

**PENGEMBANGAN EDITOR BASIS PENGETAHUAN
UNTUK SISTEM BERBASIS KASUS**



Disusun Oleh:

N a m a : Tengku Muaz Abdussalam

NIM : 16523058

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2020

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGEMBANGAN EDITOR BASIS PENGETAHUAN

UNTUK SISTEM BERBASIS KASUS

TUGAS AKHIR

ISLAM

UNIVERSITAS

INDONESIA



Disusun Oleh:

N a m a : Tengku Muaz Abdussalam

NIM : 16523058

Yogyakarta, 30 April 2020

Pembimbing,

(Sri Kusumadewi Dr. S. Si., M. T.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGEMBANGAN EDITOR BASIS PENGETAHUAN

UNTUK SISTEM BERBASIS KASUS

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk
memperoleh gelar Sarjana Komputer dari Program Studi Informatika
di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 30 April 2020

Tim Penguji

Sri Kusumadewi Dr. S.Si., M.T.

Anggota 1

Taufiq Hidayat, S.T., M.C.S.

Anggota 2

Aridhanyati Arifin, S.T., M.Cs.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Tengku Muaz Abdussalam

NIM : 16523058

Tugas akhir dengan judul:

**PENGEMBANGAN EDITOR BASIS PENGETAHUAN
UNTUK SISTEM BERBASIS KASUS**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 30 April 2020



(Tengku Muaz Abdussalam)

HALAMAN PERSEMBAHAN

Alhamdulillah dan puji syukur atas nikmat dan kesempatan yang telah Allah SWT titipkan kepada penulis sehingga penulis dapat menyelesaikan tugas akhir ini. Tidak lupa juga shalawat dan salam penulis panjatkan kepada junjungan Nabi Muhammad SAW yang telah membawa umat manusia dari kegelapan menuju cahaya.

Ucapan terima kasih yang sebesar-besarnya dipersembahkan kepada kedua orang tua penulis yang telah memberikan segalanya yang tiada henti kepada penulis hingga sekarang. Dengan penuh perasaan bangga penulis persembahkan karya ilmiah ini. Semoga dengan prestasi dan capaian kecil ini dapat membuat kedua orang tua penulis bahagia dan bangga.



HALAMAN MOTO

“Sesungguhnya sesudah kesulitan itu ada kemudahan.”

(Al-Insyirah: 5)



KATA PENGANTAR

Assalaamu'alaikum warahmatullaahi wabarakatuh

Alhamdulillah Robbil 'Alamin, puji dan syukur kehadiran Allah SWT yang mana dengan limpahan rahmat, karunia serta hidayah-Nya kita masih diberikan nikmat iman dan Islam. Dengan pertolongan-Nya juga lah penulis dapat menyelesaikan laporan tugas akhir yang berjudul “Pengembangan Editor Basis Pengetahuan Untuk Sistem Berbasis Kasus”. Tidak lupa juga kita kirimkan shalawat dan salam kepada junjungan kita Nabi Muhammad SAW beserta keluarga dan para sahabatnya. Semoga kita termasuk golongan umatnya yang selamat di dunia dan di akhirat kelak.

Laporan tugas akhir ini dibuat dan disusun dalam rangka mengakhiri masa pendidikan jenjang Strata 1 Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Laporan ini dapat terselesaikan secara sempurna berkat do'a, bimbingan, bantuan, dan dukungan yang diberikan dari berbagai kalangan. Oleh karena itu, dengan rasa bahagia penulis mengucapkan terimakasih kepada:

1. Bapak Prof. Fathul Wahid, ST., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
2. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Program Studi Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Ibu Sri Kusumadewi Dr. S.Si., M.T., selaku dosen pembimbing tugas akhir penulis yang telah sabar dalam membimbing penulis dari awal hingga akhir hingga laporan tugas akhir ini bisa terwujud.
5. Bapak dan Ibu Dosen Program Studi Informatika yang telah memberikan ilmu dan pelajaran kepada penulis.
6. Ibu Hazhira Qudsyi, S.Psi., M.A., selaku pakar dan narasumber yang telah berkenan meluangkan waktunya untuk penulis dalam rangka mengumpulkan data.
7. Ibu Hepi Wahyuningsih, S.Psi., M.Si, selaku pakar dan narasumber yang telah memberikan masukan dan saran terhadap sistem yang penulis kembangkan.
8. Kedua orang tua penulis, H. Tengku Darmawan serta Hj. Susilawati yang selalu memberikan do'a, semangat, dan dukungan kepada penulis.
9. Kakak dan adik penulis, Tengku Nurfa Rahim dan Tengku Ayyash Faisal.
10. Keluarga besar yang tidak bisa disebutkan satu persatu.

11. Rekan-rekan kontrakan, rekan-rekan badminton, dan rekan-rekan grup *chat* ABC semuanya.
12. Semua pihak yang penulis cintai dan sayangi, yang tidak dapat penulis sebutkan satu persatu.

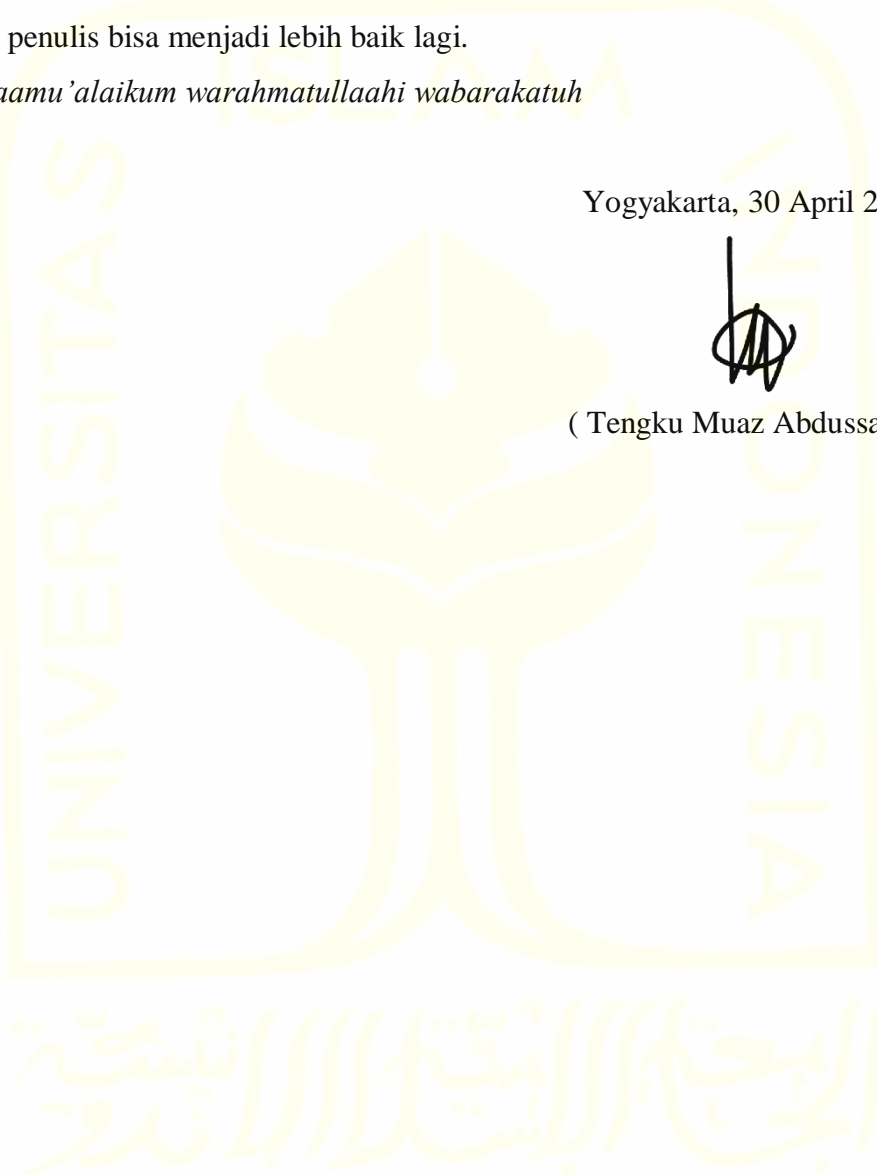
Penulis adalah manusia biasa yang tidak pernah luput dari salah sehingga laporan tugas akhir ini sudah pasti tidak lepas dari kekurangan. Oleh karena itu, penulis memohon maaf yang sebesar-besarnya. Penulis sangat mengharapkan tanggapan, kritik, dan saran dari pembaca agar kedepannya penulis bisa menjadi lebih baik lagi.

Wassalaamu'alaikum warahmatullaahi wabarakatuh

Yogyakarta, 30 April 2020



(Tengku Muaz Abdussalam)



SARI

Seorang pakar memiliki kemampuan untuk memberikan solusi dari sebuah permasalahan yang ia analisis, bahkan di era modern seperti sekarang ini komputer juga mampu memberikan solusi layaknya seorang pakar. Komputer tersebut dibekali sistem yang di dalamnya terdapat basis pengetahuan. Basis pengetahuan tersebut dirancang dan disusun menggunakan sebuah editor basis pengetahuan.

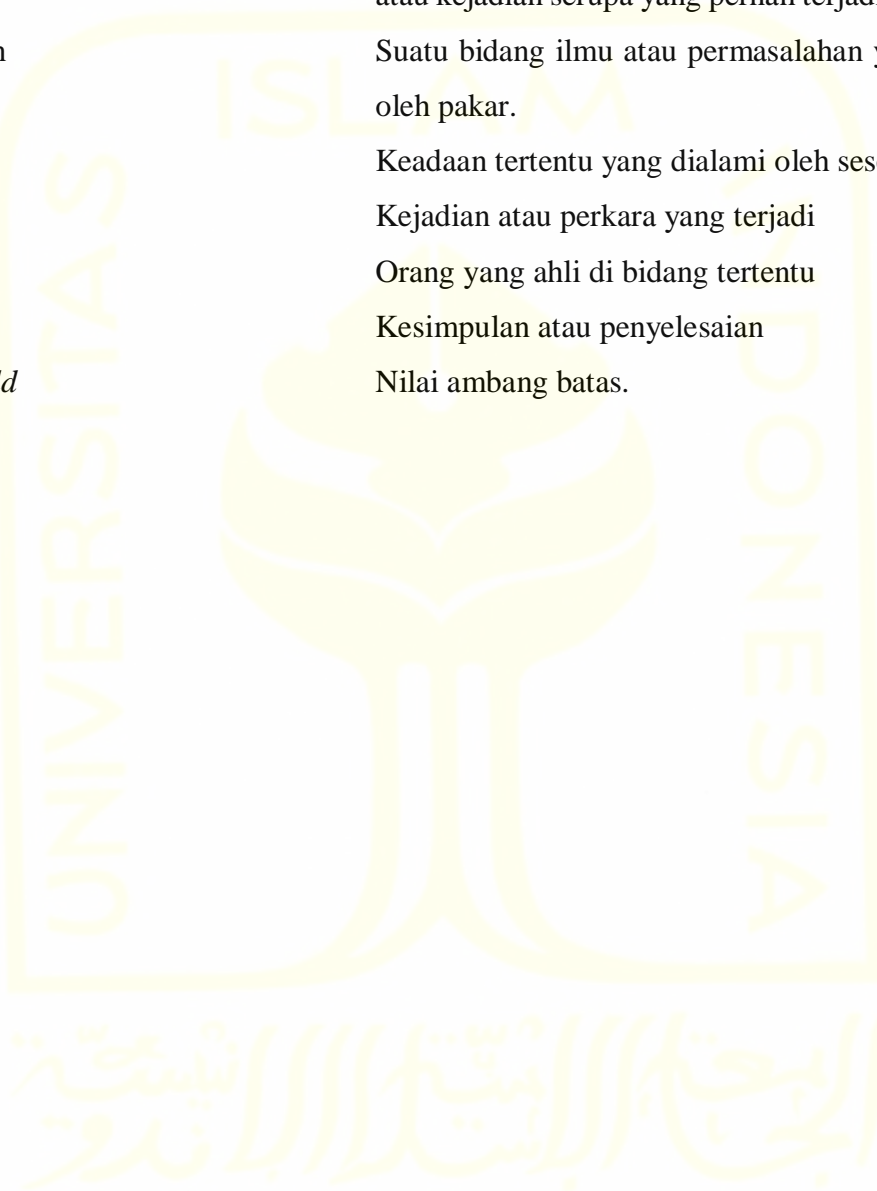
Editor basis pengetahuan untuk sistem berbasis kasus adalah sebuah sistem yang dapat menyimpan pengetahuan pakar ke dalam sistem komputer dalam bentuk basis pengetahuan dengan menggunakan metode *case based reasoning*. Namun, jumlah editor basis pengetahuan berbasis kasus lebih sedikit dibandingkan dengan yang berbasis aturan karena sebagian besar editor basis pengetahuan yang ada saat ini dikembangkan dengan menggunakan metode *rule based reasoning*.

Metode *prototyping* akan digunakan dalam mengembangkan editor basis pengetahuan ini. Agar editor basis pengetahuan ini bisa bekerja, dibutuhkanlah beberapa sumber pengetahuan yang nantinya akan digunakan dan diuji pada editor tersebut. Pengujian sistem dilakukan dengan menggunakan metode *black box testing* dan hasilnya sistem mampu merespon setiap input dengan baik dan sesuai dengan harapan. Selain itu, sistem juga diuji validitas dan usabilitasnya. Berdasarkan pengujian tersebut diperoleh nilai validitas sistem adalah 40% dan nilai usabilitas sistem adalah 86.5%.

Kata kunci: editor basis pengetahuan, *case based reasoning*, basis pengetahuan, pakar, non-pakar, domain, gejala, kasus, solusi, *threshold*.

GLOSARIUM

Basis Pengetahuan	Pengetahuan manusia atau pakar yang telah dikonversi menjadi sebuah basis data.
<i>Case Based Reasoning</i>	Sebuah metode penarikan solusi berdasarkan kasus atau kejadian serupa yang pernah terjadi di masa lalu.
Domain	Suatu bidang ilmu atau permasalahan yang ditekuni oleh pakar.
Gejala	Keadaan tertentu yang dialami oleh seseorang
Kasus	Kejadian atau perkara yang terjadi
Pakar	Orang yang ahli di bidang tertentu
Solusi	Kesimpulan atau penyelesaian
<i>Treshold</i>	Nilai ambang batas.



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI.....	ix
GLOSARIUM.....	x
DAFTAR ISI	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN	17
1.1 Latar Belakang.....	17
1.2 Rumusan Masalah.....	18
1.3 Batasan Masalah	18
1.4 Tujuan Penelitian	19
1.5 Manfaat Penelitian	20
1.6 Metodologi Penelitian	20
1.7 Sistematika Penulisan.....	20
BAB II LANDASAN TEORI.....	22
2.1 Kajian Pustaka	22
2.2 Kerangka Teoritik	22
2.2.1 Basis Pengetahuan	22
2.2.2 Sistem Basis Pengetahuan.....	23
2.2.3 Editor Basis Pengetahuan.....	23
2.2.4 Case Based Reasoning	24
2.2.5 Pengujian.....	25
2.2.6 Pengujian <i>black box testing</i>	25
2.2.7 Pengujian validitas sistem.....	25
2.2.8 Pengujian usabilitas sistem	26
BAB III METODOLOGI PENELITIAN.....	29

3.1	Metodologi Penelitian	29
3.1.1	Langkah-langkah Penelitian	29
3.1.2	Metode Pengumpulan Data	29
3.1.3	Metode Pengembangan Sistem.....	30
3.2	Subjek dan Objek Penelitian	30
3.3	Lokasi dan Waktu Penelitian	30
3.4	Instrumen Penelitian	31
BAB IV HASIL DAN PEMBAHASAN		32
4.1	Analisis Kebutuhan	32
4.1.1	Analisis Kebutuhan Pengguna.....	32
4.1.2	Analisis Kebutuhan <i>Input</i> /Masukan	32
4.1.3	Analisis Kebutuhan Proses.....	33
4.1.4	Analisis Kebutuhan <i>Output</i> /Keluaran.....	33
4.1.5	Analisis Kebutuhan <i>Interface</i>	34
4.1.6	Analisis Kebutuhan Perangkat	35
4.2	Desain Sistem	36
4.2.1	<i>Use Case Diagram</i>	36
4.2.2	<i>Activity Diagram</i>	37
4.2.3	Relasi Tabel Basis Data	42
4.3	Implementasi	45
4.3.1	Struktur Menu.....	45
4.3.2	<i>Interface</i>	46
4.3.3	<i>Source Code</i>	70
4.3.4	Contoh Proses Perhitungan	75
4.4	Uji Coba Sistem.....	79
4.4.1	Pengujian <i>black box testing</i>	79
4.4.2	Pengujian validitas sistem	83
4.4.3	Pengujian usabilitas	84
BAB V KESIMPULAN		88
5.1	Kesimpulan.....	88
5.2	Saran.....	88
DAFTAR PUSTAKA		89
LAMPIRAN		91

DAFTAR TABEL

Tabel 2.1 Kategori kelayakan sistem	26
Tabel 4.1 Tabel analisis kebutuhan pengguna	32
Tabel 4.2 Tabel aktor	37
Tabel 4.3 Tabel definisi <i>use case</i>	37
Tabel 4.4 Tabel login	43
Tabel 4.5 Tabel masalah.....	43
Tabel 4.6 Tabel gejala	44
Tabel 4.7 Tabel solusi	44
Tabel 4.8 Tabel kasus.....	44
Tabel 4.9 Tabel kasustemp	44
Tabel 4.10 Tabel pakar	45
Tabel 4.11 Tabel pesan.....	45
Tabel 4.13 Perhitungan similaritas dengan fungsi similaritas 1.1	75
Tabel 4.14 Perhitungan similaritas dengan fungsi similaritas 1.2	76
Tabel 4.15 Perhitungan similaritas dengan fungsi similaritas 1.3	77
Tabel 4.16 Perhitungan similaritas dengan fungsi similaritas 1.4	78
Tabel 4.17 Pengujian komponen pilih gejala	79
Tabel 4.18 Pengujian komponen <i>login</i>	80
Tabel 4.19 Pengujian komponen tambah domain.....	80
Tabel 4.20 Pengujian komponen pilih domain	80
Tabel 4.21 Pengujian komponen hapus domain	80
Tabel 4.22 Pengujian komponen tambah gejala	81
Tabel 4.23 Pengujian komponen ubah gejala.....	81
Tabel 4.24 Pengujian komponen tambah solusi	81
Tabel 4.25 Pengujian komponen ubah solusi	82
Tabel 4.26 Pengujian komponen tambah kasus.....	82
Tabel 4.27 Pengujian komponen ubah kasus.....	82
Tabel 4.28 Pengujian komponen ubah fungsi similaritas.....	83
Tabel 4.29 Pengujian komponen ubah <i>password</i> domain	83
Tabel 4.30 Hasil pengujian validitas sistem	83
Tabel 4.31 Kategori kelayakan sistem	85
Tabel 4.32 Hasil perhitungan persentase aspek <i>learnability</i>	85

Tabel 4.33 Hasil perhitungan persentase aspek *flexibility* 86

Tabel 4.34 Hasil perhitungan persentase aspek *effectiveness*..... 86

Tabel 4.35 Hasil perhitungan persentase aspek *attitude* 86



DAFTAR GAMBAR

Gambar 2.1 Siklus CBR	24
Gambar 4.1 <i>Use case diagram</i>	36
Gambar 4.2 <i>Activity diagram</i> konsultasi	38
Gambar 4.3 <i>Activity diagram</i> tambah domain	39
Gambar 4.4 <i>Activity diagram</i> kelola pengetahuan	40
Gambar 4.5 <i>Activity diagram</i> atur fungsi similaritas	41
Gambar 4.6 <i>Activity diagram</i> ubah <i>password</i> domain	42
Gambar 4.7 Relasi tabel basis data	43
Gambar 4.8 Struktur menu	46
Gambar 4.9 Tampilan <i>home</i>	47
Gambar 4.10 Tampilan konsultasi	48
Gambar 4.11 Tampilan pilih gejala.....	49
Gambar 4.12 Tampilan hasil diagnosis sukses	50
Gambar 4.13 Tampilan hasil diagnosis gagal.....	51
Gambar 4.14 Tampilan informasi diagnosis	52
Gambar 4.15 Tampilan <i>login</i> pakar	53
Gambar 4.16 Tampilan <i>home</i> pakar	53
Gambar 4.17 Tampilan tambah domain	54
Gambar 4.18 Tampilan pilih domain	55
Gambar 4.19 Tampilan kelola gejala	56
Gambar 4.20 Tampilan tambah gejala	57
Gambar 4.21 Tampilan ubah gejala	58
Gambar 4.22 Tampilan kelola solusi	59
Gambar 4.23 Tampilan tambah solusi	60
Gambar 4.24 Tampilan ubah solusi	61
Gambar 4.25 Tampilan antarkasus	62
Gambar 4.26 Tampilan tambah kasus	63
Gambar 4.27 Tampilan ubah kasus.....	64
Gambar 4.28 Tampilan kasus filter.....	65
Gambar 4.29 Tampilan notifikasi pesan baru.....	66
Gambar 4.30 Tampilan atur fungsi similaritas	67
Gambar 4.31 Tampilan ubah <i>password</i> domain	68

Gambar 4.32 Tampilan data pakar.....	68
Gambar 4.33 Tampilan ubah password domain	69
Gambar 4.34 Tampilan tentang	70
Gambar 4.35 <i>Source code</i> tambah domain.....	71
Gambar 4.36 <i>Source code</i> tambah gejala	72
Gambar 4.37 <i>Source code</i> ubah gejala.....	72
Gambar 4.38 <i>Source code</i> hapus gejala	74



BAB I PENDAHULUAN

1.1 Latar Belakang

Sebuah sistem pakar pada dasarnya terdiri dari 2 bagian, yakni bagian pakar dan bagian non-pakar. Bagian pakar berperan dalam proses *knowledge management* atau proses manajemen basis pengetahuan sedangkan bagian non-pakar berperan dalam proses penarikan solusi atau kesimpulan dengan menggunakan basis pengetahuan yang ada.

Pada bagian pakar, manajemen pengetahuan dilakukan dengan menggunakan alat yang disebut sebagai editor basis pengetahuan. Editor basis pengetahuan merupakan sebuah *tools* atau alat yang memberikan akses kepada pengguna pakar untuk menambah, menghapus, dan memodifikasi aturan-aturan yang ada di dalam basis pengetahuan. Keberadaan editor basis pengetahuan sangat diperlukan karena dapat membantu pakar dalam mengelola basis pengetahuan dengan mudah.

Sebuah basis pengetahuan biasanya dikembangkan dengan dua metode berikut; *rule based reasoning* (menggunakan aturan untuk mendapatkan kesimpulan atau solusi) atau dengan metode *case based reasoning* (menggunakan nilai *threshold* atau nilai batas kemiripan untuk mendapatkan kesimpulan atau solusi) (Prentzas & Hatzilygeroudis, 2003).

Umumnya, sebuah basis pengetahuan dikembangkan dengan menggunakan pendekatan *rule based reasoning* atau pendekatan berbasis aturan dan pengetahuan tersebut dipresentasikan dalam bentuk *IF-THEN* (Berka, 2011) dibandingkan dengan menggunakan pendekatan berbasis kasus atau *case based reasoning*. Selain itu, masih sedikit sekali orang yang memiliki pengalaman dalam mengoperasikan sistem berbasis kasus ini (Watson & Marir, 1994).

