model robabilistik dasar yang mengurutkan dokumen dalam urutan menurun terhadap nilai relevansi sebuah dokumen terhadap informasi yang dibutuhkan. BM25 meranking dokumen berdasarkan probabilitas dan menggunakan *term frequency* untuk meranking similarity.

BM25 merupakan sebuah algoritma yang digunakan oleh *search engine* untuk menentukan relevansi dokumen. Algoritma ini menggunakan *tf-idf* untuk melakukan prosesnya.

Tujuan akhir dari metode BM25sendiri adalah melakukan pembobotan dokumen berdasarkan *query* (atau dokumen yang menjadi patokan) yang dimasukkan dengan menghitung koefisien kemiripan (*Similarity Coefisien*)

(Sebastian et al., 2011).

|𝑞|

𝑡𝑓(𝑞𝑖. 𝑑). (𝑘₁ + 1)

∑ 𝑖𝑑𝑓 (𝑞𝑖). 𝑡𝑓(𝑞𝑖. 𝑑) + 𝑘₁. (1 − 𝑏 + 𝑏). |𝑑| (2.1)

𝑖=1 𝑎𝑣𝑔𝑑𝑙

Dimana :

*tf (qi,d)* : *Frekuensi* *term* yang muncul pada dokumen.

|d| : Jumlah kata dalam dokumen d. *avgdl* (*average document length*) : Panjang rata-rata seluruh dokumen korpus. k1, b : *Tunning Parameter* Nilai k1 = 1.2 dan b=0.75.

*idf (qi)*: Bobot dokumen dari *query* dapat dihitung sebagai berikut :

*Idf* (*qi*)=log 𝑁−𝑑𝑓 (𝑞𝑖)+0.5(2.2)

𝑑𝑓(𝑞𝑖)+0.5

Dimana :

N : Jumlah dokumen dalam korpus.

*df* : Jumlah dokumen dalam korpus yang mengandung *term* *(qi).*

Contoh Kasus : Diketahui terdapat 3 dokumen, yaitu D1, D2 dan D3 sebagai berikut:

D1 : citra identifikasi fuzzy diagnosis deteksi logika ekstraksi pakar genetika visualisasi virtual temu bm25

D2 : kriptografi keamanan robotik jaringan wireless gateway arduino protokol prototipe konversi router service cloud

D3 : geografis klasifikasi android game informasi keputusan ahp saw klastering prediksi c45 apriori cart.

Jika terdapat *query* (Q) : “Sistem Informasi Pengelohan Data Koleksi BendaBenda Bersejarah Pada Museum Negeri Provinsi Sulawesi Tenggara Berbasis Android”. Tentukan daftar dokumen yang paling relevan dengan *query* tersebut.

Penyelesaian:

Untuk menyelesaikan, pertama yang harus dikerjakan adalah melakukan tahap *preprocessing* terhadap semua (N=4) dokumen yang terlibat, yaitu D1, D2, dan D3 dan Q. Berikut adalah tabel hasil *preprocessing* dokumen.

**Tabel 2.1.** Tahap *Preprocessing*

|  |  |
| --- | --- |
| **Dokumen** | ***Term* yang mewakili Dokumen** |
| Q | sistem informasi pengelohan data koleksi benda-benda bersejarah pada museum negeri provinsi sulawesi tenggara berbasis android |
| D1 | citra identifikasi fuzzy diagnosis deteksi logika ekstraksi pakar genetika visualisasi virtual temu bm25 |
| D2 | kriptografi keamanan robotik jaringan wireless gateway arduino protokol prototipe konversi router service cloud |
| D3 | geografis klasifikasi android game informasi keputusan ahp saw klastering prediksi c45 apriori cart |

Kemudian tentukan bobot untuk setiap *term* dari dokumen tersebut. Setelah itu lakukan perhitungan IDF kemudian hitung jumlah panjang *term* pada tiap dokumen |d|. Lalu hitung jumlah keseluruhan panjang *term* tersebut kemudian dibagi dengan jumlah dokumen (N) untuk mendapatkan hasil rata-rata panjang dokumen keseluruhan (avgdl).

#### Tabel 2.2. Proses Pembobotan *Term* pada Dokumen

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | *Term* | TF | | | D F | IDF | |d| | | | avgdl |  | BM25 | |
| D1 | D2 | D3 | D1 | D2 | D3 | D1 | D2 | D3 |
| 1 | citra | 1 |  |  | 1 | 0.778 | 13 | 13 | 13 | 13 |  |  |  |
| 2 | identifikasi | 1 |  |  | 1 | 0.778 |  |  |  |
| 3 | fuzzy | 1 |  |  | 1 | 0. 778 |  |  |  |
| 4 | diagnosis | 1 |  |  | 1 | 0. 778 |  |  |  |
| 5 | deteksi | 1 |  |  | 1 | 0. 778 |  |  |  |
| 6 | logika | 1 |  |  | 1 | 0. 778 |  |  |  |
| 7 | ekstraksi | 1 |  |  | 1 | 0.778 |  |  |  |
| 8 | pakar | 1 |  |  | 1 | 0.778 |  |  |  |
| 9 | genetika | 1 |  |  | 1 | 0. 778 |  |  |  |
| 10 | visualisasi | 1 |  |  | 1 | 0. 778 |  |  |  |
| 11 | virtual | 1 |  |  | 1 | 0. 778 |  |  |  |
| 12 | temu | 1 |  |  | 1 | 0. 778 |  |  |  |
| 13 | bm25 | 1 |  |  | 1 | 0.778 |  |  |  |
| 14 | kriptografi |  | 1 |  | 1 | 0.778 |  |  |  |
| 15 | keamanan |  | 1 |  | 1 | 0. 778 |  |  |  |
| 16 | robotik |  | 1 |  | 1 | 0. 778 |  |  |  |
| 17 | jaringan |  | 1 |  | 1 | 0. 778 |  |  |  |
| 18 | wireless |  | 1 |  | 1 | 0. 778 |  |  |  |
| 19 | gateway |  | 1 |  | 1 | 0.778 |  |  |  |
| 20 | arduino |  | 1 |  | 1 | 0.778 |  |  |  |
| 21 | protokol |  | 1 |  | 1 | 0. 778 |  |  |  |  |  |  |  |
| 22 | prototipe |  | 1 |  | 1 | 0. 778 |  |  |  |
| 23 | konversi |  | 1 |  | 1 | 0. 778 |  |  |  |
| 24 | router |  | 1 |  | 1 | 0. 778 |  |  |  |
| 25 | service |  | 1 |  | 1 | 0.778 |  |  |  |
| 26 | cloud |  | 1 |  | 1 | 0.778 |  |  |  |
| 27 | geografis |  |  | 1 | 1 | 0.778 |  |  |  |
| 28 | klasifikasi |  |  | 1 | 1 | 0.778 |  |  |  |
| 29 | android |  |  | 1 | 1 | 0.3010 | 0 | 0 | 0.3292 |
| 30 | game |  |  | 1 | 1 | 0.778 |  |  |  |
| 31 | informasi |  |  | 1 | 1 | 0.3010 | 0 | 0 | 0.3292 |
| 32 | keputusan |  |  | 1 | 1 | 0.778 |  |  |  |
| 33 | ahp |  |  | 1 | 1 | 0.778 |  |  |  |
| 34 | saw |  |  | 1 | 1 | 0. 778 |  |  |  |
| 35 | klastering |  |  | 1 | 1 | 0. 778 |  |  |  |
| 36 | prediksi |  |  | 1 | 1 | 0. 778 |  |  |  |
| 37 | c45 |  |  | 1 | 1 | 0. 778 |  |  |  |
| 38 | apriori |  |  | 1 | 1 | 0.778 |  |  |  |
| 39 | cart |  |  | 1 | 1 | 0.778 |  |  |  |
|  |  | | | | | | 0 | 0 | 0.6584 |

Lalu hitung kemiripan *query* (Q) dengan dokumen yang ada. Kemiripan antar dokumen diproses dengan menggunakan *BM25*. Hasil perhitungan tersebut diperlihatkan pada tabel berikut:

#### Tabel 2.3. Hasil Akhir *BM25*

|  |  |  |
| --- | --- | --- |
| D1 | D2 | D3 |
| 0 | 0 | 0.6584 |

Maka, dokumen yang paling mirip dengan *query* (Q) adalah D3.

##### 2.7 *Unifield Modeling Language* (UML)

###### 2.7.1. Pengertian UML

Pengertian UML menurut Adi Nugroho (2010) *Unifield Modeling Language* (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan (*modeling)* sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Bahasa pemodelan grafis telah ada di industri perangkat lunak sejak lama. Pemicu utama dibalik semuanya adalah bahwa bahasa pemrograman berada pada tingkat abstrasi yang terlalu tinggi untuk memfasilitasi diskusi tentang desain dengan menggunakan notasi-notasi seperti UML, alur logika dari perangkat lunak yang akan dikembangkan bisa mudah dipahami.

**2.7.2. Diagram-diagram UML**

seperti yang telah dikemukakan sebelumnya, UML merupakan *tool* yang tepat untuk memodelkan sebuah rancangan perangkat lunak. Seperti hanya *tool* untuk memodelkan perangkat lunak terstruktur, UML pun memiliki diagramdiagram tertentu dan terstandar.

Notasi yang digunakam dalam pemodelan UML yaitu:

1. *Use case Diagram*

*Use case diagram* menampilkan sekumpulan *use case* dan *actor* (pelaku) dan hubungan diantara *use case* dan *actor* tersebut. *Use case diagram* digunakan untuk penggambaran *use case* statik dari suatu *system*. *Use case* menjelaskan apa yang dilakukan sistem (atau subsistem) tetapi tidak menspesifikasikan cara kerjanya. *Flow of event* digunakan untuk menspesifikasikan kelakuan dari *use case*. *Flow of event* menjelaskan *use case* dalam bentuk tulisan dengan sejelas-jelasnya, diantaranya bagaimana, kapan *use case* dimulai dan berakhir, ketika *use case* berinteraksi dengan aktor, obyek apa yang digunakan, alur dasar dan alur *alternative*.

