

Sistem Pakar Untuk Penyusunan Jadwal Kuliah Berbasis Forward Chaining

Feriani Astuti Tarigan, M.Kom.
STMIK Time Medan
Jalan Merbabu No. 32 AA – BB
Telp 061. 456 1932, E-mail : feriani.astuti@yahoo.com

ABSTRACT

Subject scheduling is a process of arranging exact time to avoid conflict with related rules such as lecturers, classroom, laboratories, and the subjects offered. Subject scheduling is arranged before each semester strated in order to avoid repeticion, in sekolah tinggi manajemen informatika dan komputer time,subject scheduling is made manually by staff. It is true that the process is diffiiculty & is not timely difficult in this research, correct ruler are designed to avoid any conflict that may accurs. The description of the ruler are done by using forward chaining system

Key word : expert system, forward chaining ,subject scheduling

1. Pendahuluan

Penjadwalan memegang peranan penting dalam proses perencanaan dan pengendalian, baik dalam bidang pendidikan maupun dalam bidang-bidang yang lain. Tetapi kesulitan yang dihadapi adalah tidak adanya suatu metode yang benar-benar menghasilkan jadwal yang optimal, dan ini membuat bidang ini terus berkembang. Juga adanya bermacam-macam karakteristik sistem yang ada membuat tiap sistem membutuhkan metode yang tertentu untuk menghasilkan jadwal yang optimal.

Kampus adalah suatu tempat di mana terjadinya proses belajar mengajar. Sebelumnya terjadinya proses belajar mengajar (PBM) berlangsung pada awal semester, kegiatan rutin yang dilakukan adalah penyusunan jadwal kuliah. Jadwal kuliah merupakan hal yang sangat penting dalam pelaksanaan perkuliahan. Dengan jadwal kuliah yang baik maka mahasiswa akan merasa nyaman untuk melaksanakan studinya. Berbagai aspek yang harus diperhatikan dan berkaitan dengan penjadwalan, yaitu : program studi, kelas, matakuliah, dosen yang mengampu mata kuliah, ruangan, hari, waktu/session.

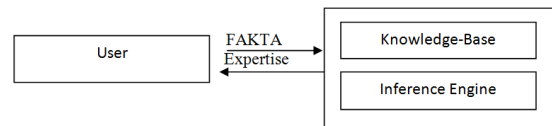
Kesalahan pada penyusunan jadwal adalah merupakan salah satu akibat dari keterbatasan manusia, baik kesalahan karena

ketidak mampuan manusia itu sendiri maupun kesalahan karena keterbatasan ketelitian yang dimiliki. Dalam penyusunan jadwal kuliah ada beberapa masalah yang dihadapi yaitu : Tidak ada jadwal paralel pada kelas yang sama, Keterbatasan jumlah ruangan.

Metode yang digunakan adalah *forward chaining* berbasis aturan. Forward chaining adalah strategi untuk memprediksi atau mencari solusi dari suatu masalah yang mulai dengan sekumpulan pakar yang diketahui, kemudian menurunkan fakta baru berdasarkan aturan yang premisnya cocok dengan fakta yang diketahui. Proses ini dilanjutkan sampai dengan mencapai goal atau tidak ada lagi aturan yang premisnya cocok dengan fakta yang diketahui.

2. Sistem Pakar

Sistem pakar merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang membuat penggunaan pengetahuan yang dikhususkan secara ekstensif untuk memecahkan masalah pada tingkatan manusia *expert*. Pakar adalah seseorang yang mempunyai keahlian dalam bidang tertentu. Pakar mempunyai pengetahuan atau ketrampilan tertentu yang tidak diketahui atau ada untuk kebanyakan orang. Teknologi sistem pakar ini mungkin mencakup bahasa sistem pakar tertentu/khusus, program, dan hardware yang ditentukan untuk tambahan dalam pengembangan dan pembuatan sistem pakar.



Gambar 1 Konsep Dasar dari Fungsi Sistem Pakar

Gambar 1 mengilustrasikan konsep pokok dari sistem pakar yang berdasarkan pada pengetahuan. User men-*supply* fakta atau informasi lainnya ke sistem pakar dan menerima nasihat *expert* atau pakar sebagai jawabannya. Secara internal, sistem pakar berisi dua komponen pokok. Yang berdasarkan

pengetahuan berisi pengetahuan yang “*inference engine*” menggambarkan kesimpulan. Kesimpulan ini adalah respon sistem pakar atas permintaan pemakai untuk *expertise*.

Sistem yang berdasarkan pengetahuan juga dirancang untuk beraksi sebagai asisten *intelligent* pada pakar manusia. Asisten *intelligent* ini dirancang dengan teknologi sistem pakar karena banyak pengetahuan ditambahkan dan akan beraksi lebih seperti pakar.

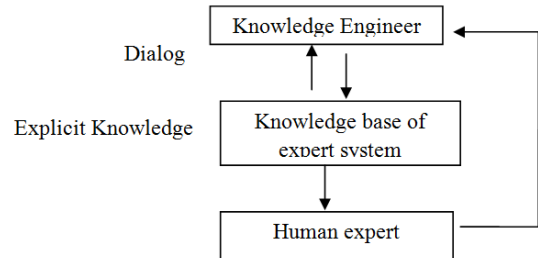
Knowledge Engineer (Perekayasa Sistem)

Adalah pihak yang membuat ES. *Knowledge engineer* ini bertugas menyerap dan mencarikan pengetahuan yang dimiliki oleh para pakar dan mengimplementasikannya kedalam sebuah *software expert sistem*. Tugas ini cukup sulit karena seorang *knowledge engineer* tidak boleh memasukkan perkiraan atau perasaannya ke dalam pengetahuan yang diperolehnya. Disamping itu dia juga harus pandai mengorek pengetahuan pakar, karena ada kalanya seorang pakar biasanya secara psikologis tidak dapat menceritakan atau menjelaskan semua keahliannya.

User (Pemakai)

Yaitu pihak yang mempergunakan program ES. Kemampuannya ES dikembangkan untuk menghemat waktu dan usaha user. ES mempunyai beberapa tipe user antara lain : Klien bukan pakar yang meminta nasihat, dalam hal ini ES bertindak sebagai konsultan atau penasihat. Siswa yang tidak tahu bagaimana harus belajar, dalam ini ES akan bertindak sebagai Partner. Jadi komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar. Pengetahuan dari sistem pakar dapat dipresentasikan dalam sejumlah cara. Salah satu metode yang paling umum untuk mempresentasikan pengetahuan adalah dalam bentuk aturan (rule) *IF...THEN* (jika...maka), seperti : **IF the light is red THEN stop** Tahap umum dalam pengembangan sistem pakar di ilustrasikan dalam gambar 2 *knowledge engineer* pertama kali membuat dialog dengan manusia ahli untuk mengeluarkan pengetahuan *expert*. Tahap ini merupakan analogi pada sistem dalam pemrograman konvensional yang mendiskusikan sistem yang diperlukan pengguna yang programnya akan dibuat.

