

PENGEMBANGAN SISTEM PAKAR BERBASIS ATURAN UNTUK MENENTUKAN MATA KULIAH YANG AKAN DIAMBIL ULANG (REMEDIAL) DENGAN METODE FORWARD CHAINING

Hariyadi¹, Sawalni²

Program Studi Teknik Elektro UMSB¹, Prodi Usaha Perjalanan Wisata UMSB²

ABSTRAK

Nilai IP (Indeks Prestasi) merupakan target pencapaian mahasiswa dalam mendapatkan hasil akademik yang nantinya menentukan kepuasan mahasiswa terhadap apa yang telah dikuasai selama perkuliahan. Untuk mencapai nilai IP yang baik dan sangat memuaskan, mahasiswa harus memaksimalkan prestasi akademiknya untuk mendapatkan nilai ujian dengan hasil yang baik terhadap matakuliah yang diambil. Jika terjadi kegagalan nilai atau perbaikan nilai matakuliah maka dibutuhkan prosedur akademik yaitu remedial. Dengan Implementasi sistem pakar dalam menentukan matakuliah remedial akan membantu pihak akademik, dosen maupun pembimbing dalam memberikan kebijakan remedial yang diambil bisa dilanjutkan atau tidak. Rancangan sistem ini menggunakan metode *Forward Chaining* dengan implementasi sistem menggunakan sistem database *Microsoft Access* dan *Software Development Visual Basic 6.0*. Dari rancangan sistem yang dibuat maka dapat diperoleh sebuah kebijakan untuk mahasiswa maupun pertimbangan dari akademik, dosen dan pembimbing atas proses remedial yang dilakukan mahasiswa.

Kata Kunci : *Sistem Pakar, Forward Chaining, remedial, matakuliah*

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Nilai IP yang didapat tentunya merupakan akumulasi dari matakuliah-matakuliah yang diambil selama semester perkuliahan. Untuk mencapai nilai IP yang baik dan sangat memuaskan, mahasiswa harus memaksimalkan prestasi akademiknya untuk mendapatkan nilai ujian dengan hasil yang baik terhadap matakuliah yang diambil, agar menyelesaikan perkuliahan tepat waktu (3,5 – 4 Tahun untuk Program S1).

Istilah remedial biasanya yang dikenal dengan Semester Pendek pada Perguruan Tinggi lainnya syarat utama dalam pengambilan Remedial ini adalah matakuliah yang diambil harus mempunyai nilai, dan hanya nilai C, D, dan E.

Adakalanya Nilai yang didapat terkadang belum memuaskan (C, D) bahkan mendapatkan nilai gagal (E). Untuk mengatasi nilai matakuliah tersebut, mahasiswa biasanya melakukan Remedial yang biasanya ditawarkan pada akhir semester setelah UAS selesai pada tiap semester.

Mahasiswa yang akan melakukan Remedial tentunya harus mengetahui apakah matakuliahnya ini wajib diulang atau tidak, dan tentunya juga bagian akademik serta dosen pembimbing pun juga harus bisa memberikan arahan terhadap pilihan yang dilakukan mahasiswa tersebut. Untuk bisa mengatasi ini semua dibutuhkan pengambilan keputusan yang tepat dan saling menguntungkan antara mahasiswa dan akademik, mahasiswa bisa memperbaiki nilainya dan akademik pun bisa memberikan yang terbaik terhadap mahasiswa.

1.2 Perumusan Masalah

Dalam penelitian ini dapat dirumuskan beberapa masalah sebagai berikut :

1. Bagaimana merancang Sistem Pakar untuk menentukan matakuliah yang akan di Remedialkan diambil, harus dengan kuliah khusus (Remedial) atau mengulang pada semester bawah ?
2. Bagaimana menggunakan metode *Forward chaining* dalam menentukan matakuliah yang akan di Remedialkan ?

1.3 Batasan Masalah

Untuk menghindari terjadinya pembahasan di luar ini maka perlu dibatasi permasalahan yang diangkat, yaitu :

1. Sistem Pakar ini berbasis aplikasi berbasis *Visual Basic*
2. Sumber pengetahuan diperoleh dari pakar (Akademik, dosen, pembimbing) dan buku-buku yang mendukung.
3. Studi kasus dilakukan pada mahasiswa Universitas Muhammadiyah Sumatera Barat.
4. Metode yang digunakan dalam penyelesaian masalah ini adalah *Forward Chaining*.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah:

1. Menganalisa data-data tentang Remedial
2. Memahami cara kerja Sistem Pakar dalam menentukan matakuliah yang akan di Remedialkan berdasarkan metode *Forward chaining*.
3. Merancang suatu Sistem Pakar untuk menentukan matakuliah yang akan di Remedialkan serta bisa menentukan matakuliah tersebut dilaksanakan apakah harus mengulang pada semester bawah.
4. Membangun Sistem Pakar yang mampu meningkatkan kemampuannya tanpa harus diprogram secara manual.
5. Menguji Sistem Pakar kedalam sebuah *software* sistem pakar.