#### Tabel 2.4 Simbol *Use Case Diagram* (Kiswanto, 2016)

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | *Use Case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*. |
|  | Aktor | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. |
|  | Asosiasi | Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor. |
|  | Ekstensi (*extend*) | Relasi *use case* tambahan ke sebuah *use case*  dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu. |
|  | Generalisasi | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya. |
|  | Menggunakan  (*include*) | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini. |

2. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses pada perangkat lunak. Yang perlu di perhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan yang di lakukan aktor .

#### Tabel 2.5 Simbol *Activity Diagram* (Kiswanto, 2016)

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | Status Awal | Status awal aktivitas sistem. |
|  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
|  | Percabangan (*decision*) | Asosiasi percabangan digunakan untuk mendefinisikan jika ada pilihan aktivitas lebih dari satu. |
|  | Penggabungan  (*join*) | Asosiasi penggabungan digunakan untuk mendefinisikan beberapa aktivitas yang digabungkan menjadi satu. |
|  | Status Akhir | Status akhir yang dilakukan sistem. |
|  | *Swimlane* | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi/ |

3. *Class Diagram*

*Class Diagram*  adalah diagram yang digunakan untuk menampilkan beberapa kelas yang ada dalam sistem perangkat lunak yang akan dikembangkan. *Class Diagram*  menunjukkan antar kelas dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Main *Class*

*Class* yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

1. *Class* yang menangani tampilan sistem (*view*)

*Class* yang mendefinisikan dan mengatur tampilan ke pemakai.

1. *Class* yang diambil dari pendefinisian *use case* (*controller*)

*Class* yang menangani fungsi – fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

1. *Class* yang diambil dari pendefinisian data (model)

*Class* yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Adapun simbol – simbol yang digunakan dalam *Class* *diagram* adalah sebagai berikut :

#### Tabel 2.6 Simbol *Class Diagram* (Kiswanto, 2016)

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | Class | Kelas pada stuktur sistem. |
|  | Antarmuka  (*interface*) | Sama dengan konsep antarmuka dalam pemrograman berorientasi objek. |
|  | Asosiasi | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multipicity*. |
|  | Asosiasi Berarah  (*directed association*) | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*. |
|  | Generalisasi | Relasi antar kelas dengan makna generalisasi–spesialisasi (umum khusus). |
|  | Kebergantungan  (*dependency*) | Relasi antar kelas dengan makna kebergantungan antar kelas. |
|  | Agregasi  (*aggreggation*) | Relasi antar kelas dengan makna semua bagian (*whole part*). |

4. *sequence Diagram*

*Sequence diagram* adalah suatu diagram interaksi yang menekankan pada pengaturan waktu dari pesan-pesan. Diagram ini menampilkan sekumpulan peran dan pesan-pesan yang dikirim dan diterima oleh instansi yang memegang peranan tersebut. *Sequence diagram* menangkap objek dan *class* yang terlibat dalam skenario dan urut-urutan pesan yang ditukar antara objek diperlukan untuk melaksanakan fungsionalitas skenario. *Sequence diagram* berasosiasi dengan *use* *case* selama proses pengembangan. Dalam *Unified Model Language* (UML), objek dalam *sequence* *diagram* digambar dengan segiempat yang berisi nama objek yang diberi garis bawah. Objek dapat diberi nama dengan tiga cara : (nama objek), (nama objek dan *class*) atau (hanya nama *class* (*anonymous* object). Berikut notasi *sequence diagram* seperti terlihat pada gambar dibawah ini :

#### Tabel 2.7 Simbol *Sequence Diagram* (Kiswanto, 2016)

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | Garis Hidup  (*lifeline*) | Menyatakan kehidupan suatu objek. |
|  | Objek | Menyatakan objek yang berinteraksi. |
|  | Waktu Aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. |
|  | Pesan Tipe *Call* | Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri. |
|  | Pesan Tipe *Send* | Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi. |

##### 2.8. *Flowchart*

*Flowchart* merupakan gambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. *Flowchart* menolong analis dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif - alternatif lain dalam pengoperasian, untuk mendesain program dan merepresentasikan program. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

#### Tabel 2.8 Simbol *flowchart* (Kiswanto, 2016)

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| *Terminator/*  *Terminal Points* | Awal atau akhir program |
| *Flow* | Arah aliran program, merepresentasikan alur kerja |
| *Preparation* | Inisialisasi / Pemberian Nilai Awal |
| *Process* | Proses / Pengolahan Data, Mempresentasikan Operasi |
| *Input/Output* | *Input/Output* Data Yang Diproses Atau Informasi |
| *Predefined Process* | Rincian Operasi Berada Di Tempat Lain |
| *Decision*  (Keputusan) | Seleksi Atau Kondisi, Keputusan Dalam Program |
| *On Page*  *Connector*  (Penghubung) | Penghubung Bagian-Bagian *Flowchart* Pada Halaman Yang Sama |

##### 2.9. Basis Data (*Database*)

*Database* adalah kumpulan informasi yang disimpan didalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. *Database* adalah kumpulan dari beberapa data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan oleh perangkat lunak untuk memanipulasinya. *Database* dapat juga diartikan kumpulan file-file yang mempunyai kaitan satu dengan yang lain sehingga membentuk satu bangunan data untuk menginformasikan data-data operasional pada satu perusahaan pada batasan tertentu. *Database* merupakan salah satu komponen yang penting pada sistem karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Perangkat lunak yang digunakan untuk mengelola dan memanggil *query* basis data disebut *Database* *Manajement* *System* (DBMS).

Penerapan *database* pada sistem disebut dengan sistem basis data. Menurut Sukamto dan Shalahudin (2015), sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Sistem *database* juga dapat diartikan sebagai suatu sistem yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya.

##### 2.10 *CodeIgniter*

Menurut Hakim (2010) *CodeIgniter* adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis semua kode program dari awal. *CodeIgniter* pertama kali dibuat oleh Rick Ellis, CEO Ellislab, Inc. (http://ellislab.com), sebuah perusahaan yang memproduksi CMS (*Content Management System*) yang cukup handal, yaitu *Expression Engine* (http://www.expressionengine.com). *CodeIgniter* adalah MVC *framework* yang di design untuk mempermudah penggunanya.

Adapun bagian dari MVC yaitu :

1. *Model*

Berisi fungsi-fungsi yang dapat digunakan untuk mengelola *database*, sepertiCRUD.

1. *View*

Yang mengatur tampilan halaman *website*. bagian inilah yang bertugas untuk mempresentasikan data kepada *user*. *View* berbentuk struktur HTML yang berisikan variabel data yang dikirimkan oleh *Controller*.

1. *Controller*

Merupakan bagian yang berfungsi sebagai penghubung antara *Model* dan *View*. *Controller* berisi fungsi-fungsi untuk memproses suatu data. *CodeIgniter* memungkinkan untuk memasukkan *script* yang di buat sendiri, atau bahkan mengembangkan *library* inti untuk sistem. MVC (*Model, View, Controller*) itu memisahkan antara logika pembuatan kode dengan pembuatan template website atau tampilan dari web. Jika menggunakan *Model-View-Controller* (MVC) akan menjadikan pembuatan sebuah *website* menjadi lebih terstruktur dan lebih singkat dalam mengerjakan pengkodingan

##### 2.11. PHP (*Hypertext Preprocessor*)

*Hypertext* *Preprocessor* atau disingkat PHP adalah suatu bahasa pemrograman yang digunakan untuk membuat web dinamis, walau bisa juga digunakan untuk membuat program lain. Tentunya bahasa pemrograman PHP berbeda dengan HTML, pada PHP script/kode yang dibuat tidak dapat ditampilkan pada halaman muka *website* begitu saja, tapi harus diproses terlebih dahulu oleh *web* *server* lalu ditampilkan dalam bentuk halaman *website* di browser, *script* PHP juga dapat disisipkan pada HTML dan *script* PHP selalu diawasi selalu diawali dengan <php dan diakhiri dengan ?>.

Manajemen *database* yang biasanya digunakan untuk pemrograman PHP misalnya seperti *MySQL*, tapi ada juga yang menggunakan Oracle, *Microsoft* *Acces* dan lain-lain. PHP disebut juga sebagai bahasa pemrograman *script* *server* side karena PHP diproses pada komputer *server*. Berikut beberapa kelebihan bahasa pemrograman PHP :

1. Banyaknya web *server* yang mendukung bahasa pemrograman PHP sehingga konfigurasnya mudah, beberapa web *server* misalnya *Apache*, IIS, *Xitami*, *Nginx* dan lain-lain
2. Pengembangan bahasa pemrograman PHP tergolong lebih mudah, karena banyak *developer* yang membantu mengembangkannya maupun menggunakannya
3. Relatif mudah untuk dipahami karena sekarang ini banyak sekali tersebar materi-materi referensi untuk mempelajari PHP
4. Bahasa pemrograman PHP juga dapat disisipkan kedalam HTML
5. Cocok digunakan untuk pemrograman web dinamis, walau bisa juga digunakan untuk membuat program komputer lainnya
6. PHP merupakan bahasa pemrograman bersifat *open* *source*, sehingga dapat digunakan dibanyak sistem operasi komputer dan tentunya gratis.

##### 2.12. XAMPP

XAMPP paket *server web* PHP dan *database* MySQL yang paling popular dikalangan pengembang web dengan menggunakan PHP dan MySQL sebagai *databasenya* (Kiswanto, 2016)

Program ini tersedia dalam GNU (*General Public License*) dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. XAMPP adalah singkatan yang masing-masing hurufnya adalah:

1. X : Simbol X di buat karena dapat dijalankan dibanyak program pada sistem operasi seperti, *windows, linux, Mac OS,* dan *Solaris*.
2. A : *Apache*, merupakan aplikasi web server. Tugas utama *Apache* menghasilkan halaman web yang benar kepada pengguna berdasarkan kode PHP yang dituliskan oleh pembuat halamnan web.
3. M : *MySQL*, merupakan aplikasi *database server*. Perkembangannya disebut *SQL* yang merupakan kepanjangan dari *Stuctured Query Language*. *SQL* merupakan bahasa terstruktur yang digunakan untuk mengolah *database*. Kita dapat memanfaatakan *MySQL* untuk menambahkan, mengubah, dan menghapus data yang berada dalam *database*.
4. P : PHP, bahasa pemrograman web. Bahasa pemrograman PHP merupakan bahasa untuk membuat halaman web yang bersifat *server-side-scripting*, PHP memungkinkan kita untuk membuat halaman web dinamais. Sistem manajemen basis data yang sering digunakan bersama PHP adalah *MySQL.*
5. P : *Pearl*, bahasa pemrograman untuk segala keperluan dikembangkan pertama oleh Larry Wall di mesin Unix.

