

LAPORAN PRAKTIKUM KECERDASAN BUATAN
SISTEM PAKAR PENDETEKSI JENIS NARKOBA YANG DIGUNAKAN
OLEH PECANDU



Disusun Oleh :

Dian Pertiwi H P	1515015180
Nadia	1515015184
Sri Uswatul Hasanah	1515015193

Asisten Praktikum :

M. Denny Irawam	Anisa N Afiah
1415015077	1415015068

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI DAN KOMUNIKASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MULAWARMAN

2017

Kata Pengantar

Puji syukur saya panjatkan kehadirat Allah SWT Tuhan Yang Maha Esa yang telah melimpahkan rahmat, hidayah, dan inaya-Nya kepada saya, sehingga saya dapat menyelesaikan laporan akhir mata kuliah Kecerdasan Buatan mengenai Sistem Pakar Pendeteksi Jenis Narkoba Yang Digunakan Pecandu.

Penyusunan laporan ini sebagai salah satu persyaratan tugas akhir mata kuliah Praktikum Kecerdasan Buatan Laporan ini saya susun dengan semaksimal mungkin dan tidak lepas dari bantuan berbagai pihak sehingga dapat memperlancar pembuatan laporan ini. Untuk itu saya menyampaikan banyak terima kasih kepada semua pihak yang telah berkontribusi dalam laporan ini.

Terlepas dari semua itu, saya menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasanya. Oleh karena itu dengan tangan terbuka saya menerima segala saran dan kritik dari pembaca agar kami lebih baik lagi ke depannya.

Akhir kata saya mengucapkan banyak terima kasih dan berharap semoga laporan ini dapat bermanfaat maupun terinspirasi terhadap pembaca.

Samarinda, 03 Mei 2017

Penulis

DAFTAR ISI

Halaman Judul	i
Kata Pengantar	i i
Daftar Isi	iii
Daftar Tabel	v
Daftar Gambar	vi
BAB I Pendahuluan	1
1.1. Latar belakang.....	1
1.2. Rumusan masalah	2
1.3. Batasan masalah.....	3
1.4. Tujuan dan Manfaat	3
BAB II Landasan Teori	4
2.1. Fungsi-Fungsi Visual Prolog yang Digunakan	4
2.1.1. Lingkungan Visual Prolog	4
2.1.2 Dasar-dasar Prolog.....	8
2.1.3 Program Visual Prolog.....	13
2.1.4. Unifikasi Dan Lacakbalik	18
2.1.5. Data Object Sederhana Dan Jamak.....	22
2.1.6. Perulangan Dan Rekursi	25

2.1.7. List	27
2.1.8. Section Facts	28
2.2 Teori Keilmuan yang Diimplementasikan....	32
2.2.1 Morfin	32
2.2.2 Heroin	32
2.2.3 Ganja	33
2.2.4 Kokain	34
2.2.5 Sabu-Sabu	35
2.2.6 Nikotin	36
BAB III Metodologi.....	38
3.1 Alur Pembuatan Sistem.....	38
BAB IV Hasil dan Pembahasan	39
4.1 Tabel Kebenaran	39
4.2 Analisis Aplikasi	40
BAB V Penutup	49
5.1 Kesimpulan	49
5.2 Saran	49
Daftar Pustaka	50
Lampiran	51
1. Source Code	51
2. Kartu Konsul	60

DAFTAR TABEL

Tabel 4.1 Tabel Kebenaran	38
Tabel 4.2 Tabel Keputusan	39

DAFTAR GAMBAR

Gambar 3.1 Alur Pembuatan Sistem	37
Gambar 4.2.1 Tampilan Awal Program	39
Gambar 4.2.2 Tampilan Pengisian Nama dan Umur	40
Gambar 4.2.3 Tampilan Untuk Memulai Program	40
Gambar 4.2.4 Tampilan Kembali Kemenu Nama dan Umur	40
Gambar 4.2.5 Tampilan Gejala-gejala Pada Narkotika Jenis Morfin	41
Gambar 4.2.6 Tampilan Gejala-gejala Pada Narkotika Jenis Heroin	42
Gambar 4.2.7 Tampilan Gejala-gejala Pada Narkotika Jenis Ganja	43
Gambar 4.2.8 Tampilan Gejala-gejala Pada Narkotika Jenis Kokain	44
Gambar 4.2.9 Tampilan Gejala-gejala Pada Narkotika Jenis Sabu-sabu	45
Gambar 4.2.10 Tampilan Gejala-gejala Pada Narkotika Jenis Nikotin	46
Gambar 4.2.11 Tampilan Jika Gejala-gejala taksesuai	47

BAB I

PENDAHULUAN

1.1 Latar Belakang

Narkoba adalah zat yang dapat menimbulkan pengaruh tertentu bagi mereka yang menggunakannya dengan cara memasukkan obat tersebut ke dalam tubuhnya, pengaruh tersebut berupa pembiasaan, hilangnya rasa sakit rangsangan, semangat dan halusinasi. Dengan timbulnya efek halusinasi inilah yang menyebabkan kelompok masyarakat terutama di kalangan remaja ingin menggunakan narkoba meskipun tidak menderita apa-apa. Hal inilah yang mengakibatkan terjadinya penyalahgunaan narkoba.

Penyalahgunaan narkoba pada remaja terjadi karena faktor pengaruh kondisi keluarga dan lingkungan. Dalam hal ini kondisi keluarga ditandai dengan keutuhan keluarga, kesibukan orang tua, hubungan interpersonal antar keluarga, dapat merupakan faktor yang berperan serta pada penyalahgunaan narkoba. Selain pada keluarga, lingkungan seperti halnya lingkungan sekolah yang tidak baik dapat menciptakan siswa tidak terbebas dari pengaruh narkoba dan dapat meningkatkan jumlah pengguna narkoba di kalangan remaja. Penyalahgunaan narkoba terutama di kalangan pelajar, pada umumnya dilakukan atau diawali dengan coba-coba, lalu ketagihan. Remaja biasanya mencoba memakai narkoba dengan anggapan Narkoba itu mengasikan. Selain itu, di masa remaja yang labil biasanya para remaja membutuhkan tempat untuk mencurahkan masalah mereka. Ketika para remaja tidak memiliki hal tersebut, maka narkoba di jadikan solusi. Terjadinya penyalahgunaan narkoba sebagian besar dimulai sejak usia remaja, karena remaja paling mudah dipengaruhi oleh teman sebayanya termasuk dalam penggunaan narkoba. Para remaja melihat hal tersebut sebagai *trend* bagi remaja. Sementara upaya pencegahan yang dilakukan orang tua dengan menasehati cenderung tidak didengarkan oleh kalangan remaja.

Untuk mengetahui jenis narkoba yang digunakan berdasarkan gejala ataupun ciri-ciri dari pengguna narkoba, maka diperlukan suatu sistem pakar (*expert system*) yang berfungsi untuk menggantikan peranan pihak kepolisian dalam menangani kenakalan remaja yang dilatarbelakangi oleh penyalahgunaan narkoba. Sistem pakar tersebut dapat mengetahui ciri-ciri dan jenis narkoba yang digunakan oleh pengguna narkoba. Sistem pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Ide dasarnya adalah kepakaran ditransfer dari seorang pakar ke komputer, pengetahuan yang ada disimpan dalam komputer, dan pengguna dapat berkonsultasi pada komputer itu untuk suatu nasehat, lalu komputer dapat mengambil inferensi atau menyimpulkannya, seperti layaknya seorang pakar, kemudian menjelaskannya ke pengguna tersebut, sistem pakar terkadang lebih baik cara kerjanya dari pada seorang pakar manusia.

Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli khususnya peranan pihak kepolisian. Dengan sistem pakar ini, orang awampun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli seperti halnya pihak kepolisian.

1.2 Rumusan Masalah

Untuk dapat mendiagnosa pengguna narkoba melalui sistem pakar maka rumusan masalahnya adalah “Bagaimana merancang dan membangun sebuah Sistem Pakar pendeteksi jenis narkoba yang digunakan pecandu. Dengan Menggunakan Program Visual Prolog”.

1.3 Batasan Masalah

Agar pembahasan penelitian ini tidak menyimpang dari apa yang telah dirumuskan, maka diperlukan batasan-batasan. Batasan-batasan dalam penelitian ini adalah :

1. Program digunakan hanya untuk mendeteksi jenis pengguna narkoba
2. Tidak memberikan solusi berupa obat-obatan dari pengguna narkoba.
3. Gejala yang dibahas adalah gejala yang dialami pengguna narkoba.
4. Perancangan program aplikasi system pakar menggunakan bahasa pemrograman *Visual Prolog*

1.4 Tujuan dan Manfaat

Tujuan yang ingin dicapai adalah :

1. Membangun suatu sistem pakar deteksi jenis narkoba yang digunakan pecandu, yang dapat membantu dan mempermudah proses mendeteksi pecandu narkoba tanpa harus pergi ke dokter langsung.
2. Memberikan informasi deteksi candu narkoba.
3. Menerapkan metode system pakar visual prolog.

BAB II

LANDASAN TEORI

2.1 Fungsi-Fungsi Visual Prolog yang Digunakan

Terdapat beberapa fungsi yang digunakan dalam program sistem pakar mendeteksi pecandu narkoba

2.1.1 Lingkungan Visual Prolog

1. Visual Prolog Environment (VPE)

VPE didesain agar seorang programmer dapat dengan mudah, nyaman dan cepat dalam membangun, menguji dan memodifikasi suatu aplikasi atau program yang ditulis dalam Visual Prolog. VPE memiliki beberapa varian sehingga mendukung untuk digunakan di beberapa platform sistem operasi seperti MS-DOS, MS Windows 3.1, MS Windows 95, MS Windows NT, Win-OS/2, atau OS/2 PM, yang digunakan pada platform prosesor 16 bit ataupun prosesor 32 bit dari keluarga prosesor Intel 80x86 ataupun kompatibelnya seperti prosesor AMD

Pengguna VPE diasumsikan mempunyai pengalaman dan pengetahuan dalam menggunakan sistem GUI (*Graphical User Interface*), seperti menggunakan menu, menutup, meminimize, memaximize ataupun mereseize suatu jendela (*window*), meload file dari jendela File Open dialog, mengklik toolbar dan lainlain. Jika praktikan belum punya pengetahuan ini, praktikan dapat mempelajari dari literature yang berkaitan dengan sistem operasi terkait, seperti buku cara menggunakan MS Windows 95 atau yang lainnya.

2. Menjalankan VPE

Sebelum menjalankan VPE, tentu saja, diasumsikan program VPE sudah terinstall di komputer yang digunakan. Program instalasi akan membuat sebuah program group yang di dalamnya terdapat icon yang digunakan untuk menjalankan VPE, yaitu dengan cara mengklik icon tersebut. Namun ada banyak cara untuk menjalankan VPE, seperti, dengan menggunakan Windows Explorer, men-double klik file VIP.EXE di direktori BIN\WIN\16 untuk platform 16 bit atau direktori BIN\WIN\32 untuk platform 32 bit yang direktori tersebut terletak di bawah direktori utama VIP. Jika Visual Prolog telah pernah membuka suatu project (dengan ekstensi .VPR) terakhir kali dan VPE ditutup, maka secara otomatis akan membuka project tersebut ketika VPE dijalankan kembali.

3. Membuka jendela editor (*editor window*)

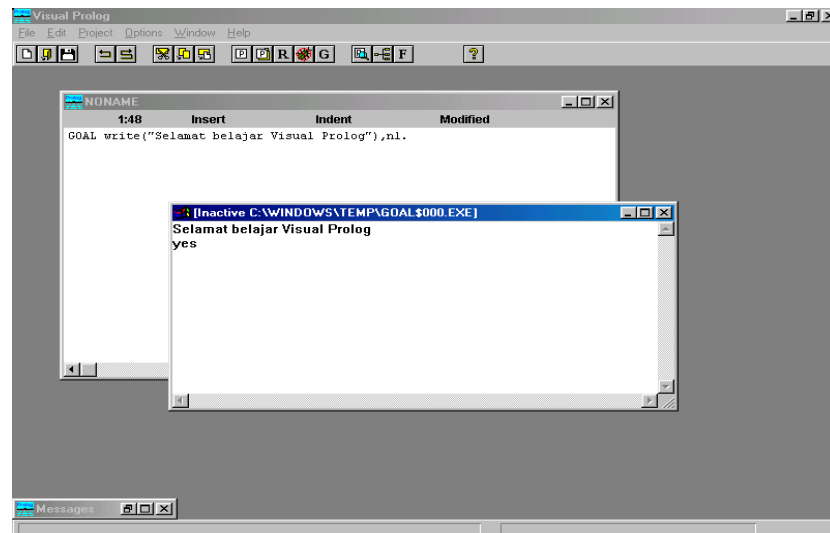
Untuk menciptakan jendela editor yang baru, praktikan dapat menggunakan menu perintah **File | New**. Setelah itu akan muncul jendela editor baru dengan judul “NONAME”. Editor ini layaknya seperti editor teks standar lainnya, seperti NOTEPAD yang dimiliki oleh MS Windows. Praktikan dapat menggunakan tombol kursor (tanda panah atas, bawah, kiri dan kanan) dan mouse untuk menggerakkan kursor seperti layaknya editor lain. Juga editor ini mendukung operasi *cut*, *copy* and *paste*, *undo/redo* yang diaktifkan dari menu **Edit** atau melalui penekanan tombol akselerator yang dapat dilihat pada menu **Edit**. Untuk lebih mendalami editor ini lebih jauh praktikan dapat membaca dari manual Visual Development Environment (VDE) pada file VDE.DOC di direktori DOC yang terletak di bawah direktori utama VIP.