2. LANDASAN TEORI

2.1 Kecerdasan Buatan

Kecerdasan buatan dapat didefinisikan sebagai mekanisme pengetahuan yang ditekankan pada kecerdasan pembentukan dan penilaian pada alat yang menjadikan mekanisme itu, serta membuat komputer berpikir secara cerdas. Kecerdasan buatan juga dapat didefinisikan sebagai salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia.

Teknologi kecerdasan buatan dipelajari dalam bidang-bidang, seperti: robotika, penglihatan komputer (*Computer Vision*), jaringan saraf tiruan (*Artificial Neural System*), pengolahan bahasa alami (*Natural Language Processing*), pengenalan suara (*Speech Recognition*), dan Sistem Pakar (*Expert System*).

2.2 Pengertian Pakar

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkan untuk memecahkan masalah atau memberi nasehat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya (Sutojo, 2011).

2.3 Pengertian Sistem Pakar

Sistem Pakar adalah cabang dari *Artificial Intelligence*, dan dikembangkan oleh komunitas *Artificial Intelligence* di pertengahan tahun 1960-an. Sebuah Sistem Pakar dapat didefinisikan sebagai sebuah program komputer cerdas yang menggunakan pengetahuan dan inferensi prosedur untuk memecahkan masalah yang cukup sulit yang memerlukan keahlian manusia untuk memecahkan solusi (Sharma, 2013)

Sistem Pakar adalah sistem yang memperkerjakan pengetahuan manusia yang ditangkap oleh komputer untuk memecahkan berbagai masalah yang biasanya membutuhkan keahlian manusia, Sistem Pakar berusaha memberikan informasi kepada manusia untuk memberikan rekomendasi (Patra, 2010). Menurut Guspa (2013) Sistem Pakar adalah sebuah program komputer yang mensimulasikan proses pemikiran seorang ahli untuk memecahkan masalah berupa keputusan yang kompleks dalam domain tertentu. Sistem Pakar adalah sebuah program yang berperilaku seperti seorang ahli pada bidang tertentu (Deepa, 2012)

2.4 Motor Inferensi

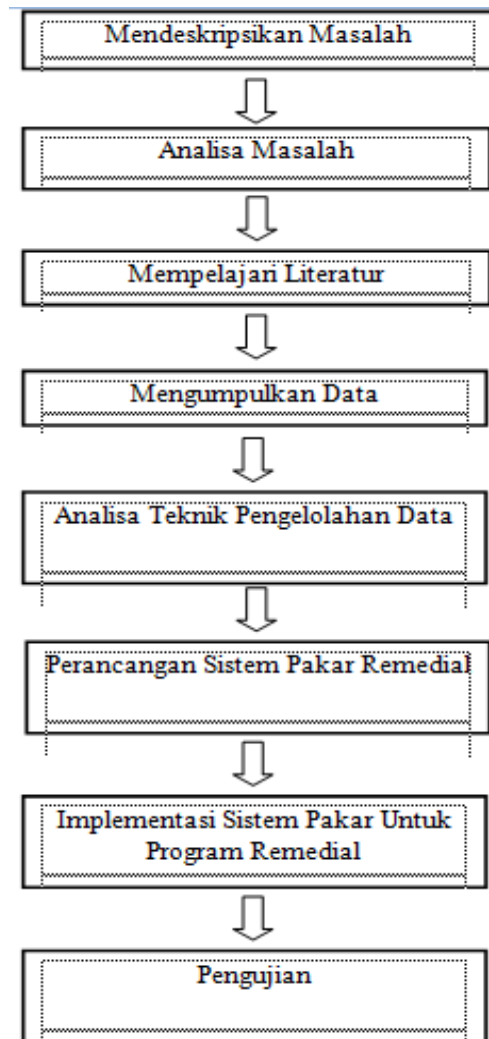
Ada 2 cara yang dapat dikerjakan dalam melakukan inferensi yaitu (Sri Kusumadewi, 2003):

1. *Forward Chaining* adalah pencocokan fakta atau pernyataan dimulai dari bagian kiri (IF dulu), dengan kata lain penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis.
2. *Bacward Chaining* adalah pencocokan fakta atau pernyataan dimulai dari bagian sebelah kanan (THEN dulu), dengan kata lain penalaran dimulai dari hipotesis terlebih dahulu dan mulai untuk menguji kebenaran hipotesis tersebut dicari fakta yang ada dalam basis pengetahuan.

Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam workplace, dan untuk memformulasikan kesimpulan (Turban, 1995).

3 METODE PENELITIAN

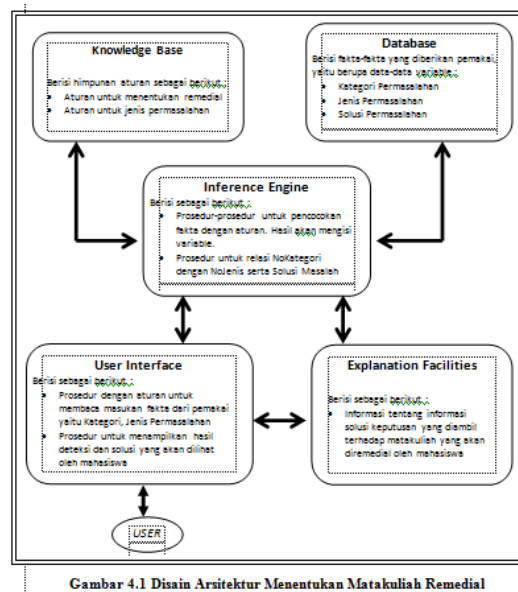
Salah satu unsur terpenting dalam metodologi penelitian adalah penggunaan metode ilmiah tertentu yang digunakan sebagai sarana yang bertujuan untuk mengidentifikasi besar kecilnya objek atau gejala dan mencari pemecahan masalah yang sedang diteliti, sehingga hasil yang diperoleh dapat dipertanggungjawabkan kebenarannya secara ilmiah. Metodologi yang akan digunakan pada tesis ini adalah metodologi yang berbasis objek (*object oriented programming*). Metodologi ini disusun dalam suatu kerangka kerja penelitian (*frame work*) seperti gambar 3.1.



4 HASIL DAN PEMBAHASAN

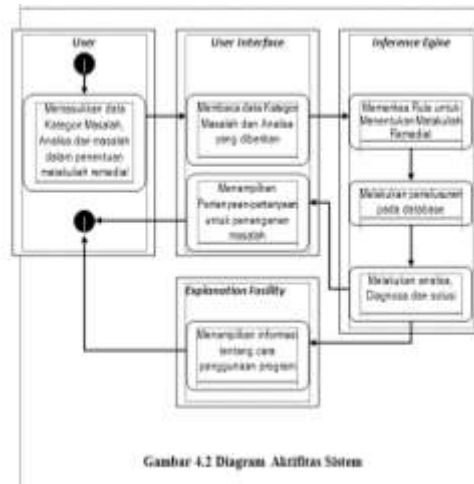
4.1 Disain Arsitektur Sistem

Dengan mengacu pada Sistem Pakar berbasis aturan yang telah dibahas pada bab sebelumnya dan dengan melakukan beberapa penyederhanaan pada beberapa komponen, maka arsitektur sistem didisain seperti terlihat pada Gambar 4.1.



4.2 Disain Aktifitas Sistem

Activity diagram menggambarkan berbagai alur aktivitas secara umum dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, proses yang dilakukan dan bagaimana proses berakhir. Activity diagram tidak menggambarkan behavior internal sebuah sistem.



4.3 Representasi Pengetahuan

Untuk mendukung penalaran dalam menentukan matakuliah, maka pengetahuan yang diperoleh dari pakar dapat direpresentasikan dalam bentuk pohon keputusan dengan penelusuran data *Depth First Search*, sebagaimana terlihat pada Gambar 4.3.

Dalam penggunaan penelusuran data dengan metode *Depth First Search* ini, dikarenakan pola sistem yang akan dibuat merupakan proses penelusuran berdasarkan pertanyaan-pertanyaan, untuk satu node akan memiliki cabang 2 bagian, dari 2 bagian cabang merupakan pilihan dari pola penelusuran data. Pertanyaan-pertanyaan yang diberikan berdasarkan fakta-fakta yang berkaitan atau berhubungan dengan konsultasi dalam mengajukan remedial matakuliah.