Pengembangan editor basis pengetahuan yang hanya terkait dengan satu domain atau satu masalah saja ternyata bukanlah hal yang praktis (Paley, et al., 1997). Akibatnya, jika ada satu domain yang ingin dijadikan sebuah basis pengetahuan, maka seorang *knowledge engineer* harus membangun sebuah editor basis pengetahuan yang baru dari awal. Oleh karena itu, dibutuhkanlah sebuah editor basis pengetahuan yang memungkinkan para pakar untuk mengakses berbagai domain atau masalah tersebut melalui satu antarmuka atau satu sistem yang tunggal saja.

1.2 Rumusan Masalah

Rumusan masalah yang dapat diambil berdasarkan latar belakang tersebut adalah Bagaimana membangun sebuah editor basis pengetahuan untuk sistem berbasis kasus yang mampu membawa pengetahuan yang dimiliki oleh seorang pakar menjadi sebuah pengetahuan yang bisa disimpan di dalam basis pengetahuan dan mampu bekerja dengan baik?

1.3 Batasan Masalah

Agar penelitian ini tetap terarah dan dilakukan dengan semestinya, ditentukanlah beberapa batasan masalah yang terdiri dari:

- a. Sistem ini tidak akan menentukan siapa yang berperan sebagai pakar dan siapa yang berperan sebagai non-pakar. Pakar merupakan tipe pengguna yang memiliki akses untuk menggunakan editor basis pengetahuan.
- b. Nilai *threshold* atau nilai batas kemiripan antara kasus baru dengan kasus serupa akan ditentukan oleh pakar itu sendiri.
- c. Sistem ini memiliki fungsi jumlah fungsi similitas yang tetap yakni sebanyak 4 buah fungsi similitas sehingga pengguna pakar tidak bisa membuat fungsi similitas sendiri saat menggunakan sistem ini. Berikut adalah fungsi-fungsi similitas yang ada di dalam sistem.
 1. Pencocokan basis kasus, yakni dengan membagi jumlah komponen yang sama antara kasus baru dan kasus yang ada pada basis pengetahuan dengan banyak komponen kasus yang ada pada basis pengetahuan. Silahkan lihat persamaan (1.1) berikut.

$$T_i = \frac{\sum_{j=1}^n \text{komponen yang sama antara } C \text{ dan } K_i}{n} \quad (1.1)$$

2. Pencocokan basis kasus dan evaluasi, yakni dengan membagi jumlah komponen yang sama antara kasus baru dan kasus yang ada pada basis pengetahuan dengan maksimum antara banyaknya komponen pada kasus baru dan pada kasus yang ada pada basis pengetahuan. Silahkan lihat persamaan (1.2) berikut.

$$T_i = \frac{\sum_{j=1}^n \text{komponen yang sama antara } C \text{ dan } K_i}{\max(m, n)} \quad (1.2)$$

3. Pencocokan basis kasus dengan bobot, yakni dengan membagi jumlah bobot dari komponen yang sama antara kasus baru dan kasus yang ada pada basis pengetahuan dengan jumlah bobot komponen kasus yang ada di basis pengetahuan. Silahkan lihat persamaan (1.1) berikut.

$$T_i = \frac{\sum_{j=1}^n \text{bobot yang sama antara } C \text{ dan } K_i}{\sum_{j=1}^n \text{bobot komponen ke } -j \text{ dari } K_i} \quad (1.3)$$

4. Pencocokan basis kasus dengan bobot dan konstanta, persamaan ini sama dengan persamaan (1.1) jika banyak komponen kasus yang ada pada basis pengetahuan lebih banyak atau sama dengan banyak komponen kasus baru. Namun, jika komponen kasus baru lebih banyak maka persamaan yang digunakan adalah persamaan yang membagi jumlah bobot dari komponen yang sama antara kasus baru dan kasus yang ada pada basis pengetahuan dengan jumlah bobot komponen kasus yang ada di basis pengetahuan ditambah dengan hasil perkalian antara konstanta positif dengan selisih antara banyak komponen kasus baru dan komponen kasus yang ada pada basis pengetahuan. Silahkan lihat persamaan (1.1) berikut.

$$T_i = \begin{cases} \frac{\sum_{j=1}^n \text{bobot yang sama antara } C \text{ dan } K_i}{\sum_{j=1}^n \text{bobot komponen } K_i}, & m \leq n \\ \frac{\sum_{j=1}^n \text{bobot yang sama antara } C \text{ dan } K_i}{\sum_{j=1}^n \text{bobot komponen } K_i + \alpha (m - n)}, & m > n \end{cases} \quad (1.4)$$

Keterangan:

1. T_i : Nilai similaritas
2. C : Kasus baru
3. K_i : Kasus lama pada basis pengetahuan
4. n : Banyak komponen K_i
5. m : Banyak komponen C
6. α : Konstanta positif

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membangun sebuah editor basis pengetahuan untuk sistem berbasis kasus yang mengubah pengetahuan yang dimiliki oleh seorang pakar

menjadi sebuah pengetahuan yang bisa disimpan di dalam basis pengetahuan. Selain itu, editor basis pengetahuan ini didesain agar dapat bekerja dengan lebih dari satu domain atau masalah, sehingga semua komponen domain tersebut bisa diatur dalam satu sistem dan antarmuka yang sama.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah diantaranya:

- a. Aplikasi editor basis pengetahuan dapat membantu dan meringankan kerja pengguna terutama pakar baik dari segi fitur hingga tampilan.
- b. Adanya sebuah editor basis pengetahuan untuk sistem berbasis kasus juga diharapkan mampu memberikan penjelasan kepada para pakar bahwa selain dengan aturan, pengetahuan pakar juga bisa dipresentasikan dalam bentuk kasus-kasus yang pernah terjadi di masa lalu beserta dengan solusinya. Penjelasan ini disampaikan kepada pakar melalui halaman tentang sistem yang ada pada sistem.

1.6 Metodologi Penelitian

Dalam menyusun skripsi ini, terdapat beberapa langkah-langkah yang ditentukan agar penelitian ini berjalan sebagaimana mestinya. Langkah-langkah tersebut meliputi analisis kebutuhan, desain sistem, implementasi sistem, serta uji coba sistem.

Selain langkah-langkah tersebut, penulis juga menggunakan metode penelitian yang berfokus pada pengumpulan data dan informasi yang nantinya akan digunakan sebagai objek penelitian. Teknik pengumpulan data yang penulis gunakan adalah wawancara kepada narasumber serta studi pustaka dengan membaca dan mempelajari buku, dokumen, artikel, ataupun jurnal yang ada di internet.

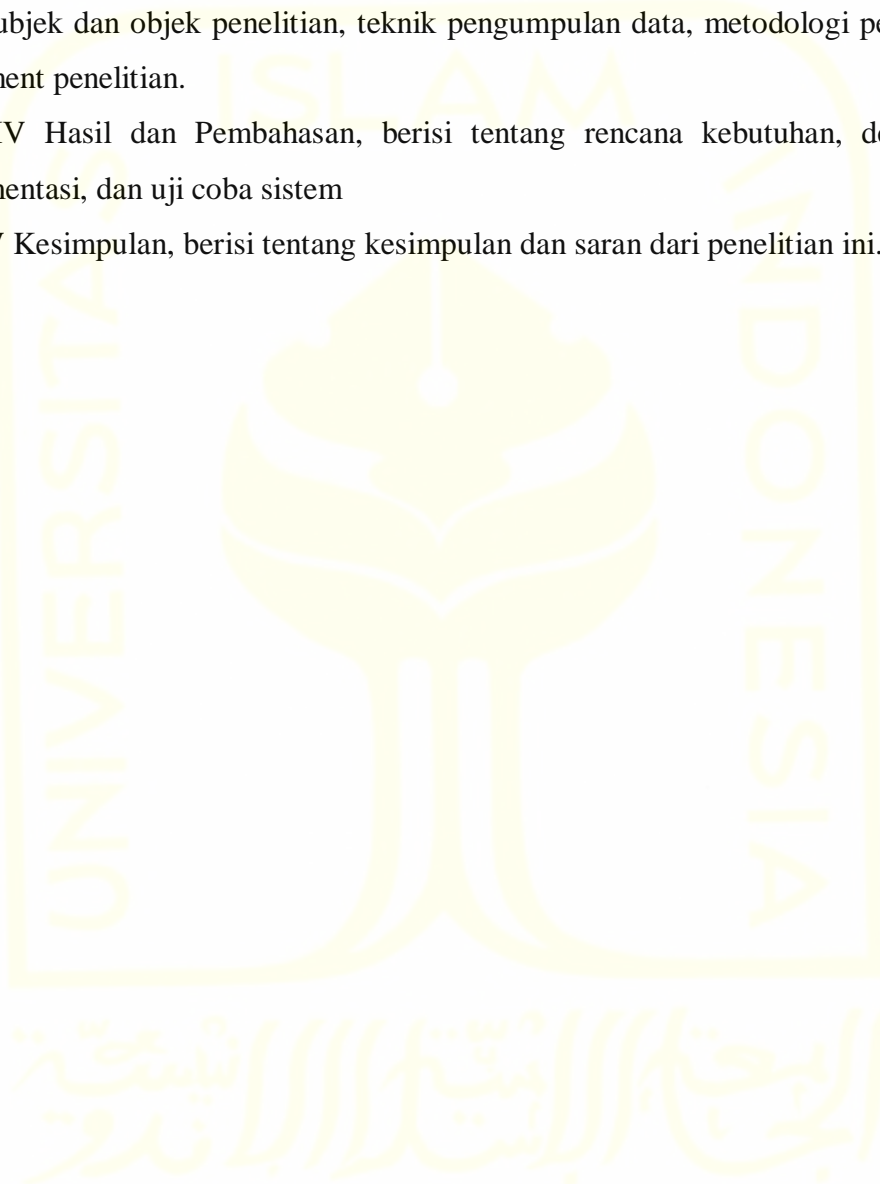
Dalam konteks pengembangan sistem, penelitian ini menggunakan metode *protoyping* yang berguna untuk mendapatkan gambaran awal mengenai gambaran sistem yang akan dikembangkan.

1.7 Sistematika Penulisan

Sistematika penulisan skripsi ini terdiri dari:

- a. BAB I Pendahuluan, merupakan bab pertama dalam laporan ini. Bab ini menjelaskan gambaran umum dari permasalahan yang diteliti.

- b. BAB II Landasan Teori, bab ini berisi tentang pengertian mengenai basis pengetahuan, sistem basis pengetahuan, editor basis pengetahuan, *case based reasoning*, dan beberapa jenis pengujian. Selain itu, bab ini juga memaparkan tentang penelitian-penelitian sebelumnya yang berkaitan dengan penelitian ini.
- c. BAB III Metodologi Penelitian, dalam bab ini berisi tentang langkah-langkah penelitian, metode pengumpulan dan pengembangan sistem, lokasi dan waktu penelitian, sumber data, subjek dan objek penelitian, teknik pengumpulan data, metodologi penelitian, dan instrument penelitian.
- d. BAB IV Hasil dan Pembahasan, berisi tentang rencana kebutuhan, desain sistem, implementasi, dan uji coba sistem
- e. BAB V Kesimpulan, berisi tentang kesimpulan dan saran dari penelitian ini.



BAB II LANDASAN TEORI

2.1 Kajian Pustaka

Dalam penelitian ini, penulis melakukan kajian terhadap penelitian-penelitian sebelumnya yang serupa dan relevan dengan penelitian ini. Kajian ini berguna sebagai bahan perbandingan, baik itu kelebihan atau kekurangan yang ada. Berikut adalah hasil kajian pustaka yang sudah penulis kumpulkan:

- a. “*Case Based Reasoning* Untuk Mendeteksi Kerusakan *Harddisk*” karya Nola Ritha dan M. Nurtanzis Sutoyo dari Universitas Maritim Raja Ali Haji. Hasil penelitian ini adalah sebuah sistem yang mampu mendiagnosis kerusakan pada *harddisk* berdasarkan gejala yang dimasukkan oleh pengguna.
- b. “Rancang Bangun Sistem Pakar Pendiagnosa Penyakit Paru-paru Menggunakan Metode *Case Based Reasoning*” karya Diki Andita Kusuma dan Chairani dari IBI Darmajaya. Hasil penelitian ini adalah sebuah sistem yang mampu mendiagnosis penyakit paru-paru yang dialami oleh pengguna.

Persamaan yang terdapat pada penelitian-penelitian ini dengan penelitian penulis terdapat pada metode penalaran yang digunakan yaitu *case based reasoning*. Terdapat juga beberapa perbedaan antara penelitian-penelitian ini dengan penelitian yang dilakukan oleh penulis, diantaranya adalah penelitian-penelitian ini hanya menghasilkan sistem yang bekerja sebagai alat diagnosis saja, sedangkan penelitian yang dilakukan oleh penulis menghasilkan sistem diagnosis lengkap dengan editor basis pengetahuannya. Selain itu, penelitian ini hanya berfokus pada satu domain saja (misalnya hanya kerusakan *harddisk*), sedangkan penelitian yang dilakukan oleh penulis adalah penelitian yang berfokus pada banyak domain.

2.2 Kerangka Teoritik

2.2.1 Basis Pengetahuan

Basis pengetahuan atau biasa disebut dengan *knowledge base* terdiri dari *facts base*. *Facts base* adalah sebuah basis yang mengandung berbagai informasi dan fakta tentang suatu permasalahan (Naykhanova & Naykhanova, 2018). Informasi dan fakta ini bisa dipresentasikan dalam bentuk aturan-aturan jika menggunakan pendekatan *rule based*

reasoning dan bisa juga berupa kasus-kasus atau kejadian serupa yang pernah terjadi di masa lalu jika menggunakan pendekatan *case based reasoning*.

Basis pengetahuan bisa mengandung pengetahuan faktual maupun heuristik (Tripathi, 2011). Pengetahuan faktual merupakan pengetahuan yang memiliki domain yang luas dan sering dibagikan, biasanya ditemukan di dalam buku atau jurnal. Sedangkan pengetahuan heuristik adalah pengetahuan yang jarang dibahas dan sebagian besar bersifat individualistis.

Sebelum masuk ke dalam basis pengetahuan. Baik pengetahuan faktual maupun heuristik harus menjalani proses akuisisi pengetahuan terlebih dahulu dimana pengetahuan-pengetahuan tersebut yang semula hanya dipahami oleh manusia (pakar) akan dikonversi dan diubah menjadi sebuah kode yang mampu dimengerti oleh komputer (Honggowibowo, 2009).

2.2.2 Sistem Basis Pengetahuan

Sistem basis pengetahuan atau *knowledge based system* (KBS) merupakan program komputer yang menggunakan basis pengetahuan (*knowledge base*) dan mengolahnya dengan tujuan untuk menemukan solusi atau rekomendasi dari sebuah permasalahan yang kompleks (Naykhanova & Naykhanova, 2018). KBS memiliki tiga tipe sub sistem yakni basis pengetahuan atau *knowledge base* yang mengandung berbagai informasi dan fakta, antarmuka pengguna atau *user interface*, dan mesin inferensi atau *inference engine*. Selain mampu menemukan solusi atau rekomendasi, KBS juga harus memiliki kemampuan untuk memberikan informasi dan penjelasan dibalik hasil solusi atau rekomendasi yang telah ditentukan (Elam & Henderson, 1983), mekanisme ini biasa disebut dengan *explanation mechanism* atau mekanisme penjelas. Sistem basis pengetahuan yang menggunakan pengetahuan dari domain atau bidang tertentu disebut sebagai sistem pakar atau *expert system*.

2.2.3 Editor Basis Pengetahuan

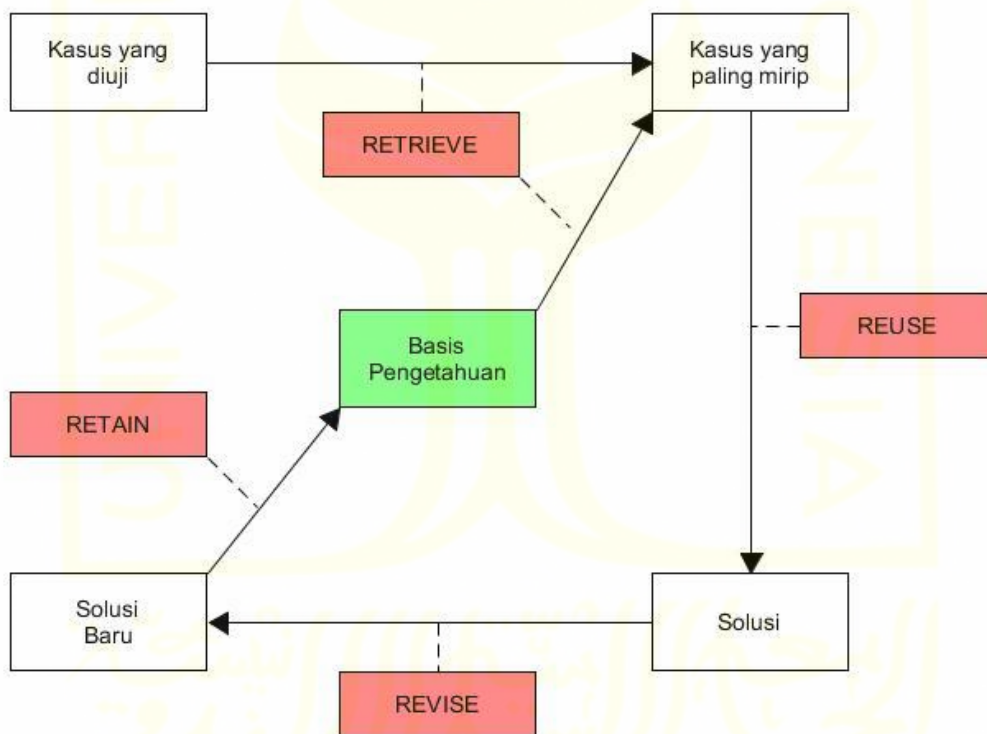
Editor basis pengetahuan atau *knowledge base editor* (KBE) memberikan akses kepada pengguna terutama pakar atau *knowledge engineer* untuk menambah, menghapus, dan memodifikasi aturan-aturan yang ada di dalam basis pengetahuan (Kayali, 2018).

KBE merupakan komponen utama dalam mekanisme akuisisi pengetahuan. Akuisisi pengetahuan sendiri merupakan proses akumulasi, transfer dan transformasi pengetahuan yang ada pada pakar atau sumber pengetahuan yang terdokumentasi ke dalam sebuah program komputer (Honggowibowo, 2009) sehingga bisa dipahami dan digunakan oleh komputer dalam

menyelesaikan suatu permasalahan. Keberadaan editor basis pengetahuan sangat diperlukan karena dapat membantu pakar dalam mengelola basis pengetahuan dengan mudah.

2.2.4 Case Based Reasoning

Case Based Reasoning (CBR) adalah suatu pendekatan dengan metode penalaran berbasis kasus yang biasa digunakan dalam proses mengambil solusi atau kesimpulan dari suatu kasus yang diuji. Dalam metode CBR ini, pengetahuan dipresentasikan dalam bentuk kasus-kasus. CBR mampu memberikan solusi atau kesimpulan dari kasus baru yang diuji dengan melihat kasus yang paling mirip dengan kasus sebelumnya yang telah tersimpan di dalam basis pengetahuan (Nurfalinda & Nikentari, 2017). Pengetahuan tersebut bisa digunakan untuk menemukan solusi dari suatu kasus serupa yang bisa saja terjadi di masa sekarang atau yang akan datang. Secara umum, CBR memiliki 4 siklus, yaitu *retrieve*, *reuse*, *revise*, dan *retain*, sebagaimana yang terlihat pada Gambar 2.1 berikut.



Gambar 2.1 Siklus CBR

Sumber: Watson (1999)

Berdasarkan Gambar 2.1 di atas, siklus CBR diawali dengan *retrieve* yang merupakan sebuah proses menemukan kasus yang paling serupa dengan kasus yang sedang diuji. Salah

satu cara untuk menentukan kasus yang paling serupa adalah dengan menggunakan fungsi similaritas. Fungsi ini akan menentukan nilai kemiripan atau similaritas antara kasus yang ada di basis pengetahuan dengan kasus baru yang sedang diuji, semakin tinggi nilai similaritasnya maka semakin mirip kedua kasus tersebut. Untuk menentukan kasus yang paling mirip ditentukanlah sebuah *threshold* atau nilai ambang batas yang dijadikan sebagai parameter bagi suatu kasus bisa dianggap serupa dengan kasus yang sedang diuji. Biasanya, *threshold* dipresentasikan dalam bentuk persen atau desimal. Sebuah kasus dikatakan mirip jika nilai similaritasnya sama atau melebihi nilai *threshold* ini.

Setelah menentukan kasus yang paling mirip, langkah selanjutnya adalah *reuse* yang merupakan proses membuat kesimpulan atau solusi untuk kasus yang sedang diuji. Solusi dibuat berdasarkan solusi dari kasus yang paling mirip. Proses selanjutnya adalah *revise* atau mengadopsi solusi tersebut ke dalam kasus yang sedang diuji setelah dilakukan pertimbangan sebelumnya oleh pakar. Langkah terakhir dari siklus CBR ini adalah *retain* di mana pakar menjadikan kasus yang sedang di uji beserta solusinya tersebut sebagai basis kasus baru dan disimpan di dalam basis pengetahuan.

Selain fungsi similaritas, ada juga metode lain yang bisa diaplikasikan ke dalam sistem berbasis kasus yakni dengan menggunakan induksi, logika fuzzy, dan pengolahan database (Watson, 1999).

2.2.5 Pengujian

Pengujian merupakan proses pelaksanaan suatu aplikasi atau sistem dengan tujuan untuk menemukan kesalahan di dalam sistem tersebut (Mustaqbal, et al., 2015).

2.2.6 Pengujian *black box testing*

Pengujian *black-box testing* adalah jenis pengujian yang mana pelaksanaan pengujian ini dilakukan dengan cara menguji setiap komponen sistem dan melihat hasilnya apakah komponen tersebut sudah beroperasi atau mengeluarkan *output* sesuai dengan yang diharapkan atau tidak.

2.2.7 Pengujian validitas sistem

Pengujian validitas sistem adalah jenis pengujian yang menguji tingkat kevalidan atau kebenaran suatu sistem atau instrumen. Nilai validitas atau keakuratan sistem dapat diukur dengan menggunakan rumus (2.1) berikut:

$$\text{Nilai validitas} = \frac{\text{jumlah yang valid}}{\text{jumlah sampel uji}} \times 100\% \quad (2.1)$$

Sumber: Aditiawarman, et al. (2017)

Nilai validitas yang diperoleh akan dikelompokkan ke dalam jenis kelayakan sistem sebagaimana yang tertera dalam tabel Tabel 2.1 berikut:

Tabel 2.1 Kategori kelayakan sistem

Interval (%)	Keterangan
0% - 19.9%	Sangat kurang
20% - 39.9%	Kurang
40% - 59.9%	Cukup
60% - 79.9%	Baik
80% - 100%	Sangat baik

2.2.8 Pengujian usabilitas sistem

Pengujian usabilitas sistem adalah salah satu jenis pengujian yang menentukan seberapa mudah pengguna menggunakan dan mengoperasikan sistem.

Pengujian usabilitas dilaksanakan dengan menggunakan kuesioner. Setiap pernyataan kuesioner terdapat lima skala yaitu Sangat Baik (SB) dengan skor 5 poin, Baik (B) dengan skor 4 poin, Cukup (C) dengan skor 3 poin, Kurang (K) dengan skor 2 poin, dan Sangat Kurang (SK) dengan skor 1 poin. Setiap komponen pada kuesioner ini dikelompokkan ke dalam salah satu dari 4 macam aspek berikut yaitu *learnability* yang berhubungan dengan kemudahan pengguna dalam mengoperasikan sistem, *flexibility* yang berhubungan dengan fitur-fitur yang ada di dalam sistem, *effectiveness* yang berhubungan dengan keberhasilan untuk mencapai tujuan dalam menggunakan sistem, dan *attitude* yang berhubungan dengan tingkat kepuasan pengguna.