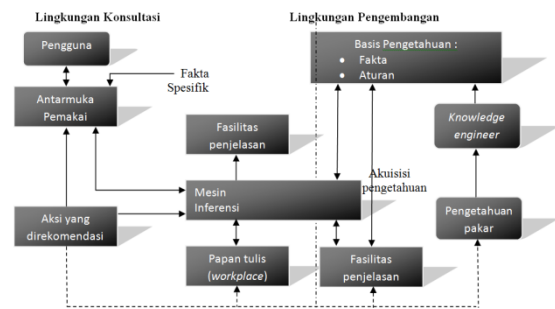
Knowledge engineer kemudian memberi *code* pengetahuan secara eksplisit dalam basis pengetahuan. Pakar kemudian mengevaluasi sistem pakar dan memberikan kritik pada *Knowledge engineer*. Proses ini berulang hingga penampilan sistem dianggap memenuhi oleh pakar.



Gambar 2 Perkembangan Sistem

Arsitektur Sistem Pakar

Arsitektur sistem pakar dapat dilihat pada gambar 3 di bawah ini, terdapat dua bagian utama sistem pakar yaitu lingkungan pengembangan (*development enviroment*) yang digunakan untuk memasukkan pengetahuan-pengetahuan pakar kedalam lingkungan sistem pakar dan lingkungan konsultasi (*consultation enviroment*) digunakan oleh pengguna yang bukan pakar untuk memperoleh pengetahuan pakar.



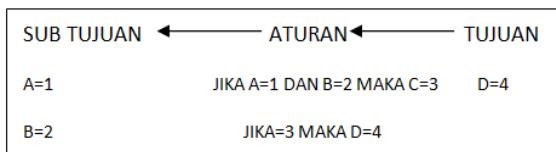
Gambar 3 Arsitektur Sistem Pakar (Sumber Turban, 2005)

Mesin Inference (Inference Engine)

Mesin inference adalah bagian yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang akan menganalisis suatu masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik. Secara deduktif mesin inferensi memilih pengetahuan yang relevan dalam rangka mencapai kesimpulan. Dengan demikian sistem ini dapat menjawab pertanyaan pemakai meskipun jawaban tersebut tidak tersimpan secara eksplisit didalam basis

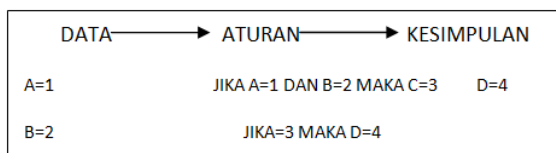
pengetahuan. Mesin inferensi memulai pelacakan dengan mencocokkan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data. Ada dua metode inferensi dalam sistem pakar yaitu pelacakan ke depan (*forward chaining*) dan pelacakan ke belakang (*backward chaining*) (Kusrini,2006).

Pelacakan kebelakang (*Backward Chaining*) yang mulai penalarannya dari sekumpulan hipotesa menuju fakta-fakta yang mendukung hipotes tersebut. Pada pelacakan kebelakang, penalaran dimulai dengan tujuan melacak balik ke jalur yang akan mengarahkan ketujuan tersebut. Tujuan dari inferensi ini adalah mengambil pilihan terbaik dari banyak kemungkinan. Gambar 4 berikut ini adalah menunjukkan proses penalaran menggunakan metode pelacakan ke belakang.



Gambar 4 Penalaran Pelacakan ke Belakang

Pelacakan ke Depan (*Forward Chaining*) yang merupakan kebalikan dari pelacakan ke belakang, yaitu memulai dari sekumpulan data menuju kesimpulan. Pelacakan ke depan berarti menggunakan himpunan aturan konsisi-aks. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan di jalankan. Gambar 5 menunjukkan cara kerja metode inferensi pelacakan ke depan.



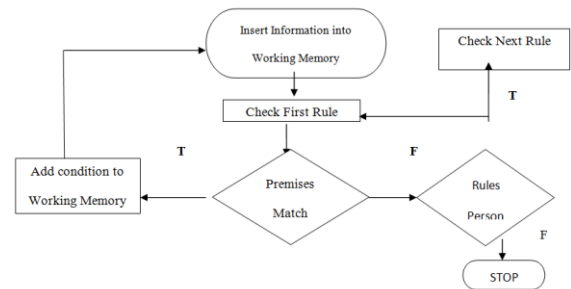
Gambar 5 Penalaran Pelacakan ke Depan

Konsep Forward Chaining

Forward chaining is the data-driven reasoning. The reasoning starts from the known data and proceeds forward with that data. Each time only the topmost rule is executed. When fired, the rule adds a new fact in the database. Any rule can be executed only once. The match-fire cycle stops when no further rules can be fired (Michael Negnevitsky, 2002).

Operasi dari sistem *forward chaining* dimulai dengan memasukkan sekumpulan fakta

yang diketahui ke dalam memori kerja (*working memory*), kemudian menurunkan fakta baru berdasarkan aturan yang premisnya cocok dengan fakta yang diketahui. Proses ini dilanjutkan sampai dengan mencapai goal atau tidak ada lagi aturan yang premisnya cocok dengan fakta yang diketahui. Operasi tersebut digambarkan seperti pada gambar 6



Gambar 1.6. Operasi Sistem Pakar Forward Chaining

Langkah-langkah yang harus dilakukan dalam membuat sistem *forward chaining* berbasis aturan, yaitu :

1. Pendefinisian Masalah. Tahap ini meliputi pemilihan domain masalah dan akuisisi pengetahuan.
2. Pendefinisian Data Input. Sistem *forward chaining* memerlukan data awal untuk memulai inferensi.
3. Pendefinisian Struktur Pengendalian Data. Aplikasi yang kompleks memerlukan premis tambahan untuk membantu mengendalikan pengaktifan suatu aturan.
4. Penulisan Kode Awal. Tahap ini berguna untuk menentukan apakah sistem telah menangkap domain pengetahuan secara efektif dalam struktur aturan yang baik.
5. Pengujian Sistem. Pengujian sistem dilakukan dengan beberapa aturan untuk menguji sejauh mana sistem berjalan dengan benar.
6. Perancangan Antarmuka. Antarmuka adalah salah satu komponen penting dari suatu sistem. Perancangan antarmuka dibuat bersama-sama dengan pembuatan basis pengetahuan.
7. Pengembangan Sistem. Pengembangan sistem meliputi penambahan antarmuka dan pengetahuan sesuai dengan prototipe sistem.
8. Evaluasi Sistem.

