

**LAPORAN PRATIKUM KECERDASAN BUATAN
SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN OBAT PADA
PENYAKIT KULIT**



Disusun oleh :

Kelvin B.	1515015100
Evi Lolita	1515015128
Nadya Octavia Islami	1515015147

Asisten Pratikum :

Alvian Nur Wahyudhi

1415015066

Annisa Nur Afiyah

1415015068

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

2017

KATA PENGANTAR

Alhamdulillah dengan memanjatkan puji syukur kehadiran Allah swt yang maha pengasih dan penyayang yang telah memberikan rahmat, hidayah dan inayahnya kepada kami, sehingga dapat menyelesaikan penyusunan laporan ini.

Laporan ini merupakan salah satu tugas yang di berikan kepada kami dalam rangka pengembangan dasar ilmu pemograman terutama Kecerdasan Buatan. Selain itu tujuan dari penyusunan laporan ini juga untuk menambah wawasan secara meluas. Sehingga besar harapan kami, laporan yang kami sajikan dapat menjadi kontribusi positif bagi pengembang wawasan pembaca.

Akhirnya kami menyadari dalam penulisan laporan ini jauh dari kesempurnaan. Oleh karena itu, dengan segala kerendahan hati kami menerima kritik dan saran agar penyusunan laporan selanjutnya menjadi lebih baik.

Samarinda, 06 Mei 2016

Kelompok 4

DAFTAR ISI

HALAMAN JUDUL

KATA PENGANTAR i

DAFTAR ISI ii

DAFTAR TABEL iv

DAFTAR GAMBAR v

BAB I PENDAHULUAN 1

1.1. Latar Belakang 1

1.2. Rumusan Masalah 1

1.3. Batasan Masalah 2

1.4. Tujuan Dan Manfaat 2

1.4.1 Tujuan 2

1.4.2 Manfaat 2

BAB II LANDASAN TEORI 3

2.1. Fungsi-fungsi Visual Prolog yang Digunakan 3

2.1.1. Fungsi Lacak Balik 3

2.1.2. Fungsi Data Objek Jamak 3

2.1.3. Fungsi Perulangan 3

2.1.4. Fungsi List 4

2.2. Unifikasi dan Lacak Balik 4

BAB III METODOLOGI 9

3.1. Alur Pembuatan Sistem 9

BAB IV HASIL DAN PEMBAHASAN 11

4.1. Tabel Keputusan 11

4.2. Analisis Aplikasi 12

BAB V PENUTUP	15
5.1. Kesimpulan	15
5.2. Saran	15
DAFTAR PUSTAKA	16
LAMPIRAN	
1. Source Code	17
2. Kartu Konsul	26

DAFTAR TABEL

Tabel 1.1. Tabel Keputusan pada pemilihan obat	10
--	----

DAFTAR GAMBAR

Gambar 3.1. Alur Pembuatan Sistem dari program	9
Gambar 4.2.1. Tampilan Credit	11
Gambar 4.2.2. Alur Sistem Pendukung Keputusan Penyakit Panu	11
Gambar 4.2.3. Alur Sistem Pendukung Keputusan Penyakit Kurap	12
Gambar 4.2.4. Alur Sistem Pendukung Keputusan Jerawat	12
Gambar 4.2.5. Tampilan ingin kembali dalam program	13

BAB I

PENDAHULUAN

1.1. Latar Belakang

Source code program aplikasi pemilihan obat yang sesuai untuk pasien atau penderita penyakit kulit sangat perlu. SPK pemilihan obat merupakan sistem yang dikembangkan untuk membantu apoteker memberikan obat saat jenis obat sangat banyak yang sesuai dengan pasien. Latar belakang dari penggunaan sistem ini adalah mengantisipasi kesalahan dalam pemberian obat.