Dalam penggunaan metode alur *Depth First Search* ini, penulis mempersingkat bagian-bagian node kedalam simbol-simbol yang dipakai dalam merepresentasikan metode ini. Adapun simbol-simbol yang dipakai di pohon keputusan ini didefinisikan sebagai berikut:

Tabel 4.1 Penjelasan Pohon Keputusan Bagian 1

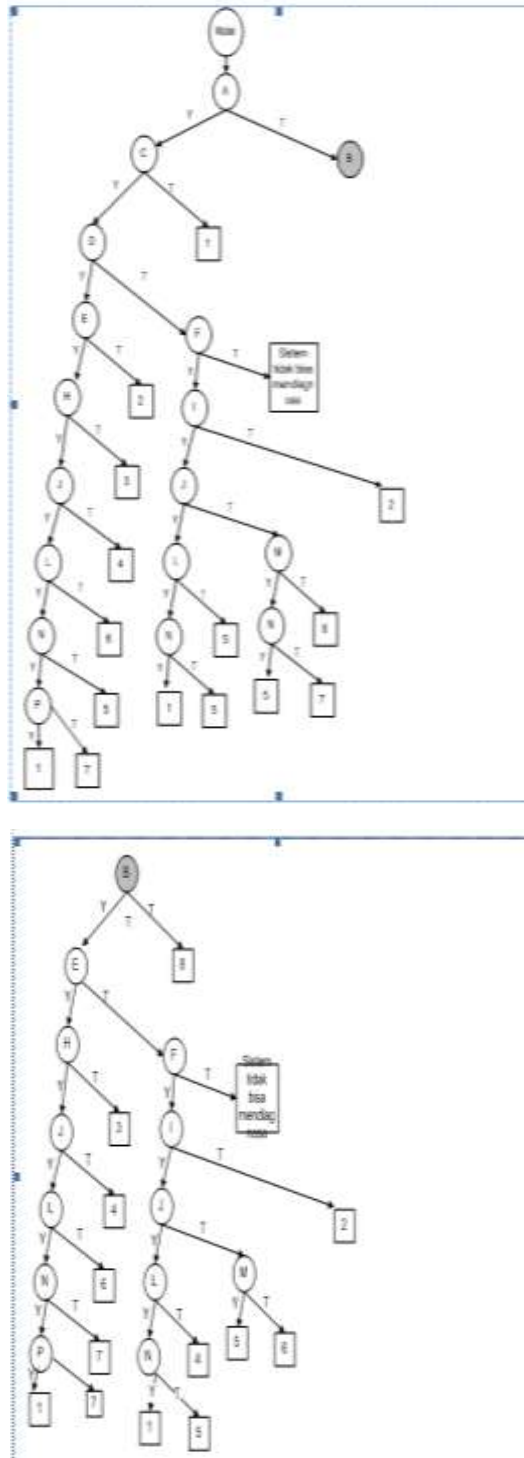
Fakta	Kode
Matakuliah di Remedial	A
Remedial Semester ini	B
Ekonomi Keluarga Sedang	C
Ekonomi Keluarga Mampu	D
Remedial Nilai Yang Gagal	E
Remedial Nilai Yang Tidak Gagal	F
Nilai Gagal	G
Nilai D/E	H
Nilai C	I
Kategori Matakuliah Keilmuan & Keterampilan (MKK)/ Keahlian Berkarya (MKB)	J
Kategori Matakuliah MKK/MKB	K
Total SKS Diambil Untuk Remedial < 10 SKS	L
Total SKS Diambil Untuk Remedial > 10 SKS	M
IPK < 3.00	N
Tidak Sesuai dengan Ketentuan	O
Sesuai dengan Ketentuan	P

Pada tabel 4.1 di atas merupakan fakta-fakta pengujian yang akan dimasukkan ke dalam pohon keputusan dengan memberikan simbol pengganti untuk kode kedalam pohon keputusan.

Tabel 4.2 Penjelasan Pohon Keputusan Bagian 2

Solusi	Kode
Disarankan Remedial	1
Perbaiki Nilai Yang Tidak Gagal	2
Nilai Tidak Gagal Hanya Perbaiki	3
Remedial Disarankan Kategori Matakuliah Keilmuan & Keterampilan (MKK) Saja	4
Dengan Remedial IPK Anda, ada kemungkinan Naik	5
Tidak Bisa Ikut Remedial, Total SKS Anda Melebihi Batas Maksimum SKS	6
Tidak Bisa Mengikuti Remedial, Karena tidak sesuai dengan ketentuan	7
Tidak Bisa mengikuti Remedial, Silahkan Ambil Semester Selanjutnya	8