Pada tahap ini dilakukan pengujian sistem dengan masalah yang sebenarnya. Jika sistem belum berjalan dengan baik maka akan dilakukan pengembangan kembali.

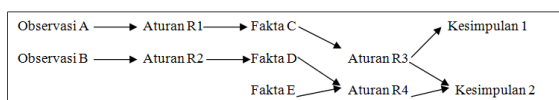
Rantai Telusur Maju (*Forward Chaining*)

Rantai Telusur Maju (*Forward Chaining*) adalah sebuah metode yang digunakan untuk mencari setiap kesimpulan yang mungkin berdasarkan dari sejumlah alasan atau dasar pemikiran yang diberikan. *Forward Chaining* terutama digunakan mendiagnosa semua kemungkinan yang didapat berdasarkan input yang diberikan berdasarkan pertimbangan-pertimbangan yang telah dilakukan. Pada *Forward Chaining*, kesimpulan yang dimaksud ering disebut *Data Driven*. *Data Driven* adalah *Engine Inference* yang bekerja dari konten awal *space* kerja menuju ke arah kesimpulan terakhir. Sebagai ilustrasi dapat dilihat pada contoh berikut :

Rule Base

- R1 : IF A AND B THEN D
- R2 : IF B THEN C
- R3 : IF C AND D THEN E

Rule-rule dalam *chaining* dibentuk berdasarkan produksi dari konsep sistem dan faktor yang dimasukkan dalam sistem. Dan pengambilan keputusan dengan menggunakan metode *forward chaining* dimulai dari inisialisasi batasan-batasan faktor berlanjut hingga kesimpulan akhir. *Forward chaining* juga disebut sebagai penalaran *forward* (*forward reasoning*). Dimana proses pencarian dimulai dari premis-premis atau informasi masukan (*IF*) dahulu kemudian menuju konklusi atau *deviverinformation* (*THEN*) atau dapat dimodelkan sebagai berikut : Informasi masukan dapat berupa data, bukti, temuan atau pengamatan. Sedangkan konklusi dapat berupa tujuan, hipotesa, penjelasan atau diagnosis. Sehingga jalannya penalaran *forward chaining* dapat dimulai dari data menuju tujuan, dari bukti menuju hipotesa, dari temuan menuju penjelasan atau dari pengamatan menuju diagnosa.

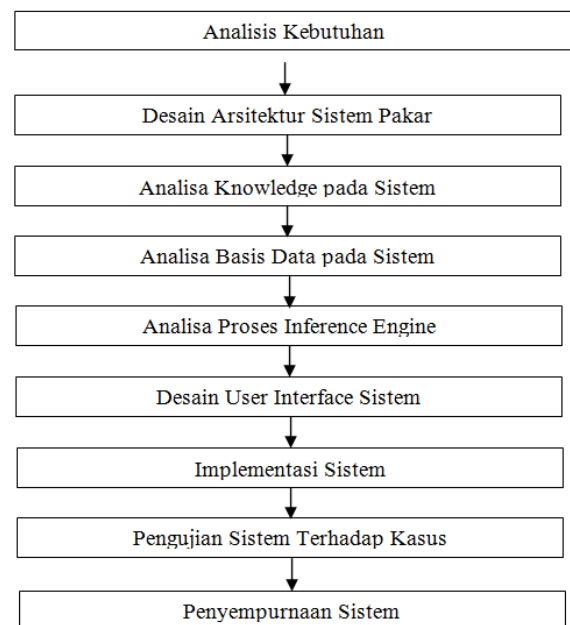


Gambar 7 Proses *Forward Chaining*

3. METODOLOGI PENELITIAN

Kerangka Kerja Penelitian

Metode penelitian adalah gambaran langkah-langkah agar penelitian dapat dilakukan dengan terstruktur. Langkah yang dilakukan mulai dari mempelajari masalah sampai dapat menghasilkan suatu sistem sehingga masalah dapat teratasi. Maka akan ditetapkan beberapa tahapan yang dilakukan, dapat dilihat pada gambar 8 :



Gambar 8 Kerangka Penelitian

Analisis Kebutuhan

Pada tahap analisa kebutuhan ini yang dilakukan adalah menganalisis siapa saja yang membutuhkan sistem, mengapa diperlukannya sistem pakar untuk menentukan penyusunan jadwal dan data berdasarkan data yang didapat yaitu program studi, kelas, matakuliah, ruangan hari dan waktu.

Desain Arsitektur Sistem Pakar

Pada tahap ini yang dilakukan adalah merancang bagaimana arsitektur dari sistem pakar yang akan dibangun. Dalam desain arsitektur ini akan ditentukan objek-objek apa saja yang dibutuhkan dan bagaiman hubungannya. Objek dari arsitektur sistem pakar ini adalah knowledge, database, inference engine, user interface, explanation facilities dan user.

Analisa Knowledge pada Sistem

Pada tahap ini yang dilakukan menentukan variabel-variabel apa saja dalam menentukan jadwal. Setelah variabel sudah ditentukan maka langkah berikutnya menentukan rule-rule untuk menganalisa variabel-variabel yang dibutuhkan. Setiap rule bisa diperlukan logika hubungan dari variabel-variabel yang ada dengan menggunakan logika AND.

Analisa Basis Data

Setelah tahap dari analisa kebutuhan dilakukan dan knowledge dari sistem diketahui maka langkah selanjutnya adalah merancang basisdata dari sistem pakarnya. Basisdata ini sudah mencakup data dari objek yang terlihat didalamnya dan apa saja proses yang dilakukan.

Analisa Proses Inference Engine Sistem

Pada tahap ini dilakukan adalah menggambarkan proses pencocokan data yang ada pada database terhadap rule yang ada di knowledge dengan menggunakan metode *forward chaining*. Proses pencocokan ini dapat lebih dari satu tahap sesuai dengan aturan rule yang ada.

Desain User Interface Sistem

Pada tahap ini yang dilakukan adalah merancang tampilan dari menu-menu yang dibutuhkan oleh sistem. Menu yang dirancang memiliki dua bagian yaitu menu pakar dan menu administrasi Menu dibagi menjadi dua karena ada dua level pengguna sistem. Masing-masing pengguna akan dibatasi oleh login password.