Sebelum menentukan hasil akhir pengujian usabilitas, persentase setiap komponen haruslah ditentukan. Untuk menemukan nilai persentase tersebut maka digunakan rumus skala Likert. Silahkan lihat persamaan (2.2) berikut ini:

$$P = \frac{X}{i} \times 100\% \quad (2.2)$$

Sumber: Riyadi (2019)

Keterangan:

1. P : Nilai persentase yang akan dicari (%)
2. X : Total skor
3. i : Skor maksimum

Nilai X atau total skor didapatkan dengan menjumlahkan semua hasil perkalian antara total jumlah responden yang memilih komponen tertentu dengan pilihan angka skor Likert.

Selain nilai X, nilai i juga akan ditentukan dengan menggunakan rumus (2.3) berikut:

$$i = \text{skor tertinggi Likert} \times \text{jumlah responden} \quad (2.3)$$

Sumber: Riyadi (2019)

Setiap nilai persentase atau nilai P yang didapatkan tersebut akan dikelompokkan ke dalam sejumlah kategori golongan yang setiap kategori memiliki interval. Interval tersebut dapat ditentukan dengan menggunakan rumus (2.4) berikut:

$$j = \frac{100}{\text{Banyaknya skor Likert}} \quad (2.4)$$

Sumber: Putra, et al. (2013)

Keterangan:

1. j : Nilai interval yang akan dicari (%)

Hasil akhir pengujian usabilitas akan ditentukan dengan menggunakan rumus (2.5) berikut:

$$U = \frac{L + F + E + A}{4} \% \quad (2.5)$$

Sumber: Riyadi (2019)

Keterangan:

1. U : Nilai *usability* yang akan dicari (%)
2. L : Hasil rata-rata aspek *learnability* (%)
3. F : Hasil rata-rata aspek *flexibility* (%)

4. E : Hasil rata-rata aspek *effectiveness* (%)
5. A : Hasil rata-rata aspek *attitude* (%)



BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian ini terdiri dari langkah-langkah penelitian, metode pengumpulan data, dan metode pengembangan aplikasi.

3.1.1 Langkah-langkah Penelitian

Penelitian ini dilaksanakan dengan menggunakan langkah-langkah berikut:

- a. Analisis kebutuhan, tahapan ini akan menentukan hal-hal yang menjadi kebutuhan sistem. Pada penelitian ini, analisis kebutuhan sistem meliputi analisis kebutuhan pengguna, analisis kebutuhan *input*/masukan, analisis kebutuhan proses, analisis kebutuhan *output*/keluaran, dan analisis kebutuhan perangkat.
- b. Desain sistem, tahapan ini berperan dalam mengubah semua kebutuhan sistem yang telah ditentukan sebelumnya menjadi atribut yang dapat menggambarkan dan menjelaskan bagaimana sistem ini dikembangkan dan dibentuk. Desain sistem meliputi *use case diagram*, *activity diagram*, *entity relationship diagram*, dan *interface*.
- c. Implementasi sistem, adalah tahapan yang akan mengubah desain sistem menjadi bahasa pemrograman yang dapat dimengerti oleh komputer.
- d. Uji coba sistem, pada tahapan ini akan dilakukan proses pengujian terhadap sistem, apakah sistem mampu bekerja sesuai dengan harapan. Pada penelitian ini, pengujian sistem meliputi *black box testing*, uji validitas sistem, dan uji usabilitas sistem.

3.1.2 Metode Pengumpulan Data

Dalam rangka mengumpulkan data dan informasi yang berguna bagi penelitian ini, penulis menggunakan metode berikut:

- a. Wawancara atau *interview*, wawancara merupakan proses tanya jawab yang bertujuan untuk mendapatkan data dan informasi dari narasumber. Jenis wawancara yang dipilih adalah wawancara semi terstruktur dimana peneliti memberikan pertanyaan yang sudah disusun sebelumnya dan kemudian peneliti dapat mengembangkan pertanyaan baru berdasarkan jawaban narasumber (Rachmawati, 2007). Hasil informasi dan data yang didapat dari proses wawancara ini nantinya digunakan untuk proses selanjutnya yakni proses pengembangan sistem. Hasil wawancara dapat dilihat pada halaman lampiran.

- b. Studi pustaka yang dilakukan dengan membaca dan mempelajari buku, dokumen, artikel, ataupun jurnal. Hasil studi pustaka dapat dilihat pada halaman daftar pustaka.

3.1.3 Metode Pengembangan Sistem

Editor basis pengetahuan dikembangkan dengan menggunakan metode *protoyping*. Metode ini akan menghasilkan sebuah *prototype* yang berguna untuk mendapatkan gambaran awal mengenai gambaran sistem yang akan dikembangkan (Pradipta, et al., 2015). Pada penelitian ini, penulis mengembangkan *prototype* dan kemudian melakukan diskusi dengan pakar apakah *prototype* ini dapat dipahami dengan mudah dan layak dikembangkan menjadi sebuah sistem.

3.2 Subjek dan Objek Penelitian

Subjek yang terlibat dalam penelitian ini adalah seorang pakar yang sudah dianggap mengerti dan paham dalam suatu bidang tertentu. Sedangkan objek dalam penelitian ini adalah bidang yang ditekuni oleh pakar atau subjek tersebut.

Dalam konteks penelitian ini, subjek penelitian adalah pakar yang ahli dalam bidang tertentu. Pakar-pakar yang dipilih menjadi narasumber adalah Ibu Hazhira Qudsyi, S.Psi., M.A., selaku tenaga ahli konseling mahasiswa UII dan Ibu Hepi Wahyuningsih, S.Psi., M.Si, selaku dosen psikologi FPSB UII. Pakar-pakar tersebut dipilih sebagai narasumber karena relevan dengan objek penelitian penulis.

Objek atau bidang yang diteliti adalah mengenai permasalahan yang dialami oleh mahasiswa atau konseling mahasiswa. Permasalahan ini meliputi kasus-kasus permasalahan yang dialami oleh mahasiswa dan yang pernah ditangani oleh Direktorat Pembinaan Kemahasiswaan Universitas Islam Indonesia. Kasus-kasus tersebut terdiri dari gejala, hasil diagnosis, dan solusi yang diberikan.

3.3 Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan di Fakultas Teknologi Industri Universitas Islam Indonesia, Direktorat Pembinaan Kemahasiswaan Universitas Islam Indonesia, dan kediaman penulis. Penelitian mulai dilaksanakan pada semester 7 (ganjil) tahun ajaran 2019/2020.

Proses wawancara terhadap narasumber dilakukan sebanyak 3 kali dengan rincian waktunya adalah hari Jum'at tanggal 6 Desember 2019 pukul 13.00, hari Jum'at tanggal 7 Februari 2020 pukul 15.30, dan hari Selasa tanggal 11 Februari 2020 pukul 09.00

3.4 Instrumen Penelitian

Instrumen penelitian merupakan sebuah instrument atau alat yang digunakan oleh peneliti selama melakukan penelitian terutama saat mengumpulkan data penelitian. Dalam penelitian ini, instrument penelitian yang digunakan diantaranya adalah panduan wawancara dan alat rekam.



BAB IV HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Analisis kebutuhan terdiri dari analisis kebutuhan user, analisis kebutuhan input, analisis kebutuhan proses, analisis kebutuhan output dan analisis kebutuhan perangkat.

4.1.1 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna akan menentukan tipe pengguna beserta hak akses yang dimiliki oleh masing-masing tipe pengguna. Pemberian hak akses bertujuan agar basis data sistem terjaga dari akses yang tidak diinginkan. Tabel 4.1 berikut memaparkan analisis kebutuhan pengguna dari sistem yang akan dikembangkan.

Tabel 4.1 Tabel analisis kebutuhan pengguna

No	Tipe pengguna	Keterangan
1.	Non-pakar	Pengguna non-pakar memiliki hak untuk mengakses halaman konsultasi dan melakukan proses konsultasi di halaman tersebut dengan tujuan untuk mendapatkan hasil diagnosis.
2.	Pakar	Pengguna pakar, pengguna pakar memiliki hak akses untuk melakukan proses modifikasi basis pengetahuan.

4.1.2 Analisis Kebutuhan *Input*/Masukan

Non-pakar

Kebutuhan *input* atau masukan tipe pengguna non-pakar yang dibutuhkan oleh sistem terdiri dari:

- a. *Input* data gejala, data yang dimasukkan berupa *checkbox*.
- b. *Input* data pesan, data yang dimasukkan berupa teks yang berisi pesan yang ingin disampaikan kepada pakar.

Pakar

Kebutuhan *input* tipe pengguna pakar yang dibutuhkan oleh sistem terdiri dari:

- a. *Input* data login, data yang dimasukkan berupa *username* dan *password*.
- b. *Input* data domain baru, data yang dimasukkan berupa data nama domain, nilai batas/*threshold*, dan *password* domain.

- c. *Input* data gejala, data yang dimasukkan berupa nama gejala.
- d. *Input* data solusi, data yang dimasukkan berupa hasil diagnosis dan solusi.
- e. *Input* data kasus, data yang dimasukkan berupa gejala dan solusi.
- f. *Input* data fungsi similaritas, data yang dimasukkan berupa tipe fungsi similaritas yang dipilih.
- g. *Input* data *threshold*, data yang dimasukkan berupa nilai batas/*threshold*.
- h. *Input* data pakar, data yang dimasukkan berupa nama pakar dan informasi pakar.
- i. *Input* data ubah *password*, data yang dimasukkan berupa *password* atau kata sandi baru.

4.1.3 Analisis Kebutuhan Proses

Non-pakar

Kebutuhan proses tipe pengguna non-pakar yang dibutuhkan oleh sistem terdiri dari:

- a. Konsultasi, merupakan proses untuk mendapatkan solusi dengan cara memilih domain dan gejala.
- b. Kirim pesan, merupakan mengirimkan pesan kepada pakar agar dapat dibaca oleh pakar.

Pakar

Kebutuhan proses tipe pengguna pakar yang dibutuhkan oleh sistem terdiri dari:

- a. Login, merupakan proses untuk masuk ke dalam sistem editor basis pengetahuan.
- b. Tambah domain baru, merupakan proses membuat dan menambahkan domain baru ke dalam sistem.
- c. Tambah gejala, merupakan proses membuat dan menambahkan gejala ke dalam sistem.
- d. Tambah solusi, merupakan proses membuat dan menambahkan hasil diagnosis dan solusi ke dalam sistem.
- e. Tambah kasus, merupakan proses membuat dan menambahkan kasus ke dalam sistem.
- f. Ubah fungsi similartias, merupakan proses mengubah tipe fungsi similaritas yang digunakan oleh domain.
- g. Ubah *password*, merupakan proses mengubah kata sandi domain.
- h. Tambah data pakar, merupakan proses membuat dan menambahkan data pakar ke dalam sistem.

4.1.4 Analisis Kebutuhan Output/Keluaran

Non-pakar

Keluaran atau *output* tipe pengguna non-pakar yang dihasilkan oleh sistem terdiri dari:

- a. *Output* informasi hasil konsultasi.

Pakar

Keluaran atau *output* tipe pengguna non-pakar yang dihasilkan oleh sistem terdiri dari:

- a. *Output* informasi domain.
- b. *Output* informasi gejala.
- c. *Output* informasi solusi.
- d. *Output* informasi kasus.
- e. *Output* informasi pesan.
- f. *Output* informasi pakar.

4.1.5 Analisis Kebutuhan Interface

Non-pakar

Kebutuhan interface yang dibutuhkan oleh sistem tipe pengguna non-pakar terdiri dari:

- a. Rancangan tampilan *home*.
- b. Rancangan tampilan konsultasi.
- c. Rancangan tampilan pilih gejala.
- d. Rancangan tampilan hasil diagnosis sukses.
- e. Rancangan tampilan hasil diagnosis gagal.
- f. Rancangan tampilan informasi diagnosis.

Pakar

Kebutuhan interface yang dibutuhkan oleh sistem tipe pengguna pakar terdiri dari:

- a. Rancangan tampilan *login* pakar.
- b. Rancangan tampilan *home* pakar.
- c. Rancangan tampilan tambah domain.
- d. Rancangan tampilan pilih domain.
- e. Rancangan tampilan kelola gejala.
- f. Rancangan tampilan tambah gejala.
- g. Rancangan tampilan ubah gejala.
- h. Rancangan tampilan kelola solusi.
- i. Rancangan tampilan tambah solusi.

- j. Rancangan tampilan ubah solusi.
- k. Rancangan tampilan kelola kasus.
- l. Rancangan tampilan tambah kasus.
- m. Rancangan tampilan ubah kasus.
- n. Rancangan tampilan notifikasi kasus baru.
- o. Rancangan tampilan notifikasi pesan baru.
- p. Rancangan tampilan atur fungsi similaritas.
- q. Rancangan tampilan atur threshold/batas.
- r. Rancangan tampilan info pakar.
- s. Rancangan tampilan ubah *password*.
- t. Rancangan tampilan ubah tentang.

4.1.6 Analisis Kebutuhan Perangkat

Alat yang digunakan terbagi menjadi 2, yakni perangkat keras dan lunak.

- a. Perangkat keras adalah semua bagian fisik pada komputer. Perangkat keras bisa dilihat dan diraba. Perangkat keras yang dibutuhkan dalam mengembangkan sistem ini meliputi:
 1. *Central Processing Unit* atau CPU merupakan komponen yang menangani setiap proses yang ada di dalam sebuah komputer. Spesifikasi CPU yang penulis gunakan adalah Intel® Core™ i3-5005U @ 2.00GHz.
 2. *Random Access Memory* atau RAM adalah media penyimpanan yang berperan dalam menampung data yang sedang diproses oleh CPU, data-data tersebut bersifat sementara dan akan hilang ketika listrik dimatikan. Kapasitas RAM yang penulis gunakan adalah 8 GB.
 3. *Harddisk* merupakan media penyimpanan yang berperan dalam menampung data yang mana data tersebut tidak hilang dan dapat diolah di lain waktu. Kapasitas *harddisk* yang penulis gunakan adalah 500 GB.
 4. Keyborad dan Mouse.
- b. Perangkat lunak adalah data yang tersimpan dan dikendalikan oleh perangkat keras. Data tersebut dapat berupa sistem operasi, program atau aplikasi. Dalam mengembangkan sistem ini, penulis menggunakan beberapa perangkat lunak yaitu:
 1. Sistem operasi Widows 10 Pro 64-bit Build 14393
 2. BIOS InsydeH2O Version 05.04.43F.14
 3. Sublime Text Version 3.2.2 Build 3211

4. Bootstrap Studio Standard Edition Version 4.1.2
5. XAMPP Version 7.1.8
6. Apache Version 2.4.27
7. phpMyAdmin Version 4.7.0
8. Google Chrome Version 79.0.3945.88

4.2 Desain Sistem

4.2.1 Use Case Diagram

Use case diagram merupakan sebuah diagram yang menampilkan informasi berupa kebutuhan fungsional dari sistem yang dikembangkan. Dalam penelitian ini, *use case* terdiri dari 2 tipe pengguna, yaitu pakar dan non-pakar sebagaimana yang terlihat pada Gambar 4.1 berikut ini.



Gambar 4.1 *Use case diagram*

Berikut ini adalah penjabaran lebih lanjut mengenai Gambar 4.1 di atas.

- a. Aktor merupakan entitas yang memiliki interaksi dengan sistem, biasanya yang berperan sebagai aktor adalah orang. Tabel 4.2 berikut memaparkan definisi aktor dari sistem yang akan dikembangkan.

Tabel 4.2 Tabel aktor

No	Aktor	Keterangan
1.	Non-pakar	Non-pakar merupakan tipe pengguna pada umumnya yang menggunakan sistem ini hanya untuk melakukan proses konsultasi saja. Pengguna ini tidak dapat melakukan proses manipulasi domain (menambah, mengubah, dan menghapus).
2.	Pakar	Pakar adalah tipe pengguna yang dapat melakukan proses manipulasi domain (menambah, mengubah, dan menghapus) dan sebagainya ketika sedang menggunakan sistem ini.

- b. *Use case* merupakan langkah yang dilakukan oleh aktor ketika sedang berinteraksi dengan sistem. *Use case* biasanya dinyatakan dengan menggunakan kata kerja. Tabel 4.3 berikut memaparkan definisi aktor dari sistem yang akan dikembangkan.

Tabel 4.3 Tabel definisi *use case*

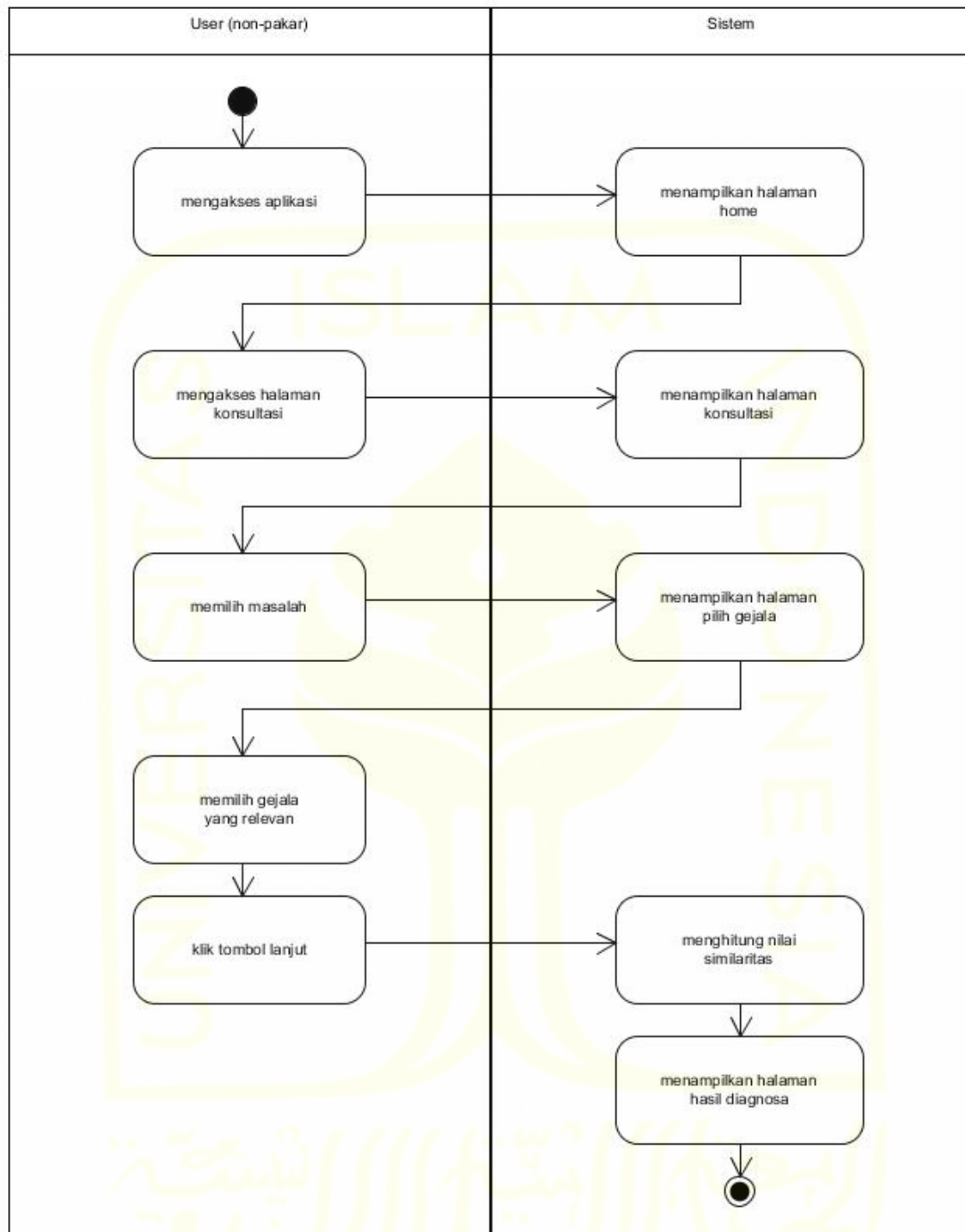
No	Event	Keterangan
1.	Konsultasi	Memilih domain dan memilih gejala.
2.	Kirim pesan	Mengirim pesan kepada pakar.
3.	Tambah domain	Memasukkan domain dan komponen domain (gejala, kasus, solusi, fungsi, dan password).
4.	Kelola domain	Melakukan perubahan terhadap komponen domain (gejala, kasus, solusi).
5.	Atur fungsi similaritas	Melakukan perubahan terhadap fungsi similaritas
6.	Tambah informasi pakar	Menambah informasi pakar yang menangani suatu domain.
7.	Ubah password domain	Mengubah kata sandi domain.

4.2.2 Activity Diagram

Activity diagram merupakan model atau diagram yang menggambarkan aliran kerja atau *workflow* dari sebuah proses bisnis yang ada di dalam sistem. Diagram ini bukan hanya menampilkan proses yang dapat dilakukan oleh *user*, namun diagram ini juga didesain agar dapat menampilkan proses yang dilakukan oleh sistem. Berikut ini merupakan rancangan *activity diagram* yang didesain untuk penelitian ini.