Pohon Keputusan

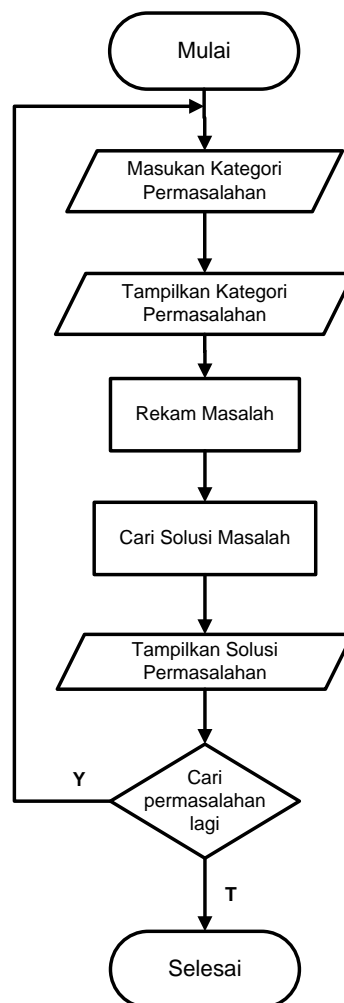


Dari pohon keputusan di atas dibagi menjadi 2 macam kategori, untuk kode huruf di dalam lingkaran merupakan simbol fakta-fakta yang diambil pada *tabel 4.1*, sedangkan kode angka didalam kotak merupakan simbol keputusan yang diambil pada *tabel 4.3* di atas. Untuk kode Y dan T merupakan pilihan dari aturan yang ada yaitu Ya atau Tidak, jika fakta ditelusuri Ya maka akan lanjut pilihan di sebelah kiri, jika fakta ditelusuri Tidak maka dilanjutkan pilihan di sebelah kanannya.

Sebagai contoh pembacaan pohon keputusan di atas, kita ambil urutan keputusan untuk A, B, dan 1. Maka kita definisikan dulu, A adalah "Matakuliah di remedial", B adalah "Remedial Semester ini" dan 1 adalah "Disarankan remedial dengan mengulang di semester ini". Jadi artinya : Goal dari pohon keputusan ini adalah Matakuliah di remedial, kemudian ditelusuri lagi dengan pilihan fakta yaitu Remedial Semester ini, setelah itu ditelusuri lagi dengan pilihan fakta Tidak, berarti sudah mencapai keputusan bahwa Disarankan remedial dengan Mengulang di semester ini.

Algoritma Deteksi Masalah Menentukan Remedial

Algoritma untuk mendeteksi masalah penentuan remedial matakuliah yang akan digunakan pada sistem ini, prosedurnya dapat dilihat pada *flowchart* berikut :



Gambar : Algoritma Menentukan Remedial Matakuliah

Dari *flowchart* di atas yang menggambarkan algoritma dari sistem aplikasi diagnosa permasalahan menentukan matakuliah remedial adalah sebagai berikut: untuk diagnosa permasalahan, aplikasi bisa langsung dijalankan oleh *user* dengan mengawali pemilihan kategori permasalahan, kemudian program akan menampilkan beberapa pilihan kategori permasalahan, saat *user* sudah memilih kategori permasalahan tersebut maka sistem akan merekam masalah dilanjutkan dengan mencari solusi permasalahan dengan hasil diagnosa yang dipilih oleh *user* tersebut, jika sistem sudah mendapatkan solusi yang tepat, maka solusi akan dimunculkan oleh sistem. Dan jika *user* ingin mencari solusi permasalahannya lagi maka

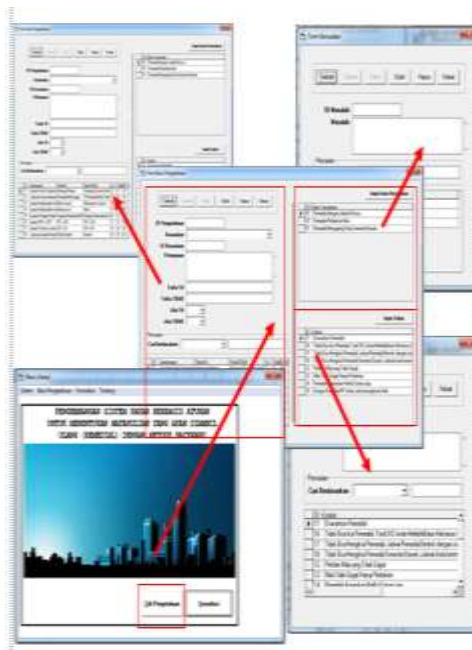
sistem akan mengarahkan kepada pemilihan kategori kembali jika tidak *user* bisa menutup aplikasinya.

5 IMPLEMENTASI DAN PENGUJIAN

5.1 Skenario Pengujian Proses Menu Utama

Form menu utama di atas terdapat beberapa tombol-tombol dan menu-menu. Untuk menu-menu berada pada bagian atas *form*, sedangkan tombol-tombol berada pada bagian tengah *form*, yang terdiri dari tombol Edit Pengetahuan, *Input* Permasalahan, *Input* Solusi dan *Diagnosis* Masalah. Untuk tombol *Edit* pengetahuan jika diklik maka akan muncul *Form* Basis Pengetahuan, untuk tombol *Input* Permasalahan jika diklik maka akan muncul *Form* Permasalahan, untuk tombol *Input* Solusi jika diklik maka akan muncul *Form* Solusi, dan untuk tombol *Diagnosis* Masalah jika diklik akan muncul *Form* pertanyaan-pertanyaan.