Proses yang ada didalam sistem ini adalah bagaimana sistem memproses data obat berupa jenis penyakit, kategori obat, harga obat dan solusi agar tidak terjadi penyakit kulit lagi. Selain data obat yang diolah, data lainnya yang ikut serta di gunakan sebagai media pengambilan keputusan untuk menentukan obat yaitu jenis penyakit. Sistem sebisa mungkin di berikan pemahaman atau kecerdasan buatan tentang penyakit dan obat yang sesuai untuk mengatasinya. Untuk melakukan hal tersebut antara jenis obat dan penyakit dilakukan perhitungan menggunakan beberapa metode yang sesuai. Dengan metode tersebut tingkat probabilitas penyakit dengan obat akan akurat.

Dalam satu jenis penyakit bisa saja terdapat banyak obat untuk menyembuhkannya. Setiap jenis obat mempunyai kandungan dan bahan masing-masing. Dari masing-masing bahan obat dijadikan sebagai variabel yang digunakan untuk memproses keputusan pemilihan obat. Maka digunakanlah sistem ini sebagai alat atau tool untuk membantu menentukan obat terbaik untuk penderita penyakit.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang dipaparkan diatas, maka rumusan masalah dalam pembuatan program “Sistem Pendukung Keputusan Pemilihan Obat pada Penyakit Kulit” adalah :

- a. Bagaimana membuat Sistem Pendukung Keputusan menggunakan Visual Prolog?
- b. Bagaimana membangun suatu program Sistem Pendukung Keputusan yang dapat membantu proses pemilihan obat pada penyakit kulit?

1.3. Batasan Masalah

Adapun batasan masalah ini adalah sebagai berikut :

- a. Sistem Pendukung Keputusan dibangun dengan menggunakan Visual Prolog.

1.4. Tujuan Dan Manfaat

1.4.1. Tujuan

Adapun tujuan ini adalah sebagai berikut :

- a. Menciptakan suatu Sistem Pendukung Keputusan untuk mempermudah pemilihan obat pada penyakit kulit.

1.4.2. Manfaat

Adapun manfaat ini adalah sebagai berikut :

- a. Membantu menentukan keputusan.
- b. Mengurangi kesalahan pengambilan keputusan.
- c. Meningkatkan efektifitas.
- d. Penghematan waktu.
- e. Peningkatan bisnis.
- f. Meningkatkan kepuasan pengambil keputusan.

BAB II

LANDASAN TEORI

2.1. Fungsi – fungsi Visual Prolog yang Digunakan

Fungsi yang digunakan adalah :

- 2.1.1. Fungsi Lacak Balik adalah 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 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.
- 2.1.2. Fungsi Data Objek Jamak adalah 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.
- 2.1.3. Fungsi Perulangan adalah 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).

2.1.4. Fungsi 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.

2.2. Unifikasi dan Lacak Balik

2.2.1. Unifikasi

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.

Unifikasi adalah usaha untuk mencoba membuat dua ekspresi menjadi identik (mempersatukan keduanya) dengan mencari substitusi-substitusi tertentu untuk mengikuti peubah-peubah dalam ekspresi mereka tersebut. Unifikasi merupakan suatu prosedur sistematis untuk memperoleh peubah-peubah instan dalam wffs. Ketika nilai kebenaran predikat adalah sebuah fungsi dari nilai-nilai yang diasumsikan dengan argumen mereka, keinstanan terkontrol dari nilai-nilai selanjutnya yang menyediakan cara memvalidasi nilai-nilai kebenaran pernyataan yang berisi predikat.

Unifikasi merupakan dasar atas kebanyakan strategi inferensi dalam Kecerdasan Buatan. Sedangkan dasar dari unifikasi adalah substitusi.

Suatu substitusi (substitution) adalah suatu himpunan penetapan istilah-istilah kepada peubah, tanpa ada peubah yang ditetapkan lebih dari satu istilah. Sebagai pengetahuan jantung dari eksekusi Prolog, adalah mekanisme unifikasi.