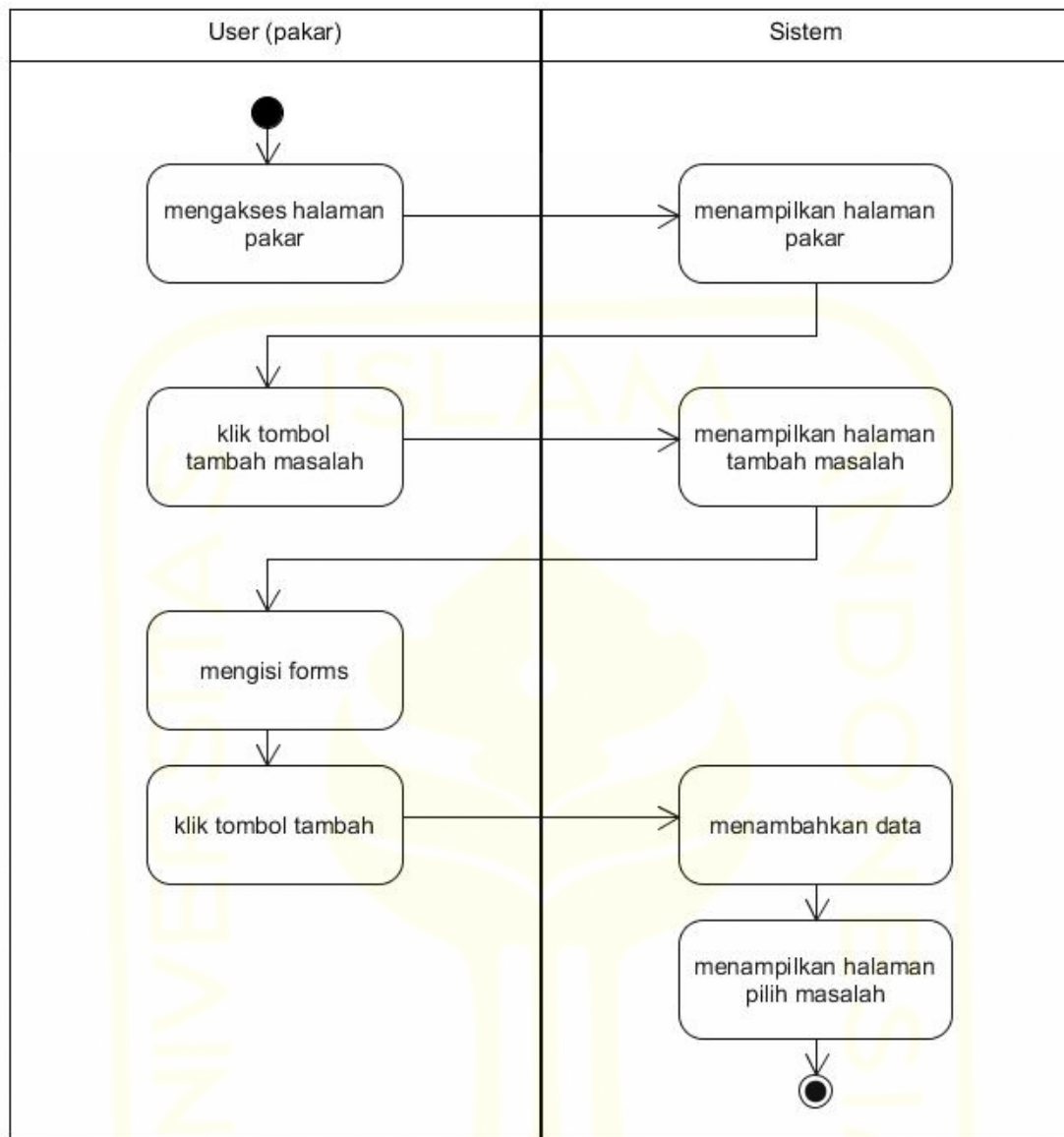
- a. *Activity diagram* konsultasi, menggambarkan proses yang dilakukan oleh pengguna non-pakar mulai dari mengakses sistem hingga muncul hasil diagnosis. Silahkan lihat Gambar 4.2 berikut.

x



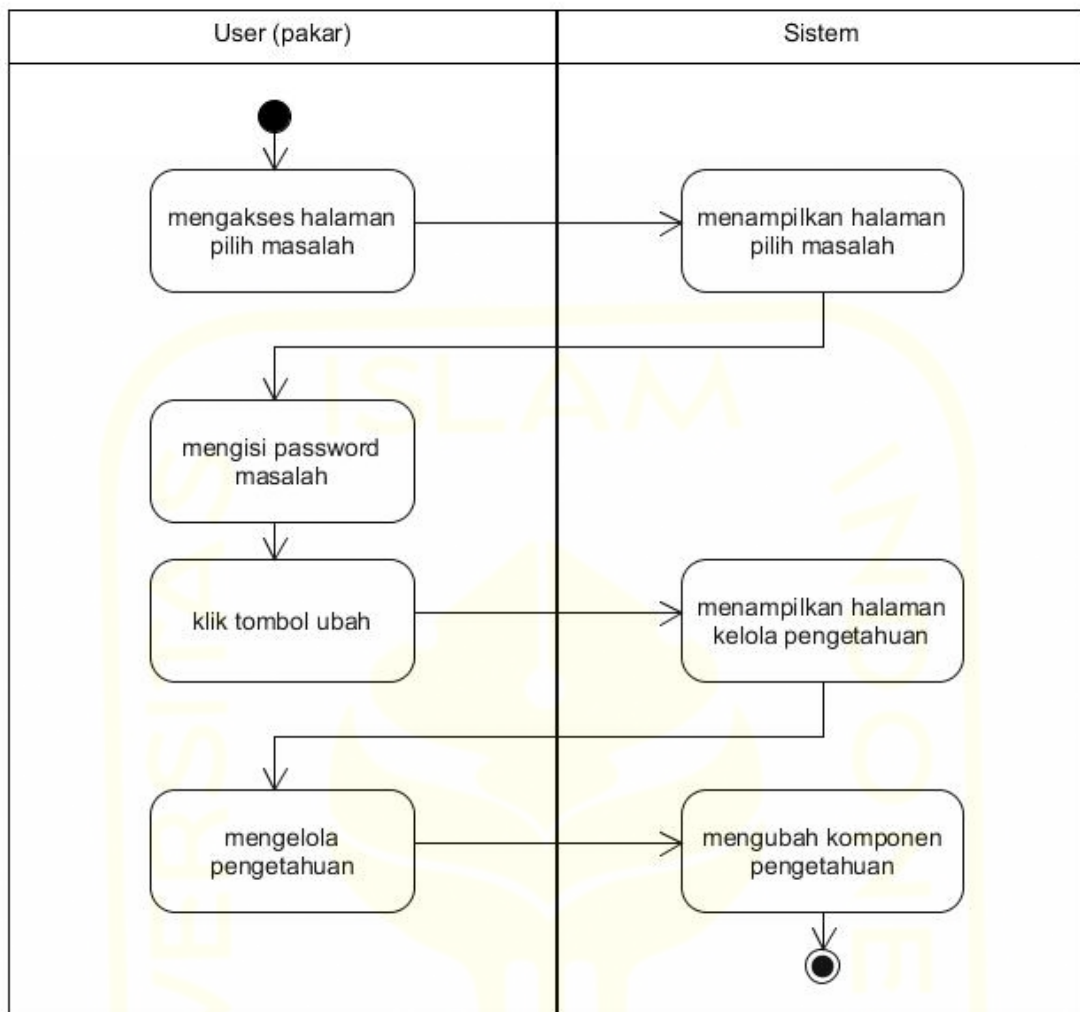
Gambar 4.2 Activity diagram konsultasi

- b. Activity diagram tambah domain, menggambarkan proses yang dilakukan oleh pengguna pakar saat menambahkan domain atau domain baru. Silahkan lihat Gambar 4.3 berikut.



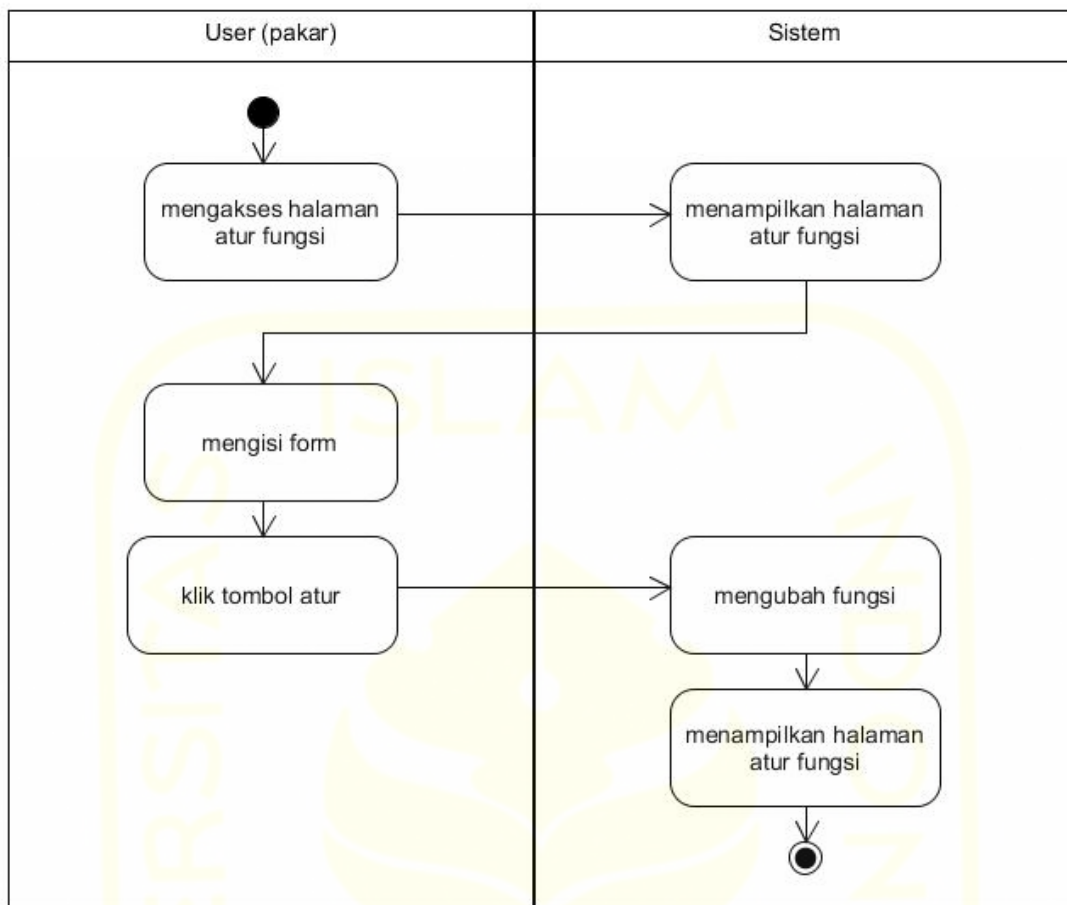
Gambar 4.3 *Activity diagram* tambah domain

- c. *Activity diagram* kelola pengetahuan, menggambarkan proses yang dilakukan oleh pengguna pakar ketika mengelola komponen-komponen pada suatu domain. Silahkan lihat Gambar 4.4 berikut.



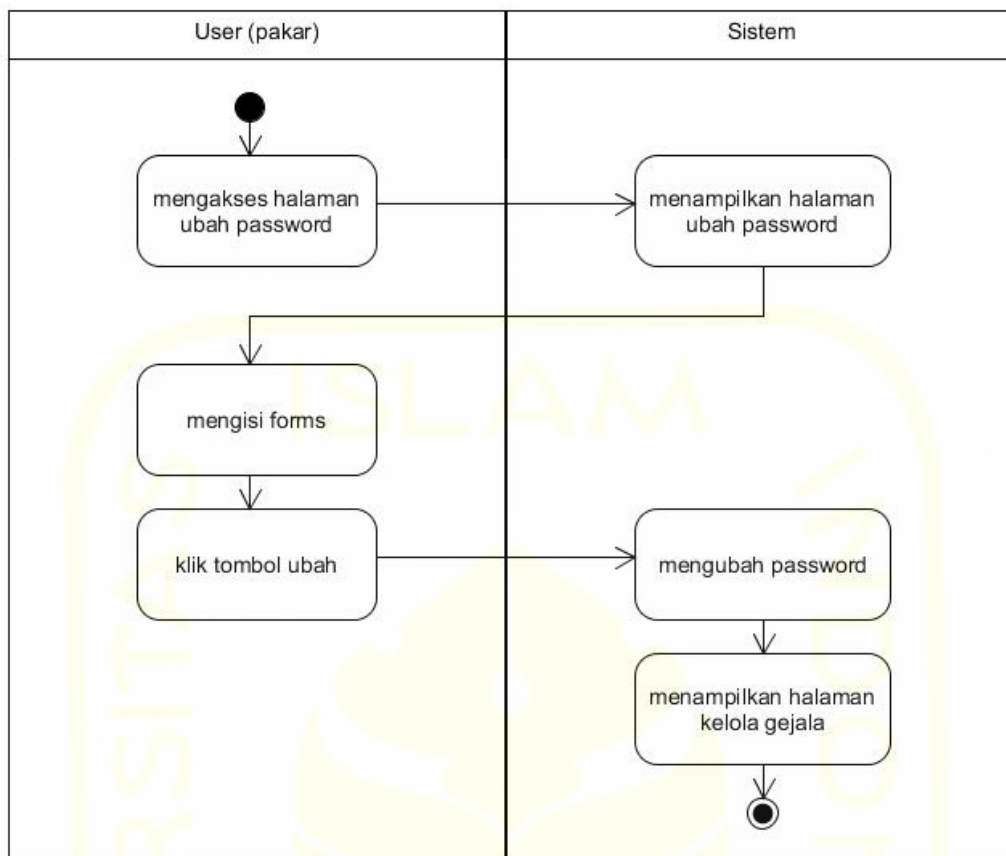
Gambar 4.4 Activity diagram kelola pengetahuan

- d. Activity diagram atur fungsi similaritas, menggambarkan proses yang dilakukan oleh pengguna pakar saat mengubah fungsi similaritas. Silahkan lihat Gambar 4.5 berikut.



Gambar 4.5 Activity diagram atur fungsi similaritas

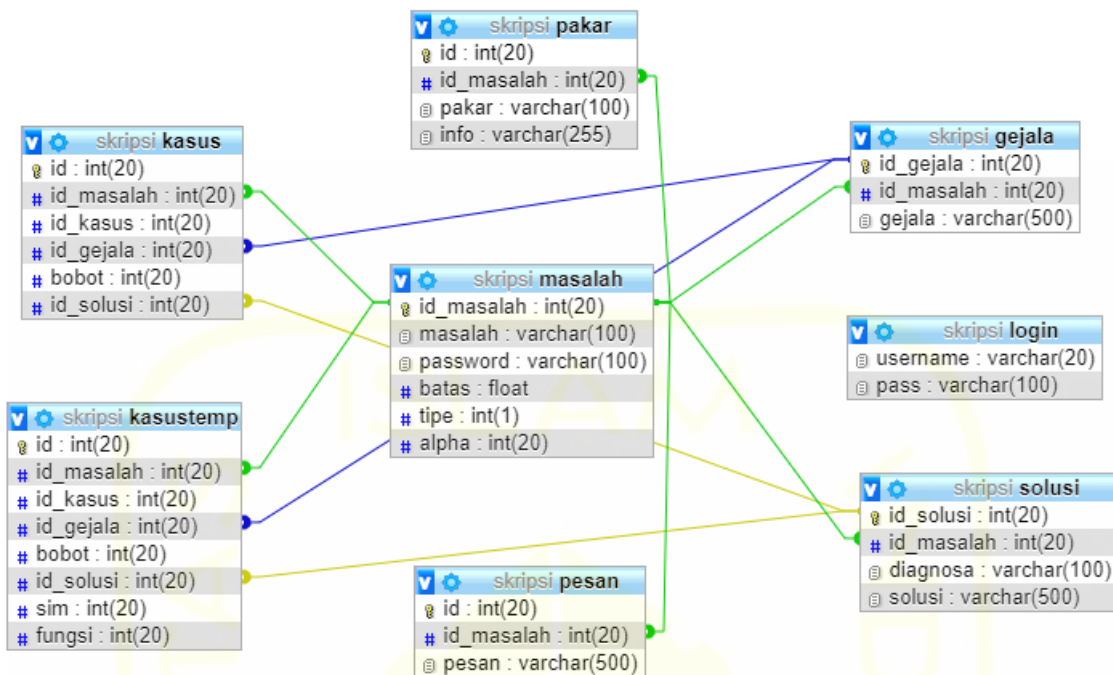
- e. Activity diagram ubah password domain, menggambarkan proses yang dilakukan oleh pengguna pakar saat mengubah kata sandi sebuah domain. Silahkan lihat Gambar 4.6 berikut.



Gambar 4.6 *Activity diagram* ubah *password* domain

4.2.3 Relasi Tabel Basis Data

Basis data merupakan kumpulan informasi yang disimpan di dalam komputer dan dapat digunakan dan diolah oleh suatu sistem atau program komputer. Untuk penelitian ini, relasi tabel basis data dapat dilihat pada Gambar 4.7 berikut.



Gambar 4.7 Relasi tabel basis data

Berikut ini adalah penjelasan lebih rinci mengenai setiap tabel pada basis data berdasarkan Gambar 4.7 diatas.

- a. Tabel login, berisi *username* dan *password* pakar. Silahkan lihat Tabel 4.4 berikut.

Tabel 4.4 Tabel login

Nama	Type	Panjang	Keterangan
Username	VARCHAR	20	Nama pengguna pakar
Password	INT	100	Kata sandi pakar

- b. Tabel masalah, berisi informasi mengenai data domain atau domain permasalahan. Silahkan lihat Tabel 4.5 berikut.

Tabel 4.5 Tabel masalah

Nama	Type	Panjang	Keterangan
id_masalah	INT	20	<i>Primary key</i>
masalah	VARCHAR	100	Nama masalah atau domain
password	VARCHAR	500	Suatu domain memiliki kata sandi sehingga hanya bisa diakses oleh pakar yang bersangkutan.
batas	FLOAT	-	Setiap domain memiliki batas atau <i>threshold</i> yang berperan dalam menentukan apakah kasus baru mirip dengan kasus yang ada di basis data.

tipe	INT	1	Setiap domain memiliki tipe atau jenis fungsi similaritas tersendiri seperti yang telah dijelaskan sebelumnya.
alpha	INT	20	Alpha berisi nilai konstanta positif.

c. Tabel gejala, berisi informasi mengenai data gejala. Silahkan lihat Tabel 4.6 berikut.

Tabel 4.6 Tabel gejala

Nama	Tipe	Panjang	Keterangan
id_gejala	INT	20	Primary key
id_masalah	INT	20	Foreign key (dari tabel masalah)
gejala	VARCHAR	500	Nama gejala

d. Tabel solusi, berisi informasi mengenai data solusi. Silahkan lihat Tabel 4.7 berikut.

Tabel 4.7 Tabel solusi

Nama	Tipe	Panjang	Keterangan
id_solusi	INT	20	Primary key
id_masalah	INT	20	Foreign key (dari tabel masalah)
diagnosis	VARCHAR	100	Berisi hasil diagnosis
solusi	VARCHAR	500	Berisi solusi yang disarankan oleh sist

e. Tabel kasus, berisi informasi mengenai data kasus. Silahkan lihat Tabel 4.8 berikut.

Tabel 4.8 Tabel kasus

Nama	Tipe	Panjang	Keterangan
id	INT	20	Primary key
id_masalah	INT	20	Foreign key (dari tabel masalah)
id_kasus	INT	20	Berisi id dari setiap kasus
id_gejala	INT	20	Foreign key (dari tabel gejala)
bobot	INT	20	Setiap gejala yang ada pada suatu kasus memiliki bobot tersendiri
id_solusi	INT	20	Foreign key (dari tabel solusi)

f. Tabel kasustemp, berisi informasi mengenai data kasus yang dimasukkan oleh pengguna non-pakar. Tabel ini akan menampung kasus-kasus tersebut sebelum pengguna pakar memutuskan apakah kasus tersebut layak untuk dijadikan basis pengetahuan baru atau tidak. Silahkan lihat Tabel 4.9 berikut.

Tabel 4.9 Tabel kasustemp

Nama	Tipe	Panjang	Keterangan
------	------	---------	------------

id	INT	20	<i>Primary key</i>
id_masalah	INT	20	<i>Foreign key</i> (dari tabel masalah)
id_kasus	INT	20	Berisi id dari setiap kasus
id_gejala	INT	20	<i>Foreign key</i> (dari tabel gejala)
bobot	INT	20	Setiap gejala yang ada pada suatu kasus memiliki bobot tersendiri
id_solusi	INT	20	<i>Foreign key</i> (dari tabel solusi)
sim	INT	20	Nilai similaritas kasus dengan kasus yang ada pada basis pengetahuan.
fungsi	INT	20	Berisi informasi tipe fungsi similaritas yang digunakan saat menghitung nilai similaritas.

- g. Tabel pakar, berisi informasi mengenai data pakar yang mengelola suatu domain. Silahkan lihat Tabel 4.10 berikut.

Tabel 4.10 Tabel pakar

Nama	Tipe	Panjang	Keterangan
id	INT	20	<i>Primary key</i>
id_masalah	INT	20	<i>Foreign key</i> (dari tabel masalah)
pakar	VARCHAR	100	Berisi informasi nama pakar
info	VARCHAR	255	Berisi informasi dari pakar, seperti nomor telepon, dan lain-lain.

- h. Tabel pesan, berisi pesan-pesan yang berasal dari pengguna non-pakar. Silahkan lihat Tabel 4.11 berikut.

Tabel 4.11 Tabel pesan

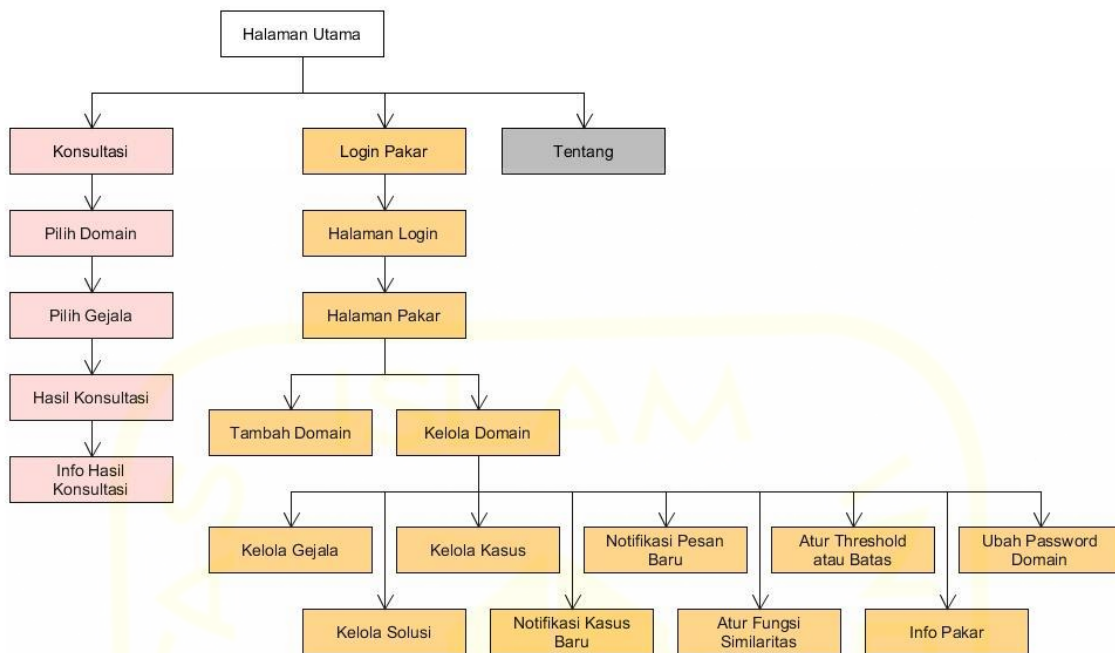
Nama	Tipe	Panjang	Keterangan
id	INT	20	<i>Primary key</i>
id_masalah	INT	20	<i>Foreign key</i> (dari tabel masalah)
pesan	VARCHAR	100	Berisi pesan yang dikirimkan dari pengguna non-pakar/

4.3 Implementasi

Pada tahap ini, penulis melakukan pengembangan sistem berdasarkan desain yang telah disusun pada tahap sebelumnya.

4.3.1 Struktur Menu

Struktur menu merupakan alur sistem yang ditampilkan dan dipaparkan dalam bentuk menu-menu yang tersedia di dalam sistem. Silahkan lihat Gambar 4.8 berikut.



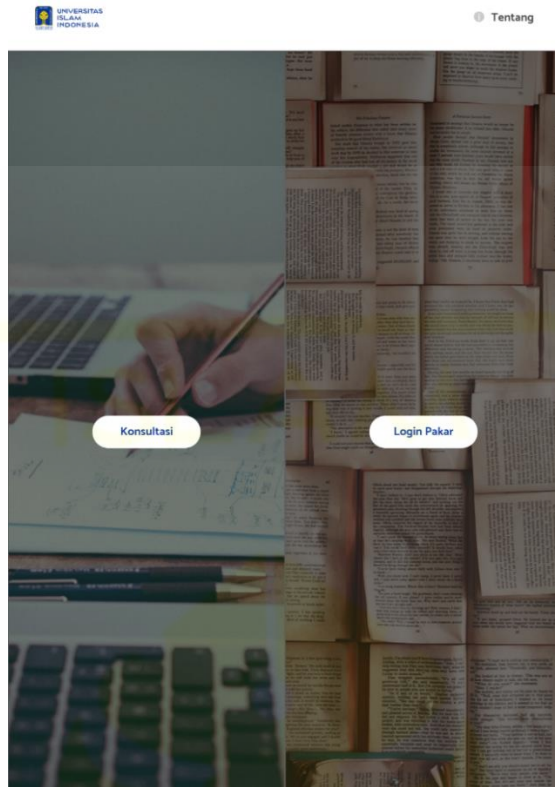
Gambar 4.8 Struktur menu

Berdasarkan Gambar 4.8 di atas, halaman utama akan muncul saat awal sistem digunakan. Halaman tersebut akan menampilkan 3 pilihan, yakni konsultasi yang akan diakses oleh tipe pengguna non-pakar, login pakar yang akan diakses oleh tipe pengguna pakar, dan halaman tentang yang menjelaskan tentang sistem.