##### 2.13. MySQL

Menurut Nugroho (2013), MySQL adalah *software* atau program *database* *server*. Sedangkan SQL adalah bahasa pemrogramannya, bahasa permintaan (*query*) dalam *database* *server* termasuk dalam MySQL itu sendiri. SQL juga dipakai dalam *software database server* lain, seperti SQL *Server*, Oracle, PostgreSQL dan lainnya.

Menurut Buana (2014), MySQLmerupakan *database* server yang paling sering digunakan dalam pemograman PHP. MySQLdigunakan untuk menyimpan data dalam *database* dan memanipulasi data-data yang diperlukan. Manipulasi data tersebut berupa menambah, mengubah, dan menghapus data yang berada dalam *database*.

##### 2.14. *Black Box Testing*

Menurut Nidhra dan Dondeti (2012), *Black box testing* juga disebut *functional testing*, sebuah teknik pengujian fungsional yang merancang *test case* berdasarkan informasi dari spesifikasi. Pengujian *Black Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Data uji dieksekusi pada perangkat lunak dan kemudian keluar dari perangkat lunak dicek apakah telah sesuai yang diharapkan.

### BAB III

### METODOLOGI PENELITIAN

#### 3.1 Metode Pengumpulan Data

Untuk pengumpulan sejumlah data yang diperlukan dalam penyusunan tugas akhir ini, penulis menggunakan beberapa metode diantaranya sebagai berikut:

1. Kepustakaan, dilakukan dengan cara mengumpulkan referensi yang berkaitan dengan objek penelitian referensi-referensi yang dirujuk dalam penelitian ini berkaitan dengan kajian data *text*.
2. Studi literatur, dilakukan dengan cara membaca berbagai literatur, dokumendokumen serta arsip yang ada kaitannya dengan judul tugas akhir.

#### 3.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada sistem ini adalah metode *Rational Unified Process* (RUP)*.* Tahap ini memiliki empat fase dalam pengembangan sistem, yaitu :

1. *Inception* (permulaan)

Pada fase ini dilakukan proses pengidentifikasian sistem dilakukan dengan menganalisis kebutuhan aplikasi, melakukan kajian terhadap penelitian yang terkait rekomendasi dosen pembimbing dan penguji menggunakan algoritma BM25.

1. *Elaboration* (perluasan/perencanaan)

Setelah menentukan ruang lingkup penelitian, tahap ini akan dilakukan perancangan dan analisis sistem menggunakan *flowchart* meliputi *flowchart* Algoritma BM25*.* Pada perancangan ini digunakan juga UML (*Unified Modeling Language*) yang meliputi *usecase, activity diagram,* dan *class diagram.* Selain melakukan analisis sistem dan perancangan menggunakan bahasa permodelan, pada tahap ini juga dilakukan perancangan tampilan *interface* dari sistem yang akan dibangun.

1. *Construction* (konstruksi)

Proses yang dilakukan pada tahap ini adalah mengimplementasikan kode sesuai perancagan yang telah dilakukan sebelumnya sehingga menjadi aplikasi yang dapat digunakan. Pada tahap ini akan dibangun semua tampilan yang ada pada

34

sistem rekomendasi dosen pembimbing dan penguji tugas akhir mengguanakan metode BM25, mulai dairu tampilan beranda, kumpulan skripsi, pemilihan dosen, daftar dosen dan daftar kata dasar .

1. *Transition* (Transisi)

Pada tahap ini akan dilakukan proses pengujian pada aplikasi yang telah dibangun.

#### 3.3 Waktu dan Tempat Penelitian

##### 3.3.1 Waktu Penelitian

Waktu pelaksanaan penelitian tugas akhir dilaksanakan pada bulan Februari 2020 . Rincian kegiatan dapat dilihat pada tabel dibawah ini :

#### Tabel 3.1 Waktu Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Uraian |  | | |  |  | |  |  | Waktu | | | |  | |  |  |  | |  |  |
| Februari | | |  | Maret | |  |  | April | | |  | Mei | |  |  | Juni | |  | Juli |
| 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 |
| 1 | *Inception* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | *Elaboration* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | *Construction* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | *Transition* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

##### 3.3.2 Tempat Penelitian

Penelitian tugas akhir bertempat di Jurusan Teknik Informatika Universitas Halu Oleo.

#### 3.4 Analisis Sistem

Analisis Sistem adalah suatu tahapan yang bertujuan untuk mengetahui dan mengamati siapa saja yang terlibat dalam sistem. Pembahasan yang ada pada analisis sistem ini yaitu analisis kebutuhan fungsional meliputi perancangan sistem menggunakan bahasa pemodelan UML (*Unified Modeling Language),* perancangan tampilan *interface* serta analisis kebutuhan nonfungsional yang meliputi kebutuhan perangkat keras dan perangkat lunak yang akan digunakan.

##### 3.4.1. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah segala bentuk data yang dibutuhkan oleh sistem agar sistem dapat berjalan sesuai dengan prosedur yang dibangun melalui perancangan sistem. Adapun kebutuhan fungsional dari sistem yang akan dibangun yaitu Perancangan diagram sistem menggunakan bahasa pemodelan UML (*Unified Modeling Language)* yang meliputi *flowchart* sistem, *flowchart* metode, *use case diagram, activity diagram, class diagram, serta sequence diagram* dan *Desain Interface* sistem.

##### 3.4.2. Analisis Kebutuhan Nonfungsional

Analisis kebutuhan nonfungsional adalah sebuah langkah dimana pembangun aplikasi menganalisis sumber daya kebutuhan untuk membangun aplikasi yang akan dibangun. Analisis kebutuhan nonfungsional yang dilakukan dibagi dalam dua tahap, yaitu anlisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak. Kebutuhan perangkat keras yaitu kebutuhan perangkat atau komponen yang dibutuhkan pada sistem dan perangkat lunak yaitu kebutuhan perangkat lunak untuk membantu agar komponen perangkat keras dapat berfungsi dan dapat dijalankan pada sistem.

##### 3.4.2.1.Kebutuhan Perangkat Keras

Untuk Menerapkan rancangan yang telah dijelaskan sebelumnya, dibutuhkan beberapa perangkat keras sebagai sarana untuk mengimplementasikan aplikasi yang dibangun. Berikut spesifikasi perangkat keras yang dibutuhkan :

#### Tabel 4.1 Spesifikasi perangkat keras

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Perangkat** | **Spesifikasi** |
| 1. | *Laptop* | *Acer Aspire V 13* |
| 2. | *Processor* | *Intel Core i3* |
| 3. | *Monitor* | *Monitor 14 inch* |
| 4. | *Memori* | *RAM 4 GB DDR3 L Memory* |
| 5. | *Harddisk* | *500 GB HDD* |

##### 3.4.2.2.Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan pada pembangunan ini, sebagai berikut: **Tabel 4.2 Spesifikasi perangkat lunak**

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Perangkat** | **Spesifikasi** |
| 1. | *Operating System* | *Windows 10* |
| 2. | *Sublime* | *Versi 3.0* |
| 3. | *Xampp* | *3.2.1* |
| 4. | *Mozila Firefox* | *58.0.2* |

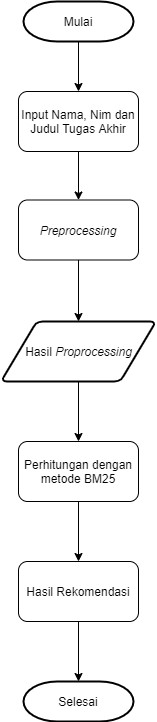
##### 3.5. Analisis Perancangan Sistem

Perancangan sistem yang akan dibangun terdiri atas perancanga *flowchart* dan perancanagan UML serta perancangan *User interface.*

###### 3.5.1. Analisis Perancangan *Flowchart*

Setelah menganalisis sistem, maka didapatkan *flowchart diagram* penerapan metode BM25 pada sistem rekomendasi dosen pembimbing dan penguji tugas akhir mahasiswa berdasarkan kompetensi dosen berbasis web. Adapun alur kerja *flowchart diagram* sistem adalah sebagai berikut:

1. *User* Menginput Nama, Nim dan Judul Tugas Akhir
2. Tahap Selanjutnya yaitu data inputan akan di proses pada tahap *preprocessing* dimana tahap tersebut bertujuan mem*filter*  kata-kata yang penting dari data input. Tahap *preprocessing*  akan menghasilkan sebuah data baru.
3. Hasil *preprocessing*  atau data baru kemudian diolah perhitungan jarak terdekat menggunakan rumus BM25*.* Jarak terdekat antara data dosen dengan data uji akan menjadi hasil rekomendasi.
4. Tahapan selanjutnya yaitu mendapatkan hasil rekomendasi dari sistem.



##### Gambar 4.1 *Flowchart Diagram* Sistem

###### 3.5.2. Analisis Perancangan UML

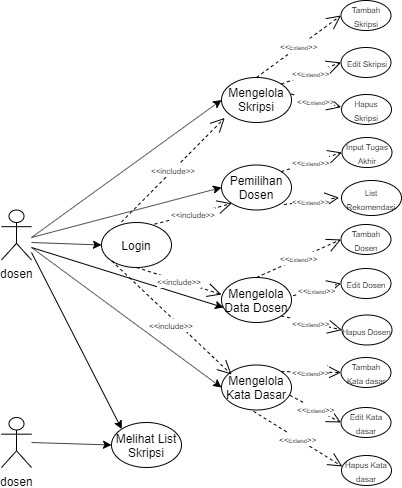
Aplikasi dibangun dengan menggunakan *Unified Modeling Language* (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram yang terdiri dari *Use Case Diagram, Class Diagram, Activty Diagram,* dan *Sequence Diagram.*

3.5.2.1. Use case diagram

*Use case diagram,* digunakan untuk memodelkan fungsionalitas-

fungsionalitas sistem yang dilihat dari pengguna yang ada di luar sistem (aktor).