4. Menjalankan dan menguji suatu program

Untuk mengecek bahwa sistem diset dengan baik, praktikan dapat mencoba mengetikkan teks berikut pada jendela editor:

GOAL write("Selamat belajar Visual Prolog"),nl.

Baris kode di atas pada Prolog dinamakan GOAL dan baris tersebut telah cukup syarat untuk menjadi program yang bisa dieksekusi. Untuk mengeksekusi GOAL, aktifkan item menu **Project | Test Goal**, atau cukup dengan menekan tombol akselerator **Ctrl+G**. Jika sistem terinstall dengan baik, maka di layar akan tampak seperti gambar berikut:



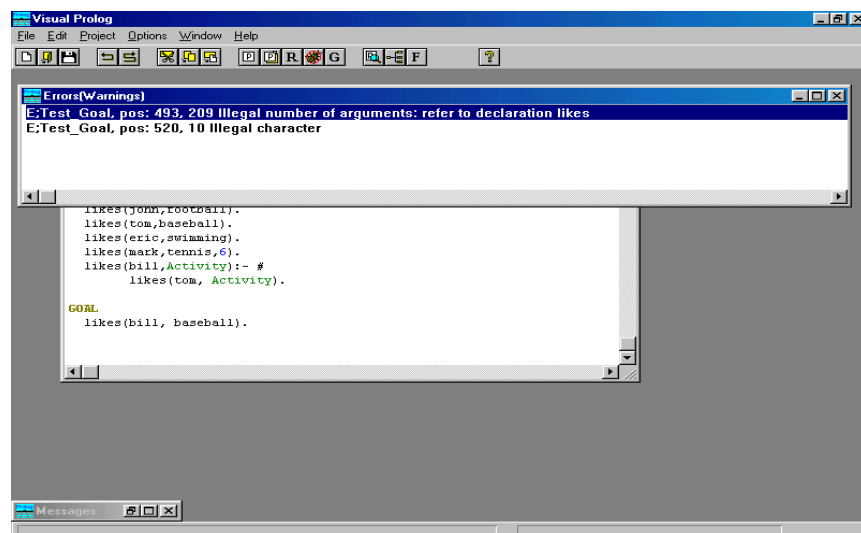
Hasil eksekusi akan ditampilkan pada jendela yang berbeda yang harus ditutup sebelum menguji GOAL lainnya. Visual Prolog memperlakukan GOAL sebagai sebuah program yang telah ter-*compile*, kemudian di-*links* dan dibangkitkan menjadi suatu bentuk jendela yang dapat dieksekusi (*executable*).

5. Membuka file dari disk

Contoh-contoh program untuk mendukung manual Language Tutorial Visual Prolog disediakan di direktori /DOC/EXAMPLES pada direktori utama VIP (dengan syarat manual beserta contohnya terinstall pada waktu menginstall Visual Prolog pertama kali). Salah satu file contoh tersebut dapat dibuka dengan menggunakan item menu **File | Open** atau menekan tombol akselerator **F8**. Pilih salah satu file (berekstensi .PRO) dan uji GOAL program tersebut dengan menekan tombol **Ctrl+G**.

6. Melihat dan memperbaiki kesalahan

Jika programmer membuat kesalahan dalam menuliskan kode Visual Prolog, maka VPE akan menampilkan jendela kesalahan (*error window*) di mana pada jendela tersebut terdapat daftar kesalahan yang dibuat. Programmer dapat mendouble klik kesalahan tersebut agar kursor beralih ke posisi di mana kesalahan tersebut dibuat pada baris kode di jendela editor, sehingga dengan lebih cepat kesalahan tersebut dapat diperbaiki. Tampilan jendela kesalahan dapat dilihat pada gambar berikut:



2.1.2 Dasar-dasar Prolog

Pemrograman Logika

Prolog dikenal sebagai bahasa deskriptif (*descriptive language*), yang berarti dengan diberikan serangkaian fakta-fakta dan aturan-aturan, Prolog, dengan menggunakan cara berpikir deduktif (*deductive reasoning*), akan dapat menyelesaikan permasalahan suatu program. Ini dikontraskan dengan bahasa komputer tradisional seperti C, BASIC, Pascal yang dikenal sebagai bahasa prosedural (*procedural language*). Dalam bahasa prosedural, programmer harus memberikan instruksi tahap demi tahap agar komputer dapat dengan pasti bagaimana menyelesaikan permasalahan yang diberikan. Dengan kata lain, programmer harus tahu lebih dahulu bagaimana cara menyelesaikan permasalahan sebelum diinstruksikan ke komputer. Lain jika dibandingkan dengan programmer Prolog. Programmer Prolog hanya membutuhkan deskripsi/gambaran permasalahan, lalu menerjemahkannya ke bahasa Prolog. Selanjutnya tinggal sistem Prolog yang menentukan bagaimana mencari solusinya.

Prolog didasarkan pada klausa-klausa Horn (*Horn clauses*), yang merupakan himpunan bagian dari sistem formal yang dinamakan logika predikat (*predicate logic*). Logika predikat menyederhanakan cara agar jelas bagaimana berpikir akan dilakukan. Prolog menggunakan variasi sintak logika predikat yang telah disederhanakan dengan demikian sintaknya mudah dimengerti dan sangat mirip dengan bahasa natural.

Prolog mempunyai mesin inferensi (*inference engine*) yang merupakan suatu proses berpikir logis mengenai informasi. Mesin inferensi mempunyai pencocok pola (*pattern matcher*) yang akan mengambil informasi yang telah disimpan (diketahui) dan kemudian mencocokkan jawaban atas pertanyaan. Satu *feature* penting dari Prolog adalah bahwa,

sehubungan mencari jawaban logis atas pertanyaan yang diajukan, ia dapat berhubungan dengan banyak alternatif dan mencari semua kemungkinan dari pada hanya satu solusi.

Logika predikat dibangun agar mudah menerjemahkan ide-ide berbasis logika menjadi bentuk tertulis. Prolog mengambil keunggulan dari sintak ini untuk membangun suatu bahasa pemrograman yang berbasis logika. Dalam logika predikat, pertama kali harus membuang semua kata-kata yang tidak dibutuhkan dari suatu kalimat. Kemudian mentransformasi kalimat tersebut dengan mencari relasi terlebih dahulu, kemudian setelah itu melakukan pengelompokkan *object*. *Object* kemudian menjadi argumen dari relasi atas *object* tersebut. Berikut ini merupakan contoh kalimat bahasa natural ditransformasikan menjadi sintak logika predikat:

Bahasa Natural	Logika Predikat
Eskrim rasanya enak	enak(eskrim)
Mawar berwarna merah	merah(mawar)
Miftah menyukai mobil	suka(miftah,mobil)
Wening suka bakso jika bakso rasanya enak	suka(wening, bakso) if enak(bakso)

a. Kalimat : Fakta dan Aturan

Programmer Prolog mendefinisikan *object-object* dan relasi-relasi, kemudian mendefinisikan aturan mengenai kapan relasi-relasi ini dikatakan benar (*true*). Contoh kalimat :

Wening suka bakso.

Memperlihatkan relasi antara *object* Wening dan bakso, relasi ini adalah suka. Berikutnya membuat aturan kapan Wening suka bakso adalah kalimat yang benar:

Wening suka bakso jika bakso rasanya enak.

Fakta : Apa yang diketahui

Dalam Prolog, relasi antara *object-object* dinamakan **predikat**. Dalam bahasa natural relasi disimbolisasikan oleh suatu kalimat. Dalam logika predikat yang Prolog gunakan, suatu relasi adalah kesimpulan dalam suatu frase sederhana. Suatu fakta memiliki nama relasi diikuti *object* atau *object-object* di dalam tanda kurung. Sebagaimana kalimat, fakta tersebut diakhiri dengan tanda titik (.). Berikut ini beberapa fakta yang mengekspresikan relasi *suka* dalam bahasa natural:

Toni suka Inung.

Inung suka Toni.

Toni suka kucing.

Dan dalam sintak Prolog, fakta di atas ditulis:

suka(toni,inung).

suka(inung,toni).

suka(toni,kucing).

Fakta juga bisa mengekspresikan sifat (*property*) dari suatu *object* sebagaimana suatu relasi. Dalam bahasa natural kalimat “Kodok berwarna hijau” dan “Komeng adalah laki-laki” diubah ke dalam sintak Prolog menjadi:

hijau(kodok).

lakilaki(komeng).

Aturan: Apa yang dapat disimpulkan dari fakta yang ada

Aturan membuat kita dapat mengambil suatu fakta dari fakta yang lain. Dengan bahasa lain, aturan adalah suatu

konklusi diketahui benar jika satu atau lebih konklusi atau fakta lain ditemukan benar. Berikut ini beberapa aturan yang berhubungan dengan relasi suka:

Inung suka apapun yang Toni suka.

Komeng suka apapun yang berwarna hijau.

Dengan aturan di atas, dapat disimpulkan dari beberapa fakta sebelumnya mengenai sesuatu yang Inung dan Komeng suka:

Inung suka Inung.

Komeng suka Kodok.

Untuk menuliskan aturan tersebut dalam sintak Prolog hanya dibutuhkan sedikit perubahan seperti ini:

suka(inung, Sesuatu):- suka(toni, Sesuatu).

suka(komeng, Sesuatu):- hijau(Sesuatu).

Simbol :- diucapkan dengan “jika” dan memisahkan dua bagian dari aturan yaitu *head* dan *body*. Aturan di atas dalam makna prosedur, bisa berarti “untuk membuktikan Inung suka sesuatu, buktikanlah bahwa Toni suka hal yang sama” dan “untuk membuktikan Komeng suka sesuatu, buktikanlah bahwa sesuatu itu berwarna hijau”.

b. Query

Kalau kita sudah memberikan Prolog sekumpulan fakta, selanjutnya dapat diajukan pertanyaan sehubungan dengan fakta-fakta. Ini dikenal dengan nama memberikan query ke sistem Prolog

(*querying the Prolog system*). Pertanyaan yang diajukan ke Prolog sama tipenya seperti dalam bahasa natural. Berdasarkan pada fakta dan aturan yang diberikan sebelumnya.

Sintak Prolog tidak berubah ketika mengajukan query dan tampak bahwa sintak query sangat mirip dengan sintak fakta. Yang penting untuk diingat adalah bahwa *object* kedua Apa diawali dengan huruf besar sedang *object* pertama *toni* tidak. Ini karena *toni* sudah *fix* sebagai konstanta *object* sedangkan Apa adalah sebuah variabel. Variabel selalu diawali dengan sebuah huruf besar atau sebuah garis bawah (*underscore*, *_*).

Prolog selalu mencari jawaban dari suatu query dengan memulai dari bagian paling atas (*top*) dari fakta-fakta dan melihat setiap fakta sampai ke bagian paling bawah (*bottom*) dimana tidak ada satupun fakta lagi.

c. Variabel

Seperti yang telah disebutkan sebelumnya, untuk memberikan nama variabel dalam Visual Prolog harus diawali dengan huruf besar (*capital letter*) atau garis bawah (*underscore*), berikutnya dapat berupa huruf (besar atau kecil), angka (“0-9”) dan garis bawah (“_”). Berikut ini merupakan contoh penamaan

d. Komentar

Programmer yang baik selalu memberikan catatan atau komentar untuk menjelaskan sesuatu yang mungkin tidak jelas bagi orang lain (atau bahkan bagi programmer sendiri dalam setahun ke depan misalnya). Komentar akan membuat program menjadi lebih mudah untuk dimengerti. Komentar dengan baris jamak harus

dimulai dengan karakter `/*` (*slash*, *asterik*) dan diakhiri dengan karakter `*/` (*asterik*, *slash*). Untuk memberikan komentar dengan baris tunggal dapat menggunakan karakter yang sama atau dapat dimulai dengan tanda persen (%). Contoh:

```
/* Ini contoh sebuah komentar */
```

```
% Ini juga contoh komentar
```

```
/* Ini komentar
```

```
dengan
```

```
tiga baris */
```

```
/* Ini contoh komentar /* dalam komentar */ di Visual Prolog */
```

2.1.3 Program Visual Prolog

1. Section dasar Visual Prolog

Secara umum, program Visual Prolog terdiri dari empat section dasar, yaitu section **clauses**, section **predicates**, section **domains**, dan terakhir section **goal**. Berikut akan dijelaskan secara singkat masing-masing section tersebut.