Implementasi Sistem

Pada tahap ini yang dilakukan adalah mengimplementasikan rancangan sistem ke dalam komputer dengan menggunakan bahasa pemrograman Visual Basic 6.0. setelah program dari aplikasi sistem pakar ini selesai dibangun maka pada tahap ini juga di bahas bagaimana cara menggunakannya agar user dapat mengoptimalkan penggunaan dari sistem tersebut.

Pengujian Sistem

Pada tahap ini, dilakukan pengujian terhadap sistem yang telah jadi dengan menggunakan data-data yang telah ada. Hal ini

bertujuan untuk mengetahui apakah sistem yang dirancang tersebut sudah sesuai dengan tujuan yang diharapkan. Hasil pengujian ini kemudian dijadikan dasar untuk membuat perbaikan-perbaikan yang diperlukan untuk menghasilkan sistem yang diharapkan.

Penyempurnaan Sistem

Sistem yang sudah diuji dan diketahui kekurangannya akan dengan mudah disempurnakan dengan memberikan kesimpulan dan menerima masukan berupa saran-saran untuk penyempurnaannya.

4. ANALISIS DAN DISAIN SISTEM

Analisa Kebutuhan

Penjadwalan mata kuliah adalah permasalahan yang cukup kompleks. Dalam permasalahan tersebut terdapat beberapa variabel yang harus dipertimbangkan agar dapat membentuk jadwal yang optimal. Ada 7 (tujuh) variabel penting yang harus diperhatikan agar penjadwalan menjadi optimal yaitu :

1. Program Studi

Program studi adalah program studi yang menyatakan dimana mahasiswa mengambil jurusan apa didalam sebuah perguruan tinggi tersebut.

2. Kelas

Jumlah mahasiswa berasal dari data registrasi mahasiswa yang telah melakukan pemilihan program studi yang diminati. Dari jumlah mahasiswa tersebut akan ditentukan banyaknya kelas berdasarkan program studi dan waktu kuliah (kuliah pagi dan sore). Dari data tersebut akan ditetapkan jumlah dosen yang akan mampu matakuliah yang ditawarkan serta pendistribusian dosen pada masing-masing matakuliah.

3. Matakuliah

Matakuliah adalah matakuliah yang berlangsung pada semester yang berjalan. Satu matakuliah dapat dijalankan dengan beberapa kelas. Untuk satu matakuliah akan dijalankan dalam tiap semester tergantung banyaknya kelas yang akan diselenggarakan untuk tiap semester.

4. Dosen

Jadwal dosen harus dibuat sedemikian rupa agar tidak terjadi dosen mengajar yang sama pada waktu yang sama. Status Dosen : Bila status dosen pengajar adalah Dosen tidak

tetap (dosen luar biasa), maka kelas yang diajarnya akan mendapat prioritas untuk dijadwal sesuai hari dan waktu yang telah ditetapkan jurusan sebelumnya. Bila terpaksa/tidak ada ruang tersisa pada hari dan jam yang ditetapkan, maka mata kuliah yang diajar oleh dosen tetap dapat diminta untuk dijadwal ulang hari atau waktunya.

5. Ruang

Ruangan yang digunakan untuk perkuliahan juga harus sesuai dengan fasilitas yang dibutuhkan oleh matakuliah dan kapasitas ruangan sesuai dengan jumlah mahasiswa tiap kelas. Kapasitas ruang tidak boleh kurang dari kapasitas kelas. Jumlah mahasiswa untuk setiap ruangan adalah sama.

6. Hari

Hari perkuliahan berlangsung selama 5 (lima) hari kerja yaitu Senin, Selasa, Rabu, Kamis, Jumat.

7. Waktu/sesi

Waktu yang digunakan dapat dibagi ke dalam dua bagian yaitu pagi dan sore. Waktu kuliah pagi dibagi menjadi 2 (dua) bagian yaitu : jam 09.01 s/d 10.00, 10.01 s/d 11.00, dan 11.01 s/d 12.00. waktu untuk perkuliahan sore hari dibagi menjadi 4 (empat) bagian yaitu : jam 17.00 s/d 18.00, 18.00 s/d 19.00, 19.00 s/d 20.00, 20.00 s/d 21.00

Pada penjadwalan matakuliah, ketujuh kriteria yang telah dibahas sebelumnya memiliki prioritas sebagai berikut :

- Prioritas 1 : Program Studi
- Prioritas 2 : Kelas
- Prioritas 3 : Matakuliah
- Prioritas 4 : Dosen
- Prioritas 5 : Ruangan
- Prioritas 6 : Hari
- Prioritas 7 : Waktu/ Sesi

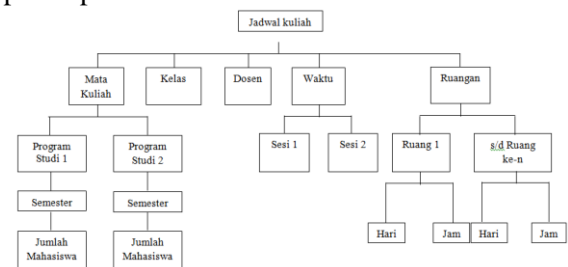
Sistem pakar *forward chaining* berbasis aturan membutuhkan pengetahuan dan mesin inferensi untuk menyusun jadwal kuliah. Basis pengetahuan ini berisikan faktor-faktor yang dibutuhkan oleh sistem. Sedangkan mesin *inferensi* digunakan untuk menganalisa faktor-faktor yang dimasukkan pengguna sehingga dapat ditemukan suatu kesimpulan basis pengetahuan yang diperlukan sistem, dan terdiri dari matakuliah, jumlah kelas, ruangan, jadwal dosen, hari, waktu. Data yang menjadi input sistem adalah data matakuliah, program studi, semester, kelas, dosen, ruang, hari, dan waktu

yang didapat dari ketua program studi. Data tersebut digunakan oleh sistem untuk menentukan ruang kuliah, jadwal dosen, hari, dan waktu.

Perancangan Sistem Pakar

Kerangka Kerja Sistem Pakar

Pembuatan block diagram dimaksudkan untuk mengetahui dan membatasi ruang lingkup permasalahan yang dibahas dengan mengetahui posisi pokok bahasan.