4.3.2 Interface

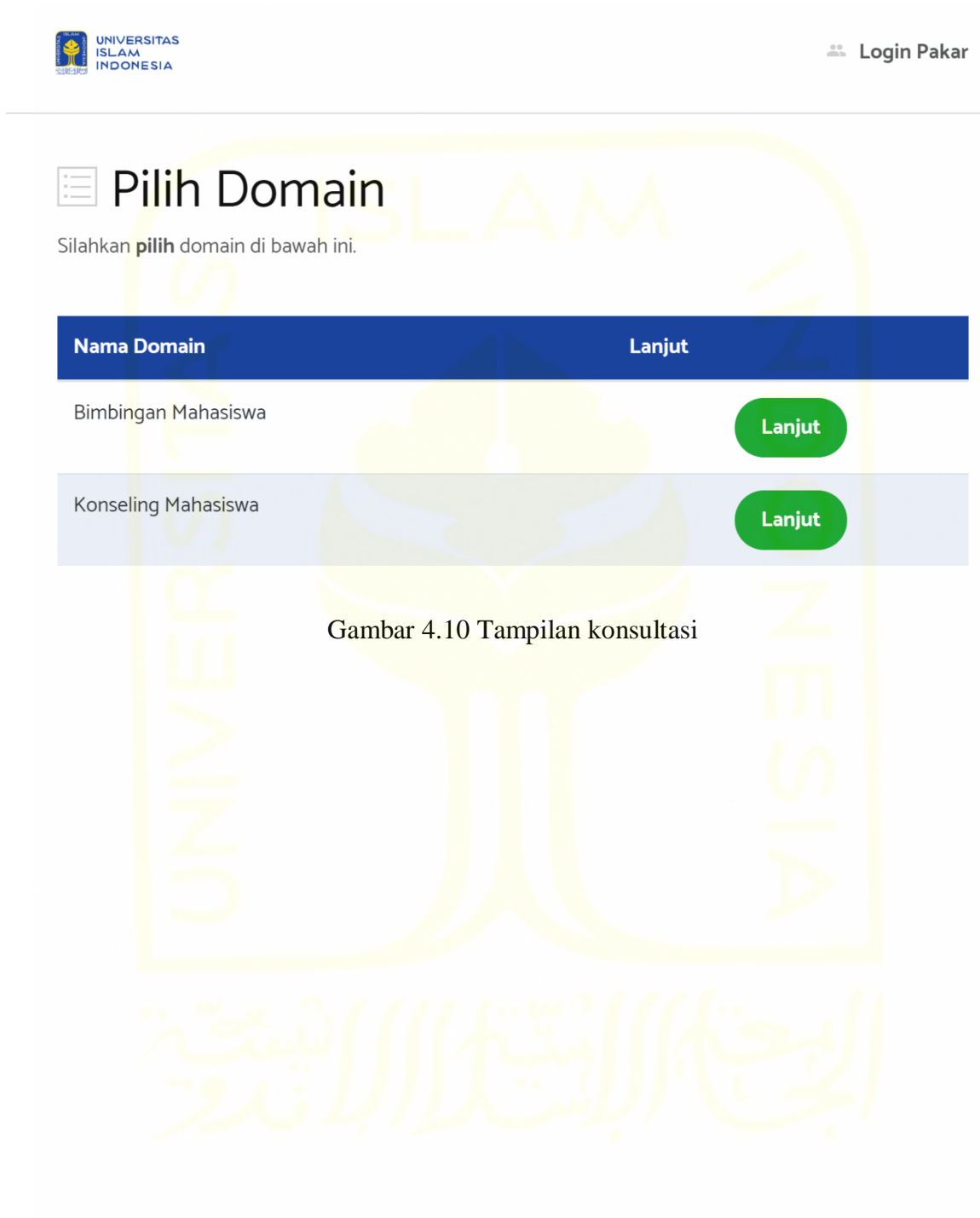
Tampilan antarmuka atau *interface* dirancang dengan menggunakan framework Bootstrap. Kriteria tampilan yang baik haruslah mudah dimengerti, bersih, konsisten, responsive, menarik, dan *familiar*. Berikut ini adalah hasil tampilan sistem yang sudah dikembangkan.

- a. Antarmuka *home*. Halaman ini merupakan halaman yang pertama kali muncul saat sistem digunakan. Silahkan lihat Gambar 4.9 berikut.





Gambar 4.9 Tampilan *home*

- b. Antarmuka konsultasi, halaman ini akan menampilkan domain yang akan dipilih oleh pengguna non-pakar. Silahkan lihat Gambar 4.10 berikut.



- c. Antarmuka pilih gejala. Pada halaman ini pengguna non-pakar akan memilih gejala-gejala yang dianggap relevan. Silahkan lihat Gambar 4.11 berikut.

 UNIVERSITAS ISLAM INDONESIA

 Login Pakar

✓ Konsultasi

Nama Domain : **Konseling Mahasiswa**
Total Gejala : **41 Gejala**

Silahkan **pilih** gejala di bawah ini yang **sesuai** dengan apa yang Anda alami.
Tidak menemukan gejala yang sesuai? Silahkan klik [link ini](#) untuk memberitahu pakar.

- Sulit tidur
- Melakukan self-harm
- Memiliki masalah asmara
- Merasa hidup tidak berguna
- Sering menangis
- Sulit berkonsentrasi
- Kawatir akan masa depan
- Tidak memahami materi kuliah
- Merasa takut
- Merasa cemas
- Nilai mengecewakan
- Tidak percaya diri
- Sulit mengendalikan emosi
- Merasa kesepian
- Suasana hati yang berubah-ubah

Gambar 4.11 Tampilan pilih gejala

- d. Antarmuka hasil diagnosis sukses. Halaman ini menampilkan hasil diagnosis yang dilakukan oleh sistem. Informasi yang ditampilkan diantaranya adalah hasil diagnosis, nilai simiaritas, dan solusi yang akan diambil. Silahkan lihat Gambar 4.12 berikut.

* Hasil Diagnosis

Nama Domain : **Konseling Mahasiswa**

Berikut adalah **hasil diagnosis sistem** berdasarkan gejala yang telah Anda pilih.

Hasil Diagnosis : **Stress**

Similaritas : **50%**

Solusi :

Disarankan mengurangi kegiatan-kegiatannya yang terlalu banyak, kemudian meningkatkan kualitas komunikasi dengan ayah dan ibu, serta banyak mendoakan orangtua.

Gejala yang dipilih

Sulit tidur

Melakukan self-harm

Sering menangis

Kawatir akan masa depan

Info

Informasi Pakar

Gambar 4.12 Tampilan hasil diagnosis sukses

- e. Antarmuka hasil diagnosis gagal. Halaman ini muncul ketika hasil diagnosis yang dilakukan oleh sistem tidak melampaui nilai batas atau *threshold*. Silahkan lihat Gambar 4.13 berikut.



Gambar 4.13 Tampilan hasil diagnosis gagal

- f. Antarmuka informasi diagnosis. Antarmuka ini memberikan informasi lebih rinci mengenai hasil diagnosis yang dilakukan oleh sistem. Silahkan lihat Gambar 4.14 berikut.

i Informasi Diagnosis

Nama Domain : **Konseling Mahasiswa**

Berikut adalah **informasi** dari hasil diagnosis Anda. Informasi ini menampilkan **tingkat kemiripan** pada kasus Anda jika dibandingkan dengan kasus-kasus sebelumnya.

Gejala yang Anda pilih sebelumnya:

- 1 - Sulit tidur
- 2 - Melakukan self-harm
- 3 - Sering menangis
- 4 - Kawatir akan masa depan

Hasil Diagnosis : **Stress**

Similaitiras : **50%**

Threshold : **30%**

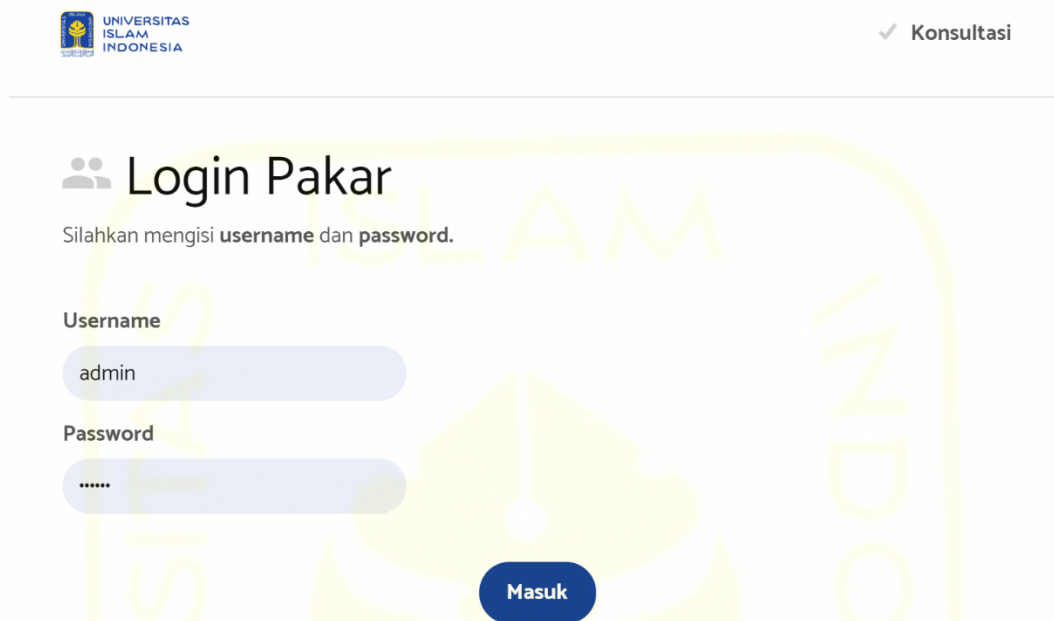
Solusi:

Disarankan mengurangi kegiatan-kegiatannya yang terlalu banyak, kemudian meningkatkan kualitas komunikasi dengan ayah dan ibu, serta banyak mendoakan orangtua.

Kasus	Gejala	Diagnosis	Solusi	%	Pilih
1	1 - Sulit tidur ✓ 2 - Melakukan self-harm ✓ 3 - Memiliki masalah asmara 4 - Merasa hidup tidak berguna	Belum mampu menerima keadaan riil diri sendiri.	Menyadari bahwa kesuksesan tidak hanya ditentukan oleh IP dan kecepatan waktu lulus. Kemudian mencoba fokus untuk melakukan apa yang dapat dilakukannya sekarang (menyelesaikan skripsi).	50	Pilih
30	1 - Sering menangis ✓ 2 - Merasa takut 3 - Memiliki masalah dengan orang tua/keluarga	Kebiasaan berpikir yang irrasional	Silahkan melakukan pertemuan dengan konselor untuk tindak lanjut.	33	Pilih
39	1 - Sering menangis ✓ 2 - Kegiatan yang overload	Stress	Disarankan mengurangi kegiatan-kegiatannya yang terlalu banyak, kemudian meningkatkan kualitas komunikasi dengan ayah dan ibu, serta banyak mendoakan orangtua.	50	Pilih

Gambar 4.14 Tampilan informasi diagnosis

- g. Antarmuka *login* pakar. Halaman ini muncul ketika pakar ingin masuk ke dalam sistem editor. Silahkan lihat Gambar 4.15 berikut.



UNIVERSITAS ISLAM INDONESIA

✓ Konsultasi

Login Pakar

Silahkan mengisi **username** dan **password**.

Username

admin

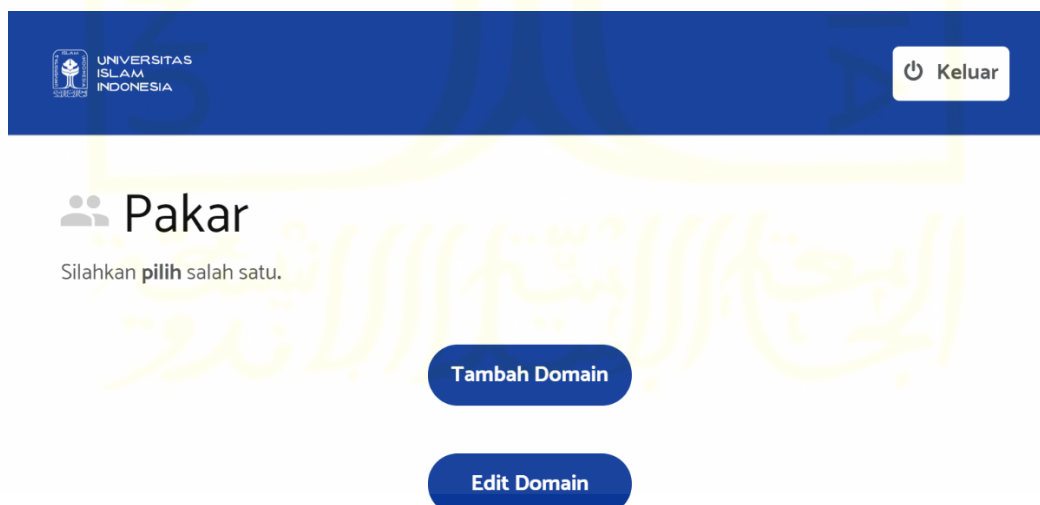
Password

.....

Masuk

Gambar 4.15 Tampilan *login* pakar

- h. Antarmuka *home* pakar. Halaman ini muncul setelah pakar sukses melakukan *login*. Silahkan lihat Gambar 4.16 berikut.



UNIVERSITAS ISLAM INDONESIA

Keluar

Pakar

Silahkan **pilih** salah satu.

Tambah Domain

Edit Domain

Gambar 4.16 Tampilan *home* pakar

- i. Antarmuka tambah domain. Halaman ini diakses oleh pakar ketika akan membuat domain baru. Silahkan lihat Gambar 4.17 berikut.

UNIVERSITAS ISLAM INDONESIA

Keluar

+ Tambah Domain

Silahkan mengisi **nama domain** dan **password**.

Nama Domain

Batas / Threshold (%)

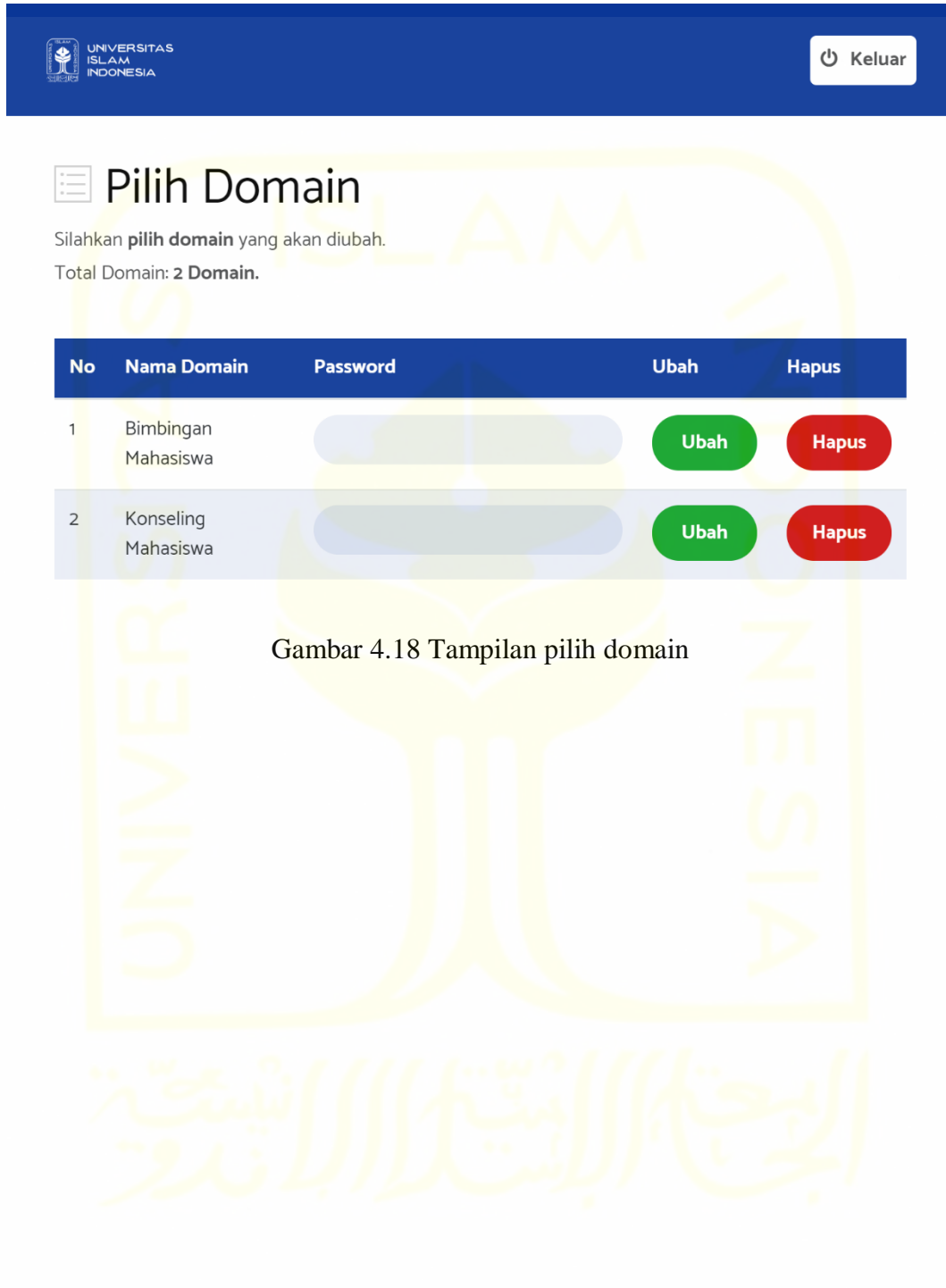
Password Domain

Ulangi Password Domain

Tambah

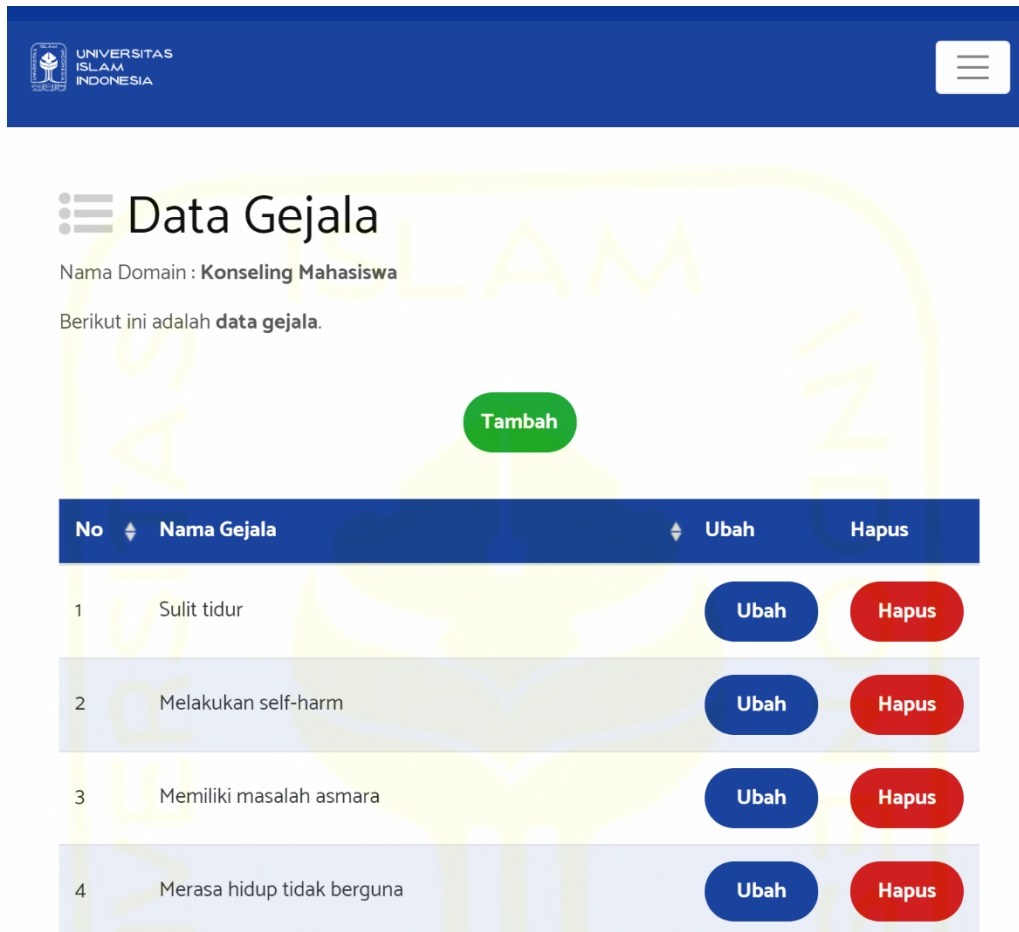
Gambar 4.17 Tampilan tambah domain

- j. Antarmuka pilih domain. Halaman ini menampilkan domain yang ada di dalam sistem. Silahkan lihat Gambar 4.18 berikut.



Gambar 4.18 Tampilan pilih domain

- k. Antarmuka kelola gejala. Antarmuka ini menampilkan daftar gejala yang ada. Silahkan lihat Gambar 4.19 berikut.