- **Section Clauses**

Section clauses merupakan section yang paling penting pada program Visual Prolog. Pada section inilah kita meletakkan fakta dan aturan. Ketika mencari jawaban, Visual Prolog akan mencari dari bagian paling atas dari section clauses, melihat setiap fakta dan aturan untuk mendapat jawaban benar, hingga ke bagian paling bawah dari section ini.

- **Section Predicates**

Sebelum mendefinisikan predikat di section clauses, maka predikat tersebut harus dideklarasikan terlebih dahulu di section predicates. Kalau tidak, Visual Prolog tidak akan mengenal predikat yang kita tuliskan tersebut. Ketika mendeklarasikan suatu predikat, kita memberitahu Visual Prolog domain dari argumen yang dimiliki predikat tersebut.

Visual Prolog mempunyai perpustakaan predikat yang kalau dipakai tidak perlu dideklarasikan, karena sudah *built-in*. Untuk melihat predikat apa saja serta manfaatnya yang ada di perpustakaan Visual Prolog dapat melihat *help* dari Visual Prolog (**Help | Contents**).

- **Section Domains**

Section domains mempunyai 2 manfaat utama, yaitu pertama, kita dapat memberikan nama yang berarti untuk domain, walaupun secara internal domain tersebut sama tipenya dengan domain yang telah ada; yang kedua, kita dapat mendeklarasi domain khusus yang digunakan untuk mendeklarasikan struktur data yang tidak didefinisikan oleh standar domain. Dengan mendeklarasikan domain juga dapat mencegah kesalahan logika pada program.

- **Section Goal**

Secara esensial, section **goal** sama dengan *body* dari sebuah aturan (*rule*), yaitu sederetan sub-sub goal. Perbedaan antara section **goal** dengan suatu aturan adalah setelah kata kunci **goal** tidak diikuti tanda :- dan Visual Prolog secara otomatis mengeksekusi goal ketika program dijalankan.

2. Section Program Lainnya.

Ada beberapa section lainnya yang digunakan di Visual Prolog yaitu section **facts**, section **constants**, dan section **global**. Kali ini akan dibahas secara singkat ketiga section tersebut sebagai pengenalan. Section **facts** akan dibahas lebih mendalam pada modul yang lain.

- **Section Facts**

Program Visual Prolog merupakan suatu koleksi dari sekumpulan fakta dan aturan. Kadang ketika program sedang berjalan, kita ingin meng-*update* (merubah, menambah, atau menghapus) beberapa fakta dari program. Pada kasus ini fakta menjadi suatu database yang dinamis atau database internal, dan fakta tersebut dapat berubah ketika program sedang berjalan. Visual Prolog menyediakan section khusus untuk mendeklarasikan fakta di program yang menjadi bagian dari database dinamis, yaitu section **facts**.

Kata kunci **facts** untuk memulai section **facts**. Visual Prolog menyediakan sejumlah predikat *built-in* yang mempermudah penggunaan section fakta dinamis ini. Kata kunci **facts** dapat digantikan dengan kata kunci **database**, untuk maksud yang sama.

- **Section Constants**

Konstanta simbolis dapat digunakan di program Visual Prolog. Untuk itu sebelumnya harus dideklarasikan terlebih dahulu. Deklarasi konstanta dilakukan pada section **constants**, diikuti dengan deklarasi menggunakan sintak :

<id> = <Macro_definition>

<id> adalah nama dari konstanta simbolis dan <Macro_definition> adalah apa yang akan diisikan (assign) kedalam konstanta. Setiap <Macro_definition> diakhiri dengan baris baru. Dengan demikian hanya ada satu deklarasi konstanta pada tiap barisnya.

- **Section Global**

Visual Prolog memperbolehkan untuk mendeklarasikan beberapa domain, predikat dan klausa menjadi global (daripada hanya lokal). Caranya dengan menset secara terpisah section **global domains**, **global predicates**, dan **global facts** pada bagian paling atas dari program. Modul ini bukan tempatnya untuk membahas secara detail mengenai section **global**.

3. *Compiler Directives*

Visual Prolog mendukung *compiler directives* yang dapat ditambahkan ke badan program untuk memberitahukan ke komputer bagaimana secara spesifik memperlakukan kode-kode waktu di-*compile*. Untuk menset *compiler directives* sebagian besar dilakukan melalui menu **Options | Project | Compiler Options**. Modul ini tidak akan membahas secara detil mengenai compiler directives, namun akan memperkenalkan salah satu diantaranya yaitu include directive.

Kalau sudah terbiasa membuat program menggunakan Visual Prolog, seringkali kita memakai suatu prosedur tertentu berulang kali, sehingga setiap kali membuat program baru dan menggunakan prosedur tersebut, prosedur tersebut harus diketikkan kembali. Untuk menghemat waktu, maka dapat digunakan include.

4. Aritas jamak (*multiple arity*)

Aritas (*arity*) suatu predikat adalah jumlah argumen yang ada pada predikat tersebut. Visual Prolog memperbolehkan kita mempunyai 2 atau lebih predikat dengan nama yang sama namun dengan aritas yang berbeda. Aritas yang berbeda dari nama predikat yang sama harus dikelompokkan bersama baik pada section **predicates** maupun pada section **clauses**. Perbedaan aritas oleh Visual Prolog akan diperlakukan secara berbeda pula.

5. Sintak Aturan (*Rule Syntax*)

Rule pada Prolog adalah ketika kebenaran sebuah fakta tergantung pada kesuksesan (kebenaran) dari satu atau lebih fakta yang lain. Seperti yang telah dijelaskan pada modul sebelumnya aturan terdiri dari 2 bagian yaitu *head* dan *body*. Berikut ini merupakan aturan generik penulisan sintak *rule* pada Visual Prolog:

HEAD:- <subgoal>, <subgoal>, ..., <subgoal>.

Bagian *body* dari *rule* terdiri dari satu atau lebih subgoal. Setiap subgoal dipisahkan oleh koma, menspesifikasikan konjungsi, dan subgoal terakhir diakhiri dengan tanda titik.

Untuk membuat suatu *Rule* dikatakan sukses (benar), Prolog harus memeriksa kebenaran dari setiap subgoal yang ada pada aturan tersebut. Jika ada subgoal yang gagal (salah), Prolog akan kembali ke atas dan mencari alternatif bagi subgoal yang paling awal, kemudian kembali memproses dengan nilai variabel yang berbeda. Ini dinamakan lacakbalik (*backtracking*). Penjelasan yang lebih rinci mengenai lacakbalik akan diberikan pada modul yang lain.

6. Konversi Tipe Otomatis (*Automatic Type Conversions*)

Ketika Visual Prolog mencocokkan 2 variabel, keduanya tidak selalu berasal dari domain yang sama. Juga kadang variabel diikat (*bound*) menjadi konstan dari domain lain. Percampuran domain ini diperbolehkan karena Visual Prolog melakukan konversi tipe otomatis dengan syarat konversi bisa terjadi bila:

- Antara **strings** dan **symbols**.
- Antara semua domain **integer** dan juga **real**. Ketika suatu karakter (**char**) dikonversikan ke nilai numeris, angka nilai ASCII dari karakter tersebut yang digunakan.

2.1.4 Unifikasi Dan Lacakbalik (*Unification And Backtracking*)

1. Unifikasi (*Unification*)

Pada waktu Visual Prolog mencoba untuk mencocokkan suatu panggilan (dari sebuah subgoal) ke klausa (pada section **clauses**), maka proses tersebut melibatkan suatu prosedur yang dikenal dengan unifikasi (*unification*), yang mana berusaha untuk mencocokkan antara struktur data yang ada di panggilan (subgoal) dengan klausa yang diberikan. Unifikasi pada Prolog mengimplementasikan beberapa prosedur yang juga dilakukan oleh beberapa bahasa tradisional seperti melewati parameter, menyeleksi tipe data, membangun struktur, mengakses struktur dan pemberian nilai (*assignment*). Pada intinya unifikasi adalah proses untuk mencocokkan dua predikat dan memberikan nilai pada variabel yang bebas untuk membuat kedua predikat tersebut identik. Mekanisme ini diperlukan agar Prolog dapat mengidentifikasi klausa-klausa mana yang dipanggil dan mengikat (*bind*) nilai klausa tersebut ke variabel.

2. Lacakbalik (*Backtracking*)

Pada waktu menyelesaikan masalah, seringkali, seseorang harus menelusuri suatu jalur untuk mendapatkan konklusi yang logis. Jika konklusi ini tidak memberikan jawaban yang dicari, orang tersebut harus memilih jalur yang lain. Perhatikan permainan *maze* berikut. Untuk mencari jalan keluar dari *maze*, seseorang harus selalu mencoba jalur sebelah kiri terlebih dahulu pada setiap percabangan hingga menemukan jalan buntu. Ketika menemukan jalan buntu maka orang tersebut harus kembali ke percabangan terakhir (*back-up*) untuk mencoba lagi (*try again*) ke jalur kanan dan jika menemukan percabangan lagi maka tetap harus mencoba jalur kiri terlebih dahulu. Jalur kanan hanya akan sekali-sekali dipilih. Dengan begitu orang tersebut akan bisa keluar dari *maze*, dan memenangkan permainan. Metode balik-ke-atas-dan-coba-lagi (*backing-up-and-trying-again*) ini pada Visual Prolog disebut lacakbalik (*backtracking*). Visual Prolog menggunakan metode ini untuk menemukan suatu solusi dari permasalahan yang diberikan. Visual Prolog dalam memulai mencari solusi suatu permasalahan (atau goal) harus membuat keputusan diantara kemungkinan-kemungkinan yang ada. Ia menandai di setiap percabangan (dikenal dengan titik lacak balik) dan memilih subgoal pertama untuk telusuri. Jika subgoal tersebut gagal (ekivalen dengan menemukan jalan buntu), Visual Prolog akan lacakbalik ke titik lacakbalik (*back-tracking point*) terakhir dan mencoba alternative subgoal yang lain.

3. Pengendalian proses lacakbalik

Mekanisme lacak balik dapat menghasilkan pencarian yang tidak perlu, akibatnya program menjadi tidak efisien. Misalnya adanya beberapa jawaban yang muncul ketika kita hanya membutuhkan solusi tunggal dari masalah yang diberikan. Pada kasus lain, suatu kebutuhan untuk memaksa Visual Prolog untuk melanjutkan mencari jawaban tambahan walaupun goal tersebut sudah terpenuhi. Untuk kasus-kasus tersebut, kita harus mengontrol proses lacakbalik.

Visual Prolog menyediakan 2 alat yang memperbolehkan kita untuk mengendalikan mekanisme lacak balik yaitu predikat *fail* yang digunakan untuk memaksa lacakbalik dan predikat *cut* (ditandai dengan **!**) yang digunakan untuk mencegah lacakbalik.

Predikat *fail*

Visual Prolog akan memulai lacakbalik jika ada panggilan yang gagal. Pada situasi tertentu, ada kebutuhan untuk memaksa lacakbalik dalam rangka mencari alternative solusi. Visual Prolog menyediakan predikat khusus *fail* untuk memaksa kegagalan sehingga memicu terjadinya lacakbalik. Efek dari *fail* sama dengan efek memberikan perbandingan $2=3$ atau subgoal yang tidak mungkin (*impossible*) lainnya.

Predikat *Cut* (!)

Visual Prolog memiliki *cut* yang digunakan untuk mencegah lacakbalik, ditulis berupa sebuah tanda seru (!). Efek dari *cut* adalah sederhana, yaitu tidak akan memungkinkan terjadinya lacakbalik melewati sebuah *cut*. Kita menempatkan *cut* dalam program sama persis seperti menempatkan sebuah subgoal pada *body* dari suatu *rule*.

Ketika proses melewati *cut*, pemanggil ke *cut* dinyatakan sukses dan subgoal berikutnya (jika ada) dipanggil. Sekali sebuah *cut* dilewati, adalah menjadi tidak mungkin untuk melakukan lacakbalik pada subgoal yang berada pada sebelum *cut* pada klausa yang sedang diproses dan adalah menjadi tidak mungkin untuk melakukan lacakbalik ke predikat lain yang mendefinisikan predikat yang sekarang diproses (predikat yang mengandung *cut*).