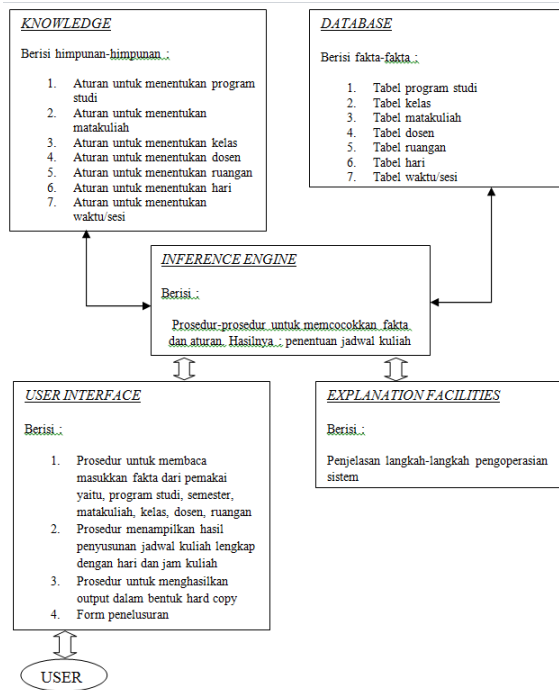
Gambar 9 Kerangka Kerja Area Permasalahan

Kerangka kerja pada gambar 9 merupakan gambar yang menjelaskan tentang beberapa fakta yang termasuk ke dalam kelompok penyusunan jadwal dan merupakan pokok permasalahan yang akan dibahas.

Salah satu penelusuran yang dirancang dalam sistem pakar penyusunan jadwal adalah bentuk pohon penelusuran *depth first search*, dimana penelusuran pertama dilakukan dengan mengidentifikasi setiap kelompok matakuliah untuk tiap program studi .

Desain Arsitektur Sistem

Setelah dilakukan penyederhanaan terhadap arsitektur sistem pakar maka desain arsitektur sistem pakar dalam penyusunan jadwal kuliah dapat dilihat pada gambar 10.



Gambar 10 Disain Arsitektur Sistem Pakar

Komponen-komponen yang ada pada arsitektur sistem pakar adalah sebagai berikut :

Knowledge Base

Knowledge base terdiri dari dua elemen dasar yaitu fakta dan rules. Dalam kasus ini adalah memasukkan fakta-fakta yang dibutuhkan oleh sistem. Fakta-fakta yang ditemukan adalah program studi (Prodi), kelas, matakuliah, dosen, hari kuliah, dan sesi/waktu kuliah. Dari fakta-fakta tersebut maka untuk menyusun jadwal perkuliahan ditemukan 5 (lima) *rule* yang didefinisikan sebagai berikut :

- $$R1 : A \wedge B \rightarrow C$$
- $$R2 : C \rightarrow D$$
- $$R3 : C \wedge D \rightarrow E$$
- $$R4 : E \rightarrow F \wedge G$$
- $$R5 : C \wedge D \wedge E \wedge F \wedge G \rightarrow H$$

Keterangan :

Untuk menyederhanakan aturan-aturan dan fakta-fakta di atas dapat dilambangkan dengan :

- A = Prodi
- B = Kelas
- C = Matakuliah
- D = Dosen
- E = Ruangan
- F = Hari
- G = Session
- H = Jadwal Tersusun

- Rule R1** membutuhkan data masukan berupa data program studi dan data kelas untuk mengetahui data matakuliah apa saja yang akan dijadwalkan.
- Rule R2** membutuhkan data matakuliah apa saja yang akan dijadwalkan untuk menentukan dosen-dosen yang akan mengjarakan matakuliah tersebut.
- Rule R3** membutuhkan data matakuliah dan data dosen untuk menentukan ruangan yang akan digunakan.
- Rule R4** membutuhkan data ruangan untuk menentukan hari dan sesi/waktu perkuliahan.
- Rule R5** membutuhkan data matakuliah, dosen, ruangan, hari dan sesi/waktu untuk menghasilkan jadwal.

Database

Database yang dibangun terdiri dari fakta-fakta antara lain : A (prodi), B (kelas), C (matakuliah), D (dosen), E (ruangan), F (hari), G (waktu/sesi), H (jadwal tersusun). Dalam perancangan sebuah basis data, diperlukan adanya pemodelan terhadap kebutuhan dan aktifitas yang terjadi pada basis data tersebut. Sehingga pada pelaksanaannya basis data tersebut dapat menjalankan proses dengan baik sesuai dengan kebutuhannya. Adapun perancangan basis data pada sistem pakar untuk menentukan jadwal kuliah adalah sebagai berikut :

1. Tabel Login

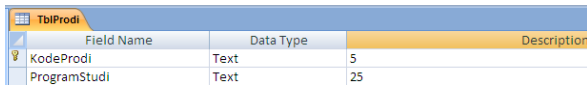
Tabel ini berfungsi untuk merekam *user name* dan *password* dari pengguna sistem. Pengguna sistem ini dapat lebih dari satu orang oleh karena itu setiap pengguna harus mendaftarkan *user name* dan *password*. Untuk keterangan *field-field* yang dirancang seperti gambar gambar 11.

Field Name	Data Type	Description
Namauser	Text	10
Password	Text	10
Status	Text	14

Gambar 11 Tabel Pakar

2. Tabel Program Studi

Tabel program studi adalah tabel data untuk menyimpan data-data program studi. Di mana ada dua program studi yaitu sistem informasi dan teknik informatika. Untuk keterangan *field-field* dapat dilihat pada gambar 12.

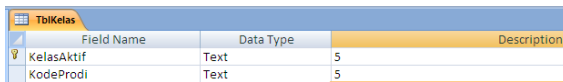


Field Name	Data Type	Description
KodeProdi	Text	5
ProgramStudi	Text	25

Gambar 12 Tabel Program Studi

3. Tabel Kelas

Tabel kelas berfungsi untuk menyimpan pembagian kelas dari mahasiswa. Setiap angkatan dapat memiliki jumlah mahasiswa yang lebih besar dari 25 orang sehingga dapat terbentuk kelas paralel. Untuk keterangan *field-field* dari tabel kelas dapat dilihat pada gambar 13.

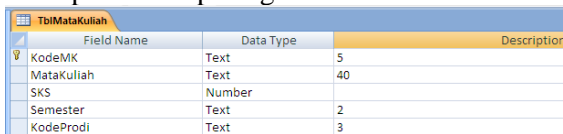


Field Name	Data Type	Description
KelasAktif	Text	5
KodeProdi	Text	5

Gambar 13 Tabel Kelas

4. Tabel Mata Kuliah

Tabel mata kuliah ini berfungsi sebagai tempat penyimpanan data mata kuliah yang diambil berdasarkan semester yang berlangsung. Untuk keterangan *field-field* dapat dilihat pada gambar 14.