UNIVERSITAS ISLAM INDONESIA

Data Gejala

Nama Domain : **Konseling Mahasiswa**

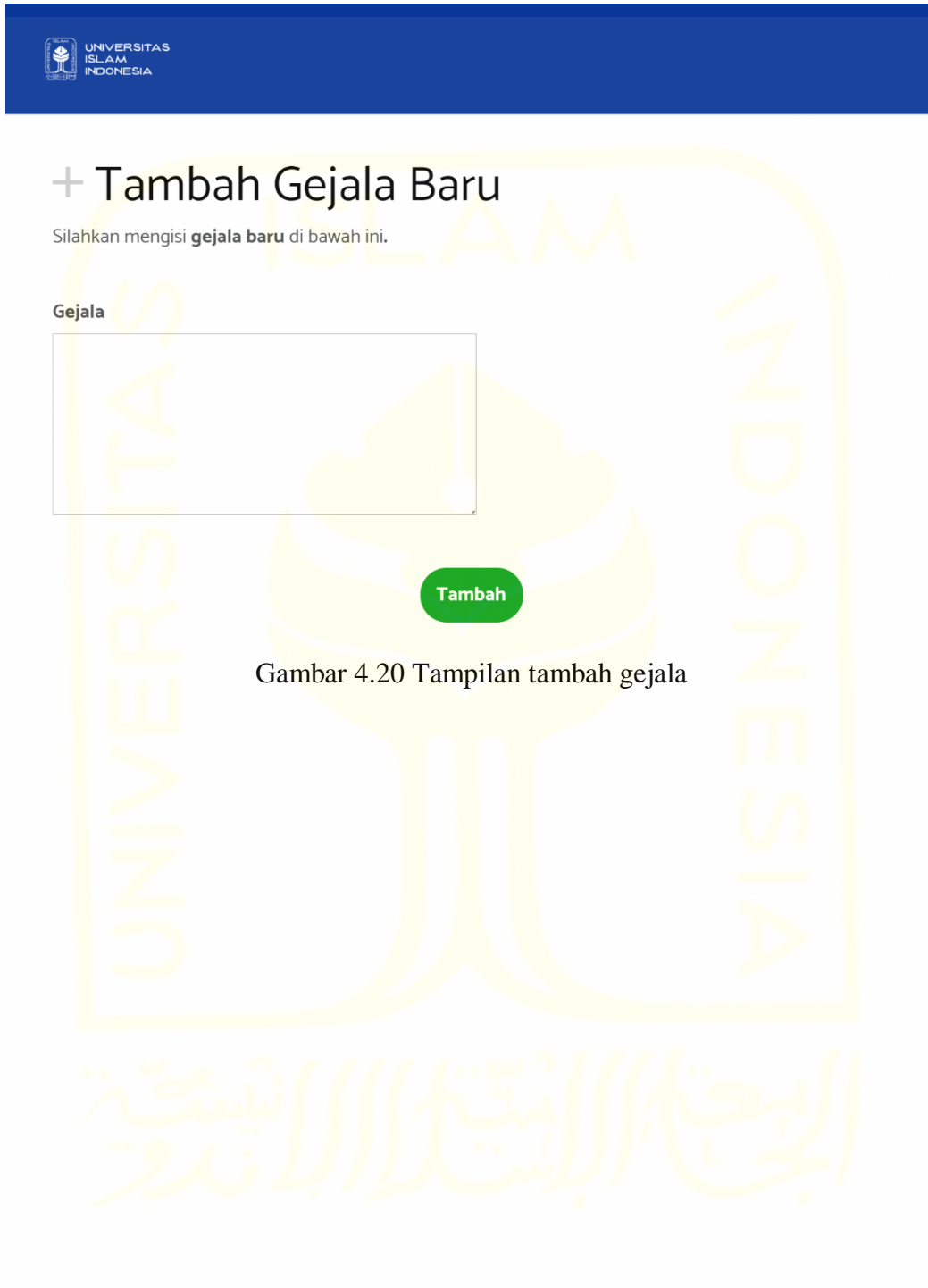
Berikut ini adalah **data gejala**.

Tambah

No	Nama Gejala	Ubah	Hapus
1	Sulit tidur	Ubah	Hapus
2	Melakukan self-harm	Ubah	Hapus
3	Memiliki masalah asmara	Ubah	Hapus
4	Merasa hidup tidak berguna	Ubah	Hapus

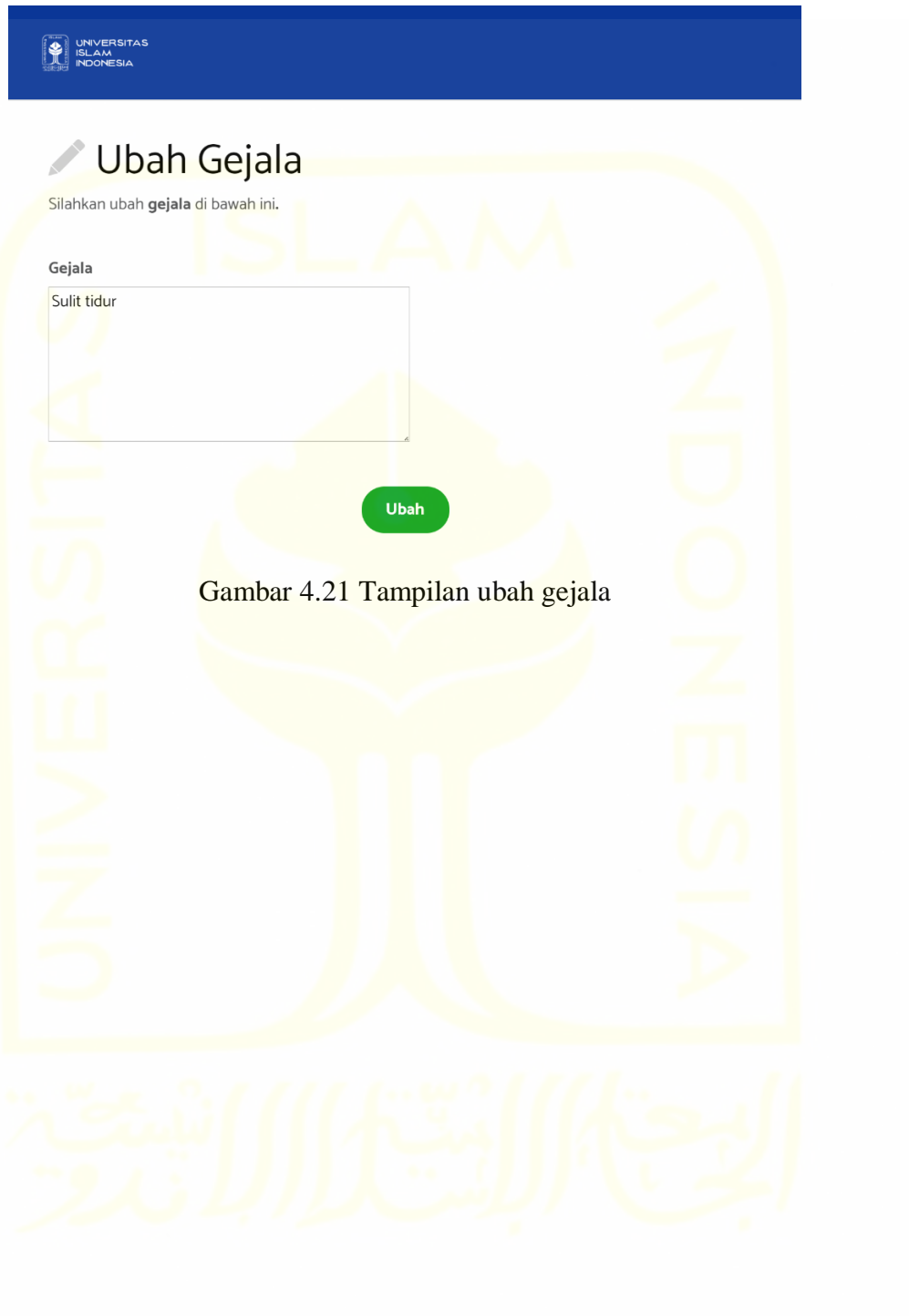
Gambar 4.19 Tampilan kelola gejala

1. Antarmuka tambah gejala. Pakar mengakses halaman ini ketika akan membuat gejala baru. Silahkan lihat Gambar 4.20 berikut.



Gambar 4.20 Tampilan tambah gejala

- m. Antarmuka ubah gejala. Halaman ini menampilkan gejala tertentu yang ingin diubah, Silahkan lihat Gambar 4.21 berikut.



Gambar 4.21 Tampilan ubah gejala

- n. Antarmuka kelola solusi. Antarmuka ini menampilkan daftar solusi yang ada. Silahkan lihat Gambar 4.22 berikut.

UNIVERSITAS ISLAM INDONESIA

✓ Data Solusi

Nama Domain : **Konseling Mahasiswa**

Berikut ini adalah **data diagnosis dan solusi**.

Tambah

No	Diagnosis	Solusi	Ubah	Hapus
1	Depresif ringan	Melakukan pengembangan diri dan penggalan potensi dengan kegiatan-kegiatan positif.	Ubah	Hapus
2	Problem Penyesuaian Pada Masa Transisi Siklus Kehidupan	Mengganti pikiran-pikiran terdahulu yang keliru dengan pemikiran positif, melakukan kegiatan yang lebih bermanfaat dan disenangi sebagai bagian dari upaya mengatasi masalah. Mencoba melawan rasa malas.	Ubah	Hapus
3	Gangguan Panik	Membiasakan diri untuk berfikir positif sebagai pengganti pikiran-pikiran yang keliru. Membiasakan diri melakukan relaksasi nafas tiap harinya. Melakukan kegiatan positif yang menyenangkan	Ubah	Hapus

Gambar 4.22 Tampilan kelola solusi

- o. Antarmuka tambah solusi. Pakar mengakses halaman ini ketika akan membuat solusi baru. Silahkan lihat Gambar 4.23 berikut.



UNIVERSITAS ISLAM INDONESIA

+ Tambah Solusi Baru

Silahkan mengisi **solusi baru** di bawah ini.

Diagnosa

Solusi

Tambah

Gambar 4.23 Tampilan tambah solusi

- p. Antarmuka ubah solusi. Halaman ini menampilkan solusi tertentu yang ingin diubah. Silahkan lihat Gambar 4.24 berikut.

UNIVERSITAS ISLAM INDONESIA

Ubah Solusi

Silahkan ubah **diagnosis dan solusi** di bawah ini.

Diagnosis

Depresif ringan

Solusi

Melakukan pengembangan diri dan penggalian potensi dengan kegiatan-kegiatan positif.

Ubah

Gambar 4.24 Tampilan ubah solusi

- q. Antarmuka kelola kasus. Antarmuka ini menampilkan daftar solusi yang ada. Silahkan lihat Gambar 4.25 berikut.

UNIVERSITAS ISLAM INDONESIA

Data Kasus

Nama Domain : **Konseling Mahasiswa**

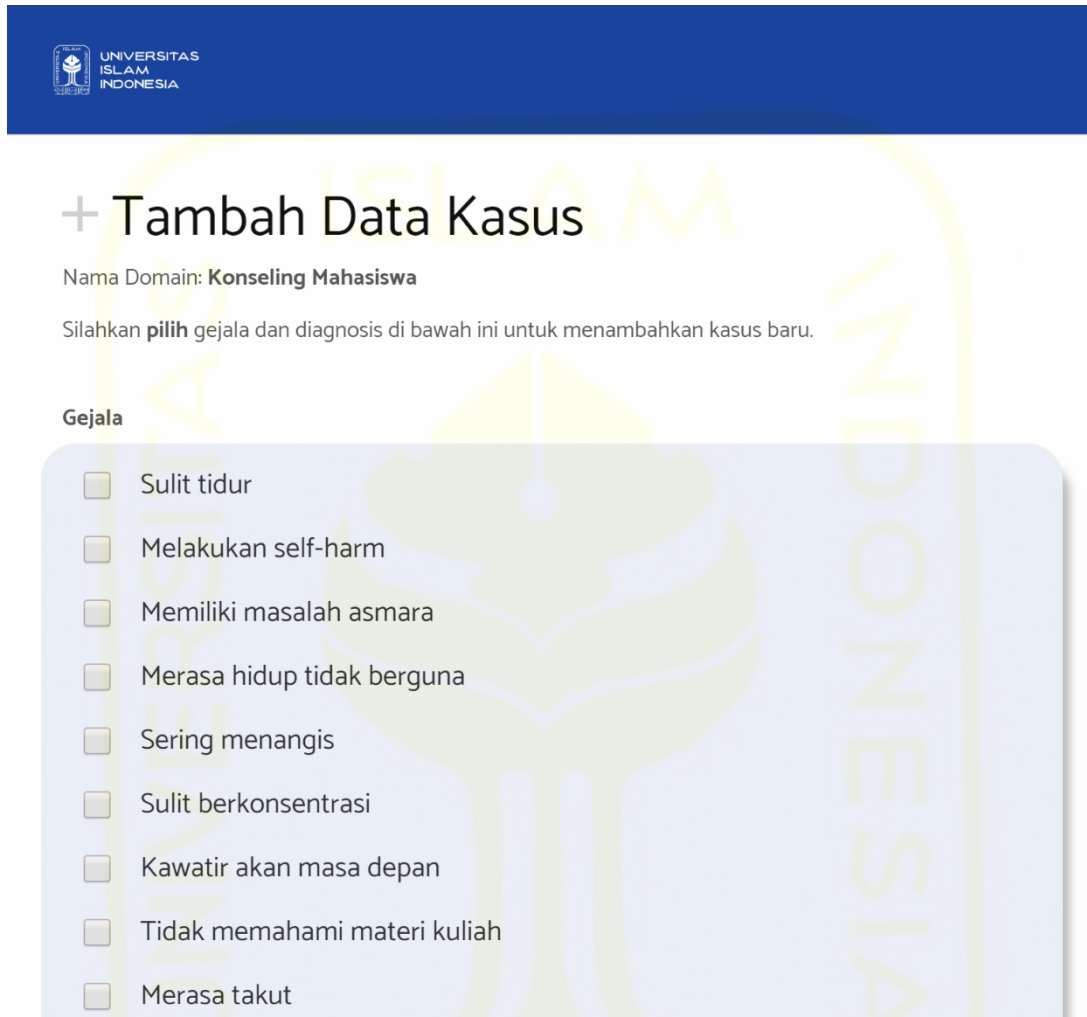
Berikut ini adalah **data kasus** yang terdiri dari gejala-gejala dan solusi.

[Tambah](#)

No	Gejala	Diagnosis	Solusi	Ubah	Hapus
1	<ul style="list-style-type: none"> 1 - Sulit tidur 2 - Melakukan self-harm 3 - Memiliki masalah asmara 4 - Merasa hidup tidak berguna 	Belum mampu menerima keadaan riil diri sendiri.	Menyadari bahwa kesuksesan tidak hanya ditentukan oleh IP dan kecepatan waktu lulus. Kemudian mencoba fokus untuk melakukan apa yang dapat dilakukannya sekarang (menyelesaikan skripsi).	Ubah	Hapus
2	<ul style="list-style-type: none"> 1 - Sulit tidur 2 - Memiliki masalah asmara 3 - Sering menangis 4 - Sulit berkonsentrasi 	Depresif ringan	Melakukan pengembangan diri dan penggalian potensi dengan kegiatan-kegiatan positif.	Ubah	Hapus

Gambar 4.25 Tampilan antarkasus

- r. Antarmuka tambah kasus. Pakar mengakses halaman ini ketika akan membuat solusi baru. Silahkan lihat Gambar 4.26 berikut.



UNIVERSITAS
ISLAM
INDONESIA

+ Tambah Data Kasus

Nama Domain: **Konseling Mahasiswa**

Silahkan **pilih** gejala dan diagnosis di bawah ini untuk menambahkan kasus baru.

Gejala

- Sulit tidur
- Melakukan self-harm
- Memiliki masalah asmara
- Merasa hidup tidak berguna
- Sering menangis
- Sulit berkonsentrasi
- Kawatir akan masa depan
- Tidak memahami materi kuliah
- Merasa takut

Gambar 4.26 Tampilan tambah kasus

- s. Antarmuka ubah kasus. Halaman ini menampilkan solusi tertentu yang ingin diubah. Silahkan lihat Gambar 4.27 berikut.

UNIVERSITAS ISLAM INDONESIA

Ubah Data Kasus

Nama Domain: Konseling Mahasiswa.

Silahkan ubah data kasus bawah ini.

Gejala

Gejala sebelumnya:

- 1 - Sulit tidur
- 2 - Melakukan self-harm
- 3 - Memiliki masalah asmara
- 4 - Merasa hidup tidak berguna

- Sulit tidur
- Melakukan self-harm
- Memiliki masalah asmara
- Merasa hidup tidak berguna
- Sering menangis
- Sulit berkonsentrasi
- Kawatir akan masa depan
- Tidak memahami materi kuliah
- Merasa takut
- Merasa cemas
- Nilai mengecewakan
- Tidak percaya diri
- Sulit mengendalikan emosi

Gambar 4.27 Tampilan ubah kasus

- t. Antarmuka notifikasi kasus baru. Halaman ini menampilkan daftar kasus-kasus baru yang berasal dari pengguna non-pakar yang melakukan proses diagnosis sebelumnya. Silahkan lihat Gambar 4.28 berikut.

UNIVERSITAS ISLAM INDONESIA

⚠ Notifikasi Kasus Baru

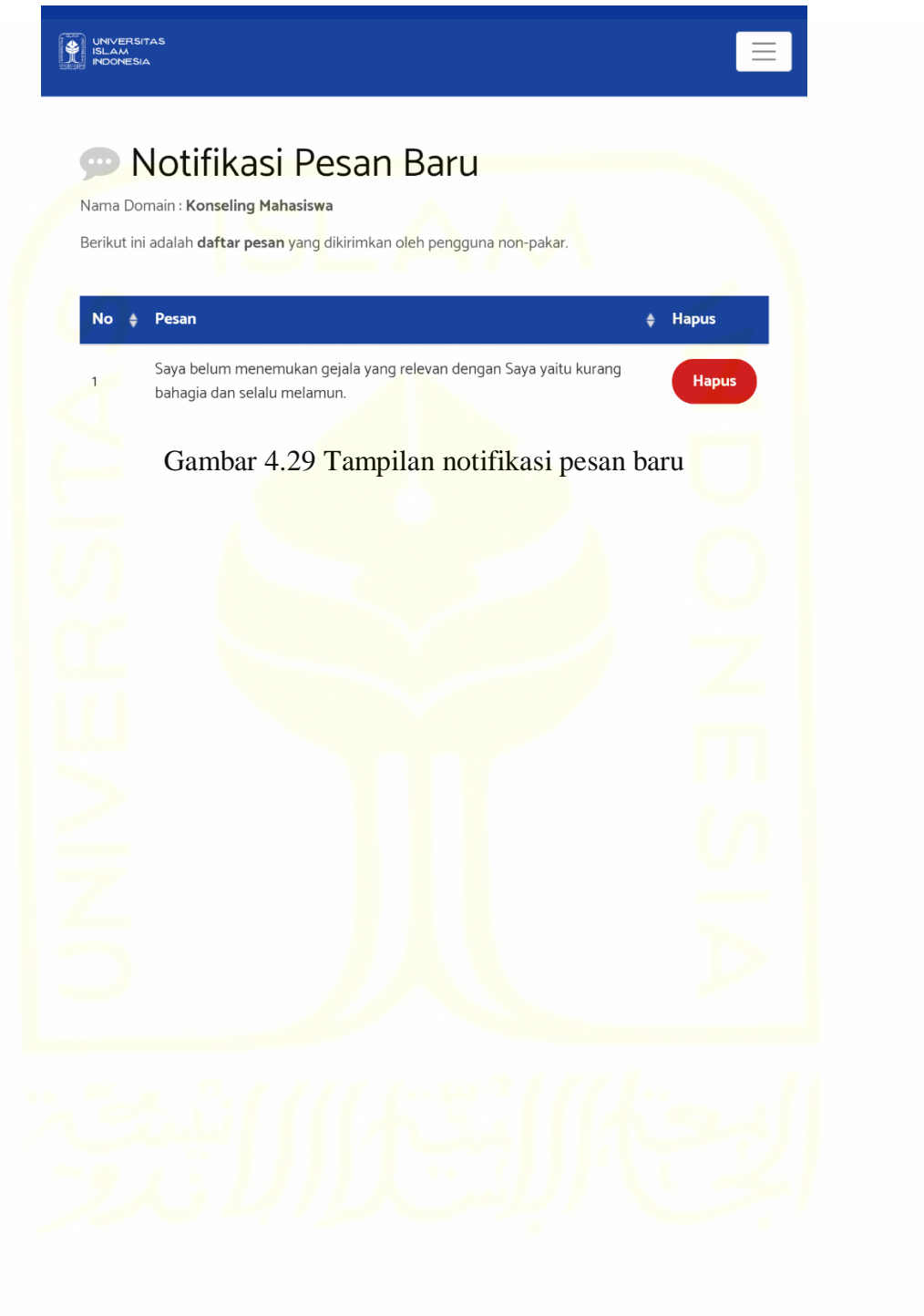
Nama Domain : **Konseling Mahasiswa**
 Batas / Threshold : **30%**

Data kasus berikut ini berasal dari **hasil diagnosis** yang dijalankan oleh sistem. Pakar bisa **menambahkan** kasus ke dalam basis pengetahuan, **menghapus** kasus, atau **mengganti** solusi jika dirasa kurang akurat dan tepat.

No	Fungsi	%	Gejala	Diagnosis	Solusi	Tambah	Hapus
1	2	50	1 - Memiliki masalah dengan orang tua/keluarga 2 - Tidak memiliki motivasi 3 - Memiliki masalah dengan skripsi	Konflik dengan diri se	Melakukan empty chair untuk mengeluarkan semua perasaan yang tidak pernah tersampaikan.	Tambah	Hapus
			1 - Sering menangis 2 - Merasa kesepian		Silahkan melakukan pertemuan dengan		

Gambar 4.28 Tampilan kasus filter

- u. Antarmuka notifikasi pesan baru. Halaman ini menampilkan daftar pesan yang berasal dari pengguna non-pakar. Silahkan lihat Gambar 4.29 berikut.



Gambar 4.29 Tampilan notifikasi pesan baru

- v. Antarmuka atur fungsi similaritas. Halaman atur fungsi similaritas menampilkan tipe-tipe fungsi similaritas yang akan dipilih oleh pakar. Selain itu, Setiap algoritma atau metode perhitungan fungsi similaritas juga ditampilkan di halaman ini. Silahkan lihat Gambar 4.30 berikut.

UNIVERSITAS ISLAM INDONESIA

Atur Fungsi Similaritas

Nama Domain : **Konseling Mahasiswa**

Silahkan **pilih** fungsi bawah ini.

Fungsi Similaritas

2. Pencocokan basis I ▾

α (hanya untuk fungsi similaritas 4)

1

Algoritma

Persamaan ini membagi jumlah komponen yang sama antara kasus baru dan kasus yang ada pada basis pengetahuan dengan maksimum antara banyaknya komponen pada kasus baru dan pada kasus yang ada pada basis pengetahuan.

$$T_i = \frac{\sum_{j=1}^n \text{komponen yang sama antara } C \text{ dan } K_i}{\text{Max}(m, n)}$$

Keterangan:

1. T_i : Nilai similaritas
2. C : Kasus baru
3. K_i : Kasus pada basis pengetahuan
4. n : Banyak komponen K_i
5. m : Banyak komponen C
6. α : Konstanta positif (hanya untuk fungsi similaritas 4)

Atur

Gambar 4.30 Tampilan atur fungsi similaritas

- w. Antarmuka atur threshold/batas. Halaman ini berguna untuk mengubah nilai *threshold* dari domain yang dikelola. Silahkan lihat Gambar 4.31 berikut.



% Atur Threshold / Batas

Nama Domain : **Konseling Mahasiswa**

Silahkan isi nilai bawah ini.

Threshold / Batas (%)

30

Atur

Gambar 4.31 Tampilan ubah password domain

- x. Antarmuka data pakar. Halaman ini menampilkan informasi dari pakar yang mengelola domain tersebut. Silahkan lihat Gambar 4.32 berikut.