4. Predikat *Not*

Program berikut memperlihatkan bagaimana penggunaan predikat *not* untuk mengidentifikasi seorang mahasiswa teladan yaitu mahasiswa yang mempunyai Indeks Prestasi Kumulatif (IPK) minimal 3,5 dan tidak sedang dalam masa percobaan (sedang menjalani masa hukuman karena melakukan tindak kejahatan).

Ada satu catatan ketika menggunakan *not* yaitu predikat *not* akan sukses ketika subgoal tidak bisa dibuktikan kebenarannya. Hal ini untuk mencegah suatu situasi variabel yang belum diikat akan diikat menggunakan *not*. Jika subgoal dengan variable bebas dipanggil melalui *not*, maka Visual Prolog akan mengeluarkan pesan kesalahan *Free variables not allowed in 'not' or 'retractall'*.

2.1.5 Data Object Sederhana Dan Jamak (Simple Dan Compound Data Object)

1. Data *Object* sederhana

Data *object* sederhana terdiri dari 2 yaitu variabel atau konstanta. Konstanta yang dimaksud tidak sama dengan konstanta simbolis yang ditulis di *section constants* pada bagian program. Yang dimaksud dengan konstanta di sini adalah apapun yang diidentifikasi sebagai sebuah *object* bukan *subject* yang nilainya bisa bervariasi, seperti sebuah karakter (**char**), angka (integer atau real) atau sebuah **atom** (symbol atau string).

A. Variabel

Variable harus dimulai dengan sebuah huruf kapital (A ..Z) atau sebuah *underscore* (_). Sebuah *underscore* tunggal merepresentasikan sebuah variable anonim. Variabel dalam prolog bersifat lokal bukan global, oleh karena itu jika ada dua klausa yang mengandung sebuah variabel X maka X pada kedua klausa tersebut adalah variabel yang berbeda.

B. Konstanta

Konstanta meliputi karakter, angka, dan atom. Suatu nilai konstanta juga merupakan nama dari konstanta tersebut. Konstanta 2 merepresentasikan angka 2 dan konstanta halo merepresentasikan simbol halo.

- **Karakter**

Karakter bertipe **char**, yaitu karakter-karakter yang bisa tercetak (ASCII 32 – 127), karakter angka (0 – 9), huruf kecil (a – z), huruf kapital (A – Z) dan tanda baca. Konstanta karakter ditulis dengan diapit oleh tanda petik tunggal (‘). Contoh : ‘a’, ‘*’, ‘{’, ‘W’. Jika kita menginginkan sebuah backslash atau petik tunggal menjadi karakter menuliskannya harus didahului dengan

sebuah tanda backslash. Contoh : ‘\’ (backslash), ‘\’ (tanda petik tunggal). Beberapa karakter mempunyai fungsi khusus ketika didahului oleh karakter escape (\). Contoh : ‘\n’ (ganti baris), ‘\t’ (tabulasi). Konstanta karakter dapat juga ditulis berdasarkan kode ASCII-nya, dengan didahului backslash. Contoh : ‘\64’ (@), ‘\90’ (Z)

- **Angka**

Angka merupakan salah satu dari integer atau real.

- **Atom**

Sebuah atom terdiri dari sebuah simbol atau sebuah string. Perbedaan antara simbol dan string berkaitan dengan representasi dan implementasi mesin. Visual Prolog melakukan konversi tipe otomatis diantara domain string dan domain simbol. Jadi simbol dapat digunakan untuk domain string dan string dapat digunakan untuk domain simbol. Namun string dinyatakan sebagai sesuatu yang berada diantara tanda petik ganda sedang simbol tidak membutuhkan tanda petik ganda. Simbol dimulai dengan sebuah huruf kecil dan hanya boleh berisikan huruf, angka, dan *underscore*. String adalah sesuatu yang diapit tanda petik ganda dan berisikan kombinasi dari karakter, kecuali ASCII NULL (0), yang dipakai untuk menandai akhir dari string.

2. Data object jamak

Data object jamak memperlakukan beberapa informasi sebagai sebuah item tunggal. Contohnya : tanggal 1 desember 1999. Tanggal tersebut terdiri dari 3 jenis informasi yaitu hari, bulan dan tahun.

Penulisan ini mirip dengan penulisan suatu fakta tetapi ini bukan fakta. Ini adalah sebuah data object. Data object dimulai dengan sebuah nama yang biasa disebut functor (dalam contoh adalah tanggal) yang diikuti oleh 3 argumen. Sebuah functor dalam Visual Prolog tidak seperti sebuah fungsi pada bahasa pemrograman lain. Functor tidak melakukan apa-apa, hanya sebuah nama yang mengidentifikasi sebuah jenis data object jamak yang didalamnya terdapat argumen. Argumen dari sebuah data object jamak bisa dalam bentuk data object jamak pula.

3. Deklarasi Domain-Campuran Jamak (*Compound Mix-Domain*)

Deklarasi domain-campuran jamak bermaksud :

- memiliki sebuah argumen dengan kemungkinan lebih dari satu tipe argumen;
- memiliki beberapa macam argumen, masing-masing dengan tipe yang berbeda;
- memiliki beberapa macam argumen, beberapa diantaranya dengan kemungkinan lebih dari satu tipe argumen.

Agar suatu predikat Visual Prolog dapat menerima suatu argumen yang memberikan informasi dengan tipe yang berbeda maka functor tersebut harus dideklarasikan.

2.1.6 Perulangan Dan Rekursi (Repetition And Recursion)

A. Perulangan dan Rekursi

Komputer memiliki bermacam kemampuan yang berguna salah satunya adalah kemampuan melakukan sesuatu berulang-ulang. Prolog dapat melakukan perulangan dalam dua hal yaitu berupa prosedur dan struktur data. Ide dari struktur data repetitif (rekursif) adalah bagaimana menciptakan struktur data yang ukuran (*size*) akhirnya belum diketahui ketika struktur tersebut pertama kali dibuat (*create*).

1. Proses Perulangan

Prolog menyediakan dua jenis perulangan yaitu lacakbalik (mencari jawaban jamak dari satu pertanyaan) dan rekursi (prosedur pemanggilan dirinya sendiri).

- **Lacakbalik**

Ketika suatu prosedur melakukan lacakbalik, prosedur akan mencari alternatif jawaban dari sebuah goal yang sudah terpenuhi. Lacakbalik merupakan salah satu cara untuk melakukan proses perulangan.

- **Implementasi Lacakbalik dengan Loop**

Lacakbalik merupakan cara yang baik untuk mencari alternatif jawaban dari sebuah goal. Namun jika suatu goal tidak memiliki alternatif jawaban, lacakbalik masih dapat digunakan untuk melakukan perulangan. Berikut ini didefinisikan predikat duaklausa.

ulang.

ulang:-ulang.

Ini untuk mengakali struktur kendali Prolog agar berpikir bahwa terdapat sejumlah jawaban berbeda yang tak terbatas (cara kerjanya akan dibahas pada bagian mengenai rekursi ekor / *tail recursion*). Kegunaan ulang adalah agar lacakbalik terjadinya tak terhingga).

- **Rekursi**

Cara lain untuk melakukan perulangan adalah melalui rekursi. Prosedur rekursi adalah prosedur yang di dalamnya ada pemanggilan terhadap dirinya sendiri. Prosedur rekursi dapat merekam perkembangannya karena ia melewati (*passing*) pencacah, total, dan hasil sementara sebagai argumen dari satu iterasi ke iterasi berikutnya.

- **Rekursi Ekor (*Tail Recursion*)**

Rekursi mempunyai kelemahan yaitu memakan memori. Ketika suatu prosedur memanggil dirinya, keadaan pemanggil prosedur dari eksekusi harus disimpan sehingga prosedur pemanggil dapat meresume keadaan tersebut setelah prosedur pemanggil selesai. Ini berarti jika ada suatu prosedur memanggil dirinya 100 kali, maka ada 100 keadaan dari eksekusi yang harus disimpan. Keadaan (*state*) yang disimpan tersebut dikenal dengan nama *stack frame*. Ukuran *stack* maksimum pada platform 16 bit, seperti IBM-PC dengan sistem operasi DOS, adalah 64KByte yang bisa mengandung sekitar 3000 atau 4000 *stack frame*. Pada platform 32 bit, secara teoritis bisa sampai ukuran Giga Byte.

Untuk mengatasi kelemahan tersebut, maka digunakan optimasi rekursi ekor (*tail recursion optimization*). Diumpamakan, selain memanggil prosedur C, prosedur B memanggil dirinya sendiri pada langkah terakhir. Ketika prosedur B memanggil B, *stack frame* dari pemanggilan B akan ditimpa nilainya oleh sebuah *stack frame* dari pemanggil B, jadi tidak menambah *stack frame* baru. Hanya argumen yang perlu di-*update* nilainya dan kemudian proses akan melompat ke awal prosedur. Dalam perspektif procedural adalah sama seperti memperbaharui variabel pengendali perulangan. Syarat dari rekursi ekor adalah :

- Pemanggil merupakan subgoal terakhir dari klausa tersebut.
- Tidak ada titik lacak balik sebelumnya pada klausa.

2. Struktur Data Rekursif

Tidak hanya *rule* yang bisa rekursif tapi juga struktur data. Prolog merupakan satu-satunya bahasa pemrograman yang digunakan secara luas yang memperbolehkan mendefinisikan tipe struktur data rekursif. Salah satu tipe struktu data rekursif yaitu struktur data pohon (*tree*).

2.1.7 List

Pada Prolog, yang dimaksud dengan *list* adalah sebuah *object* yang didalamnya mengandung sejumlah *object* yang lain (jumlahnya dapat berubah-ubah). *List* dalam bahasa pemrograman lain bisa disamakan dengan tipe data *pointer* (C dan Pascal). Berikut ini cara penulisan *list* pada Prolog.

```
[ 1, 2, 3 ] /* list yang mengandung integer 1, 2 dan 3 */
```

[kucing, anjing, tikus] /* list yang terdiri dari 3 buah symbol */

[“Syarif Musadek”, “Yusida Andriani”, “Diana Putri”]

/* list yang terdiri dari 3 buah string */

Tanda asterik (*) berarti domain tersebut merupakan sebuah *list*. Tanda asterik juga dipakai di bahasa C untuk pendeklarasian tipe data *pointer*. Pada Pascal pendeklarasian *pointer* menggunakan tanda ^. Elemen *list* bisa berupa apapun, termasuk suatu *list* yang lain, namun semua elemen dari suatu *list* harus berasal dari domain yang sama.

Head dan Tail (Kepala dan Ekor)

List adalah suatu data *object* jamak rekursif (*recursive compound object*). *List* terdiri dari 2 bagian yaitu *head*, yang merupakan elemen pertama dari *list* dan *tail*, elemen sisanya. *Tail* dari *list* adalah juga merupakan sebuah *list*, sedangkan *head* dari *list* merupakan sebuah elemen.

2.1.8 Section Facts

Section facts terdiri dari fakta-fakta yang mana fakta-fakta tersebut dapat ditambah dan dihapus secara langsung dari sebuah program pada saat program sedang berjalan (*at run time*). Kita dapat mendeklarasikan sebuah predikat pada *section facts* dan predikat tersebut dapat digunakan sama halnya seperti kalau dideklarasikan pada *section predicates*. Visual Prolog menyediakan beberapa predikat *built-in* untuk menangani hal yang berkaitan dengan penggunaan *section facts*, antara lain :

- `assert`, `asserta` dan `assertz` untuk menambah fakta baru pada *section facts*.
- `retract` dan `retractall` untuk menghapus fakta yang ada.
- `consult` untuk membaca fakta dari sebuah file dan menyertakan fakta tersebut ke dalam fakta internal.
- `save` menyimpan isi fakta internal ke dalam sebuah file.

➤ **Deklarasi Section Facts**

Kata kunci facts atau bisa juga database menandai permulaan sederetan deklarasi dari predikat yang ada pada section facts. Kita dapat menambahkan fakta-fakta (bukan rule) pada suatu section facts dari keyboard pada saat run time dengan menggunakan asserta dan assertz atau memanggil predikat consult untuk mengambil fakta tambahan dari sebuah file.

➤ **Menambah fakta pada saat *run time***

Pada saat *run time*, fakta-fakta dapat ditambah ke *section facts* dengan menggunakan predikat assert, asserta dan assertz atau me-load sebuah file yang berisikan fakta menggunakan predikat consult. Cara penulisannya adalah sebagai berikut:

- asserta(<fakta>[, nama_section_facts])
- assertz(<fakta>[, nama_section_facts])
- assert(<fakta>[, nama_section_facts])
- consult(namafile[, nama_section_facts])

Perbedaan assert, asserta dan assertz adalah asserta menyertakan sebuah fakta baru pada *section facts* sebelum fakta-fakta yang telah ada untuk predikat tersebut, sedangkan assertz menyertakan sebuah fakta baru setelahnya, sedangkan assert berfungsi sama seperti assertz. Sedangkan consult membaca dari sebuah file dan menyertakan fakta-fakta yang ada di file tersebut sesudah fakta-fakta yang telah ada.