Aturan – aturan unifikasi :

1. Dua atom adalah identik.
2. Dua daftar identik atau ekspresi dikonversi ke dalam satu buah daftar.
3. Sebuah konstanta dan satu peubah terikat dipersatukan, sehingga peubah menjadi terikat kepada konstanta.
4. Sebuah perubah tak terikat dipersatukan dengan sebuah peubah terikat.
5. Sebuah peubah terikat dipersatukan dengan sebuah konstanta jika pengikatan pada peubah terikat dengan konstanta tidak ada konflik.
6. Dua peubah tidak terikat disatukan. Jika peubah yang satu lainnya menjadi terikat dalam upa-urutan langkah unifikasi, yang lainnya juga menjadi terikat ke atom yang sama (peubah atau konstanta).
7. Dua peubah terikat disatukan jika keduanya terikat (mungkin melalui pengikatan tengah) ke atom yang sama (peubah atau konstanta).

2.2.2. Lacak Balik

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 di antara 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 alternatif subgoal yang lain.

Untuk mengetahui terjadinya lacakbalik, akan diamati langkah demi langkah bagaimana Visual Prolog mencari solusi dari goal yang diberikan.

1. Visual Prolog memanggil predikat `dapat_berenang` dengan variabel bebas `Apa`. Untuk mencoba menjawab panggilan ini, Visual Prolog mencari di program untuk pencocokan. Ditemukan kecocokan dengan klausa `dapat_berenang`, dan variabel `Apa` diunifikasikan dengan variabel `Y`.
2. Kemudian, Visual Prolog berusaha untuk memenuhi bagian *body* dari *rule*. Untuk melakukannya, Visual Prolog memanggil subgoal yang pertama pada *body* dari *rule* tersebut yaitu `jenis(X, hewan)`, dan mencari pencocokan untuk panggilan ini. Ditemukan pencocokan dengan fakta pertama dari klausa relasi `jenis`. Pada titik ini `X` diikat dengan nilai `vertebrata`. Kemudian Visual Prolog menset titik lacakbalik pada fakta `jenis(vertebrata, hewan)`.
3. Dengan `X` diikat pada nilai `vertebrata`, Visual Prolog membuat panggilan untuk subgoal yang kedua yaitu `adalah(Y, vertebrata)`, dan

mencari pencocokan. Dan menemukan dengan fakta yang pertama adalah(zebra, vertebrata). Y diikat dengan nilai zebra dan Prolog menset titik lacakbalik pada adalah(zebra, vertebrata).

4. Kemudian Prolog mencoba untuk memenuhi subgoal yang terakhir yaitu, hidup(zebra, di_air). Karena tidak ditemukan fakta tersebut maka panggilan gagal dan Prolog memulai lacakbalik untuk menemukan solusi lain.
5. Ketika Visual Prolog lacakbalik, proses kembali ke titik lacakbalik terakhir. Pada kasus ini, titik lacakbalik terakhir berada pada subgoal yang kedua dari *rule* yaitu fakta adalah(zebra, vertebrata).
6. Ketika Visual Prolog berada pada titik lacakbalik, ia akan membebaskan variabel yang diberi nilai pada titik lacakbalik tersebut dan berusaha mencari jawaban lain untuk panggilan tersebut dalam kasus ini adalah(Y, vertebrata).
7. Visual Prolog melanjutkan pencarian ke bawah dari titik lacakbalik tersebut untuk pencocokan klausa yang lain. Karena tidak ada klausa yang ditemukan, maka terjadi lacak balik dan proses kembali ke titik lacakbalik terakhir yaitu di jenis(vertebrata, hewan).
8. Visual Prolog membebaskan variabel dan mencoba untuk menemukan solusi lain dari panggilan jenis(X, hewan). Pencarian dimulai setelah titik lacakbalik dan menemukan kecocokan dengan fakta jenis di program pada klausa jenis(ikan, hewan). X kemudian diikat pada nilai ikan, dan titik lacakbalik baru di set pada fakta tersebut.
9. Kemudian Visual Prolog bergerak ke subgoal berikutnya pada *rule*. Karena ini panggilan baru, maka pencarian dimulai dari bagian atas program dengan panggilan adalah(Y, ikan). Visual Prolog menemukan kecocokan dengan panggilan ini. Y diikat dengan nilai lele, dan sebuah titik lacakbalik baru diset pada fakta adalah(lele, ikan).