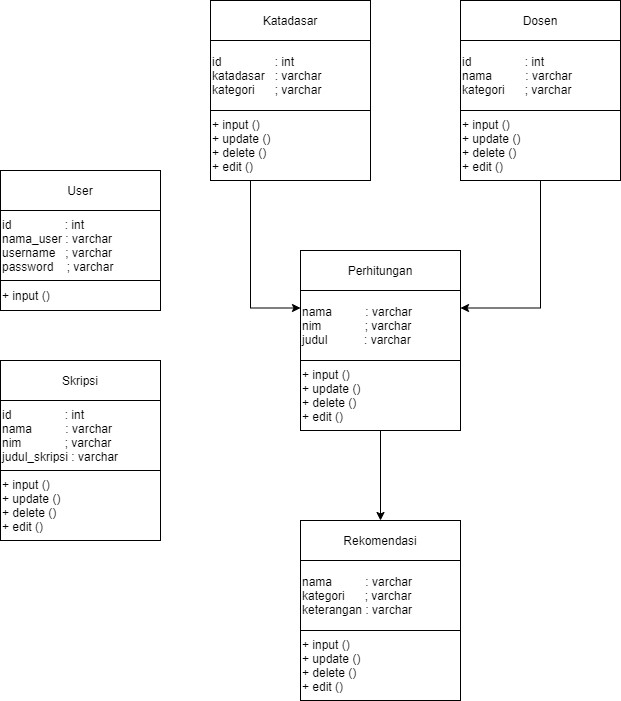
Berikut adalah *use case* *diagram* untuk sistem yang akan dibangun:



Mahasiswa

#### Gambar 4.2 *Usecase Diagram* Sistem

##### 3.5.2.2*. Class Diagram*



String

String

String

String

String

String

String

String

String

String

String

String

String

String

String

String

#### Gambar 4.3 *Class Diagram* Sistem

Pada sistem penentuan rekomendasi dosen pembimbing dan penguji memiliki

6 *class Diagram* yaitu *class user*, *class* Katadasar, *class* Dosen, *class* Skripsi, *class* Perhitungan dan *class* Rekomendasi.

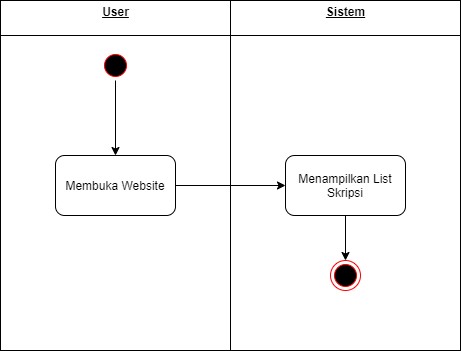
##### 3.5.2.3*. Activity Diagram*

**3.5.2.3.1*. Activity Diagram* Untuk *User***

##### 1. Beranda

Setelah menganalisis sistem, maka didapatkan rancangan diagram *activity* beranda yang ditunjukkan oleh Gambar 3.4. Adapun alur kerja menu beranda (*user*) adalah sebagai berikut:

1. *User* mengakses sistem.
2. Sistem menampilkan daftar List skripsi

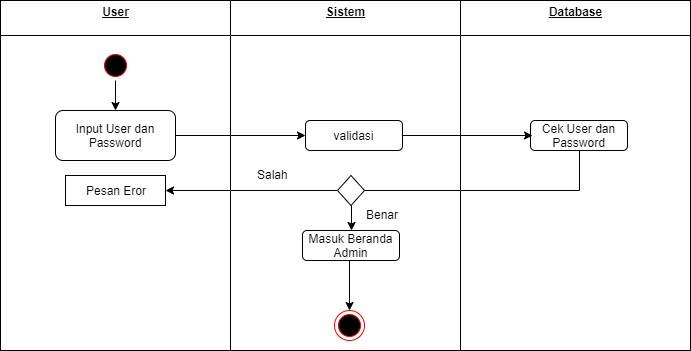


##### Gambar 4.4 *Activity Diagram* Beranda *User*

**3.5.2.3.2. *Activity Diagram* Untuk *Admin***

###### 1. Login (*Admin*)

Setelah menganalisis sistem, maka didapatkan rancangan diagram *activity* *Login* yang ditunjukkan oleh Gambar 4.6.



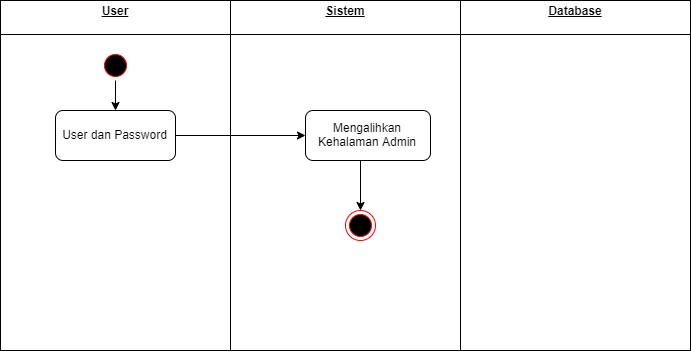
##### Gambar 4.5 *Activity Diagram* *Login Admin*

Adapaun alur kerja *login* adalah sebagai berikut :

1. *Admin* memasukkan *admin id* dan *password*.
2. Data akan divalidasi oleh sistem dengan mengecek kecocokan data yang ada pada *database*. Jika data *valid*, maka sistem akan menampilkan halaman beranda. Namu jika data tidak *valid*, maka sistem akan menampilkan pesan *error*, yang meminta *admin* memasukkan ulang *user id* dan *password*.

###### 2. Menu Beranda Admin

Setelah menganalisis sistem, maka didapatkan rancangan diagram *activity* beranda yang ditunjukkan oleh Gambar 4.6. Jika *admin id* dan *password* yang dimasukkan dalam *form login* benar, maka sistem akan mengalihkan halaman ke beranda khusus *admin*.

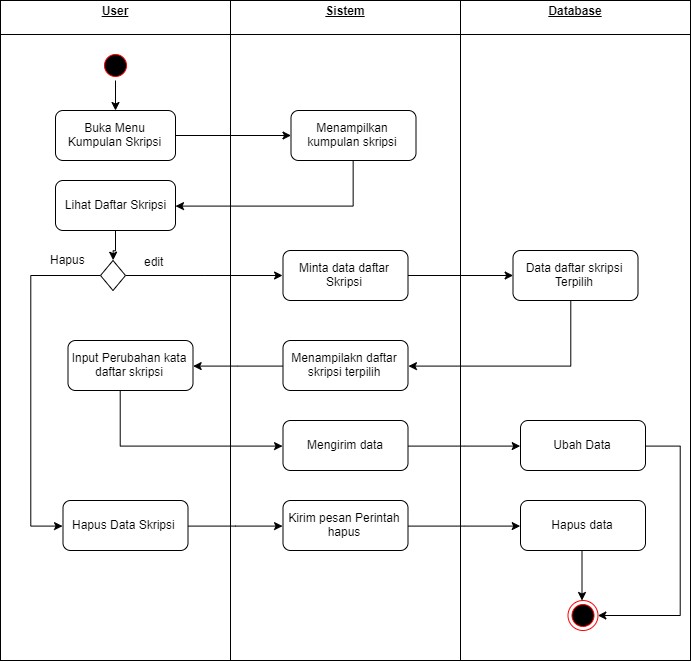


##### Gambar 4.6 *Activity Diagram* Beranda *Admin*

###### 3. Menu Kumpulan Skripsi

Berdasarkan analisis sistem, maka didapatkan rancangan diagram *activity* Kumpulan Skripsi yang ditunjukkan oleh Gambar 4.7.

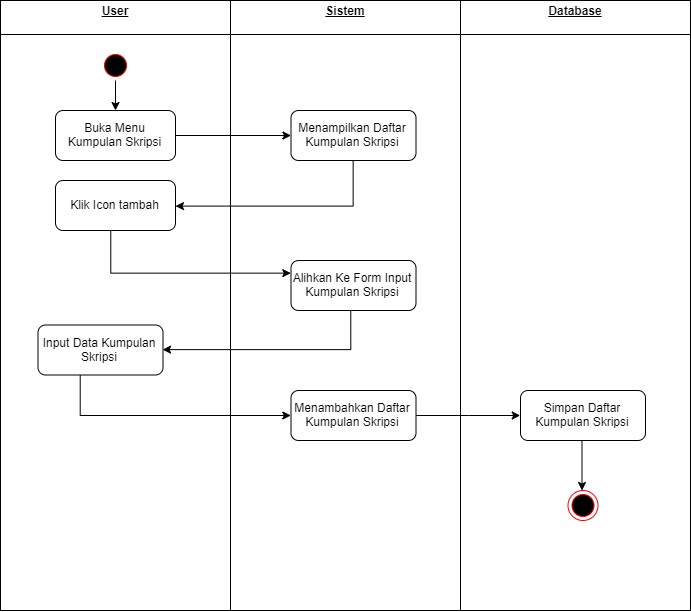
1. *Admin* membuka *menu* Kumpulan skripsi, yang secara bersamaan sistem akan mengirimkan perintah ke *database* untuk memilih tabel Kumpulan skripsi.
2. Kumpulan skripsi selanjutnya akan ditampilkan pada halaman *menu* daftar skripsi berupa tabel.
3. Jika *admin* memilih aksi ubah, maka akan diarahkan ke halaman *form* ubah skripsi.
4. Jika *admin* memilih aksi hapus, maka data yang terpilih untuk aksi tersebut akan segera dihapus.



#### Gambar 4.7 *Activity Diagram* Kumpulan Skripsi

##### 4. Menu Tambah Skripsi

Setelah menganalisis sistem, maka didapatkan rancangan diagram *activity* tambah skripsi yang ditunjukkan oleh Gambar 4.8. *Admin* memilih menu tambah skripsi. Kemudian sistem akan mengalihkan ke *form input* skripsi. Selanjutnya *admin* akan mengisi *textfield* serta atribut-atribut lain yang diperlukan dalam penginputan Tambah Skripsi.