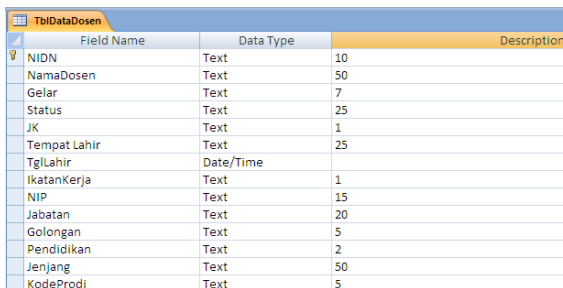


Field Name	Data Type	Description
KodeMK	Text	5
MataKuliah	Text	40
SKS	Number	
Semester	Text	2
KodeProdi	Text	3

Gambar 14 Tabel Mata Kuliah

5. Tabel Dosen

Tabel dosen adalah tabel data untuk menyimpan data dosen yang mengajar per program studi dan per semester. Untuk keterangan *field-field* dapat dilihat pada gambar 15.

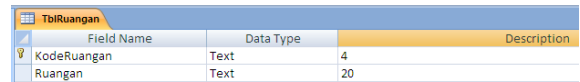


Field Name	Data Type	Description
NIDN	Text	10
NamaDosen	Text	50
Gelar	Text	7
Status	Text	25
JK	Text	1
Tempat Lahir	Text	25
TglLahir	Date/Time	
IkatanKerja	Text	1
NIP	Text	15
Jabatan	Text	20
Golongan	Text	5
Pendidikan	Text	2
Jenjang	Text	50
KodeProdi	Text	5

Gambar 15 Tabel Dosen

6. Tabel Ruang

Tabel ruangan berfungsi untuk merekam data ruangan, baik itu ruangan belajar maupun ruangan laboratorium beserta kapasitas *jumlah* siswa yang dapat belajar di ruangan tersebut. Adapun *field-field* yang digunakan dapat dilihat pada gambar 16.

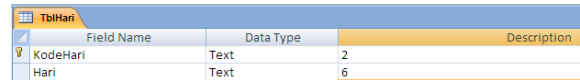


Field Name	Data Type	Description
KodeRuang	Text	4
Ruang	Text	20

Gambar 16 Tabel ruangan

7. Tabel Hari

Tabel hari adalah tabel data untuk menyimpan hari-hari yang digunakan untuk melakukan proses belajar-mengajar di STMIK TIME. Untuk keterangan *field-field* dilihat pada gambar 17.

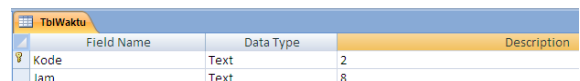


Field Name	Data Type	Description
KodeHari	Text	2
Hari	Text	6

Gambar 1.7 Tabel Hari

8. Tabel Waktu

Tabel waktu adalah tabel data untuk menyimpan pembagian waktu jam mengajar di STMIK TIME. Untuk keterangan *field-field* dapat dilihat pada gambar 18.

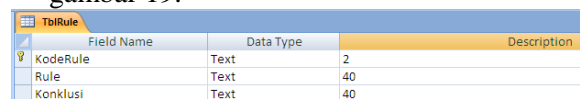


Field Name	Data Type	Description
Kode	Text	2
Jam	Text	8

Gambar 18 Tabel Waktu

9. Tabel Rule

Tabel ini berfungsi untuk merekam data *rule-rule* yang akan digunakan beserta kesimpulan dari setiap rule. *Field-field* yang digunakan pada tabel ini dapat dilihat pada gambar 19.

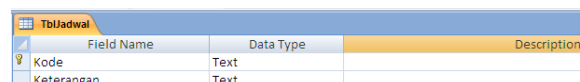


Field Name	Data Type	Description
KodeRule	Text	2
Rule	Text	40
Konklusi	Text	40

Gambar 19 Tabel Rule

10. Tabel Jadwal

Tabel ini berfungsi untuk merekam kode ruangan, beserta keterangan yang berisi hari dan sesi/waktu yang merupakan hasil ketentuan *rule-rule* yang dibentuk. *Field-field* yang digunakan pada tabel ini dapat dilihat pada gambar 20.



Field Name	Data Type	Description
Kode	Text	
Keterangan	Text	

Gambar 20 Tabel Jadwal

Inference Engine

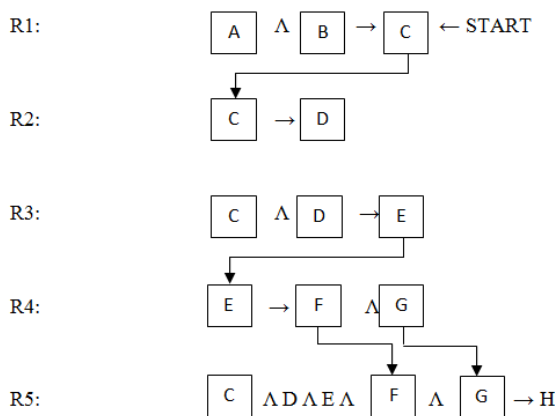
Inference engine merupakan perangkat lunak yang melakukan penalaran dengan menggunakan pengetahuan yang ada seperti prosedur untuk mencocokkan fakta ruangan dengan hari dan jam matakuliah yang ditawarkan. Mekanisme *inferensi* mengandung

suatu mekanisme pola pikir dan penalaran yang digunakan dalam menyelesaikan suatu masalah, dalam hal ini bagaimana sistem dapat mengambil suatu kesimpulan berdasarkan data (program studi, kelas, matakuliah, dosen, ruangan, hari, waktu/sesi) yang dimasukkan oleh *user*. Selanjutnya pendekatan yang dipakai sistem pakar untuk menyusun jadwal dengan menggunakan pelacakan ke depan (*forward chaining*) dimana pelacakan tersebut didasarkan pada data masukkan dan selanjutnya akan dibuat kesimpulannya berupa jadwal kuliah.

Pada proses penyusunan jadwal, dalam keadaan tertentu mungkin saja satu matakuliah diselenggarakan untuk lebih dari satu kelas, satu dosen dapat membawakan lebih dari satu matakuliah dan satu ruangan dapat digunakan untuk banyak matakuliah. Solusi yang dapat diambil untuk menyelesaikan adalah melakukan pengelompokkan matakuliah. Selanjutnya sistem pakar akan melakukan pelacakan kelas dan dosen untuk tiap matakuliah. Selanjutnya sistem pakar akan melakukan identifikasi untuk menentukan ruang, hari dan waktu yang akan digunakan untuk tiap matakuliah sehingga dihasilkan suatu susunan jadwal kuliah.