Data Pakar

Nama Domain : **Konseling Mahasiswa**

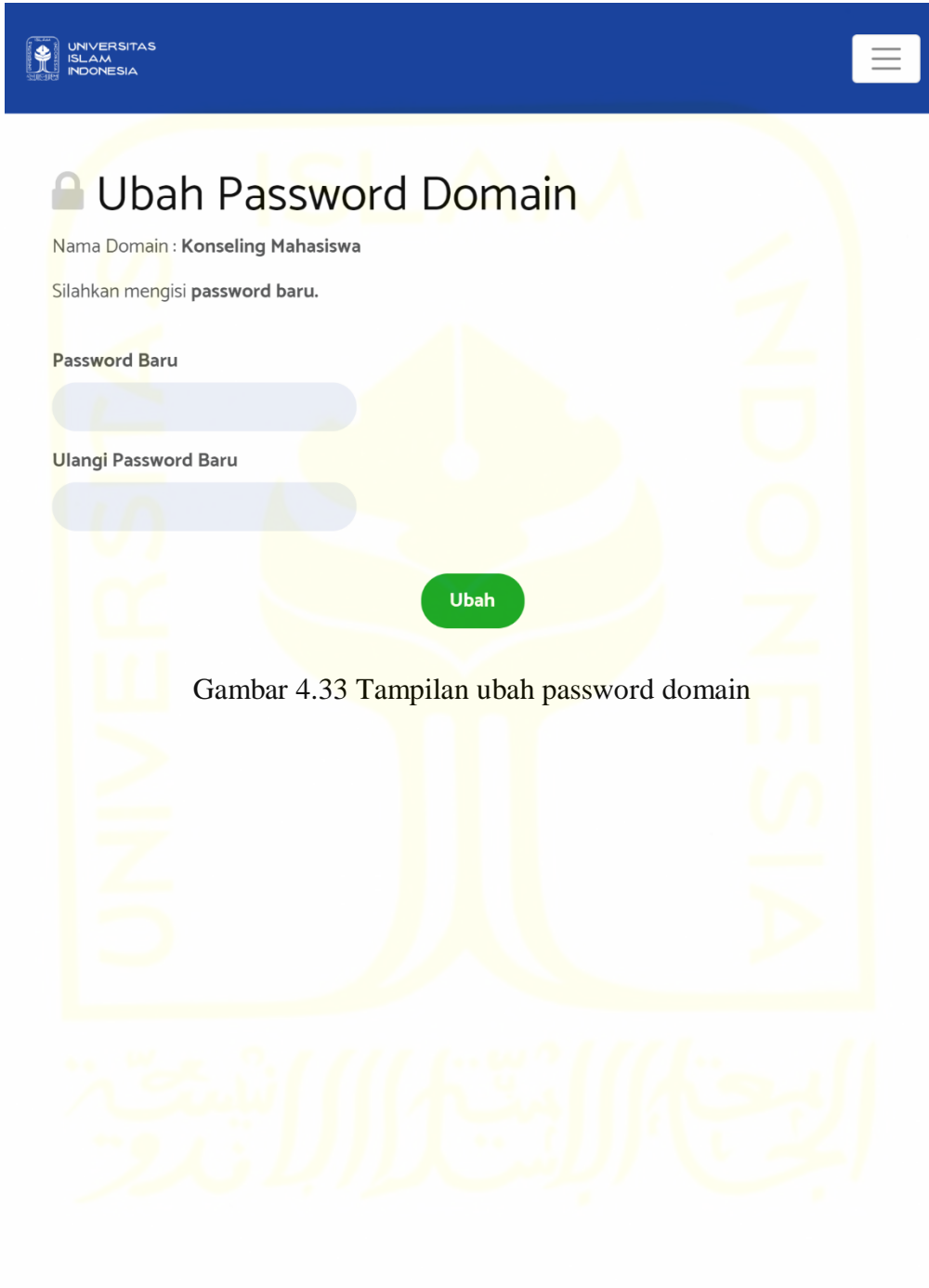
Berikut ini adalah **pakar**. Pengguna non-pakar dapat menggunakan informasi ini untuk menghubungi pakar.

Tambah

No	Pakar	Info	Ubah	Hapus
1	Pakar1	InfoPakar1	Ubah	Hapus
2	Pakar2	InfoPakar2	Ubah	Hapus

Gambar 4.32 Tampilan data pakar

- y. Antarmuka ubah password domain. Halaman ini berperan dalam memfasilitasi pakar dalam mengubah *password* atau kata sandi domain yang dikelola. Silahkan lihat Gambar 4.33 berikut.



The screenshot shows a web interface for changing a domain password. At the top, there is a dark blue header with the Universitas Islam Indonesia logo and name on the left, and a white menu icon on the right. The main content area has a white background with a large, faint watermark of the university's logo and name. The title 'Ubah Password Domain' is displayed in a large, bold font, preceded by a lock icon. Below the title, the text 'Nama Domain : Konseling Mahasiswa' is shown. A prompt 'Silahkan mengisi password baru.' is followed by two input fields: 'Password Baru' and 'Ulangi Password Baru'. A green 'Ubah' button is positioned below the input fields. At the bottom of the page, there is a line of Arabic calligraphy.

Gambar 4.33 Tampilan ubah password domain

- z. Antarmuka tentang. Halaman ini menampilkan tentang sistem. Silahkan lihat Gambar 4.34 berikut.

Tentang Sistem

Sebuah sistem pakar pada dasarnya terdiri dari 2 bagian, yakni bagian pakar dan bagian non-pakar. Bagian pakar berperan dalam proses knowledge management atau proses manajemen basis pengetahuan sedangkan bagian non-pakar berperan dalam proses penarikan solusi atau kesimpulan dengan menggunakan basis pengetahuan yang ada.

Pada bagian pakar, manajemen pengetahuan dilakukan dengan menggunakan alat yang disebut sebagai editor basis pengetahuan. **Editor basis pengetahuan merupakan sebuah tools atau alat yang memberikan akses kepada pengguna pakar untuk menambah, menghapus, dan memodifikasi aturan-aturan yang ada di dalam basis pengetahuan. Keberadaan editor basis pengetahuan sangat diperlukan karena dapat membantu pakar dalam mengelola basis pengetahuan dengan mudah.**

Sebuah basis pengetahuan biasanya dikembangkan dengan dua metode berikut; rule based reasoning (menggunakan aturan untuk mendapatkan kesimpulan atau solusi) atau dengan metode case based reasoning (menggunakan nilai threshold atau nilai batas kemiripan untuk mendapatkan kesimpulan atau solusi) (Prentzas & Hatzilygeroudis, 2003).

Umumnya, sebuah basis pengetahuan dikembangkan dengan menggunakan pendekatan rule based reasoning atau pendekatan berbasis aturan dan pengetahuan tersebut dipresentasikan dalam bentuk IF-THEN (Berka, 2011) dibandingkan dengan menggunakan pendekatan berbasis kasus atau case based reasoning. Selain itu, masih sedikit sekali orang yang memiliki pengalaman dalam mengoperasikan sistem berbasis kasus ini (Watson & Marir, 1994).

Pengembangan editor basis pengetahuan yang hanya terkait dengan satu domain atau satu masalah saja ternyata bukanlah hal yang praktis (Paley, et al., 1997). Akibatnya, jika ada satu domain yang ingin dijadikan sebuah basis pengetahuan, maka seorang knowledge engineer harus membangun sebuah editor basis pengetahuan yang baru dari awal. Oleh karena itu, dibutuhkanlah sebuah editor basis pengetahuan yang memungkinkan para pakar untuk mengakses berbagai domain atau masalah tersebut melalui satu antarmuka atau satu sistem yang tunggal saja.

Gambar 4.34 Tampilan tentang

4.3.3 Source Code

Source code merupakan kumpulan kode program yang berisi perintah atau instruksi terhadap sistem dan mampu dipahami oleh komputer. Berikut adalah beberapa contoh potongan *source code* yang memiliki menambah domain dan mengelola gejala.

- a. Tambah domain. Silahkan lihat Gambar 4.35 berikut.

```
<?php
    include "connP.php";

    if($_SERVER["REQUEST_METHOD"] == "POST") {

        $masalah = trim($_POST['masalah']);
        $pass = trim($_POST['pass']);
        $passUlang = trim($_POST['passUlang']);
        $pass = md5($pass);
        $passUlang = md5($passUlang);
        $batas = trim($_POST['batas']);

        $batas = $batas / 100;
```



```

        echo "$masalah<br>";
        echo "$pass<br>";
        echo "$passUlang<br>";
        echo "$batas<br>";

    }

    $select = "INSERT INTO masalah (masalah, password, batas, tipe) VALUES
('$masalah', '$passUlang', $batas, 1)";

    if ($koneksi->query($select) === TRUE) {
        echo "masalah sukses ditambahkan ";
        header("Location: pakarPilih.php");
    } else {
        echo "Error: " . $select . "<br>" . $koneksi->error;
        // header("Location: home.html");
    }

    echo "<br><br>";

    $koneksi->close();

?>

```

Gambar 4.35 Source code tambah domain

b. Tambah gejala. Silahkan lihat Gambar 4.36 berikut.

```

<?php

    include "connP.php";

    if($_SERVER["REQUEST_METHOD"] == "POST") {

        $gejala = trim($_POST['gejala']);
        $id_masalah = trim($_POST['id_masalah']);

        $masalah = trim($_POST['masalah']);
        $pass = trim($_POST['pass']);

        echo "$masalah<br>";
        echo "$pass<br>";
        echo "$id_masalah<br>";

    }

    $select = "INSERT INTO gejala (id_masalah, gejala) VALUES ('$id_masalah',
'$gejala')";

    if ($koneksi->query($select) === TRUE) {
        echo "gejala sukses ditambahkan ";
    } else {
        echo "Error: " . $select . "<br>" . $koneksi->error;
    }

    echo "<br><br>";

```

```

echo '<button type="button" onclick="history.back();">Kembali</button>';

$koneksi->close();

header("Location:
pakarEditGejala.php?id_masalah=$id_masalah&masalah=$masalah&pass=$pass");
exit();

?>

```

Gambar 4.36 Source code tambah gejala

c. Ubah gejala. Silahkan lihat Gambar 4.37 berikut.

```

<?php

include "connP.php";

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $id_gejala = trim($_POST['id_gejala']);
    $gejala = trim($_POST['gejala']);

    $id_masalah = trim($_POST['id_masalah']);

    $masalah = trim($_POST['masalah']);
    $pass = trim($_POST['pass']);

    echo "Data yang Anda kirimkan: <br><br>";
    echo "gejala : $gejala <br>";
    echo "id : $id <br>";
}

$sql = mysqli_query($koneksi, "UPDATE gejala SET gejala = '$gejala'
WHERE id_gejala = '$id_gejala' ");

if ($sql=== TRUE) {
    echo "****Data telah diubah.**** <br>";
} else {
    echo "****Gagal mengubah gejala.****<br><br> ".$koneksi->error;
}

echo '
<p><br><br><button type="button"
onclick="history.back();">Kembali</button></p>';

$koneksi -> close();

header("Location:
pakarEditGejala.php?id_masalah=$id_masalah&masalah=$masalah&pass=$pass");
exit();

?>

```

Gambar 4.37 Source code ubah gejala

d. Hapus gejala. Silahkan lihat Gambar 4.38 berikut.

```

<?php

include "connP.php";

$id_gejala = $_GET['id_gejala'];
$gejala = $_GET['gejala'];
$id_masalah = $_GET['id_masalah'];

$masalah = $_GET['masalah'];
$pass = $_GET['pass'];

// echo "$id_masalah<Br>";

$sql = "DELETE FROM gejala WHERE id_gejala = '$id_gejala' ";

if ($koneksi->query($sql)) {
    echo "gejala <b>$gejala</b> sukses dihapus";
} else {
    echo "Error: " . $koneksi->error;

    $sql = mysqli_query($koneksi, "SELECT DISTINCT(id_kasus) FROM
`kasus` WHERE id_gejala = $id_gejala");

    $id_kasus = array();

    while ($row = $sql->fetch_assoc()) {
        $id_kasus[] = $row['id_kasus'];
    }

    // print_r($id_kasus);

    for ($i=0; $i < count($id_kasus); $i++) {

        $sql10 = mysqli_query($koneksi, "DELETE FROM kasus WHERE
id_masalah = $id_masalah AND id_kasus = $id_kasus[$i] - $i");

        $sql11 = mysqli_query($koneksi, "UPDATE kasus SET
kasus.id_kasus = kasus.id_kasus - 1 where kasus.id_kasus > $id_kasus[$i] -
$i");

        if ($sql10 === TRUE && $sql11 === TRUE) {
            // echo "<br>masalah sukses ditambahkan ";
        } else {
            echo "Error: " . $sql10 . "<br>" . $koneksi->error;
        }

    }

    $sql = mysqli_query($koneksi, "SELECT DISTINCT(id_kasus) FROM
`kasustemp` WHERE id_gejala = $id_gejala");

```

```

$id_kasus = array();

while ($row = $sql->fetch_assoc()) {
    $id_kasus[] = $row['id_kasus'];
}

// print_r($id_kasus);

for ($i=0; $i < count($id_kasus); $i++) {

    $sql12 = mysqli_query($koneksi, "DELETE FROM kasustemp
WHERE id_masalah = $id_masalah AND id_kasus = $id_kasus[$i] - $i");

    $sql13 = mysqli_query($koneksi, "UPDATE kasustemp SET
kasustemp.id_kasus = kasustemp.id_kasus - 1 where kasustemp.id_kasus >
$id_kasus[$i] - $i");

    if ($sql12 === TRUE && $sql13 === TRUE) {
        // echo "<br>masalah sukses ditambahkan ";
    } else {
        echo "Error: " . $sql10 . "<br>" . $koneksi->error;
    }

}

$sqlx = "DELETE FROM gejala WHERE id_gejala = '$id_gejala' ";

if ($koneksi->query($sqlx)) {
    echo "gejala <b>$gejala</b> sukses dihapus";
}

}

echo '
<p><br><br><button type="button"
onclick="history.back();">Kembali</button></p>';

$koneksi->close();

header("Location:
pakarEditGejala.php?id_masalah=$id_masalah&masalah=$masalah&pass=$pass");
exit();

?>

```

Gambar 4.38 Source code hapus gejala

4.3.4 Contoh Proses Perhitungan

Proses perhitungan berikut akan menjelaskan alur penarikan kesimpulan yang berawal dari pemilihan gejala hingga nilai similaritas.

Misal diketahui kasus baru dengan gejala sulit tidur, sering menangis, merasa takut, dan tidak percaya diri. Gejala-gejala yang ada pada kasus baru tersebut akan dibandingkan dengan gejala-gejala yang ada pada kasus-kasus lama yang terdapat di dalam basis pengetahuan. Nilai *threshold* atau batas yang digunakan adalah 30%. Berikut adalah contoh perhitungan penarikan kesimpulan dengan menggunakan 4 jenis fungsi similaritas yang berbeda.

- a. Pencocokan basis kasus, fungsi similaritas ini menggunakan persamaan (1.1) untuk menentukan nilai similaritas. Silahkan lihat Tabel 4.12 berikut.

Tabel 4.12 Perhitungan similaritas dengan fungsi similaritas 1.1

Gejala Kasus Baru	Gejala Kasus Lama	Solusi Kasus Lama	Gejala yang Sama	Jumlah Gejala Kasus Lama	Nilai Similaritas	Melewati threshold (Y/T)
<ul style="list-style-type: none"> • Sulit tidur • Sering menangis • Merasa takut • Kawatir akan masa depan 	<ul style="list-style-type: none"> • Sulit tidur • Melakukan self-harm • Memiliki masalah asmara • Merasa hidup tidak berguna 	Belum mampu menerima keadaan riil diri sendiri	1	4	25%	T
	<ul style="list-style-type: none"> • Sulit tidur • Memiliki masalah asmara • Sering menangis • Sulit berkonsentrasi 	Depresif ringan	2	4	50%	Y
	<ul style="list-style-type: none"> • Kawatir akan masa depan • Tidak memahami materi kuliah 	Problem penyesuaian pada masa transisi siklus kehidupan	1	2	50%	Y

Berdasarkan perhitungan di atas, maka solusi atau kesimpulan dari kasus baru tersebut adalah Depresif ringan atau Problem penyesuaian pada masa transisi siklus kehidupan dengan nilai similaritas adalah 50%.

- b. Pencocokan basis kasus dan evaluasi, fungsi similaritas ini menggunakan persamaan (1.2) untuk menentukan nilai similaritas. Silahkan lihat Tabel 4.13 berikut.

Tabel 4.13 Perhitungan similaritas dengan fungsi similaritas 1.2

Gejala Kasus Baru	Gejala Kasus Lama	Solusi Kasus Lama	Gejala yang Sama	Max(Gejala Kasus Baru, Gejala Kasus Lama)	Nilai Similaritas	Melewati threshold (Y/T)
<ul style="list-style-type: none"> • Sulit tidur • Sering menangis • Merasa takut • Kawatir akan masa depan 	<ul style="list-style-type: none"> • Sulit tidur • Melakukan self-harm • Memiliki masalah asmara • Merasa hidup tidak berguna 	Belum mampu menerima keadaan riil diri sendiri	1	4	25%	T
	<ul style="list-style-type: none"> • Sulit tidur • Memiliki masalah asmara • Sering menangis • Sulit berkonsentrasi 	Depresif ringan	2	4	50%	Y
	<ul style="list-style-type: none"> • Kawatir akan masa depan • Tidak memahami materi kuliah 	Problem penyesuaian pada masa transisi siklus kehidupan	1	4	25%	T

Berdasarkan perhitungan di atas, maka solusi atau kesimpulan dari kasus baru tersebut adalah Depresif ringan dengan nilai similaritas adalah 50%.

- c. Pencocokan basis kasus dengan bobot, fungsi similaritas ini menggunakan persamaan (1.3) untuk menentukan nilai similaritas. Silahkan lihat Tabel 4.14 berikut.

Tabel 4.14 Perhitungan similaritas dengan fungsi similaritas 1.3

Gejala Kasus Baru	Gejala Kasus Lama	Solusi Kasus Lama	Total Bobot yang Sama	Total Bobot di Kasus Lama	Nilai Simil aritas	Melewati threshold (Y/T)
<ul style="list-style-type: none"> • Sulit tidur • Sering menangis • Merasa takut • Kawatir akan masa depan 	<ul style="list-style-type: none"> • Sulit tidur (<i>Bobot = 4</i>) • Melakukan self-harm (<i>Bobot = 3</i>) • Memiliki masalah asmara (<i>Bobot = 5</i>) • Merasa hidup tidak berguna (<i>Bobot = 2</i>) 	Belum mampu menerima keadaan riil diri sendiri	4	14	29%	T
	<ul style="list-style-type: none"> • Sulit tidur (<i>Bobot = 5</i>) • Memiliki masalah asmara (<i>Bobot = 3</i>) • Sering menangis (<i>Bobot = 8</i>) • Sulit berkonsent rasi (<i>Bobot = 4</i>) 	Depresif ringan	13	20	65%	Y
	<ul style="list-style-type: none"> • Kawatir akan masa depan (<i>Bobot = 5</i>) 	Problem penyesuaian pada masa transisi siklus kehidupan	5	9	56%	Y

	<ul style="list-style-type: none"> Tidak memahami materi kuliah (<i>Bobot = 4</i>) 					
--	---	--	--	--	--	--

Berdasarkan perhitungan di atas, maka solusi atau kesimpulan dari kasus baru tersebut adalah Depresif ringan atau Problem penyesuaian pada masa transisi siklus kehidupan dengan nilai similaritas masing-masing adalah 65% dan 56%.

- d. Pencocokan basis kasus dengan bobot dan konstanta, fungsi similaritas ini menggunakan persamaan (1.4) untuk menentukan nilai similaritas. Silahkan lihat Tabel 4.15 berikut.

Tabel 4.15 Perhitungan similaritas dengan fungsi similaritas 1.4

Gejala Kasus Baru	Gejala Kasus Lama	Solusi Kasus Lama	Total Bobot yang Sama	Total Bobot di Kasus Lama	Nilai Similaritas	Melewati threshold (Y/T)
<ul style="list-style-type: none"> Sulit tidur Sering menangis Merasa takut Kawatir akan masa depan 	<ul style="list-style-type: none"> Sulit tidur (<i>Bobot = 4</i>) Melakukan self-harm (<i>Bobot = 3</i>) Memiliki masalah asmara (<i>Bobot = 5</i>) Merasa hidup tidak berguna (<i>Bobot = 2</i>) 	Belum mampu menerima keadaan riil diri sendiri	4	14	29%	T
	<ul style="list-style-type: none"> Sulit tidur (<i>Bobot = 5</i>) Memiliki masalah asmara (<i>Bobot = 3</i>) 	Depresif ringan	13	20	65%	Y

	<ul style="list-style-type: none"> • Sering menangis (<i>Bobot</i> = 8) • Sulit berkonsentrasi (<i>Bobot</i> = 4) 					
	<ul style="list-style-type: none"> • Kawatir akan masa depan (<i>Bobot</i> = 5) • Tidak memahami materi kuliah (<i>Bobot</i> = 4) 	Problem penyesuaian pada masa transisi siklus kehidupan	5	9	26%	T

Berdasarkan perhitungan di atas, maka solusi atau kesimpulan dari kasus baru tersebut adalah Depresif ringan dengan nilai similaritas adalah 65%.

4.4 Uji Coba Sistem

4.4.1 Pengujian *black box testing*

Berikut adalah hasil pengujian dengan menggunakan metode *black box testing*:

- Komponen pilih gejala. Pengujian ini berhubungan dengan proses konsultasi yang dilakukan oleh pengguna non-pakar. Pengujian ini akan menentukan apakah sistem mampu menangani kemungkinan *error* yang bisa saja terjadi saat melakukan proses konsultasi. Silahkan lihat Tabel 4.16 berikut.

Tabel 4.16 Pengujian komponen pilih gejala

Masukan	Yang diharapkan	Keterangan
<i>Checkbox</i> dibiarkan kosong semua	Muncul peringatan “ <i>Checkbox</i> masih kosong!”	Sukses
<i>Checkbox</i> diisi setidaknya satu butir	Muncul halaman hasil diagnosa	Sukses

- Komponen *login*. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses *login*. Silahkan lihat Tabel 4.17 berikut.

Tabel 4.17 Pengujian komponen *login*

Masukan	Yang diharapkan	Keterangan
Salah satu <i>form</i> dibiarkan kosong	Muncul peringatan “Please fill out this field.”	Sukses
Salah satu <i>form</i> diisi dengan data yang salah	Muncul halaman gagal login	Sukses
<i>Forms</i> diisi dengan data yang benar	Muncul halaman <i>home</i> pakar	Sukses

- c. Komponen tambah domain. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses menambahkan data domain baru. Silahkan lihat Tabel 4.18 berikut.

Tabel 4.18 Pengujian komponen tambah domain

Masukan	Yang diharapkan	Keterangan
Salah satu <i>form</i> dibiarkan kosong	Muncul peringatan “Please fill out this field.”	Sukses
<i>Forms</i> diisi dengan data yang benar	Muncul halaman kelola gejala	Sukses

- d. Komponen pilih domain. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses memilih domain yang mana komponennya akan dikelola. Silahkan lihat Tabel 4.19 berikut.

Tabel 4.19 Pengujian komponen pilih domain

Masukan	Yang diharapkan	Keterangan
<i>Form</i> password dibiarkan kosong	Muncul halaman proses gagal	Sukses
<i>Form</i> password diisi dengan data yang salah	Muncul halaman proses gagal	Sukses
<i>Form</i> password diisi dengan data yang benar	Muncul halaman kelola gejala	Sukses

- e. Komponen hapus domain. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses penghapusan salah satu domain. Silahkan lihat Tabel 4.20 berikut.

Tabel 4.20 Pengujian komponen hapus domain

Masukan	Yang diharapkan	Keterangan
<i>Form</i> password dibiarkan kosong	Muncul halaman proses gagal	Sukses
<i>Form</i> password diisi dengan data yang salah	Muncul halaman proses gagal	Sukses

<i>Form</i> password diisi dengan data yang benar	Domain dihapus	Sukses
---	----------------	--------

- f. Komponen tambah gejala. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses menambah gejala baru. Silahkan lihat Tabel 4.21 berikut.

Tabel 4.21 Pengujian komponen tambah gejala

Masukan	Yang diharapkan	Keterangan
<i>Form</i> dibiarkan kosong	Muncul peringatan “Please fill out this field.”	Sukses
<i>Form</i> diisi dengan data yang benar	Muncul halaman kelola gejala dengan gejala yang baru saja ditambahkan	Sukses

- g. Komponen ubah gejala. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses mengubah gejala. Silahkan lihat Tabel 4.22 berikut.