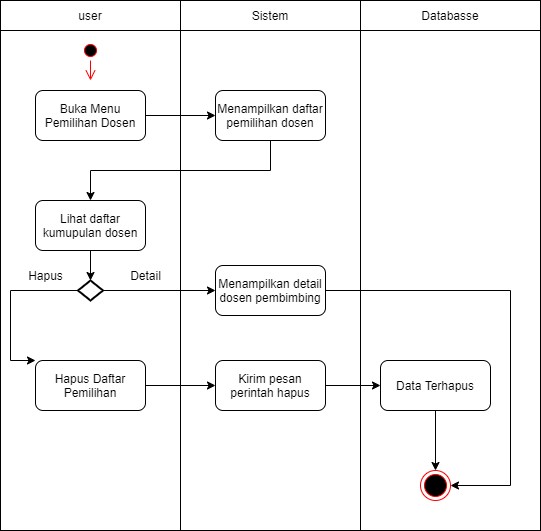


#### Gambar 4.8 *Activity Diagram* Tambah Skripsi

##### 5. Menu Pemilihan Dosen

Berdasarkan analisis sistem, maka didapatkan rancangan diagram *activity* Pemilihan Dosen yang ditunjukkan oleh Gambar 4.9.

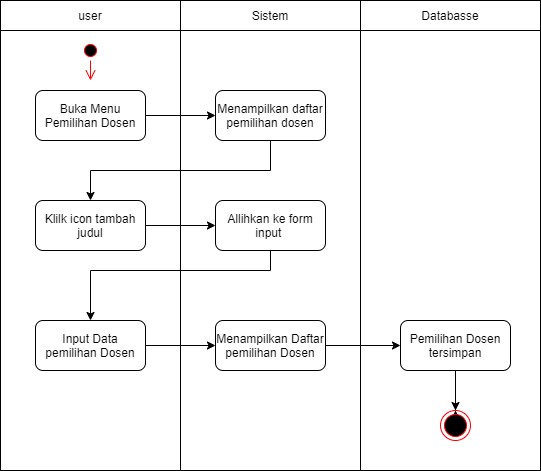
1. *Admin* membuka *menu* Pemilihan Dosen, yang secara bersamaan sistem akan mengirimkan perintah ke *database* untuk memilih tabel Pemilihan Dosen.
2. Pemilihan Dosen selanjutnya akan ditampilkan pada halaman *menu* Pemilihan Dosen berupa tabel.
3. Jika *admin* memilih aksi ubah, maka akan diarahkan ke halaman *form* ubah Pemilihan Dosen.
4. Jika *admin* memilih aksi hapus, maka data yang terpilih untuk aksi tersebut akan segera dihapus.



#### Gambar 4.9 *Activity Diagram* Pemilihan Dosen

##### 6. Menu Tambah Pemilihan Dosen

Setelah menganalisis sistem, maka didapatkan rancangan diagram *activity* tambah pemilihan dosen yang ditunjukkan oleh Gambar 4.10. *Admin* memilih menu tambah pemilihan dosen. Kemudian sistem akan mengalihkan ke *form input* Tambah Pemilihan dosen. Selanjutnya *admin* akan mengisi *textfield* serta atributatribut lain yang diperlukan dalam penginputan pemilihan dosen.

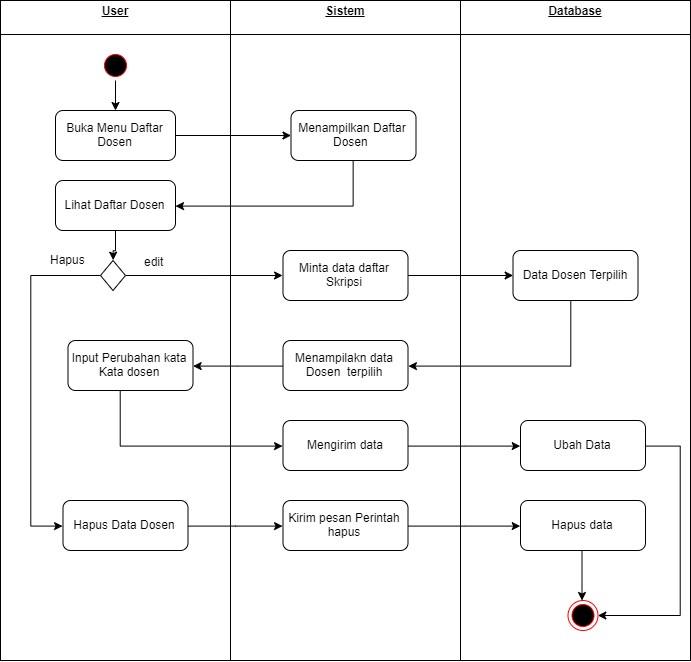


#### Gambar 4.10 *Activity Diagram* Tambah Pemilihan Dosen

##### 7. Menu Daftar Dosen

Berdasarkan analisis sistem, maka didapatkan rancangan diagram *activity* Daftar Dosen yang ditunjukkan oleh Gambar 4.11.

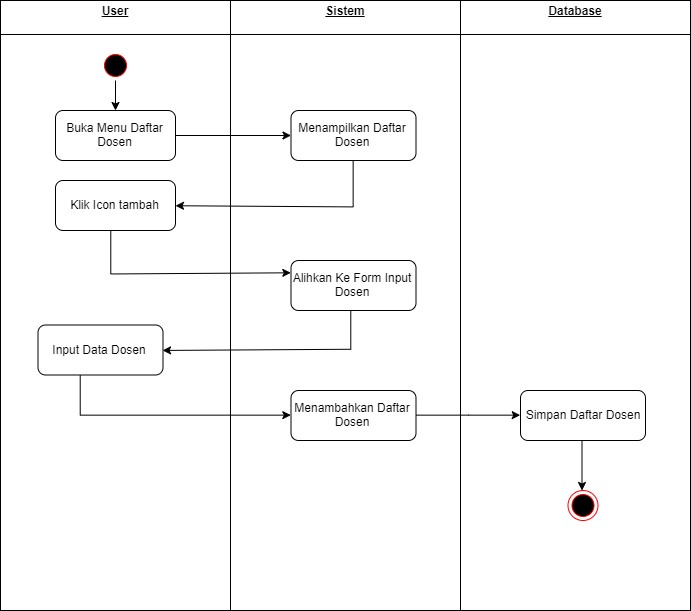
1. *Admin* membuka *menu* Daftar Dosen, yang secara bersamaan sistem akan mengirimkan perintah ke *database* untuk memilih tabel Daftar Dosen.
2. Daftar Dosen selanjutnya akan ditampilkan pada halaman *menu* Daftar Dosen berupa tabel.
3. Jika *admin* memilih aksi ubah, maka akan diarahkan ke halaman *form* ubah s Daftar Dosen.
4. Jika *admin* memilih aksi hapus, maka data yang terpilih untuk aksi tersebut akan segera dihapus.



#### Gambar 4.11 *Activity Diagram* Daftar Dosen

##### 8. Menu Tambah Daftar Dosen

Setelah menganalisis sistem, maka didapatkan rancangan diagram *activity* tambah Daftar Dosen yang ditunjukkan oleh Gambar 4.12. *Admin* memilih menu tambah Daftar Dosen. Kemudian sistem akan mengalihkan ke *form input* Tambah Pemilihan Dosen. Selanjutnya *admin* akan mengisi *textfield* serta atribut-atribut lain yang diperlukan dalam penginputan Daftar Dosen.

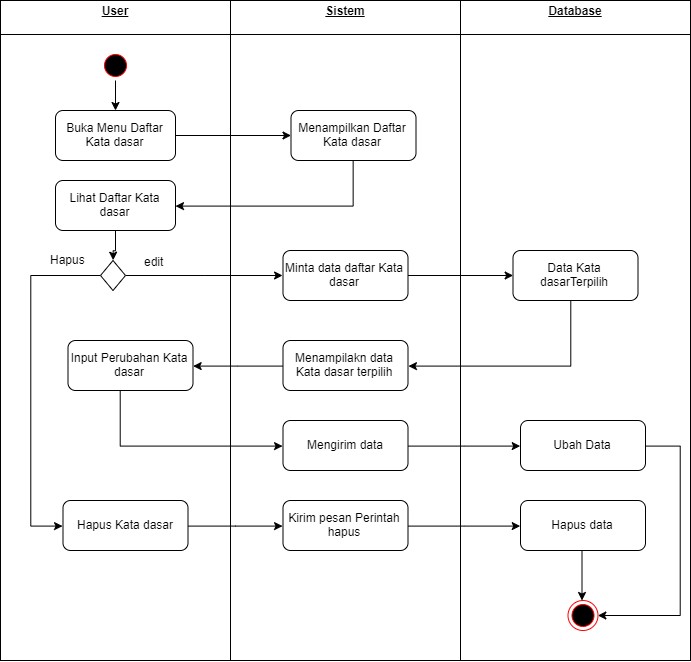


#### Gambar 4.12 *Activity Diagram* Tambah Daftar Dosen

##### 9. Menu Daftar Kata Dasar

Berdasarkan analisis sistem, maka didapatkan rancangan diagram *activity* daftar kata dasar yang ditunjukkan oleh Gambar 4.13. Adapun alur kerja *form* daftar kata dasar adalah sebagai berikut :

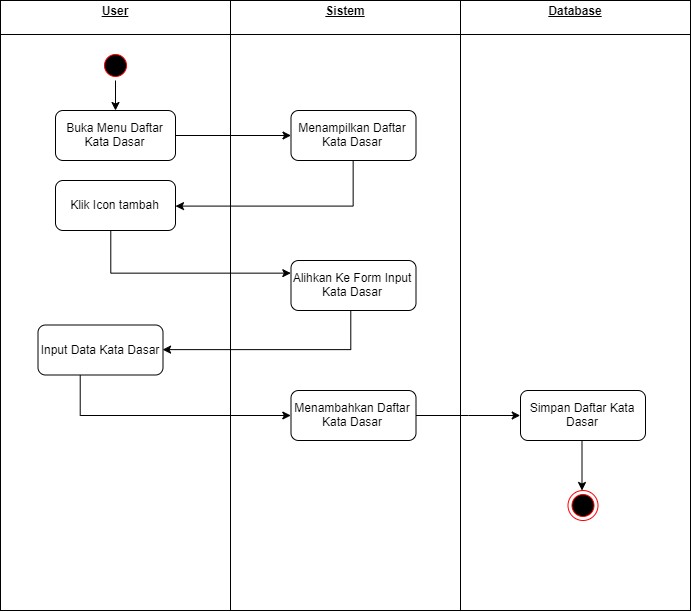
1. *Admin* membuka *menu* daftar kata dasar, yang secara bersamaan sistem akan mengirimkan perintah ke *database* untuk memilih tabel daftar kata dasar.
2. Daftar kata dasar selanjutnya akan ditampilkan pada halaman *menu* daftar kata dasar berupa tabel.
3. Jika *admin* memilih aksi ubah, maka akan diarahkan ke halaman *form* ubah kata dasar.
4. Jika *admin* memilih aksi hapus, maka data yang terpilih untuk aksi tersebut akan segera dihapus.