Tidak seperti assertz, jika consult dipanggil hanya dengan satu argumen (tidak ada nama *section facts*) maka hanya akan menyertakan fakta-fakta yang predikatnya telah dideklarasikan di *section facts default* yaitu **dbasedom**. Jika memanggil consult dengan dua argumen (nama file dan nama *section facts*) maka hanya akan menyertakan fakta-fakta yang

predikatnya dideklarasikan pada *section facts* dengan nama yang sesuai. Jika file tersebut mengandung fakta-fakta yang bukan milik dari *section facts* tersebut, maka akan terjadi *error* pada saat membaca bagian tersebut. Perlu diperhatikan bahwa consult membaca fakta satu demi satu, jika pada file ada 10 fakta dan pada fakta ke-7 terjadi *error*, maka consult akan menyertakan 6 fakta pertama pada *section facts* kemudian menampilkan pesan kesalahan.

Sebagai catatan, consult hanya bisa membaca sebuah file dengan syarat format file tersebut sama persis dengan format file yang disimpan menggunakan predikat *save*, yaitu :

- tidak ada karakter kapital kecuali dalam tanda petik dua (penulisan string).
- tidak spasi kecuali dalam tanda petik dua.
- tidak ada komentar.
- tidak ada baris kosong.
- tidak ada symbol tanpa di dalam tanda petik dua.

➤ **Menghapus fakta pada saat *run time***

Predikat *retract* mengunifikasi suatu fakta dan menghapus fakta tersebut dari *section facts*. Cara penulisannya adalah sebagai berikut :

`retract(<fakta>[, nama_section_facts])`

retract akan menghapus fakta pertama yang cocok dengan argumen <fakta> yang diberikan. Karena *retract* merupakan predikat nondeterministik maka selama lacakbalik, *retract* akan menghapus fakta-fakta yang cocok dengan argumen <fakta>, kecuali jika fakta yang akan dihapus, predikatnya dideklarasikan deterministik. Ketika semua fakta yang cocok sudah terhapus, pemanggilan *retract* berikutnya akan gagal.

Predikat `retractall` akan menghapus semua fakta yang cocok dengan argumen `<fakta>` dan penulisannya sebagai berikut :

```
retractall(<fakta>[, nama_section_facts])
```

`retractall` berperilaku sama seperti kalau didefinisikan sebagai berikut :

```
retractall(X):- retract(X), fail. %fail untuk memaksa lacak balik  
retractall(_).
```

➤ **Menyimpan database fakta-fakta pada saat *run time***

Predikat `save` berfungsi untuk menyimpan fakta-fakta yang ada pada *section facts* ke dalam sebuah file. Cara penulisannya sebagai berikut

```
save(nama_file[, nama_section_facts])
```

Jika memanggil `save` hanya dengan satu argumen (tidak ada nama *section facts*), maka akan menyimpan fakta-fakta dari *section facts default dbasedom* ke file dengan nama yang sesuai dengan argumen. Jika memanggil `save` dengan dua argumen (nama file dan nama *section facts*), maka akan menyimpan semua fakta yang ada pada *section facts* yang sesuai ke dalam file dengan nama yang sesuai pula.

➤ **Kata kunci pada *section facts***

Fakta-fakta pada *section facts* dapat dideklarasikan dengan beberapa kata kunci opsional berikut:

`nondeterm` menentukan bahwa kemungkinan ada sejumlah fakta dari suatu

2.2 Teori Keilmuan yang Diimplementasikan

Terdapat beberapa keilmuan yang di implementasikan dalam mendeteksi pecandu narkoba. Teori keilmuan yang di implementasikan pada program kali ini yaitu :

2.2.1 Morfin

Morfin adalah hasil olahan dari opium atau candu mentah dan merupakan alkaloida yang terdapat dalam opium berupa serbuk putih. Konsumsi morfin biasanya dilakukan dengan cara dihisap atau disuntikkan. Karena morfin tergolong dalam jenis depresan, maka ia bekerja dengan cara menekan susunan syaraf pusat, menyebabkan turunnya aktifitas neuron, pusing, perubahan perasaan dan kesadaran berkalut. Konsumsi morfin secara kontinyu memiliki resiko tinggi berujung kematian.

Pecandu yang terdeteksi menggunakan jenis narkoba Morfin mengalami gangguan, antara lain yaitu :

- Pupil mata menyempit,
- Denyut nadi semakin lama melambat,
- Tekanan darah menurun,
- Suhu badan menurun,
- Produksi air seni berkurang,
- Merasa kebingungan,

2.2.2 Heroin

Heroin adalah bentuk tingkat rendah dari heroin. Heroin berasal dari bunga opium, sejenis bunga di iklim panas dan kering. Bunga tersebut menghasilkan zat lengket yang menjadi cikal bakal dari heroin, opium, morfin dan kodein. Heroin adalah zat depresan. Obat-obatan depresan tidak langsung membuat Anda merasa tertekan. Zat-zat tersebut memperlambat pesan dari otak ke tubuh dan sebaliknya. Beberapa nama lain dari zat tersebut adalah bedak, putih.

Pecandu yang terdeteksi menggunakan jenis narkoba Heroin mengalami gangguan, antara lain yaitu :

- Merasa gatal pada hidung,
- Merasa mual dan muntah-muntah,
- Pupil mata menyempit,
- Suka menyendiri,
- Tekanan darah menurun.

2.2.3 Ganja

Ganja (*Cannabis sativa* syn. *Cannabis indica*) adalah tumbuhan budidaya penghasil serat, namun lebih dikenal karena kandungan zat narkotika pada bijinya, tetrahidrokanabinol (*THC, tetra-hydro-cannabinol*) yang dapat membuat pemakainya mengalami *euforia* (rasa senang yang berkepanjangan tanpa sebab). Tanaman ganja biasanya dibuat menjadi rokok mariyuana.

Tanaman semusim ini tingginya dapat mencapai 2 meter. Berdaun menjari dengan bunga jantan dan betina ada di tanaman berbeda (berumah dua). Bunganya kecil-kecil dalam dompolan di ujung ranting. Ganja hanya tumbuh

di pegunungan tropis dengan ketinggian di atas 1.000 meter di atas permukaan laut.

Ganja menjadi simbol budaya hippies yang pernah populer di Amerika Serikat. Hal ini biasanya dilambangkan dengan daun ganja yang berbentuk khas. Selain itu ganja dan opium juga didengungkan sebagai simbol perlawanan terhadap arus globalisme yang dipaksakan negara kapitalis terhadap negara berkembang. Di India, sebagian Sadhuyang menyembah dewa Shiva menggunakan produk derivatif ganja untuk melakukan ritual penyembahan dengan cara menghisap hashish melalui pipa chlam/chillum, dan dengan meminum bhang.

Pecandu yang terdeteksi menggunakan jenis narkoba Ganja mengalami gangguan, antara lain yaitu :

- Sulit diajak berkomunikasi,
- Suka berfantasi (mengkhayal),
- Merasa denyut nadi dan jantung lebih cepat,
- Sulit mengingat.

2.2.4 Kokain

Kokain adalah zat yang adiktif yang sering disalahgunakan dan merupakan zat yang sangat berbahaya. Kokain merupakan alkaloid yang didapatkan dari tanaman belukar *Erythroxylon coca*, yang berasal dari Amerika Selatan, dimana daun dari tanaman belukar ini biasanya dikunyah-kunyah oleh penduduk setempat untuk mendapatkan efek stimulan. Saat ini Kokain masih digunakan sebagai anestetik lokal, khususnya untuk pembedahan mata, hidung dan tenggorokan, karena efek vasokonstriksinya juga membantu. Kokain diklasifikasikan sebagai suatu narkotik, bersama dengan morfin dan heroin karena efek adiktif dan efek merugikannya telah dikenali. Nama lain untuk

Kokain : *Snow, coke, girl, lady dan crack* (kokain dalam bentuk yang paling murni dan bebas basa untuk mendapatkan efek yang lebih kuat).

Pecandu yang terdeteksi menggunakan jenis narkoba Kokain mengalami gangguan, antara lain yaitu :

- Sering kejang-kejang,
- Sering mengeluarkan dahak,
- Merasa berat badan menurun,
- Merasa kebingungan.

2.2.5 Sabu-Sabu

Shabu-shabu berbentuk kristal, biasanya berwarna putih, dan dikonsumsi dengan cara membakarnya di atas aluminium foil sehingga mengalir dari ujung satu ke arah ujung yang lain. Kemudian asap yang ditimbulkannya dihirup dengan sebuah Bong (sejenis pipa yang didalamnya berisi air). Air Bong tersebut berfungsi sebagai filter karena asap tersaring pada waktu melewati air tersebut. Ada sebagian pemakai yang memilih membakar Sabu dengan pipa kaca karena takut efek jangka panjang yang mungkin ditimbulkan aluminium foil yang terhirup. Sabu sering dikeluhkan sebagai penyebab paranoid (rasa takut yang berlebihan), menjadi sangat sensitif (mudah tersinggung), terlebih bagi mereka yang sering tidak berpikir positif, dan halusinasi visual. Masing-masing pemakai mengalami efek tersebut dalam kadar yang berbeda.

Jika sedang banyak mempunyai persoalan / masalah dalam kehidupan, sebaiknya narkoba jenis ini tidak dikonsumsi. Hal ini mungkin dapat dirumuskan sebagai berikut: **MASALAH + SABU = SANGAT BERBAHAYA**. Selain itu, pengguna Sabu sering mempunyai kecenderungan untuk memakai dalam jumlah banyak dalam satu sesi dan sukar berhenti kecuali

jika Sabu yang dimilikinya habis. Hal itu juga merupakan suatu tindakan bodoh dan sia-sia mengingat efek yang diinginkan tidak lagi bertambah (*The Law Of Diminishing Return*). Beberapa pemakai mengatakan Sabu tidak mempengaruhi nafsu makan. Namun sebagian besar mengatakan nafsu makan berkurang jika sedang mengkonsumsi Sabu. Bahkan banyak yang mengatakan berat badannya berkurang drastis selama memakai Sabu.

Pecandu yang terdeteksi menggunakan jenis narkoba Sabu-Sabu mengalami gangguan, antara lain yaitu :

- Mengalami insomnia(susah tidur),
- Kekurangan kalsium,
- Merasa jantung berdebar-debar,
- Merasa nafsu makan berkurang,
- Merasa suhu tubuh naik.

2.2.6 Nikotin

Nikotin adalah senyawa kimia organik kelompok alkaloid yang dihasilkan secara alami pada berbagai macam tumbuhan, terutama suku terung-terungan (*Solanaceae*) seperti tembakau dan tomat. Nikotina berkadar 0,3 sampai 5,0% dari berat kering tembakau berasal dari hasil biosintesis di akar dan terakumulasi di daun. Nikotin merupakan berpotensi sebagai racun saraf yang potensial dan digunakan sebagai bahan baku berbagai jenis insektisida dalam konsentrasi tinggi. Pada konsentrasi rendah, zat ini dapat menimbulkan kecanduan, khususnya pada rokok.

Nikotina memiliki daya karsinogenik terbatas yang menjadi penghambat kemampuan tubuh untuk melawan sel-sel kanker, akan tetapi nikotina tidak menyebabkan perkembangan sel-sel sehat menjadi sel-sel

kanker/ bukan penyebab kanker. Nikotin secara alami ada dalam tanaman tembakau, demikian halnya kafein pada kopi, tein, katekin pada teh, theobromin dalam coklat. Mereka termasuk dalam bahan penyegar, maksudnya konsumsi bahan tersebut dalam makanan tertentu dengan kadar tertentu akan menimbulkan efek giat dan segar. Karena itu bagi tidak disarankan konsumsi bahan penyegar pada penderita insomnia/ kesulitan tidur, berulang kali terbangun dari tidur.