Tabel 4.22 Pengujian komponen ubah gejala

Masukan	Yang diharapkan	Keterangan
<i>Form</i> dibiarkan kosong	Muncul peringatan “Please fill out this field.”	Sukses
<i>Form</i> diisi dengan data yang benar	Muncul halaman kelola gejala dengan gejala yang baru saja diubah	Sukses
<i>Form</i> diisi dengan data yang sama dengan data sebelumnya	Muncul peringatan “Data tidak boleh sama!”	Sukses

- h. Komponen tambah solusi. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses menambah solusi baru. Silahkan lihat Tabel 4.23 berikut.

Tabel 4.23 Pengujian komponen tambah solusi

Masukan	Yang diharapkan	Keterangan
<i>Forms</i> dibiarkan kosong	Muncul peringatan “Please fill out this field.”	Sukses
<i>Forms</i> diisi dengan data yang benar	Muncul halaman kelola solusi dengan solusi yang baru saja ditambahkan	Sukses

- i. Komponen ubah solusi. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses mengubah solusi. Silahkan lihat Tabel 4.24 berikut.

Tabel 4.24 Pengujian komponen ubah solusi

Masukan	Yang diharapkan	Keterangan
<i>Forms</i> dibiarkan kosong	Muncul peringatan “Please fill out this field.”	Sukses
<i>Forms</i> diisi dengan data yang benar	Muncul halaman kelola solusi dengan solusi yang baru saja diubah	Sukses
<i>Forms</i> diisi dengan data yang sama dengan data sebelumnya	Muncul peringatan “Data tidak boleh sama!”	Sukses

- j. Komponen tambah kasus. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses menambah kasus baru. Silahkan lihat Tabel 4.25 berikut.

Tabel 4.25 Pengujian komponen tambah kasus

Masukan	Yang diharapkan	Keterangan
<i>Forms</i> dibiarkan kosong	Muncul peringatan “Silahkan isi form yang masih kosong.”	Sukses
<i>Forms</i> diisi dengan data yang benar	Muncul halaman kelola kasus dengan kasus yang baru saja ditambahkan	Sukses

- k. Komponen ubah kasus. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses mengubah kasus. Silahkan lihat Tabel 4.26 berikut.

Tabel 4.26 Pengujian komponen ubah kasus

Masukan	Yang diharapkan	Keterangan
<i>Forms</i> dibiarkan kosong	Muncul peringatan “Silahkan isi form yang masih kosong.”	Sukses
<i>Forms</i> diisi dengan data yang benar	Muncul halaman kelola kasus dengan kasus yang baru saja diubah	Sukses
<i>Forms</i> diisi dengan data yang sama dengan data sebelumnya	Muncul peringatan “Data tidak boleh sama!”	Sukses

- l. Komponen ubah fungsi similaritas. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses mengubah atau mengganti fungsi similaritas. Silahkan lihat Tabel 4.27 berikut.

Tabel 4.27 Pengujian komponen ubah fungsi similaritas

Masukan	Yang diharapkan	Keterangan
Form diisi dengan data yang sama dengan data sebelumnya	Muncul peringatan “Data tidak boleh sama!”	Sukses

m. Komponen ubah *password* domain. Pengujian komponen ini akan menentukan apakah sistem mampu menangani *error* yang mungkin terjadi saat melakukan proses mengubah kata sandi domain. Silahkan lihat Tabel 4.28 berikut.

Tabel 4.28 Pengujian komponen ubah *password* domain

Masukan	Yang diharapkan	Keterangan
Forms dibiarkan kosong	Muncul peringatan “Please fill out this field.”	Sukses
Form password baru dengan ulangi password baru diisi dengan data yang berbeda	Muncul peringatan “Kombinasi password salah!”	Sukses

4.4.2 Pengujian validitas sistem

Validitas sistem diuji dengan membandingkan hasil diagnosis yang dilakukan oleh pakar dan hasil diagnosis yang dilakukan oleh sistem. Jika hasil diagnosisnya sama maka dianggap *valid* dan jika berbeda maka tidak *valid*.

Rangkuman hasil uji validitas dapat dilihat pada Tabel 4.29 berikut yang menggambarkan perbandingan hasil diagnosis antara kenyataan dan sistem.

Tabel 4.29 Hasil pengujian validitas sistem

No	Kenyataan	Sistem	Keterangan
1	Belum mampu menerima keadaan riil diri sendiri.	Belum mampu menerima keadaan riil diri sendiri.	VALID
2	Kebiasaan berpikir yang irrasional	Depresif ringan	TIDAK VALID
3	Problem Penyesuaian Pada Masa Transisi Siklus Kehidupan	Problem Penyesuaian Pada Masa Transisi Siklus Kehidupan	VALID
4	Gangguan Panik	Gangguan Panik	VALID
5	Ketidakmampuan mengontrol emosi	Problem Penyesuaian Pada Masa Transisi Siklus Kehidupan	TIDAK VALID
6	Problem yang berkaitan dengan lingkungan sosial	Problem yang berkaitan dengan lingkungan sosial	VALID
7	Belum mampu menerima keadaan riil diri sendiri.	Belum mampu menerima keadaan riil diri sendiri.	VALID

8	Kecemasan	Kecemasan	VALID
9	Kebiasaan berpikir yang irrasional	Kesepian, belum menemukan makna hidup.	TIDAK VALID
10	Kebiasaan berpikir yang irrasional	Kebiasaan berpikir yang irrasional	VALID
11	Gangguan stres pasca-trauma	Kebiasaan berpikir yang irrasional	TIDAK VALID
12	Masalah yang berhubungan dengan hubungan orang tua & anak	Belum mampu menerima keadaan riil diri sendiri.	TIDAK VALID
13	Ketidakmampuan mengontrol emosi	Ketidakmampuan mengontrol emosi	VALID
14	Ketidakmampuan mengontrol emosi	Ketidakmampuan mengontrol emosi	VALID
15	Ketidakmampuan mengontrol emosi	Ketidakmampuan mengontrol emosi	VALID
16	Motivasi yang rendah	Motivasi yang rendah	VALID
17	Gangguan stres pasca-trauma	Konflik dengan diri sendiri	TIDAK VALID
18	Gangguan stres pasca-trauma	Kecemasan	TIDAK VALID
19	Kecemasan	Kecemasan	VALID
20	Kebingungan memilih karir	Kebingungan memilih karir	VALID

Berdasarkan Tabel 4.29 diatas, maka nilai validitas sistem dapat dihitung dengan menggunakan rumus (2.1) berikut:

$$\text{validitas} = \frac{13}{20} \times 100\%$$

$$\text{validitas} = 65\%$$

Sistem mendapatkan nilai validitas sebesar 65%. Dengan menggunakan pedoman Tabel 2.1 maka dapat disimpulkan bahwa sistem ini masuk ke dalam kategori baik karena memiliki nilai usabilitas di antara 60% dan 79.9%.

4.4.3 Pengujian usabilitas

Pengujian usabilitas dilakukan dengan melibatkan 5 orang responden. 5 orang responden tersebut terdiri dari dua orang pakar, dan tiga mahasiswa.

Sebelum menggunakan rumus (2.2), nilai X atau total skor harus didapatkan terlebih dahulu dengan menjumlahkan semua hasil perkalian antara total jumlah responden yang memilih komponen tertentu dengan pilihan angka skor Likert. Untuk lebih jelasnya, berikut ini adalah langkah perhitungan menentukan nilai X terhadap komponen pertama kuesioner:

- a. Responden yang menjawab Sangat Baik/SB (skor 5) berjumlah 2 orang = $5 \times 2 = 10$

- b. Responden yang menjawab Baik/B (skor 4) berjumlah 3 orang = $4 \times 3 = 12$
- c. Responden yang menjawab Cukup/C (skor 3) berjumlah 0 orang = $3 \times 0 = 0$
- d. Responden yang menjawab Kurang/K (skor 2) berjumlah 0 orang = $2 \times 0 = 0$
- e. Responden yang menjawab Sangat Kurang/SB (skor 1) berjumlah 0 orang = $1 \times 0 = 0$
- f. X (total skor, hanya untuk komponen pertama) = 22

Pengujian usabilitas ini memiliki lima skor Likert dengan nilai tertinggi adalah Sangat Baik (SB) yang memiliki bobot 5 poin dan melibatkan 5 orang responden sehingga dengan menggunakan rumus (2.3), maka nilai i adalah 25.

Pada pengujian ini, banyaknya skor Likert adalah lima sehingga dengan menggunakan rumus (2.4), maka nilai j adalah:

$$j = \frac{100}{5}$$

$$j = 20\%$$

Berdasarkan perhitungan di atas maka interval yang ditentukan berjumlah 5 buah dan setiap interval memiliki rentang nilai 20%. Silahkan lihat Tabel 4.30 berikut.

Tabel 4.30 Kategori kelayakan sistem

Interval (%)	Keterangan
0% - 19.9%	Sangat kurang
20% - 39.9%	Kurang
40% - 59.9%	Cukup
60% - 79.9%	Baik
80% - 100%	Sangat baik

Hasil lengkap perhitungan nilai P atau persentase dengan aspek *flexibility* dapat dilihat pada Tabel 4.31 berikut.

Tabel 4.31 Hasil perhitungan persentase aspek *learnability*

No	Komponen Penilaian	Nilai	Keterangan
1	Kemudahan dalam memahami dan menggunakan sistem.	88.0%	Sangat baik
2	Sistem ini sangat membantu pengguna.	84.0%	Sangat baik
3	Bahasa yang digunakan pada sistem mudah untuk dimengerti.	88.0%	Sangat baik
4	Tata letak tombol dan icon mudah untuk mudah untuk dijangkau dan dipahami maknanya.	84.0%	Sangat baik

5	Responden dapat memprediksi apa yang akan terjadi setelah sebuah tombol ditekan.	80.0%	Sangat baik
---	--	-------	-------------

Hasil lengkap perhitungan nilai P atau persentase dengan aspek *flexibility* dapat dilihat pada Tabel 4.32 berikut.

Tabel 4.32 Hasil perhitungan persentase aspek *flexibility*

No	Komponen Penilaian	Nilai	Keterangan
6	Fitur yang ada pada sistem sudah sesuai dengan harapan.	92.0%	Sangat baik
7	Sistem sangat informatif.	88.0%	Sangat baik

Hasil lengkap perhitungan nilai P atau persentase dengan aspek *effectiveness* dapat dilihat pada Tabel 4.33 berikut.

Tabel 4.33 Hasil perhitungan persentase aspek *effectiveness*

No	Komponen Penilaian	Nilai	Keterangan
8	Pengguna tidak mengalami kesulitan dalam melakukan sebuah aktifitas pada sistem.	84.0%	Sangat baik
9	Respon yang diberikan sistem terhadap aksi yang dilakukan oleh pengguna baik.	88.0%	Sangat baik

Hasil lengkap perhitungan nilai P atau persentase dengan aspek *attitude* dapat dilihat pada Tabel 4.34 berikut.

Tabel 4.34 Hasil perhitungan persentase aspek *attitude*

No	Komponen Penilaian	Nilai	Keterangan
10	Tampilan sistem menarik dan tidak mengganggu pandangan mata.	84.0%	Sangat baik
11	Sistem ini menarik dan inovatif.	88.0%	Sangat baik
12	Pengguna merasa puas dengan sistem ini.	84.0%	Sangat baik

Setelah melakukan perhitungan terhadap setiap komponen, langkah selanjutnya adalah menentukan nilai rata-rata dari setiap aspek pengujian. Hasil rata-rata setiap aspek adalah sebagai berikut:

- a. Hasil rata-rata aspek *learnability* adalah 84.8%
- b. Hasil rata-rata aspek *flexibility* adalah 90.0%
- c. Hasil rata-rata aspek *effectiveness* adalah 86.0%
- d. Hasil rata-rata aspek *attitude* adalah 85.3%

Dengan menggunakan rumus (2.5) diatas, maka nilai akhir *usability* adalah:

$$U = \frac{84.8 + 90.0 + 86.0 + 85.3}{4} \%$$
$$U = 86.5\%$$

Setelah proses pengujian selesai dilakukan, sistem mendapatkan nilai usabilitas senilai 86.5%. Dengan menggunakan pedoman Tabel 4.30 maka dapat disimpulkan bahwa sistem ini masuk ke dalam kategori sangat baik karena memiliki nilai usabilitas di atas 80%.



BAB V

KESIMPULAN

5.1 Kesimpulan

Penelitian ini menghasilkan keluaran berupa sebuah sistem editor basis pengetahuan untuk sistem berbasis kasus-kasus yang mampu membawa pengetahuan yang dimiliki oleh seorang pakar menjadi sebuah pengetahuan yang bisa disimpan di dalam basis pengetahuan. Editor ini dikembangkan dengan menggunakan metode pengembangan sistem *prototyping*. Studi kasus yang digunakan adalah bimbingan atau konseling mahasiswa UII. Berdasarkan pengujian yang telah dilakukan, sistem mampu bekerja sesuai dengan harapan. Selain itu, sistem juga memiliki nilai validitas sistem sebesar 65% (baik) serta nilai usability sistem sebesar 86.5% (sangat baik).

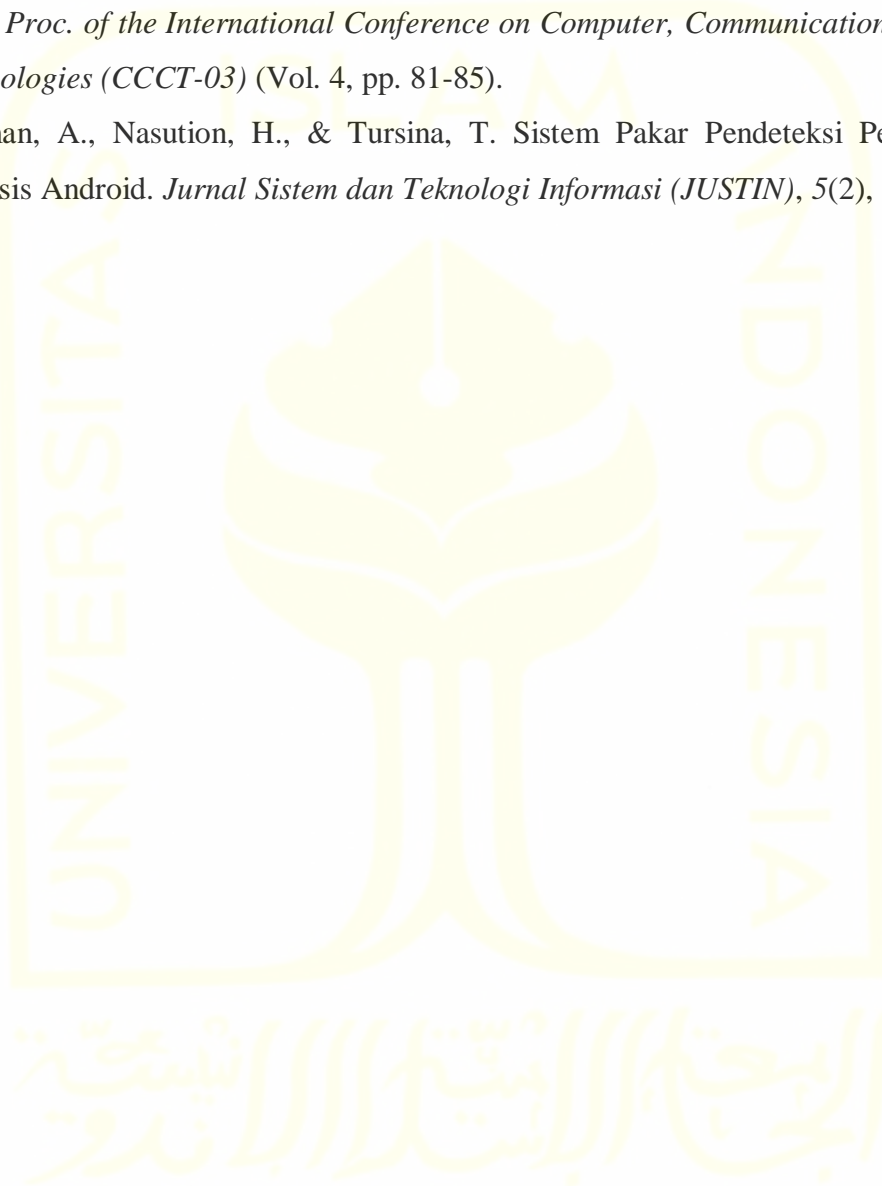
5.2 Saran

Saran untuk sistem ini kedepannya diharapkan mampu memiliki fitur untuk membuat fungsi similaritas sendiri, sehingga fungsi similaritas yang dimiliki sistem lebih banyak. Selain itu, perlu ditambahkan basis pengetahuan dalam jumlah yang lebih banyak dan lebih variatif sehingga sistem ini bisa bekerja dengan lebih akurat saat proses pengambilan keputusan. Sistem kedepannya juga diharapkan mampu dikembangkan dengan tampilan yang lebih familiar dan mudah dimengerti bagi pengguna baik non-pakar maupun pakar.

DAFTAR PUSTAKA

- Berka, P. (2011). NEST: a compositional approach to rule-based and case-based reasoning. *Advances in Artificial Intelligence*, 2011.
- Elam, J. J., & Henderson, J. C. (1983). Knowledge engineering concepts for decision support system design and implementation. *Information & Management*, 6(2), 109-114.
- Honggowibowo, A. S. (2009). Sistem pakar diagnosis penyakit tanaman padi berbasis web dengan forward dan backward chaining. *Telkomnika*, 7(3), 187.
- Kayali, I. (2018). Expert System for Diagnosis of Chest Diseases Using Neural Networks. *arXiv preprint arXiv:1802.06866*.
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2016). Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SMNPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3).
- Naykhanova, L. and Naykhanova, I. (2018). Conceptual model of knowledge base system. *Journal of Physics: Conference Series*, 1015, p.032097.
- Nurfalinda, N., & Nikentari, N. (2017). Case Based Reasoning untuk Diagnosis Penyakit Gizi Buruk pada Balita. *Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan*, 6(2), 53-60.
- Rachmawati, I. N. (2007). Pengumpulan data dalam penelitian kualitatif: wawancara. *Jurnal Keperawatan Indonesia*, 11(1), 35-40.
- Pradipta, A. A., Prasetyo, Y. A., & Ambarsari, N. (2015). Pengembangan Web E-Commerce Bojana Sari Menggunakan Metode Prototype. *eProceedings of Engineering*, 2(1).
- Watson, I., & Marir, F. (1994). Case-based reasoning: A review. *The knowledge engineering review*, 9(4), 327-354.
- Watson, I. (1999). Case-based reasoning is a methodology not a technology. In *Research and Development in Expert Systems XV* (pp. 213-223). Springer, London.
- Paley, S. M., Lowrance, J. D., & Karp, P. D. (1997, July). A generic knowledge-base browser and editor. In *AAAI/IAAI* (pp. 1045-1051).
- Wong, D. F., & Kwok, S. L. (1997). Difficulties and patterns of social support of mature college students in Hong Kong: Implications for student guidance and counselling services. *British Journal of Guidance and Counselling*, 25(3), 377-387.

- Riyadi, N. R. (2019). PENGUJIAN USABILITY UNTUK MENINGKATKAN ANTARMUKA APLIKASI MOBILE myUMM STUDENTS. *SISTEMASI: Jurnal Sistem Informasi*, 8(1), 226-232.
- Putra, F. A., Satoto, K. I., & Martono, K. T. (2013). Analisis Usabilitas pada Permainan The Zoo Berbasis Kinect. *Jurnal Teknologi dan Sistem Komputer*, 1(4), 104-112.
- Prentzas, J., & Hatzilygeroudis, I. (2003). Integrations of rule-based and case-based reasoning. In *the Proc. of the International Conference on Computer, Communication and Control Technologies (CCCT-03)* (Vol. 4, pp. 81-85).
- Aditiawarman, A., Nasution, H., & Tursina, T. Sistem Pakar Pendeteksi Penyakit Mata Berbasis Android. *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, 5(2), 118-122.



LAMPIRAN

PERTANYAAN WAWANCARA DENGAN PAKAR

Wawancara dilakukan kepada narasumber Ibu Hazhira Qudsyi, S.Psi., M.A. sebagai tenaga konseling mahasiswa UII. Adapun pertanyaan dari proses wawancara ini adalah sebagai berikut:

1. Apa saja yang dibutuhkan oleh sebuah editor basis pengetahuan agar bisa bekerja dengan optimal sesuai dengan apa yang pakar inginkan? Seperti apa karakteristik sebuah editor basis pengetahuan agar bisa dianggap baik?
2. Apa saja fitur-fitur penting yang harus ada pada sebuah editor basis pengetahuan?
3. Apa saja yang bisa pakar lakukan saat menggunakan editor basis pengetahuan?
4. Komponen-komponen apa saja yang harus ada di dalam editor basis pengetahuan?
5. Bagaimana alur atau proses penambahan basis pengetahuan yang dilakukan oleh seorang pakar dari awal hingga akhir?
6. Apakah penggunaan bobot pada gejala penting bagi pakar saat menggunakan editor basis pengetahuan untuk sistem berbasis kasus? Apakah boleh dan sah-sah saja sebuah editor basis pengetahuan tidak menggunakan bobot di dalam sistemnya?
7. Bagaimana jika pengguna biasa atau non pakar melakukan konsultasi sendiri (tanpa pakar), namun saat memilih gejala ternyata gejala yang dialami tidak ada pada masalah tersebut? Apa yang harus ia lakukan?
8. Jika ada sebuah gejala atau solusi dari basis pengetahuan yang dirasa tidak lagi relevan, tindakan apa yang harus dilakukan? Manakah yang lebih dianjurkan, mengubah atau menghapus gejala atau solusi tersebut?
9. Elemen-elemen atau komponen apa saja yang bisa dimodifikasi di dalam sebuah basis pengetahuan? Bagaimana alurnya?
10. Bagaimana alur atau proses penghapusan suatu elemen di dalam basis pengetahuan?
11. Editor basis pengetahuan ini dirancang mampu menampung lebih dari 1 masalah/domain. Apakah diperlukan pengamanan bagi setiap domain tersebut? Jika iya, cara apa yang paling tepat untuk mengamankannya?
12. Biasanya saat mendiagnosis kasus seperti apa yang sering dianalisis? Gejala apa yang sering muncul dan apa solusinya?

Dari kegiatan wawancara di atas, penulis dapat menarik kesimpulan sebagai berikut.

1. Pakar menginginkan sistem yang mudah dipahami, dengan fitur yang simpel, sederhana dan tidak terlalu banyak. Sistem yang baik menurut pakar adalah sistem yang bisa digunakan dan dimengerti tanpa harus membutuhkan waktu yang lama.
2. Fitur-fitur penting yang harus ada di sistem tersebut meliputi kelola pengetahuan (tambah, ubah, dan hapus) yang dilakukan oleh pakar serta fitur bagi non-pakar untuk melakukan konsultasi.
3. Komponen yang harus ada di dalam basis pengetahuan meliputi gejala, solusi (hasil diagnosis dan solusi), dan kasus-kasus yang tersusun dari gejala dan solusi.
4. Menurut pakar, penggunaan bobot tidak terlalu penting bagi bidangnya. Namun, keberadaan bobot di dalam basis pengetahuan dianggap tidak menjadi masalah dan boleh-boleh saja asalkan diberi pilihan untuk memilih menggunakan bobot atau tidak.
5. Jika ada sebuah gejala atau solusi dari basis pengetahuan yang dirasa tidak lagi relevan, maka pakar akan tetap menyimpan gejala atau solusi tersebut dan memilih untuk tidak menghapusnya karena bisa jadi suatu saat nanti gejala atau solusi tersebut digunakan kembali.
6. Dikarenakan sistem ini bersifat *multi* domain, pakar menginginkan pengamanan bagi setiap basis pengetahuan agar tidak dikelola oleh orang yang bukan bidangnya.