#### Gambar 4.13 *Activity Diagram* Daftar Kata Dasar

##### 10. Menu Tambah Daftar Kata Dasar

Setelah menganalisis sistem, maka didapatkan rancangan diagram *activity* tambah kata dasar yang ditunjukkan oleh Gambar 4.14.



#### Gambar 4.14 *Activity Diagram* Tambah Daftar Kata Dasar

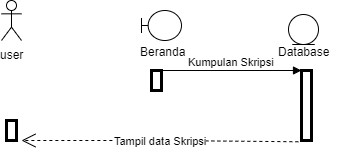
*Admin* memilih menu tambah kata dasar. Kemudian sistem akan mengalihkan ke *form input* tambah kata dasar. Selanjutnya *admin* akan mengisi *textfield* serta atribut-atribut lain yang diperlukan dalam penginputan tambah kata dasar.

**3.5.2.4*. Sequence Diagram***

##### 1. Beranda *User*

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequance* beranda yang ditunjukkan oleh Gambar 4.15. Adapun alur kerja menu beranda adalah sebagai berikut :

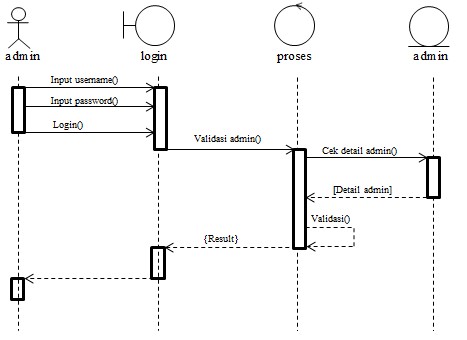
a. *User* mengakses menu beranda.



##### Gambar 4.15 *Sequence Diagram* Beranda *User*

###### 1. *Login*

Setelah menganalisis sistem, maka didapatkan rancangan diagram *Sequance* *login*  yang ditunjukkan oleh Gambar 4.16.



##### Gambar 4.16 *Sequence Diagram* *Login*

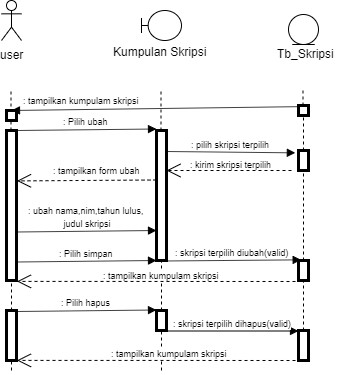
Adapun alur kerja *login* adalah sebagai berikut :

1. *Admin* memasukkan *admin id* dan *password*.
2. Data akan divalidasi oleh sistem dengan mengecek kecocokan data yang ada pada *database*. Jika data *valid*, maka sistem akan menampilkan halaman beranda. Namun jika data tidak *valid*, maka sistem akan menampilkan pesan *error*, yang meminta *admin* memasukkan ulang *admin id* dan *password*.

###### 2. Kumpulan Skripsi

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequance* daftar skripsiyang ditunjukkan oleh Gambar 4.17. Adapun alur kerja daftar skripsiadalah sebagai berikut :

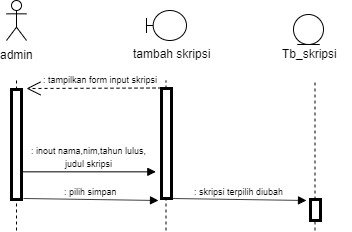
1. *Admin* membuka *menu* skripsi berupa tabel.
2. Jika *admin* memilih aksi ubah, maka akan diarahkan ke halaman *form* ubah skripsi.
3. Jika *admin* memilih aksi hapus, maka data yang terpilih untuk aksi tersebut akan segera dihapus.



#### Gambar 4.17 *Sequence Diagram* Kumpulan Skripsi

##### 3. Tambah Skripsi

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequance* tambah skripsiyang ditunjukkan oleh Gambar 4.18. *Admin* memilih menu tambah skripsi. Kemudian sistem akan mengalihkan ke *form input* skripsi. Selanjutnya *admin* akan mengisi *textfield* serta atribut-atribut lain yang diperlukan dalam penginputan skripsi.

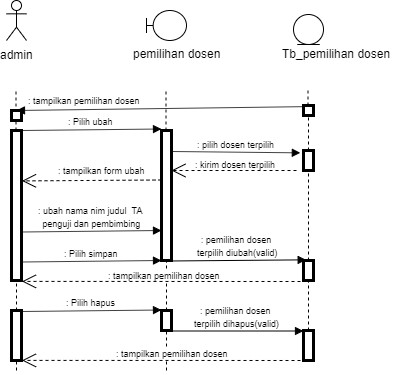


#### Gambar 4.18 *Sequence Diagram* Tambah Skripsi

##### 4. Pemilihan Dosen

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequance* pemilihan dosenyang ditunjukkan oleh Gambar 4.19. Adapun alur kerja pemilihandosen adalah sebagai berikut :

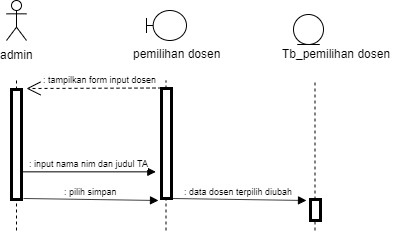
1. *Admin* membuka menu pemilihan dosen berupa tabel.
2. Jika *admin* memilih aksi ubah, maka akan diarahkan ke halaman *form* ubah pemilihan dosen.
3. Jika *admin* memilih aksi hapus, maka data yang terpilih untuk aksi tersebut akan segera dihapus.



#### Gambar 4.19 *Sequence Diagram* Pemilihan Dosen

##### 5. Tambah Pemilihan Dosen

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequance* Tambah pemilihan dosenyang ditunjukkan oleh Gambar 4.20. *Admin* memilih menu tambah skripsi. Kemudian sistem akan mengalihkan ke *form input* pelihan dosen. Selanjutnya *admin* akan mengisi *textfield* serta atribut-atribut lain yang diperlukan dalam penginputan pemihan dosen.

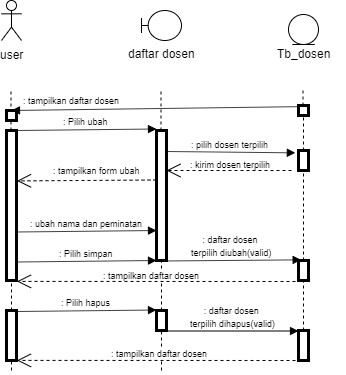


#### Gambar 4.20 *Sequence Diagram* Tambah Pemilihan Dosen

##### 6. Daftar Dosen

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequance* Daftar dosenyang ditunjukkan oleh Gambar 4.21. Adapun alur kerja daftar dosenadalah sebagai berikut :

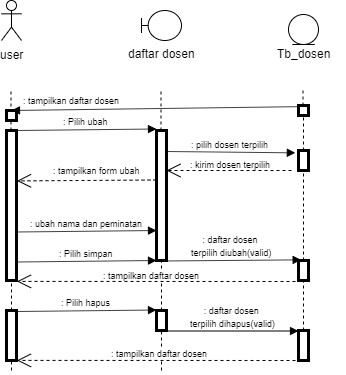
1. *Admin* membuka *menu* pemilihan dosen berupa tabel.
2. Jika *admin* memilih aksi ubah, maka akan diarahkan ke halaman *form* ubah Daftar dosen.
3. Jika *admin* memilih aksi hapus, maka data yang terpilih untuk aksi tersebut akan segera dihapus.



#### Gambar 4.21 *Sequence Diagram* Daftar Dosen

##### 7. Tambah Daftar Dosen

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequance* Tambah Daftar dosenyang ditunjukkan oleh Gambar 4.22. *Admin* memilih menu tambah skripsi. Kemudian sistem akan mengalihkan ke *form input* daftar dosen. Selanjutnya *admin* akan mengisi *textfield* serta atribut-atribut lain yang diperlukan dalam penginputan Tambah Daftar dosen .

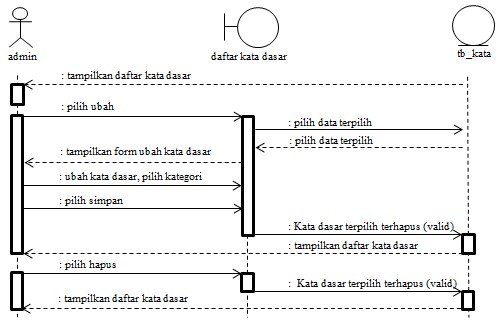


#### Gambar 4.22 *Sequence Diagram* Tambah Daftar Dosen

##### 8. Daftar kata Dasar

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequance* daftar kata dasar yang ditunjukkan oleh Gambar 4.23. Adapun alur kerja *form* daftar kata dasar adalah sebagai berikut :

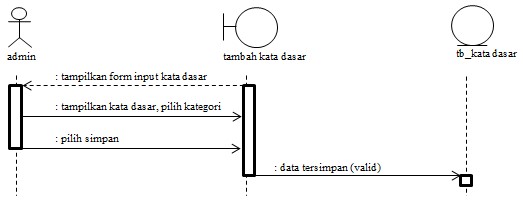
1. *Admin* membuka *menu* daftar kata dasar, yang secara bersamaan sistem akan mengirimkan perintah ke *database* untuk memilih tabel daftar kata dasar.
2. Daftar kata dasar selanjutnya akan ditampilkan pada halaman *menu* daftar kata dasar berupa tabel.
3. Jika *admin* memilih aksi ubah, maka akan diarahkan ke halaman *form* ubah kata dasar.
4. Jika *admin* memilih aksi hapus, maka data yang terpilih untuk aksi tersebut akan segera dihapus.