Berdasarkan aturan dan fakta yang ada, disusunlah suatu *inference engine* dapat dilihat pada gambar 4.3 di mana *inference engine* ini menyediakan mekanisme fungsi penalaran sistem yang digunakan oleh seorang pakar sebagai berikut :

1. Mekanisme akan menjalankan masalah tertentu dan selanjutnya mencari kesimpulan atau jawaban yang terbaik
2. Mesin ini akan dimulai pelacakannya dengan mencocokkan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data



Gambar 21 Sistem *Inference* dengan *Forward Chaining*

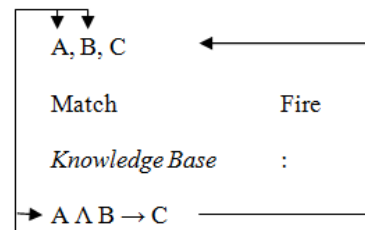
Berikut adalah penjelasan dari Sistem *Inference Engine* penyusunan jadwal dari tabel 4.11 dengan *Forward Chaining* :

Tabel 1 Aturan/Rule

ATURAN	RULE
R1	IF A \wedge B THEN C
R2	IF C THEN D
R3	IF C \wedge D THEN E
R4	IF E THEN F \wedge G
R5	IF F \wedge G \wedge E \wedge D \wedge C THEN H

Berikut ini adalah tahap-tahap penelusuran dengan inferensi *engine cycles procedure match fire* dengan menggunakan *Forward chaining* :

1. Database



$C \rightarrow D$

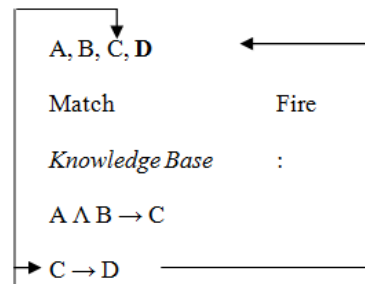
$C \wedge D \rightarrow E$

$E \rightarrow F \wedge G$

$C \wedge D \wedge E \wedge F \wedge G \rightarrow H$

R1 membutuhkan data A dan B dari database sehingga menghasilkan data C didalam database terdapat atau ditemukan data A dan B sehingga data C di-*-fired* dan ditambahkan ke dalam database.

2. Database

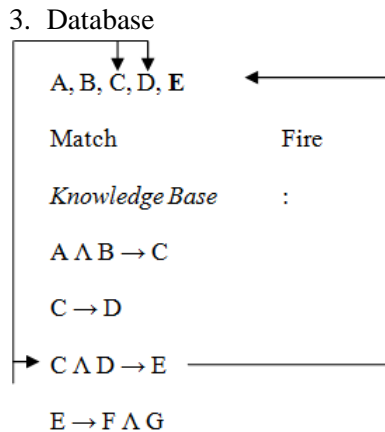


$C \wedge D \rightarrow E$

$E \rightarrow F \wedge G$

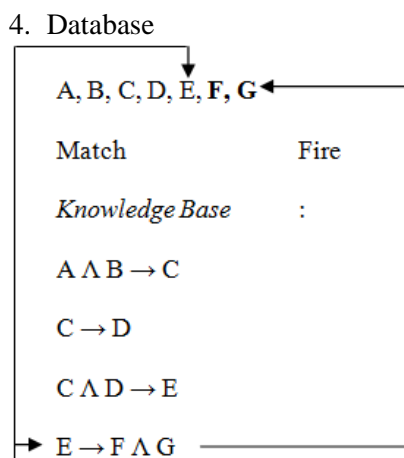
$C \wedge D \wedge E \wedge F \wedge G \rightarrow H$

R2 membutuhkan data C dari database sehingga akan menghasilkan data D, didalam database terdapat atau ditemukan data C sehingga data D akan di-fired dan ditambahkan kedalam database.



$$C \wedge D \wedge E \wedge F \wedge G \rightarrow H$$

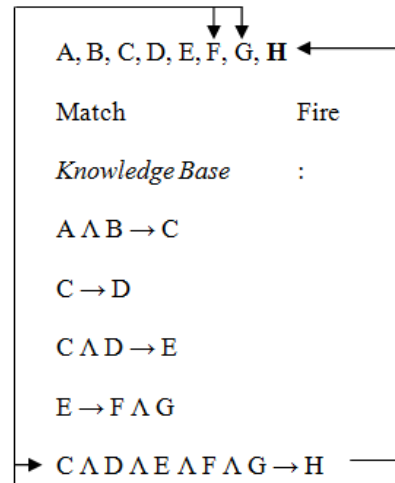
R3 membutuhkan data C dan D dari database sehingga akan menghasilkan data E, didalam database terdapat atau ditemukan data C sehingga data D akan di-fired dan ditambahkan kedalam database.



$$C \wedge D \wedge E \wedge F \wedge G \rightarrow H$$

R4 membutuhkan data E dari database sehingga akan menghasilkan data F dan G, didalam database terdapat atau ditemukan data E sehingga data F dan G akan di-fired dan ditambahkan kedalam database.

5. Database



R5 membutuhkan data C, D, E, F, dan G sehingga akan menghasilkan data H didalam database terdapat atau ditemukan C, D, E, F dan G data di-fired dan ditambahkan kedalam database.

Untuk memahami cara kerja dari sistem ini maka diberikan contoh sebagai berikut.

Langkah pertama adalah dengan cara memilih program studi misalnya kita akan pilih program studi Teknik Informatika dan Sistem Informasi

Langkah dua adalah setelah program studi ditentukan maka kelas akan tampil sesuai dengan program studi yang ditentukan diatas misalnya kelas **TIA 07** dan **SIA 07**

Langkah tiga adalah setelah kelas ditentukan pilih semester yang diambil misalnya semester I, maka akan tampil daftar maka kuliah sesuai dengan program studi yang ditentukan pada langkah pertama dipilih matakuliah yang akan dibuat jadwalnya dengan memilih mata kuliah misalnya : **algoritma dan pemograman dan pengantar sistem informasi.**

Langkah empat setelah mata kuliah ditentukan maka, dosen dapat dipilih sesuai dengan pengelompokkan mata kuliah yang dipilih, misalnya : Jackri Hendrik

Langkah lima setelah langkah I, II, III, IV, terpenuhi maka user hanya memilih tombol jadwal perkuliahan maka sistem akan mencari hari dan ruangan secara otomatis.