10. Karena Y diikat dengan nilai lele, panggilan subgoal berikutnya adalah hidup(lele, di_air). Karena panggilan baru, maka pencarian dimulai dari bagian atas program. Prolog mencoba setiap fakta namun gagal menemukan kecocokan dan subgol dinyatakan gagal.
11. Visual Prolog, kemudian kembali ke titik lacakbalik terakhir yaitu pada adalah(lele, ikan). Variabel yang diikat pada pencocokan ini kemudian dibebaskan. Beranjak dari titik ini, Visual Prolog mencari solusi baru untuk panggilan adalah(Y, ikan).
12. Visual Prolog menemukan kecocokan dengan klausa adalah berikutnya. Y kemudian diikat dengan simbol tuna. Kemudian Visual Prolog mencoba lagi subgoal yang terakhir dengan variabel Y diikat pada nilaituna. Panggilannya menjadi hidup(tuna, di_air). Pencarian kembali dimulai dari bagian atas program karena merupakan panggilan baru dan menemukan kecocokan dengan fakta yang ada dan subgoal terakhir sukses (true).
13. Pada titik ini, *body* dari *rule* dapat_berenang dapat dipenuhi. Visual Prolog mengembalikan nilai Y pada pemanggil (goal) dapat berenang(Apa). Karena Apa diikat nilainya ke Y dan Y diikat nilainya tuna, maka sekarang Apa juga diikat nilainya ke tuna pada goal tersebut. Visual Prolog kan menampilkan jawaban:

Apa=tuna

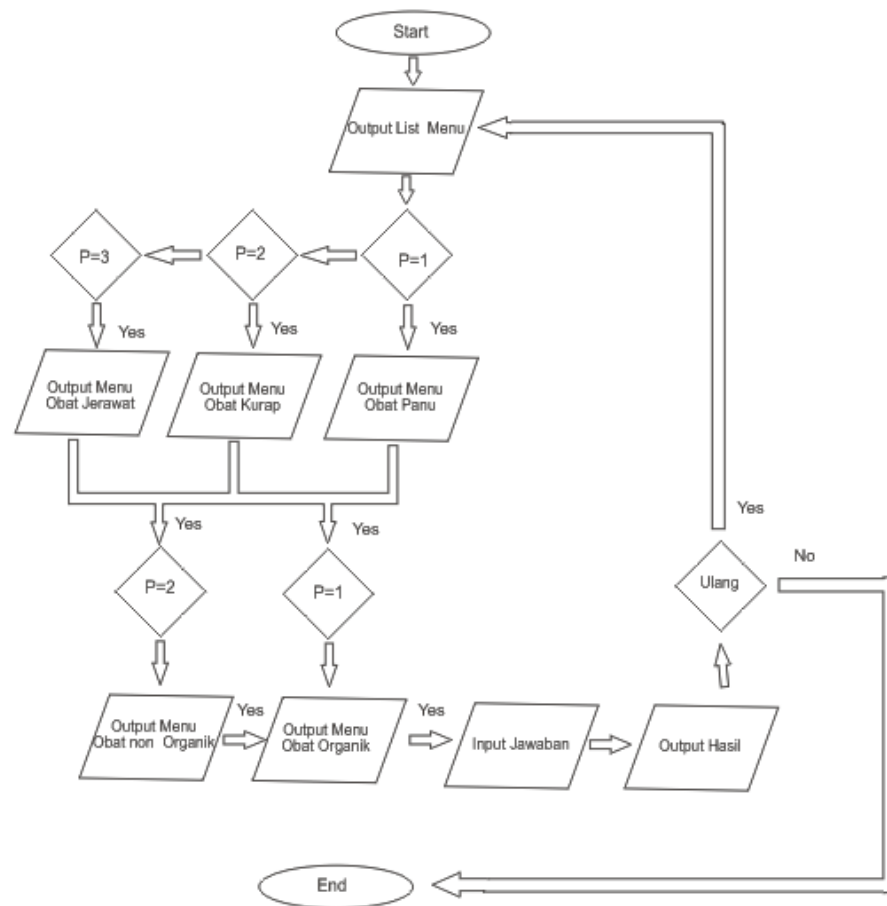
1 Solution

dan program berhenti dengan sukses.