#### Gambar 4.23 *Sequance Diagram* Daftar Kata Dasar

##### 9. Tambah Kata Dasar

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequance* tambah kata dasar yang ditunjukkan oleh Gambar 4.24. *Admin* memilih menu tambah kata dasar. Kemudian sistem akan mengalihkan ke *form input* tambah kata dasar. Selanjutnya *admin* akan mengisi *textfield* serta atribut-atribut lain yang diperlukan dalam penginputan tambah kata dasar.



#### Gambar 4.24 *Sequence Diagram* Tambah Kata Dasar

##### 3.5.3. Perancangan *User Interface*

Rancangan antar muka merupakan desain awal dari sistem yang akan berinteraksi langsung dengan *user.* Rancangan ini diperlukan agar sistem yang dibangun bisa lebih terarah dan jelas dari sisi tampilan. **4.2.3.1.Antarmuka *User***

**1.**

**Beranda**

**Gambar**

**4**

**.**

**2**

**5**

**Menu Beranda**

***User***



***2. Login***

##### Gambar 4.26 merupakan rancangan antarmuka halaman *login*

Rekomendasi dosen pembimbing dan penguji. Halamantersebut berisi:

1. *Textfield* *username*. Berfungsi sebagai tempat mengisikan *username* pengguna.
2. *Textfield password*. Berfungsi sebagai tempat mengisikan *password* pengguna.
3. *Button* masuk. Berfungsi sebagai pemberi perintah kepada sistem untuk mencocokkan data *username* dan *password* dengan data pengguna yang telah terdaftar di *database*.

**Gambar**

**4**

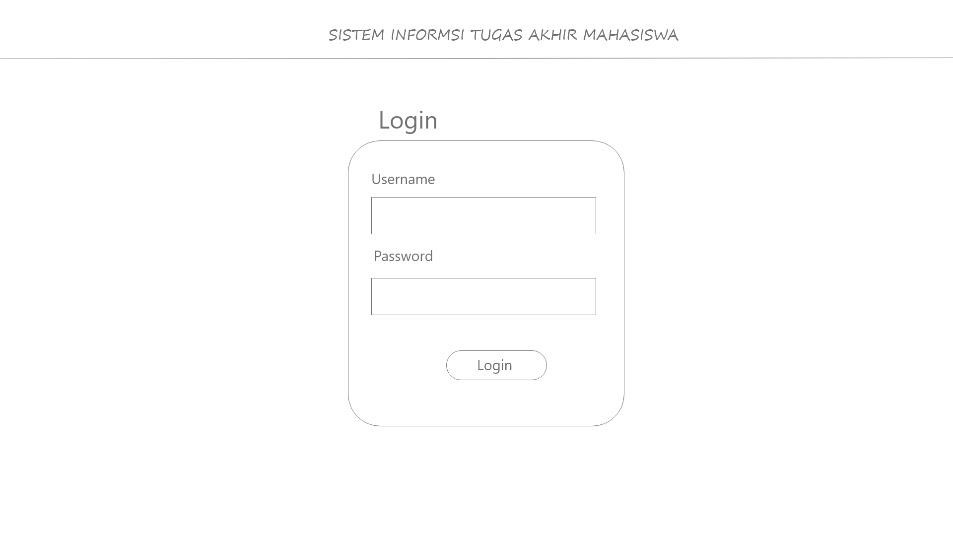
**.**

**2**

**6**

**Menu**

***Login***



**3.5.3.2.Antarmuka *Admin***

###### 1. Beranda

Pada halaman beranda, ditampilkan tulisan selamat datang. Selain itu, pada halaman *administrator* terdapat beberapa menu yaitu, beranda,

Kumpulan skripsi, Pemilihan Dosen, daftar dosen dan daftar kata dasar,

**Gambar**

**4**

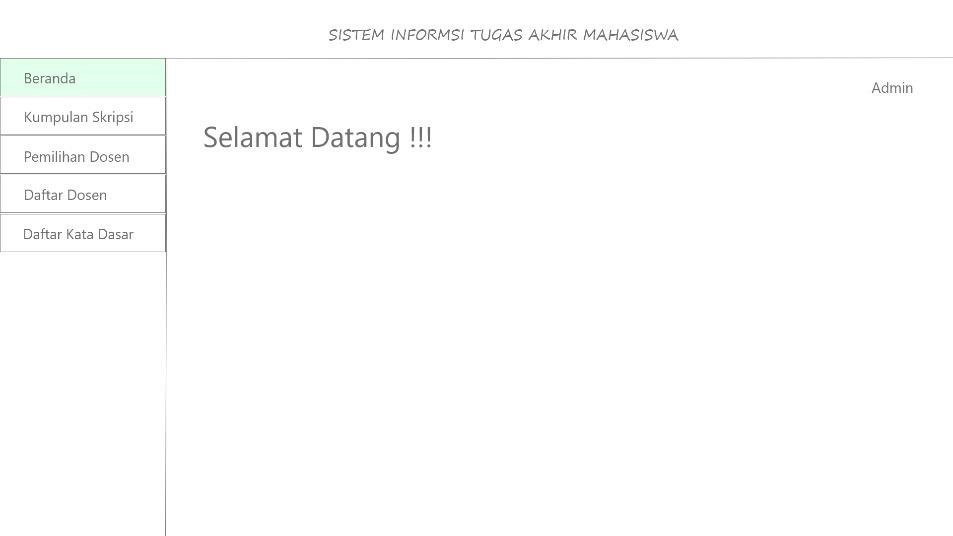
**.**

**2**

**7**

**Menu Beranda**

***Admin***



###### 2. Kumpulan Skripsi

Gambar 4.28 merupakan rancangan antarmuka halamanKumpulan skripsi. Halaman tersebut berisi :

1. Tabel yang berisi daftar skripsi yang datanya telah tersimpan pada *database*.
2. *Field* aksi. Berisi dua *button* yaitu ubah, dan hapus.
3. *Button* ubah, berfungsi sebagai pemberi perintah ke sistem agar menampilkan halaman ubah skripsi yang tampilannya sama dengan halaman *input* skripsi.
4. *Button* hapus, berfungsi menghapus skripsi terpilih pada *database*.

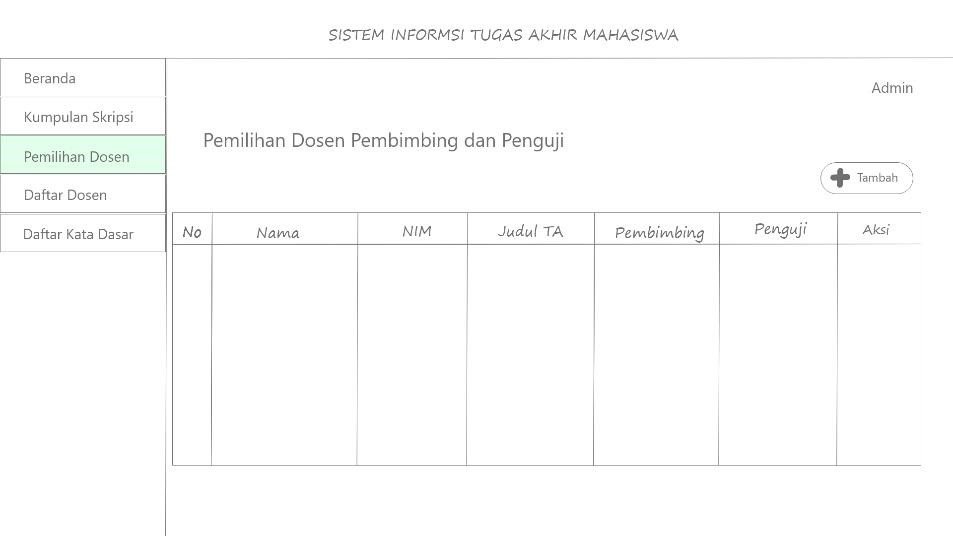


#### Gambar 4.28 Menu Kumpulan Skripsi

##### 3. Pemilihan Dosen

Gambar 4.29 merupakan rancangan antarmuka halamanPemilihan Dosen. Halaman tersebut berisi :

1. Tabel yang berisi Pemilihan Dosen yang datanya telah tersimpan pada *database*.
2. *Field* aksi. Berisi dua *button* yaitu ubah, dan hapus.
3. *Button* ubah, berfungsi sebagai pemberi perintah ke sistem agar menampilkan halaman ubah Pemilihan Dosen yang tampilannya sama dengan halaman *input* Pemilihan Dosen.
4. *Button* hapus, berfungsi menghapus Pemilihan Dosen terpilih pada *database*.



#### Gambar 4.29 Menu Pemilihan Dosen

##### 4. Tambah Pemilihan Dosen

Gambar 4.30 merupakan rancangan antarmuka halamantambah daftar Dosen. Halaman tersebut berisi :

1. *Textfield* Nama. Berfungsi sebagai tempat mengisi Nama dan NIM mahasiswa yang mengajukan judul tugas akhir.
2. *Text Area* kategori*.* Berfungsi sebagai tempat memilih menginput Judul Tugas Akhir.
3. *Button* Simpan. Berfungsi sebagai pemberi perintah kepada sistem untuk menyimpan data yang telah dimasukkan.



#### Gambar 4.30Menu Tambah Pemilihan Dosen

##### 5. Daftar Dosen

Gambar 4.31 merupakan rancangan antarmuka halamandaftar kata dasar. Halaman tersebut berisi :

1. Tabel yang berisi daftar Dosen yang datanya telah tersimpan pada *database*.
2. Pada tabel ditampilkan daftar dosen,kategori 1,kategori 2,kategori 3.
3. *Field* aksi. Berisi dua *button* yaitu ubah, dan hapus.
4. *Button* ubah, berfungsi sebagai pemberi perintah ke sistem agar menampilkan halaman ubah Daftar Dosen yang tampilannya sama dengan halaman *input* daftar dosen.
5. *Button* hapus, berfungsi menghapus Daftar Dosen terpilih pada *database*.