Adapun skenario untuk *testing form* menu utama diatas sebagai berikut : Untuk tombol *Edit* Pengetahuan, *Input* Permasalahan, *Input* Solusi dan *Diagnosis* Masalah dilakukan eksekusi dengan mengklik masing-masing tombol tersebut, Setelah ditest maka hasilnya program merespon tombol tersebut sesuai dengan *settingan* yang telah diprogram.



Tampilan Proses Pengujian Form Menu Utama

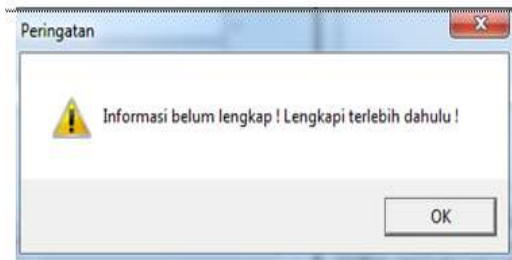
5.2 Skenario Pengujian Proses Form Basis Pengetahuan

Form ini berfungsi untuk menghubungkan permasalahan – permasalahan yang ada sehingga menampilkan solusi yang tepat bagi si pengguna sistem.

1. Untuk pengisian data-data yang dibutuhkan sebagai berikut :
 - a. ID Pengetahuan merupakan kode pembeda dari data-data yang akan dipnut nantinya, contoh ID Pengetahuan “T1”
 - b. Kolom masalah & ID Masalah ini fungsinya sejalan, sumber datanya dari penginputan yang dilakukan dari pada Form Permasalahan, yang akan dieksekusi di sini dengan mengKlik Combo Box ID Masalah, maka akan muncul data Masalah yang akan dipilih
 - c. Kolom pertanyaan diisi dengan pertanyaan-pertanyaan yang diberikan untuk ditelusuri oleh si pengguna nantinya.
 - d. Kolom isian “Fakta Ya” dan “Fakta Tidak” diisi dengan komentar dari pertanyaan.
 - e. Untuk Combo Box “Jika Ya” dan “Jika Tidak” dengan cara memilih data-data yang dimunculkan, yang merupakan keterkaitan dari pertanyaan yang diberikan, jika

pertanyaan yang diberikan mengarah kepada jawaban “Ya” maka dilanjutkan ketahapan pilihan berikutnya apakah ke pertanyaan atau solusi.

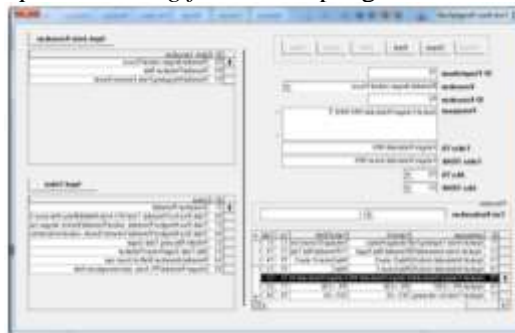
2. Untuk kolom pencarian berfungsi untuk *proses* pencarian data-data yang tersimpan pada *database*
3. Tombol-tombol untuk *kontrol* program yang berada di sebelah kanan berfungsi sebagai berikut :
 - a. Tombol “Tambah” Untuk menambahkan data.
 - b. Tombol “ Simpan” untuk menyimpan data, yang akan aktif setelah mengklik tombol “Tambah”. Jika data terisi sebagian maka tombol simpan tidak bisa berfungsi. Maka akan keluar pesan *warning* sebagai berikut :



Tampilan Pesan *Warning* Data isian Tidak Lengkap

- c. Tombol “Batal” untuk membatalkan isian data, akan aktif jika tombol “Tambah” sudah diklik.
- d. Tombol “Ubah” untuk eksekusi data-data yang akan diubah dengan memilih dulu *record* yang ada pada *datagrid*.
- e. Tombol “Hapus” untuk menghapus data pada database dengan memilih dulu *record* yang ada pada *datagrid*.
- f. Tombol “Keluar” untuk keluar dari *form* basis pengetahuan

Berikut gambaran *proses testing form* basis pengetahuan :



Tampilan *Testing Form* Basis Pengetahuan

5.3 Skenario Pengujian *Proses Form* Parameter Kategori Permasalahan

Fungsi dari *form* permasalahan ini adalah untuk menyimpan data kategori-kategori dari permasalahan remedial. Adapun *skenario testing* dari *Form* Kategori Permasalahan ini sebagai berikut :