BAB III

METODOLOGI

3.1. Alur Pembuatan Sistem



Gambar 3.1. Alur Pembuatan Sistem dari program

- 3.1.1. Mulai dari domains mendeklarasikan berbagai macam variabel yang akan digunakan untuk fungsi list.
- 3.1.2. Facts untuk mendeklarasikan sebuah fungsi lacak balik.

- 3.1.3. Predicates mendeklarasikan berbagai macam variabel yang akan digunakan.
- 3.1.4. Clauses untuk menjabarkan sebuah variabel yang telah dideklarasikan di predicates dengan menampilkan menggunakan fungsi write.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Tabel Keputusan

Tabel 1.1. Tabel Keputusan pada pemilihan obat

No	Atribut*																		Laboratorium
	O1	O2	O3	O4	O5	O6	O7	O8	O9	N1	N2	N3	N4	N5	N6	N7	N8	N9	
1	Y	Y	Y																Obat Organik Penyakit Panu
2				Y	Y	Y													Obat Organik Penyakit Kurap
3							Y	Y	Y										Obat Organik Jerawat
4										Y	Y	Y							Obat Non Organik Penyakit Panu
5													Y	Y	Y				Obat Non Organik Penyakit Kurap
6																Y	Y	Y	Obat Non Organik Jerawat

Penjelasan :

- O1-O9 adalah Obat Organik dari ketiga jenis penyakit kulit (penyakit panu, penyakit kurap dan jerawat).
- N1-N9 adalah Obat Non Organik dari ketiga jenis penyakit kulit (penyakit panu, penyakit kurap dan jerawat).

4.2. Analisis Aplikasi

=====ANGGOTA KELOMPOK=====		
NAMA : EVI LOLITA APRIYANI	NIM : 1515015128	KELAS : C'2015
NAMA : KELVIN BRAMASTOVIC	NIM : 1515015100	KELAS : C'2015
NAMA : NADYA OCTAVIA ISLAMI	NIM : 1515015100	KELAS : C'2015
===== =====		

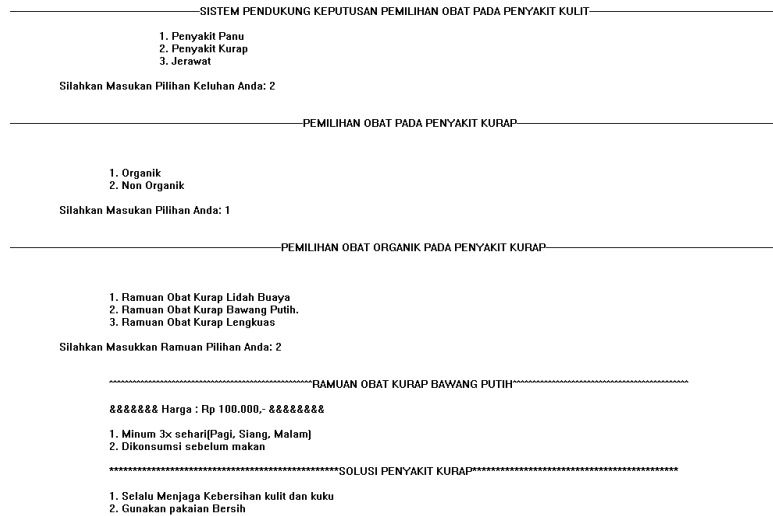
Gambar 4.2.1. Tampilan Credit

(Menampilkan anggota kelompok menggunakan list dengan memanggil di form berbeda).

SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN OBAT PADA PENYAKIT KULIT
1. Penyakit Panu 2. Penyakit Kurap 3. Jerawat
Silahkan Masukan Pilihan Keluhan Anda: 1
PEMILIHAN OBAT PADA PENYAKIT PANU
1. Organik 2. Non Organik
Silahkan Masukan Pilihan Anda: 1
PEMILIHAN OBAT ORGANIK PADA PENYAKIT PANU
1. Ramuan Obat Panu Daun Ketepeng China 2. Ramuan Obat Panu Bawang Putih. 3. Ramuan Obat Panu Bunga Melati
Silahkan Masukan Ramuan Obat Pilihan Anda: 1
RAMUAN OBAT PANU DAUN KETEPENG CHINA
~~~~~
~~~~~ Harga : Rp 250.000,- ~~~~~
1. Minum 3x sehari(Pagi, Siang, Malam) 2. Sebelum diminum, dihangatkan kembali ramuan tersebut 3. Dikonsumsi setelah makan
*****SOLUSI OBAT PANU*****