**Gambar**

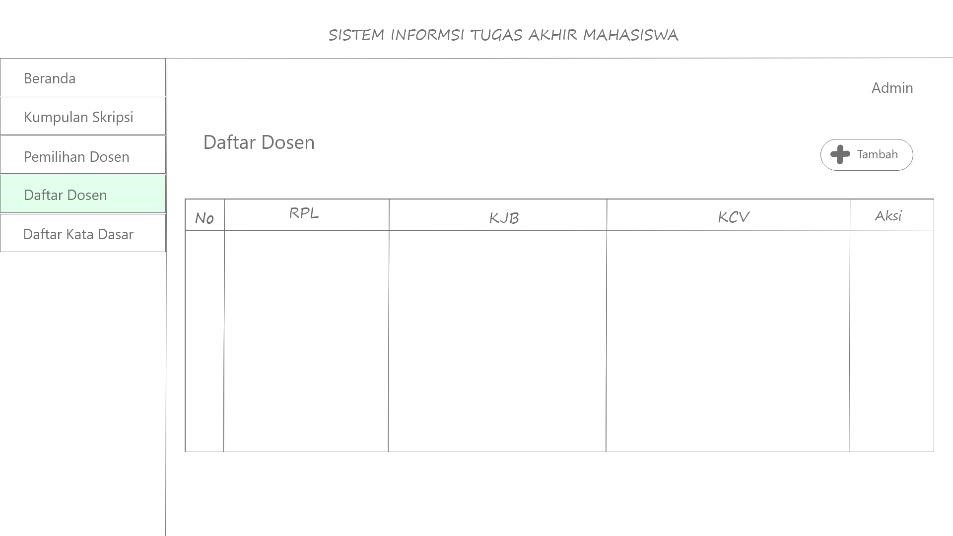
**4**

**.**

**3**

**1**

**Menu Daftar Dosen**



##### 6. Tambah Daftar Dosen

Gambar 4.32 merupakan rancangan antarmuka halamantambah daftar Dosen. Halaman tersebut berisi :

1. *Textfield* Nama. Berfungsi sebagai tempat mengisi Nama Dosen.
2. *Checkbox* kategori*.* Berfungsi sebagai tempat memilih kategori yang sesuai dengan Nama Dosen..
3. *Button* Simpan. Berfungsi sebagai pemberi perintah kepada sistem untuk menyimpan data yang telah dimasukkan.

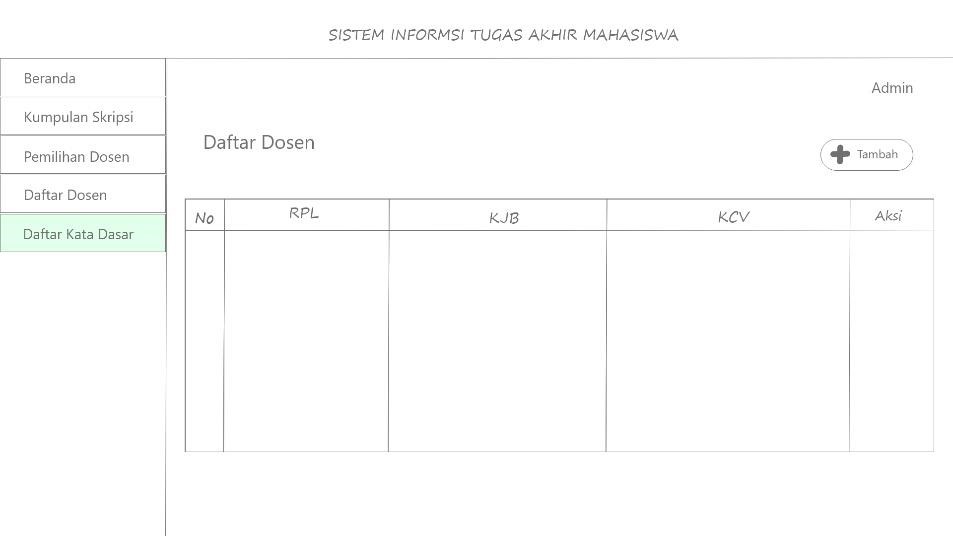


#### Gambar 4.32 Menu Tambah Daftar Dosen

##### 7. Daftar Kata Dasar

Gambar 4.33 merupakan rancangan antarmuka halamandaftar kata dasar. Halaman tersebut berisi :

1. Tabel yang berisi daftar kata dasar yang datanya telah tersimpan pada *database*.
2. Pada tabel ditampilkan kata dasar, kategori 1, kategori 2, kategori 3.
3. *Field* aksi. Berisi dua *button* yaitu ubah, dan hapus.
4. *Button* ubah, berfungsi sebagai pemberi perintah ke sistem agar menampilkan halaman ubah kata dasar yang tampilannya sama dengan halaman *input* kata dasar.
5. *Button* hapus, berfungsi menghapus kata dasar terpilih pada *database*.



#### Gambar 4.33Menu Daftar Kata Dasar

##### 8. Tambah Daftar Kata Dasar

Gambar 4.34 merupakan rancangan antarmuka halamantambah kata dasar. Halaman tersebut berisi :

1. *Textfield* kata dasar. Berfungsi sebagai tempat mengisi kata dasar.
2. *Checkbox* kategori*.* Berfungsi sebagai tempat memilih kategori yang sesuai dengan kata dasar.
3. *Button* Simpan. Berfungsi sebagai pemberi perintah kepada sistem untuk menyimpan data yang telah dimasukkan.



**Gambar 4.34Menu Tambah Daftar Kata Dasar**

102

### DAFTAR PUSTAKA

Adi Nugroho. 2010. Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP. Andi. Yogyakarta

Anastasia Diana, Lilis Setiawati. 2016. Sistem Informasi Akuntansi, Perancangan, Prosedur dan Penerapan. Edisi 1. Yogyakarta: Andi Yogyakarta

Arshall B. Romney dan Paul John Steinbart. 2014, Sistem Informasi Akuntansi: Accounting Information Systems (Edisi 13), Prentice Hall

Aswati, S., Mulyani, N., Siagian, Y., & Syah, A.Z. 2015. Peranan Sistem Informasi

Dalam Perguruan T inggi. Jurnal Teknologi Dan Sistem Informasi, Vol. 1. No.2 , 79-86

Bintana, R. R. (2012). Penerapan Model Okapi Bm25 Pada Sistem Temu Kembali Informasi, 273–279*.*

Buana, I Komang Setia. 2014. Jago pemrograman PHP. Dunia Komputer, Jakarta, Indonesia.

Darmawan, Deni. 2013. Metode Penelitian Kuantitatif. Bandung: PT Remaja Rosdakarya.

Firayati, Sarita, M. I., & Statiswaty. (2016). Metode Weighted Product (WP). *semanTik*, *2*(1), 247–256. http://doi.org/10.1109/42.640738

Hakim, Lukmanul.2010.Membangun Web Berbasis PHP dengan Framework Codeigniter.Yogyakarta : Lokomedia.

Harlian, Milkha. 2006. Text Mining. http://lecturer.eepis-

its.edu/~iwanarif/kuliah/dm/6Text%20Mining.pdf

Hartono, Bambang. 2013. Sistem Informasi Manajemen Berbasis Komputer.

Jakarta: Rineka Cipta

Hearst, Marti. (2003). What is Text Mining? *Text Mining in Practice with R*, 1–15.

<https://doi.org/10.1002/9781119282105.ch1>

Hutahaean, Jeperson. 2015. Konsep Sistem Informasi. Yogyakarta.

Kiswanto, Christine. Florence. 2016. “Sistem Informasi Akademik *Sub-*Sistem : *Utility* Dan Epsbed”. *Skripsi*. Fakultas Teknik, Teknik Informatika, Universitas Halu Oleo.

Kurniawan, B., Effendi, S., Sitompu, O.S. 2012. *Klasifikasi Konten Berita Dengan Metode Text Mining*. Universitas Sumatera Utara, Medan.

Laengge, I., Wowor, H. F., & Putro, M. D. (2016). Sistem Pendukung Keputusan Dalam Menentukan Dosen Pembimbing Skripsi. *Jurnal Teknik Informatika*,

*9*(1).

Lukman, Andi, 2012. Penentuan Pembimbing dan Penguji Skripsi Berdasarkan

Spesifikasi Keahlian Dosen Menggunakan Logika Fuzzy. STIMED Nusa

Palapa.

M. Dra. Hj. Yulia Djahir dan S. M. Dewi Pratita, bahan Ajar Sistem Informasi Manajemen, Yogyakarta.

Mulyani, Sri. 2016. Sistem Informasi Manajemen. Bandung: Abdi Sistematika.

Nidhra, Srinivas dan Jagruthi Dondeti. 2012. Black box and White box Testing Techniques – A Literature Review. Internasional Journal of Embedded System and Applications (IJESA) Vol.2, No.2.

Oleo Halu U (2019). Peraturan Rektor Universitas Halu Oleo Nomor : 1 Tahun 2019 Tentang: Tugas Akhir. Kendari : Universitas Halu Oleo. Hal 56.

Pardede, J., Gustiana Husada, M., & Riansyah, R. (2015). Implementasi Dan

Perbandingan Metode Okapi BM25 Dan PLSA Pada Aplikasi Information Retrieval, 1–10.

Sebastian, D., Rachmat, A., & Sudiarto Raharjo, W. (2011). Implementasi Algortima Okapi BM25 dan K-Means untuk Mencari Relevansi Artikel.

Septiana, I., Irfan, M., Atmadja, A. R., & Subaeki, B. (2016). Sistem Pendukung Keputusan Penentu Dosen Penguji Dan Pembimbing Tugas Akhir Menggunakan Fuzzy Multiple Attribute Decision Making dengan Simple Additive Weighting (Studi Kasus: Jurusan Teknik Informatika UIN SGD

Bandung). *Jurnal Online Informatika*, *1*(1), 43. http://doi.org/10.15575/join.v1i1.10

Sukamto, Rosa A. dan M. Salahuddin. 2015. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika.

Triawati, C. 2009. *Metode Pembobotan Statistical Concept Based untuk Klastering dan Kategorisasi Dokumen Berbahasa Indonesia*. Institut Teknologi Telkom, Bandung.

Yudanto. (2009). Materi PKI Pengertian Tugas Akhir */ FIK UNY Yudanto / FIK UNY*. 6–7.

.