1. Data-data yang akan diinputkan di *form* ini adalah : ID Masalah dan Jenis Masalah.
2. Untuk ID Masalah berisi dengan kode masalah untuk membedakan kategori masalah dengan yang lainnya. Contoh isiannya “R1”.
3. Untuk Jenis Masalah berisi dengan narasi kategori dari masalah-masalah yang dikelompokkan.
4. Tombol-tombol untuk kontrol program yang berada di bagian atas berfungsi sebagai berikut :
 - a. Tombol “Tambah” Untuk menambahkan data.

- b. Tombol “ Simpan” untuk menyimpan data, yang akan aktif setelah mengklik tombol “Tambah”. Jika data terisi sebagian maka tombol simpan tidak bisa berfungsi maka akan keluar pesan *warning*.
 - c. Tombol “Batal” untuk membatalkan isian data, akan aktif jika tombol “Tambah” sudah diklik.
 - d. Tombol “Ubah” untuk eksekusi data-data yang akan diubah dengan memilih dulu *record* yang ada pada *datagrid*.
 - e. Tombol “Hapus” untuk menghapus data pada database dengan memilih dulu *record* yang ada pada *datagrid*.
 - f. Tombol “Keluar” untuk keluar dari *form* permasalahan.
- Berikut gambaran *proses testing form* kategori permasalahan :

ID	Isi
S1	Disarankan Remedial
S6	Tidak Bisa ikut Remedial, Total SKS Anda Melebihi Batas Maksimum !
S7	Tidak Bisa Mengikuti Remedial, Jadwal Remedial Bertolak dengan Jadwal
S8	Tidak Bisa Mengikuti Remedial Semester Berikut, Jadwal Anda bertolak
S2	Perbaiki Nilai yang Tidak Gagal
S3	Nilai Tidak Gagal Hanya Perbaikan
S4	Remedial Akan diberikan Maksimal 10 menit waktu

Tampilan *Testing Form* Parameter Kategori Permasalahan

5.4 Skenario Pengujian Proses *Form* Parameter Solusi

Fungsi dari *form* solusi ini adalah untuk menyimpan data-data solusi dari permasalahan remedial. Adapun *skenario* untuk *testing form* kategori ini sebagai berikut :

1. Data-data yang akan diinputkan di *form* ini adalah : ID Solusi dan Solusi.
2. Untuk “ID Solusi” berisi dengan kode solusi untuk membedakan solusi-solusi dengan yang lainnya. Contoh isiannya “S1”.
3. Untuk “Solusi” berisi dengan narasi solusi-solusi permasalahan.
4. Tombol-tombol untuk kontrol program yang berada di bagian atas berfungsi sebagai berikut :
 - a. Tombol “Tambah” Untuk menambahkan data.
 - b. Tombol “ Simpan” untuk menyimpan data, yang akan aktif setelah mengklik tombol “Tambah”. Jika data terisi sebagian maka tombol simpan tidak bisa berfungsi maka akan keluar pesan *warning*.
 - c. Tombol “Batal” untuk membatalkan isian data, akan aktif jika tombol “Tambah” sudah diklik.
 - d. Tombol “Ubah” untuk eksekusi data-data yang akan dirubah dengan memilih dulu *record* yang ada pada *datagrid*.
 - e. Tombol “Hapus” untuk menghapus data pada *database* dengan memilih dulu *record* yang ada pada *datagrid*.
 - f. Tombol “Keluar” untuk keluar dari *form* permasalahan.

Berikut gambaran *proses testing form* parameter solusi :



Tampilan Testing Form Parameter Solusi

5.5 Skenario Pengujian Proses Form Pemilihan Kategori Permasalahan

Fungsi dari *form* ini untuk pemilihan kategori-kategori permasalahan, sebelum menampilkan pertanyaan-pertanyaan dari permasalahan, berguna untuk mengelompokkan pertanyaan agar memudahkan dalam penelusuran fakta. Sumber data yang dimunculkan dari *form* ini adalah dari hasil penginputan pada “Form kategori permasalahan” yang ada di bawah ini. Adapun skenario testing dari *form* ini sebagai berikut, dalam menjalankan *form* ini, *user* memilih kategori yang dimunculkan pada *combo box* kemudian setelah itu mengklik tombol “Proses” untuk lanjut ke tahapan berikutnya.