Untuk melihat hasil pengujian dari langkah-langkah rule pengujian dari *rule* tersebut dapat dilihat pada bab V

User Interface

User interface adalah perangkat lunak yang menyediakan media komunikasi antar

pengguna dengan sistem, seperti prosedur untuk membaca masukkan fakta dari pemakai yaitu program studi, kelas, matakuliah, dosen, ruangan, waktu prosedur menampilkan hasil penyusunan jadwal kuliah lengkap dengan hari dan jam kuliah. Prosedur untuk menghasilkan output dalam bentuk hard copy, *form* penelusuran.

Desain menu utama

Merupakan desain yang dirancang sedemikian rupa untuk menjalankan sistem yang telah dibangun, baik untuk login, penelusuran penyusunan jadwal yang telah dibangun, maupun bagaimana cara penyusunan yang telah dirancang.

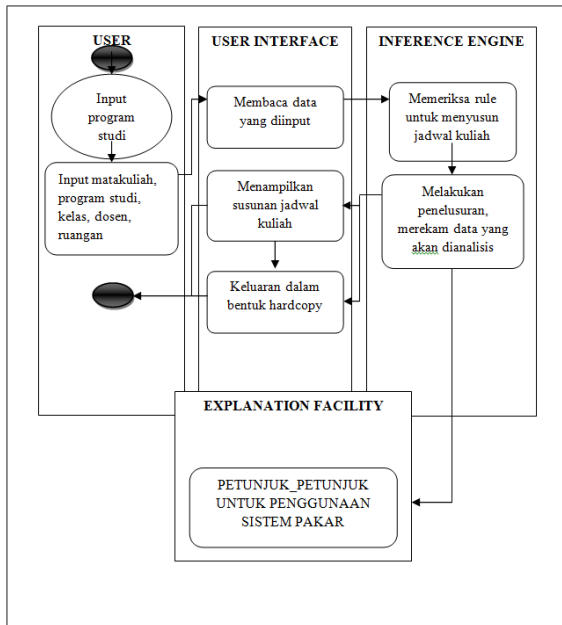
Adapun tampilan menu utama. Ada dua menu utama yang akan dibentuk yaitu :

- a. Menu Utama Pakar
Menu ini berfungsi untuk menambah kepakaran dalam hal penyusunan jadwal atau juga untuk melakukan koreksi terhadap *rule-rule* yang telah dibuat.
- b. Menu Admin
Menu ini berfungsi untuk Akademik dimana pada menu ini administrasi hanya dapat melihat jadwal yang sudah diinput oleh pakar.
- c. Desain *Form* Login
Merupakan yang tersedia dalam menu login yakni pakar dan admin agar pakar dan admin maupun user bisa melakukan login atau menjalankan sistem pakar penyusunan jadwal yang telah ditentukan.
- d. Desain *Form* Program Studi
Form ini berfungsi untuk mengisi tabel program studi yang terdiri dari data kode program studi dan nama program studinya. Pada *form* ini disediakan tombol untuk menyimpan data yang baru, membatalkan aksi yang diinginkan, mengubah data(update), menghapus data dan keluar dari run *form* program studi.
- e. Desain *Form* Kelas
Pada desain *form* kelas akan digunakan untuk mencatat kelas dari masing-masing kelas yang ada dan jumlah mahasiswa.
- f. Desain *Form* Matakuliah
Pada desain *form* matakuliah ini adalah menyimpan semua matakuliah yang wajib diambil atau yang telah ditentukan sesuai dengan masing-masing program studi dan semester yang telah ditetapkan.
- g. Desain *form* Dosen
Pada desain *form* dosen ini digunakan untuk menyimpan data dosen yang mengampu

matakuliah yang telah ditetapkan oleh program studi. Setiap kata dosen yang disimpan akan ditetapkan kode masing-masing dosen sesuai dengan kelompok dosen yang sudah ditetapkan.

- h. esain *Form* Ruangan
Pada desain *form* ruangan bahwa setiap ruangan akan diberi kode atau nama sehingga dapat digunakan untuk membantu dalam menetapkan jadwal sehingga tidak ada pemakaian ruangan yang sama dalam waktu bersamaan.
- i. Desain *Form* hari
Pada desain *form* hari, untuk setiap hari diberi kode mulai dari senin sampai dengan jumat diberi kode dengan nama H1 sampai H5.
- j. Desain *Form* Waktu
Pada desain *form* waktu ini penelusuran yang terdapat pada tabel ini adalah menentukan waktu pada session berapa mahasiswa tersebut mengikuti waktu perkuliahan dan dalam waktu tersebut dibuat adanya jam awal dan jam akhir perkuliahan.
- k. Desain *Form* Rule
Form ini dirancang untuk menyimpan rule-rule yang akan digunakan pada sistem penjadwalan. Data yang dibutuhkan berupa kode rule-rule dan konklusinya. Tombol-tombol yang digunakan berupa tombol simpan, batal, update, hapus dan keluar.
- l. Desain *Form* Detail Rule
Form untuk detail rule ini akan menginput tahun akademik, kode program studi, kelas yang sesuai dengan program studi yang terpilih beserta semester yang sesuai dengan kode tahun ajaran. Tahap ini dengan aturan dari rule 1. Tahap dari rule yaitu menginput mata kuliah, rule ini dapat dilakukan setelah rule 1 diberikan sehingga mata kuliah yang akan dimunculkan adalah mata kuliah yang sesuai dengan program studi dan semester yang terpilih. Tahap dari rule 3 adalah menentukan nama dosen yang sesuai dengan bidang matakuliah yang terpilih. Setelah semua detail rule ini diisi maka program nanti akan menentukan langsung ruangan, jam dan hari untuk jadwal perkuliahan oleh karena itu dibutuhkan tombol untuk menyimpan, mengubah, dan menghapus data dari detail rule.
- m. Explanation facilities
Explanation facilities penjelasan merupakan komponen tambahan yang dibuat agar pemakai dapat memanfaatkan sistem dengan benar.

- n. User adalah pemakai yang menggunakan program sistem pakar ini untuk menyusun jadwal kuliah berdasarkan matakuliah dan ruangan yang akan digunakan. Berikut ini merupakan diagram aktifitas secara umum yang menggambarkan sistem yang dirancang mulai dari kegiatan awal sistem, keputusan yang dihasilkan sampai sistem berakhir.



Gambar 22 Desain Aktivitas Sistem

5. DAFTAR PUSTAKA

- Arhami (2005). "Konsep Dasar Sistem Pakar" , Andi Offset, Yogyakarta.
- Giarattano, J. & Riley, G. (2005). " Expert System Principles and Programming". Giarattano, 4th Edition, PWS Publishing Company, Boston.
- Kusumadewi (2003). "Artificial Intelegence" , Graha ilmu Yogyakarta.