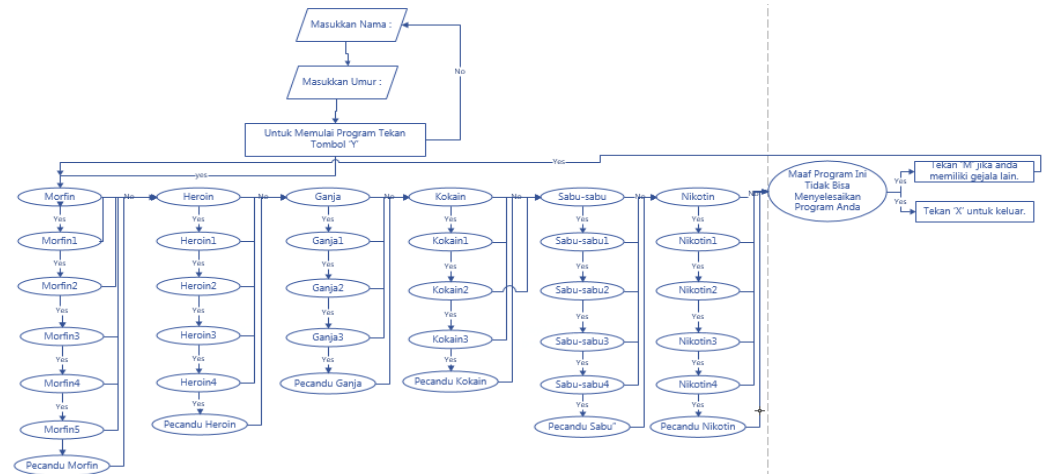
Pecandu yang terdeteksi menggunakan jenis narkoba Nikotin mengalami gangguan, antara lain yaitu :

- Sering sakit kepala,
- Merasakan nafas terasa berat,
- Merasa meningkatnya denyut jantung,
- Merasakan suhu badan menurun,
- Sering mengantuk.

BAB III

METODOLOGI

3.1 Alur Pembuatan Sistem



Gambar 3.1 Alur Pembuatan Sistem.

Pertama kita jalan dulu programnya setelah itu akan ada tampilan untuk memasukkan nama dan umur, setelah mengisi nama dan umur akan ada pilahan apakah ingin memulai program? jika ya maka tekan tombol ‘Y’. Jika tidak maka akan kembali ke menu untuk memasukkan nama dan umur.

Setelah memilih program untuk dijalankan maka gejala – gejala yang ada pada program akan ditampilkan satu per satu, untuk memilih gejala yang anda hadapi maka tekan tombol ‘y’ apabila gejala tersebut tidak terjadi pada anda maka tekan tombol ‘t’ dan begitu seterusnya. Contoh apabila gejala morfin 1 sampai terakhir terjadi pada anda maka anda terdeteksi menggunakan atau mengkonsumsi narkorika jenis morfin begitu seterusnya. Tetapi apa bila semua gejala yang ada tidak terjadi pada anda berarti anda tidak mengkonsumsi narkotika. Setelah program selesai maka akan muncul pilihan baru yaitu, tekan tombol ‘M/m’ jika anda memiliki gejala lain. Dan tekan tombol ‘X/x’ untuk keluar dari program.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Tabel Kebenaran dan Keputusan

Tabel 4.1 Tabel Kebenaran

No	Nama Gejala	Narkoba				
		Morfin	Heroin	Ganja	Kokain	Sabu-sabu
1	Pupil mata menyempit	√	√			
2	Merasa mual dan muntah-muntah		√			
3	Merasa denyut nadi dan jantung lebih cepat			√		
4	Merasa kebingungan	√				
5	Kekurangan kalsium				√	
6	Merasa meningkatnya denyut jantung					√
7	Tekanan darah menurun	√	√			
8	Suhu badan menurun	√				
9	Denyut nadi semakin lama melambat	√				
10	Merasa gatal pada hidung		√			
11	Suka menyendiri		√			
12	Produksi air seni berkurang	√				
13	Mengalami insomnia(susah tidur)				√	
14	Sering kejang-kejang				√	
15	Sering sakit kepala					√
16	Merasa jantung berdebar-debar				√	
17	Merasa berat badan menurun				√	
18	Sulit diajak berkomunikasi			√		
19	Merasa suhu tubuh naik				√	
20	Merasakan suhu badan menurun					√
21	Sering mengeluarkan dahak				√	
22	Sering mengantuk					√
23	Suka berfantasi(mengkhayal)			√		
24	Merasa napsu makan berkurang				√	
25	Merasakan nafas terasa berat					√
26	Sulit mengingat			√		

Tabel Kebenaran adalah tabel yang digunakan melihat nilai kebenaran dari suatu premis/ Pernyataan. Terdapat gejala-gejala yang di alami oleh pecandu Narkoba di dalam tabel tersebut.

Tabel 4.2 Tabel Keputusan

No	Atribut*																										Narkoba
	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	g13	g14	g15	g16	g17	g18	g19	g20	g21	g22	g23	g24	g25	g26	
1	V			V			V	V	V			V															Morfin
2	V	V					V			V	V																Heroin
3			V															V						V		V	Ganja
4					V								V	V		V	V		V		V			V			Kokain
5						V									V					V		V			V		Sabu-sabu

Tabel keputusan (decision table) adalah tabel yang digunakan sebagai alat bantu untuk menyelesaikan logika dalam program. Algoritma yang berisi keputusan bertingkat yang banyak sekali sangat sulit untuk digambarkan langsung dengan structured English atau pseudocode dan dapat dibuat terlebih dahulu dengan menggunakan tabel keputusan. Dengan demikian tabel keputusan efektif digunakan bilamana kondisi yang akan diseleksi didalam program jumlahnya cukup banyak dan rumit. Di dalam table tersebut gejala-gejala pada pecandu narkoba di inisialka menjadi g1,g2,g3,...dan seterusnya.

4.2 Analisis Aplikasi

```

*****
                        SELAMAT DATANG DI PROGRAM
                SISTEM PAKAR PENDETEKSI JENIS NARKOTIKA YANG DIGUNAKAN PECANDU
*****
Nama Kelompok :
1.Dian Pertiwi H.P  1515015180
2.Nadia            1515015184
3.Sri Uswatul H    1515015193
*****
Masukkan Nama Anda :|

```

Gambar 4.2.1 : Tampilan Awal Program.

Dimana pada saat program pertmakali dijalankan maka akan muncul tampilan seperti diatas, yang berisi Nama kelompok dan pengimputan nama.

```

*****
                        SELAMAT DATANG DI PROGRAM
                SISTEM PAKAR PENDETEKSI JENIS NARKOTIKA YANG DIGUNAKAN PECANDU
*****
Nama Kelompok :
1.Dian Pertiwi H.P 1515015180
2.Nadia           1515015184
3.Sri Uswatul H   1515015193
*****
Masukkan Nama Anda : Danis
Masukkan Umur Anda : 34

```

Gambar 4.2.2 : Tampilan Pengisian Nama dan Umur.

Setelah menu tampilan awal, disini juga ada pengisian data diri dari seseorang yang menggunakan program ini, jadi sebelum masuk untuk memulai program pengguna wajib mengisi nama dan umur terlebih dahulu.

```

*****
Selamat Datang Diprogram Sistem Pakar Pendeteksi Narkotika Yang Digunakan Pecandu
*****
Nama Pengguna Danis Berumur 34 Tahun.

*****
                Untuk Memulai Program, Tekan Tombol 'Y'
*****

```

Gambar 4.2.3 : Tampilan Untuk Memulai Program.

Jadi, setelah mengisi nama dan umur anda, setelah itu tekan enter maka anda akan masuk kedalam program, setelah itu untuk memulai proses menjalankan program. Anda harus menekan tombol 'Y/y'.

```

*****
Selamat Datang Diprogram Sistem Pakar Pendeteksi Narkotika Yang Digunakan Pecandu
*****
Nama Pengguna Danis Berumur 34 Tahun.

*****
                Untuk Memulai Program, Tekan Tombol 'Y'
*****
Masukkan Nama Anda : |

```

Gambar 4.2.4 : Tampilan Kembali Kemenu Nama dan Umur.

Jadi apabila anda tidak menekan tombol selain 'Y/y' maka program akan mengarah anda ke tampilan sebelumnya yaitu pengisian nama dan umur.

```

*****
Selamat Datang Diprogram Sistem Pakar Pendeteksi Narkotika Yang Digunakan Pecandu
*****
Nama Pengguna Danis Berumur 34 Tahun.

*****
Untuk Memulai Program. Tekan Tombol 'Y'
*****

Apakah Pupil Mata Menyempit? {y/t}y
Apakah Denyut Nadi Semakin Lama Melambat? {y/t}y
Apakah Tekanan Darah Menurun? {y/t}y
Apakah Suhu Badan Menurun? {y/t}y
Apakah Produksi Air Seni Berkurang? {y/t}y
Apakah Anda Merasa Kebingungan? {y/t}y

A. Gejala Yang Terjadi Pada Anda Dikarenakan : Mengkonsumsi Narkotika Jenis Morfin
B. Dampaknya adalah :
1. Peradangan Pada Hati.
2. Kerusakan Pada Sistem Otak.
3. Kadar Gula Yang Menurun.
4. Kesulitan Saat Berbicara [cadell].

*****
Terima Kasih Sudah Menggunakan Program Kami
Tekan 'M' Jika Anda Masih Memiliki Gejala Lain.
Tekan 'X' Jika Anda Ingin Keluar.
*****

```

Gambar 4.2.5 : Tampilan Gejala – gejala Pada Narkotika Jenis Morfin.

Untuk memulai program tekan tombol 'Y/y'. Setelah itu gejala – gejala yang ada akan tampil satu per satu, seperti contoh diatas gejala yang pertama muncul yaitu Apakah pupil mata menyempit ? disini kami memilih 'y' maka program akan mengarahkan ke gejala yang selanjutnya yaitu Denyut nadi semakin lama melambat ? jika 'y' maka akan diarahkan ke gejala tekanan darah menurun? jika 'y' maka akan diarahkan ke gejala produksi air seni berkurang? jika 'y' maka akan diarahkan ke gejala selanjutnya yaitu apakah anda merasa bingung? jika 'y' maka dari semua gejala yang ada bisa disimpulkan bahwa anda mengkonsumsi narkotika jenis morfin. Dan setelah mengetahui jenis narkoba yang anda konsumsi program ini juga memberitahukan dampak yang akan terjadi apa bila ada masih mengkonsumsinya dalam jangka waktu yang panjang. Setelah itu ada menu pilihan jika anda ingin kembali ke gejala – gejala narkotika silahkan tekan tombol 'M/m' dan jika ingin keluar tekan tombol 'X/x'.

Apakah Pupil Mata Menyempit? [y/t]t
 Apakah Merasa Gatal Pada Hidung? [y/t]y
 Apakah Merasa Mual dan Muntah-muntah? [y/t]y
 Apakah Suka Menyendiri? [y/t]y
 Apakah Pupil Mata Mengecil? [y/t]y
 Apakah Tekanan Darah Menurun? [y/t]y
 A. Gejala Yang Terjadi Pada Anda Dikarenakan : Mengkonsumsi Narkotika Jenis Heroin
 B. Dampaknya adalah :
 1. Pembuluh Darah Akan Pecah.
 2. Masalah Pada Jantung dan Cabang Tenggorokan.
 3. Impotensi.

```

*****
Terima Kasih Sudah Menggunakan Program Kami
Tekan 'M' Jika Anda Masih Memiliki Gejala Lain.
Tekan 'X' Jika Anda Ingin Keluar.
*****
1
  
```

Gambar 4.2.6 : Tampilan Gejala – gejala Pada Narkotika Jenis Heroin.

Untuk memulai program tekan tombol ‘Y/y’. Setelah itu gejala – gejala yang ada akan tampil satu per satu, seperti contoh diatas gejala yang pertama muncul yaitu Apakah pupil mata menyempit ? disini kami memilih ‘t’ maka program akan mengarahkan ke gejala yang selanjutnya yaitu merasa gatal pada hidung? jika ‘y’ maka akan diarahkan ke gejala merasa mual dan muntah – muntah? jika ‘y’ maka akan diarahkan ke gejala suka menyendiri? jika ‘y’ maka akan diarahkan ke gejala selanjutnya yaitu apakah pupil mata mengecil? jika ‘y’ maka akan diarahkan ke gejala tekanan darah menurun? jika ‘y’ maka dari semua gejala yang ada bisa disimpulkan bahwa anda mengkonsumsi narkotika jenis heroin. Dan setelah mengetahui jenis narkoba yang anda konsumsi program ini juga memberitahukan dampak yang akan terjadi apa bila ada masih mengkonsumsinya dalam jangka waktu yang panjang. Setelah itu ada menu pilihan jika anda ingin kembali ke gejala – gejala narkotika silahkan tekan tombol ‘M/m’ dan jika ingin keluar dari program tekan tombol ‘X/x’.