Dari hasil eksekusi program untuk *form* ini, berjalan sesuai dengan *proses* yang telah diprogram dan tidak ada kesalahan dari fungsinya. Berikut gambaran proses *testing form* kategori masalah :



Tampilan Testing Form Pemilihan Kategori Masalah

5.6 Skenario Pengujian Proses Form Diagnosis Masalah

Form Diagnosis masalah ini merupakan *form* tampilan pilihan-pilihan pertanyaan untuk menelusuri fakta-fakta dari permasalahan. Sumber pertanyaan yang muncul dari *form* ini berasal dari *input* data pada “Form pengetahuan” pada Gambar 5.6. *Form* ini dimunculkan sebelum “*form* pemilihan kategori permasalahan” pada Gambar 5.9. untuk testing dari tombol-tombol yang dimunculkan di *form* ini sebagai berikut :

- Tombol “Ya” diklik jika pertanyaan yang diberikan sesuai fakta
- Tombol “Tidak” diklik jika pertanyaan yang diberikan tidak sesuai dengan fakta
- Tombol “Selesai” diklik jika ingin mengakhiri proses dari *form*.

Berikut gambaran proses *testing form* diagnosis masalah :



Tampilan Testing Form Diagnosis Masalah

5.4 Form Hasil Diagnosis / Solusi

Form di atas merupakan *finishing proses* dari penelusuran fakta-fakta dari permasalahan yang menampilkan sebuah solusi terhadap masalah yang ditelusuri. Untuk tombol “Alasan” jika diklik akan memunculkan pertanyaan-pertanyaan yang telah ditelusuri, sedangkan tombol “Selesai” berfungsi untuk mengakhiri program.

Berikut gambaran *proses testing form* hasil diagnosis :



Tampilan *Testing Form* Hasil Diagnosis

6.SIMPULAN DAN SARAN

6.1 Simpulan

Berdasarkan uraian pada bab-bab sebelumnya dan setelah melakukan pembahasan, perancangan dan implementasi maka penulis dapat menarik beberapa kesimpulan sebagai berikut :

1. Sistem Pakar yang dibangun untuk input maupun aturan-aturan (*rule*) masih memiliki kekurangan di dalam *knowledge base*, sehingga untuk penelusuran fakta-fakta yang ada tidak terlalu mendalam, hanya data-data yang masih bersifat umum saja, yang biasa diterapkan pada lingkungan akademik.
2. Sistem Pakar yang dirancang ini, untuk penambahan Pengetahuan baru hanya bisa dilakukan oleh *Administrator* atau Pakar.
3. Sistem Pakar ini hanya mampu untuk *personal user* saja, dan belum mengarah kepada sistem *client server*.

6.2 Saran

Saran yang dapat disampaikan dalam rangka operasional dan pengembangan Sistem Pakar sebagai berikut

1. Untuk mendiagnosis penentuan matakuliah diremedial perlu dilakukan perbaikan input data dan kaidah (*rule*) sehingga didapatkan *goal* yang lebih tepat dan beragam. Artinya, program Sistem Pakar ini agar dikembangkan untuk dapat mendiagnosis permasalahan penentuan matakuliah remedial dimasa yang akan datang.
2. Sistem Pakar dikembangkan lagi agar *User* bisa menambahkan pengetahuan baru ke sistem secara otomatis tanpa melibatkan *administrator* atau pakar.
3. Sistem Pakar ini dikembangkan lagi secara *client server* yaitu kemudahan dalam mengakses program aplikasi ini terutama di lingkungan kampus Universitas Muhammadiyah Sumatera Barat maupun secara *online* yaitu dapat mengakses sistem ini dimanapun berada dengan memakai koneksi internet.

REFERENSI

- Anita Desiani & Muhammad Arhami (2006), "*Konsep Kecerdasan Buatan*", Andi Yogyakarta, Yogyakarta
- Kusrini (2006), "*Sistem Pakar (Teori & Aplikasi)*", Andi Yogyakarta, Yogyakarta
- Kusumadewi, Sri (2003), "*Artificial Intelligence (Teknik dan Aplikasinya)* ", Graha Ilmu, Yogyakarta
- Michael Negnevitsky (2002), "*Artificial Intelligence, A Guide to Intelligent Systems*"
- Sri Hartati & Sari Iswanti (2008), "*Sistem Pakar & Pengembangannya*", Graha Ilmu, Yogyakarta
- Sutojo, Edy Mulyanto & Vincent Suhartono (2010), "*Kecerdasan Buatan*", Andi Offset, Yogyakarta
- Sukardi, H.M (2008), "*Evaluasi Pendidikan, Prinsip & Operasionalnya*", Bumi Aksara, Yogyakarta
- Tim Penerbit Andi (2003), "*Pengembangan Sistem Pakar Menggunakan Visual Basic*", Andi Offset, Yogyakarta