Apakah Pupil Mata Menyempit? [y/t]t

Apakah Merasa Gatal Pada Hidung? [y/t]t

Apakah Anda Sulit Diajak Berkomunikasi? [y/t]y

Apakah Anda Sering Berfantasi (Mengkhalayak)? [y/t]y

Apakah Denyut Nadi dan Jantung Lebih Cepat? [y/t]y

Apakah Anda Sulit Mengingat? [y/t]y

A. Gejala Yang Terjadi Pada Anda Dikarenakan : Mengonsumsi Narkotika Jenis Ganja

B. Dampaknya adalah :

1. Kanker Paru-paru.
2. Gejala Gangguan Kejiwaan.
3. Kerusakan Sistem Kekebalan Tubuh.

```
*****
Terima Kasih Sudah Menggunakan Program Kami
Tekan 'M' Jika Anda Masih Memiliki Gejala Lain.
Tekan 'X' Jika Anda Ingin Keluar.
*****
|
```

Gambar 4.2.7 : Tampilan Gejala – gejala Pada Narkotika Jenis Ganja.

Untuk memulai program tekan tombol ‘Y/y’. Setelah itu gejala – gejala yang ada akan tampil satu per satu, seperti contoh diatas gejala yang pertama muncul yaitu Apakah pupil mata menyempit ? disini kami memilih ‘t’ maka program akan mengarahkan ke gejala yang selanjutnya yaitu merasa gatal pada hidung? jika ‘t’ maka akan diarahkan ke gejala Apakah anda sulit diajak berkomunikasi? jika ‘y’ maka akan diarahkan ke gejala apakah anda sering berfantasi? jika ‘y’ maka akan diarahkan ke gejala selanjutnya yaitu apakah denyut nadi dan jantung lebih cepat? jika ‘y’ maka akan diarahkan ke gejala apakah anda sulit mengingat? jika ‘y’ maka dari semua gejala yang ada bisa disimpulkan bahwa anda mengonsumsi narkotika jenis ganja. Dan setelah mengetahui jenis narkoba yang anda konsumsi program ini juga memberitahukan dampak yang akan terjadi apa bila ada masih mengkonsumsinya dalam jangka waktu yang panjang. Setelah itu ada menu pilihan jika anda ingin kembali ke gejala – gejala narkotika silahkan tekan tombol ‘M/m’ dan jika ingin keluar dari program tekan tombol ‘X/x’.

Apakah Pupil Mata Menyempit? [y/t]t
 Apakah Merasa Gatal Pada Hidung? [y/t]t
 Apakah Anda Sulit Diajak Berkomunikasi? [y/t]t
 Apakah Anda Sering Kejang-Kejang? [y/t]y
 Apakah Anda Sering Mengeluarkan Dahak? [y/t]y
 Apakah Berat Badan Anda Menurun? [y/t]y
 Apakah Anda Sering Merasa Kebingungan? [y/t]y
 A. Gejala Yang Terjadi Pada Anda Dikarenakan : Mengkonsumsi Narkotika Jenis Kokain
 B. Dampaknya adalah :
 1. Kerusakan Pada Ginjal, Hati, dan Paru-paru.
 2. Kontraksi Pembuluh Darah.
 3. Kerusakan Pada Jaringan dalam Hidung.
 4. Tekanan Darah Tinggi.
 5. Kerusakan Pada Gigi.

 Terima Kasih Sudah Menggunakan Program Kami
 Tekan 'M' Jika Anda Masih Memiliki Gejala Lain.
 Tekan 'X' Jika Anda Ingin Keluar.

 |

Gambar 4.2.8 : Tampilan Gejala – gejala Pada Narkotika Jenis Kokain.

Untuk memulai program tekan tombol 'Y/y'. Setelah itu gejala – gejala yang ada akan tampil satu per satu, seperti contoh diatas gejala yang pertama muncul yaitu Apakah pupil mata menyempit ? disini kami memilih 't' maka program akan mengarahkan ke gejala yang selanjutnya yaitu merasa gatal pada hidung? jika 't' maka akan diarahkan ke gejala Apakah anda sulit diajak berkomunikasi? jika 't' maka akan diarahkan ke gejala apakah anda sering kejang – kejang? jika 'y' maka akan diarahkan ke gejala selanjutnya yaitu apakah anda sering mengeluarkan dahak? jika 'y' maka akan diarahkan ke gejala apakah berat badan anda menurun? jika 'y' maka akan diarahkan ke gejala selanjutnya yaitu apakah anda sering merasa gelisah? Jika 'y' maka dari semua gejala yang ada bisa disimpulkan bahwa anda mengkonsumsi narkotika jenis kokain. Dan setelah mengetahui jenis narkoba yang anda konsumsi program ini juga memberitahukan dampak yang akan terjadi apa bila ada masih mengkonsumsinya dalam jangka waktu yang panjang. Setelah itu ada menu pilihan jika anda ingin kembali ke gejala – gejala narkotika silahkan tekan tombol 'M/m' dan jika ingin keluar dari program tekan tombol 'X/x'.

Apakah Pupil Mata Menyempit? [y/t]t
 Apakah Merasa Gatal Pada Hidung? [y/t]t
 Apakah Anda Sulit Diajak Berkomunikasi? [y/t]t
 Apakah Anda Sering Kejang-Kejang? [y/t]t
 Apakah Anda Mengalami Insomnia[Susah Tidur]? [y/t]y
 Apakah Anda Kekurangan Kalsium? [y/t]y
 Apakah Jantung Anda Berdebar-debar? [y/t]y
 Apakah Nafsu Makan Anda Berkurang? [y/t]y
 Apakah Suhu Tubuh Anda Naik? [y/t]y
 A. Gejala Yang Terjadi Pada Anda Dikarenakan : Mengonsumsi Narkotika Jenis Sabu-sabu
 B. Dampaknya adalah :
 1. Tekanan Darah Meningkat.
 2. Infeksi Pada Paru-paru.
 3. Pupil Membesar.

 Terima Kasih Sudah Menggunakan Program Kami
 Tekan 'M' Jika Anda Masih Memiliki Gejala Lain.
 Tekan 'X' Jika Anda Ingin Keluar.

Gambar 4.2.9 : Tampilan Gejala – gejala Pada Narkotika Jenis Sabu-sabu.

Untuk memulai program tekan tombol 'Y/y'. Setelah itu gejala – gejala yang ada akan tampil satu per satu, seperti contoh diatas gejala yang pertama muncul yaitu Apakah pupil mata menyempit ? disini kami memilih 't' maka program akan mengarahkan ke gejala yang selanjutnya yaitu merasa gatal pada hidung? jika 't' maka akan diarahkan ke gejala Apakah anda sulit diajak berkomunikasi? jika 't' maka akan diarahkan ke gejala apakah anda sering kejang – kejang? jika 't' maka akan diarahkan ke gejala selanjutnya yaitu apakah anda mengalami insomnia(susah tidur)? jika 'y' maka akan diarahkan ke gejala apakah anda kekurangan kalsium? jika 'y' maka akan diarahkan ke gejala selanjutnya yaitu apakah jantung anda berdebar-debar? Jika 'y' maka anda akan diarahkan ke gejala selanjutnya yaitu apakah nafsu makan anda berkurang? Jika 'y' maka program akan mengarahkan ke gejala berikutnya apakah suhu tubuh anda naik? Jika 'y' maka dari semua gejala yang ada bisa disimpulkan bahwa anda mengonsumsi narkotika jenis sabu-sabu. Dan setelah mengetahui jenis narkoba yang anda konsumsi program ini juga memberitahukan dampak yang akan terjadi apa bila anda masih mengkonsumsinya dalam jangka waktu yang panjang. Setelah itu ada menu pilihan jika anda ingin kembali ke gejala – gejala narkotika silahkan tekan tombol 'M/m' dan jika ingin keluar dari program tekan tombol 'X/x'.

```

Apakah Pupil Mata Menyempit? [y/t]t
Apakah Merasa Gatal Pada Hidung? [y/t]t
Apakah Anda Sulit Diajak Berkomunikasi? [y/t]t
Apakah Anda Sering Kejang-Kejang? [y/t]t
Apakah Anda Mengalami Insomnia(Susah Tidur)? [y/t]t
Apakah Anda Sering Sakit Kepala? [y/t]y
Apakah Saat Nafas Terasa Berat? [y/t]y
Apakah Anda Merasa Meningkatnya Denyut Jantung? [y/t]y
Apakah Suhu Badan Anda Menurun? [y/t]y
Apakah Anda Sering Mengantuk? [y/t]y
A. Gejala Yang Terjadi Pada Anda Dikarenakan : Mengonsumsi Narkotika Jenis Nikotin
B. Dampaknya adalah :
1. Gangguan Pada Pembuluh Darah.
2. Kerusakan Pada Paru-paru, Hati dan Jantung.
3. Emfisema (Menbuat Paru-paru sulit mengeluarkan udara kotor).
*****
Terima Kasih Sudah Menggunakan Program Kami
Tekan 'M' Jika Anda Masih Memiliki Gejala Lain.
Tekan 'X' Jika Anda Ingin Keluar.
*****

```

Gambar 4.2.10 : Tampilan Gejala – gejala Pada Narkotika Jenis Nikotin.

Untuk memulai program tekan tombol ‘Y/y’. Setelah itu gejala – gejala yang ada akan tampil satu per satu, seperti contoh diatas gejala yang pertama muncul yaitu Apakah pupil mata menyempit ? disini kami memilih ‘t’ maka program akan mengarahkan ke gejala yang selanjutnya yaitu merasa gatal pada hidung? jika ‘t’ maka akan diarahkan ke gejala Apakah anda sulit diajak berkomunikasi? jika ‘t’ maka akan diarahkan ke gejala apakah anda sering kejang – kejang? jika ‘t’ maka akan diarahkan ke gejala selanjutnya yaitu apakah anda mengalami insomnia(susah tidur)? jika ‘t’ maka akan diarahkan ke gejala apakah anda sering sakit kepala? jika ‘y’ maka akan diarahkan ke gejala selanjutnya yaitu apakah saat bernafas terasa berat?? Jika ‘y’ maka anda akan diarahkan kegejala selanjutnya yaitu apakah anda merasa meningkatnya denyut jantung? Jika ‘y’ maka program akan mengarahkan ke gejala berikutnya apakah suhu badan anda menurun? Jika ‘y’ maka program akan mengarahkan anda kegejala selanjutnya apakah anda sering mengantuk? Jika ‘y’ maka dari semua gejala yang ada bisa disimpulkan bahwa anda mengonsumsi narkotika jenis nikotin. Dan setelah mengetahui jenis narkoba yang anda konsumsi program ini juga memberitahukan dampak yang akan terjadi apa bila ada masih mengkonsumsinya dalam jangka waktu yang panjang. Setelah itu ada menu pilihan jika anda ingin kembali ke gejala – gejala narkotika silahkan tekan tombol ‘M/m’ dan jika ingin keluar dari program tekan tombol ‘X/x’.

Apakah Pupil Mata Menyempit? [y/t]t
Apakah Merasa Gatal Pada Hidung? [y/t]t
Apakah Anda Sulit Diajak Berkomunikasi? [y/t]t
Apakah Anda Sering Kejang-Kejang? [y/t]t
Apakah Anda Mengalami Insomnia[Susah Tidur]? [y/t]t
Apakah Anda Sering Sakit Kepala? [y/t]t
Maaf Program Ini Tidak Dapat Menyelesaikan Gejala Yang Anda Hadapi

```
*****
Terima Kasih Sudah Menggunakan Program Kami
Tekan 'M' Jika Anda Masih Memiliki Gejala Lain.
Tekan 'X' Jika Anda Ingin Keluar.
*****
```

Gambar 4.2.11 : Tampilan jika gejala – gejala taksesuai.

Jadi apabila gejala – gejala yang ada pada program tidak sesuai dengan yang anda rasakan maka program akan berikan notif yaitu “maaf program ini tidak dapat menyelesaikan gejala yang anda hadapi. Berarti gejala anda buka pecandu narkoba. Setelah itu ada menu pilihan jika anda ingin kembali ke gejala – gejala narkoba silahkan tekan tombol ‘M/m’ dan jika ingin keluar dari program tekan tombol ‘X/x’.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melalui tahap pengujian pada sistem pakar untuk mendeteksi jenis pecandu narkoba, maka dapat diambil beberapa kesimpulan sebagai berikut :

System pakar adalah system yang berusaha mengadopsi pengetahuan manusia kekomputer, agar komputer dapat menyelesaikan masalah yang seperti biasa dilakukan oleh para ahli.

Ada beberapa hal penting yang perlu diperhatikan bahwa secanggih apapun suatu system atau sebesar apapun basis pengetahuan yang dimiliki, tentu saja ada kelemahannya sebagai konsekuensi logis kelemahan manusia sebagai penyusun element-elementnya. Bahwa system tidak memiliki inisiatif untuk melakukan suatu tindakan diluar dari apa yang telah diprogramkan untuknya.

Berdasarkan hasil pengujian, program system pakar ini berguna untuk membantu dan mempermudah pengguna dalam memperoleh informasi mengenai pecandu narkoba serta mendapatkan hasil diagnosa jenis narkoba yang dikonsumsi pecandu melalui gejala-gejalanya.

5.2 Saran

Beberapa hal yang dapat diungkap sebagai saran untuk pengembangan sistem pakar mendeteksi jenis pecandu narkoba adalah sebagai berikut :

Dilakukan pengembangan program sejenis dengan permasalahan domain yang lebih luas.

Data mengenai gejala dan semua mengenai narkoba kiranya dapat lebih dimaksimalkan sampai mendapatkan informasi atau perhitungan yang lebih akurat serta dicari alternative lain yang memungkinkan penyelesaian yang jauh lebih baik.

DAFTAR PUSTAKA

[https://id.wikipedia.org/wiki/Tabel_k](https://id.wikipedia.org/wiki/Tabel_kebenaran)
[ebenaran](https://id.wikipedia.org/wiki/Tabel_kebenaran)

<http://nandathe123go.blogspot.co.id/>

<https://core.ac.uk/download/pdf/33323081.pdf>

Lampiran

1. Source Code

```
DOMAINS
narkoba = symbol
gejala = symbol
tanya = string
jawab = char
kondisi = cond*
cond = string

FACTS
nama(kondisi)
false(cond)
xcari(gejala)
xgagal(gejala)

PREDICATES
nondeterm mulai
nondeterm mulai1
nondeterm balik
    cari(tanya,gejala)
    gagal(tanya,gejala)
nondeterm gejala(gejala)
    selesai
    simpan(gejala,jawab)
    tanya(tanya,gejala,jawab)
    go_once
nondeterm diagnosa(narkoba)
```

```

        jenis(narkoba)
        dampak(narkoba)
nondeterm oke(char)
nondeterm ya(char)
nondeterm lihat(kondisi)
zz(char,char)

CLAUSES
zz(A,A):-!.
zz(_,_):-fail.

nama(["1.Dian Pertiwi H.P    1515015180", "2.Nadia
      1515015184", "3.Sri Uswatul H    1515015193"]).
mulai:-
        write("*****
*****"),nl,
        write("\t\tSELAMAT DATANG DI PROGRAM"),nl,
        write("\tSISTEM PAKAR PENDETEKSI JENIS NARKOTIKA
YANG DIGUNAKAN PECANDU"),nl,
        write("*****
*****"),nl,
        write("Nama Kelompok :"),nl,
        nama(Kelompok),
        lihat(Kelompok),
        balik.
balik:-
        write("*****
*****"),nl,
        write("Masukkan Nama Anda : "),

```

```

readln(Orang),
write("Masukkan Umur Anda : "),
readln(Umur),nl,
write("*****
*****"),nl,
write("        Selamat Datang Diprogram Sistem Pakar Pendeteksi
Narkotika Yang Digunakan Pecandu        "),nl,
write("*****
*****"),nl,
write("Nama Pengguna ",Orang," Berumur ",Umur," Tahun."),nl,nl,nl,
write("*****
*****"),nl,
write("\t\tUntuk Memulai Program, Tekan Tombol 'Y'"),nl,
write("*****
*****"),nl,
readchar(Y), ya(Y).
ya(Y):-zz(Y,'y'), mulai1.
ya(_):-balik.

lihat ([H|T]):-
    not(false(H)),
    write(H),nl,
    lihat(T).

lihat([H|_]):-
    asserta(false(H)).

mulai1:-
    go_once,nl,nl,

```

```

write("*****
*****"),nl,
        write("\t\t Terima Kasih Sudah Menggunakan Program
Kami"),nl,
        write("\t\t Tekan 'M' Jika Anda Masih Memiliki Gejala
Lain."),nl,
        write("\t\t Tekan 'X' Jika Anda Ingin Keluar."),nl,

write("*****
*****"),nl,
        readchar(Y),oke(Y).
        oke(Y):-zz(Y,'M'),gejala(morfin).
        oke(Y):-zz(Y,'m'),gejala(morfin).
        oke(Y):-zz(Y,'X'),exit.
        oke(Y):-zz(Y,'x'),exit.
        oke(_):-mulai1.

go_once:-
        diagnosa(_),!,
        save("test.dat"),
        selesai.

go_once:-
        write("\nMaaf Program Ini Tidak Dapat Menyelesaikan Gejala
Yang Anda Hadapi"),nl,
        selesai.

cari(_,Gejala):-
        write("\nApakah "),

```

```
xcari(Gejala),!.
```

```
cari(Tanya,Gejala):-
```

```
    not(xgagal(Gejala)),  
    tanya(Tanya,Gejala,Jawab),  
    Jawab = 'y'.
```

```
gagal(_,Gejala):-
```

```
    xgagal(Gejala),!.
```

```
gagal(Tanya,Gejala):-
```

```
    not(xcari(Gejala)),  
    tanya(Tanya,Gejala,Jawab),  
    Jawab = 't'.
```

```
tanya(Tanya,Gejala,Jawab):-
```

```
    write(Tanya),  
    readchar(Jawab),  
    write(Jawab),nl,  
    simpan(Gejala,Jawab).
```

```
simpan(Gejala,'y'):-
```

```
    asserta(xcari(Gejala)).
```

```
simpan(Gejala,'t'):-
```

```
    asserta(xgagal(Gejala)).
```

```
selesai:-
```

```
    retract(xcari(_)),fail.
```

selesai:-

retract(xgagal(_)),fail.

selesai.

gejala(Gejala):-

xcari(Gejala),!.

gejala(Gejala):-

xgagal(Gejala),!,fail.

gejala(morfin):-

cari(" Pupil Mata Menyempit? (y/t)",morfin),

cari(" Denyut Nadi Semakin Lama Melambat? (y/t)",morfin1),

cari(" Tekanan Darah Menurun? (y/t)",morfin2),

cari(" Suhu Badan Menurun? (y/t)",morfin3),

cari(" Produksi Air Seni Berkurang? (y/t)",morfin4),

cari(" Anda Merasa Kebingungan? (y/t)",morfin5).

gejala(heroin):-

cari(" Merasa Gatal Pada Hidung? (y/t)",heroin),

cari(" Merasa Mual dan Muntah-muntah? (y/t)",heroin1),

cari(" Suka Menyendiri? (y/t)",heroin2),

cari(" Pupil Mata Mengecil? (y/t)",heroin3),

cari(" Tekanan Darah Menurun? (y/t)",heroin4).

gejala(ganja):-

cari(" Anda Sulit Diajak Berkomunikasi? (y/t)",ganja),

cari(" Anda Sering Berfantasi (Mengkhayal)? (y/t)",ganja1),

cari(" Denyut Nadi dan Jantung Lebih Cepat? (y/t)",ganja2),

cari(" Anda Sulit Mengingat? (y/t)",ganja3).

gejala(kokain):-

```

cari(" Anda Sering Kejang-Kejang? (y/t)",kokain),
cari(" Anda Sering Mengeluarkan Dahak? (y/t)",kokain1),
cari(" Berat Badan Anda Menurun? (y/t)",kokain2),
cari(" Anda Sering Merasa Kebingungan? (y/t)",kokain3).
gejala(sabusabu):-
    cari(" Anda Mengalami Insomnia(Susah Tidur)?
(y/t)",sabusabu),
    cari(" Anda Kekurangan Kalsium? (y/t)",sabusabu1),
    cari(" Jantung Anda Berdebar-debar? (y/t)",sabusabu2),
    cari(" Nafsu Makan Anda Berkurang? (y/t)",sabusabu3),
    cari(" Suhu Tubuh Anda Naik? (y/t)",sabusabu4).
gejala(nikotin):-
    cari(" Anda Sering Sakit Kepala? (y/t)",nikotin),
    cari(" Saat Nafas Terasa Berat? (y/t)",nikotin1),
    cari(" Anda Merasa Meningkatnya Denyut Jantung?
(y/t)",nikotin2),
    cari(" Suhu Badan Anda Menurun? (y/t)",nikotin3),
    cari(" Anda Sering Mengantuk? (y/t)",nikotin4).
diagnosa("Narkoba_morfin"):-
    gejala(morfin),
    gejala(morfin1),
    gejala(morfin2),
    gejala(morfin3),
    gejala(morfin4),
    gejala(morfin5),
    jenis("Mengonsumsi Narkotika Jenis Morfin"),
    dampak("1. Peradangan Pada Hati.\n 2. Kerusakan Pada Sistem
Otak.\n 3. Kadar Gula Yang Menurun.\n 4. Kesulitan Saat Berbicara
(cadel).\n").

```

diagnosa("Narkoba_heroin):-

gejala(heroin),

gejala(heroin1),

gejala(heroin2),

gejala(heroin3),

gejala(heroin4),

jenis("Mengkonsumsi Narkotika Jenis Heroin"),

dampak("1. Pembuluh Darah Akan Pecah.\n 2. Masalah Pada Jantung dan Cabang Tenggorokan.\n 3. Impotensi.\n").

diagnosa("Narkoba_ganja):-

gejala(ganja),

gejala(ganja1),

gejala(ganja2),

gejala(ganja3),

jenis("Mengkonsumsi Narkotika Jenis Ganja"),

dampak("1. Kanker Paru-paru.\n 2. Gejala Gangguan Kejiwaan.\n 3. Kerusakan Sistem Kekebalan Tubuh.\n").

diagnosa("Narkoba_kokain):-

gejala(kokain),

gejala(kokain1),

gejala(kokain2),

gejala(kokain3),

jenis("Mengkonsumsi Narkotika Jenis Kokain"),

dampak("1. Kerusakan Pada Ginjal, Hati, dan Paru-paru.\n 2. Kontraksi Pembuluh Darah.\n 3. Kerusakan Pada Jaringan dalam Hidung.\n 4. Tekanan Darah Tinggi.\n 5. Kerusakan Pada Gigi.\n").

diagnosa("Narkoba_sabusabu):-

gejala(sabusabu),

gejala(sabusabu1),


```
gejala(sabusabu2),
gejala(sabusabu3),
gejala(sabusabu4),
jenis("Mengkonsumsi Narkotika Jenis Sabusabu"),
dampak("1. Tekanan Darah Meningkat.\n 2. Infeksi Pada Paru-
paru.\n 3. Pupil Membesar.\n").
```


```
diagnosa("Narkoba_nikotin):-
gejala(nikotin),
gejala(nikotin1),
gejala(nikotin2),
gejala(nikotin3),
gejala(nikotin4),
jenis("Mengkonsumsi Narkotika Jenis Nikotin"),
dampak("1. Gangguan Pada Pembuluh Darah. \n 2. Kerusakan
Pada Paru-paru, Hati dan Jantung.\n 3. Emfisema (Menbuat Paru-paru sulit
mengeluarkan udara kotor).\n").
```

```
jenis(Narkoba):-
write("\n A. Gejala Yang Terjadi Pada Anda Dikarenakan :
",Narkoba),nl.
```

```
dampak(Narkoba):-
write("\n B. Dampaknya adalah : \n ",Narkoba),nl.
```

```
GOAL
mulai,
mulai1.
```

2. Kartu Konsul

<p>PERATURAN</p> <ol style="list-style-type: none">1. Praktikan wajib konsultasi Programing Project dengan Asisten Yang telah ditentukan.2. Praktikan yang tidak konsultasi DILARANG mengikuti ujian.3. Konsultasi I, membahas tentang konsep PP yang akan dibuat.4. Konsultasi II, membahas tentang hal yang berkaitan dengan PP yang sedang dikerjakan.	<p>"SISTEM PAKAR PENDETEKSI JENIS NARKOTIKA YANG DIGUNAKAN PECANDU"</p> <p>PRAKTIKUM KECERDASAN BUATAN</p> <p>KELOMPOK : 1/D2/2015</p> <p>DIAN PERTIWI I.P / 1515015180 NADIA / 1515015184 SRI USWATUL HASANAH / 1515015193</p>  <p>LABORATORIUM BAHASA PEMROGRAMAN FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI 2017</p>
--	--

<p>Tanggal Konsultasi : 27/04/2017</p> <p>Uraian :</p> <p>- Konsul Program dan Pengesuanan Fakta -</p> <p>- Jenis narkotika :</p> <ol style="list-style-type: none">1. Morfin2. heroin3. ganja4. kokain5. Sabu-sabu6. Nikotin <p>Tanda Tangan Asisten : <i>[Signature]</i></p>	<p>Tanggal Konsultasi : 02/05/2017</p> <p>Uraian :</p> <p>- Konsul tentang list</p> <p>- Tabel keputusan</p> <p>- Simpan dengan asert</p> <p>Tanda Tangan Asisten : <i>[Signature]</i></p>
---	--