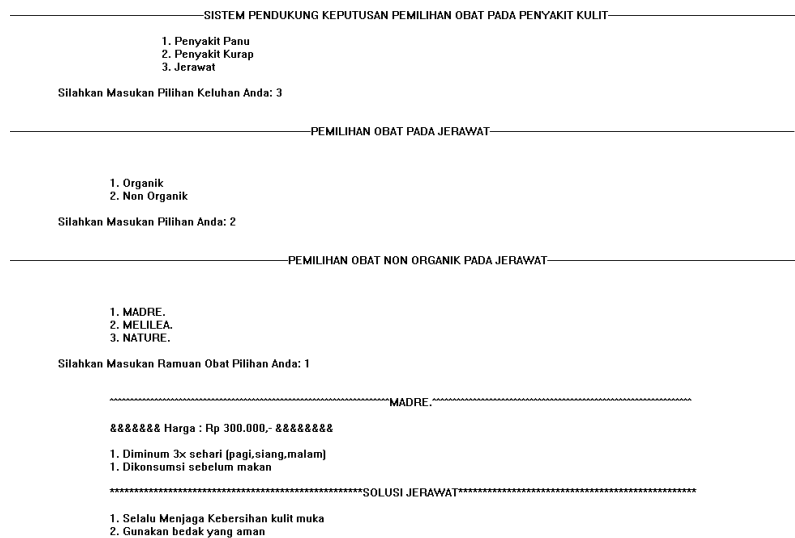
Gambar 4.2.2. Alur sistem pendukung keputusan penyakit panu

(Alur pertama memilih penyakit panu, kemudian memilih obat organik atau non organik lalu memilih ramuan mana yang akan dibeli, kemudian mendapatkan resep dari sistem tersebut. Proses tersebut menggunakan fungsi lacak balik, perulangan dan data objek jamak).



Gambar 4.2.3. Alur sistem pendukung keputusan penyakit kurap

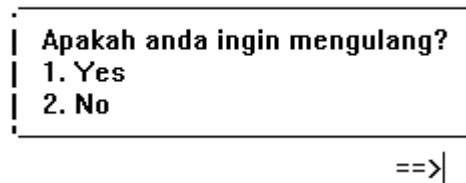
(Alur pertama memilih penyakit kurap, kemudian memilih obat organik atau non organik lalu memilih ramuan mana yang akan dibeli, kemudian mendapatkan resep dari sistem tersebut. Proses tersebut menggunakan fungsi lacak balik, perulangan dan data objek jamak).



Gambar 4.2.4. Alur sistem pendukung keputusan jerawat

(Alur pertama sama dengan penyakit panu dan kurap yaitu memilih pilihan jerawat, kemudian memilih obat organik atau non organik lalu memilih ramuan mana yang akan dibeli, kemudian mendapatkan resep

dari sistem tersebut. Proses tersebut menggunakan fungsi lacak balik, perulangan dan data objek jamak).



Apakah anda ingin mengulang?

1. Yes
2. No

==>

Gambar 4.2.5. Tampilan ingin kembali dalam program

(Proses dimana sistem jika ingin kembali maka pilih yes kemudian akan kembali dari awal, jika tidak maka no untuk mengakhiri eksekusi program).

BAB V

PENUTUP

5.1. Kesimpulan

SPK pemilihan obat merupakan sistem yang dikembangkan untuk membantu apoteker memberikan obat saat jenis obat sangat banyak yang sesuai dengan pasien. Latar belakang dari penggunaan sistem ini adalah mengantisipasi kesalahan dalam pemberian obat.

Proses yang ada didalam sistem ini adalah bagaimana sistem memproses data obat berupa jenis penyakit, kategori obat, harga obat dan solusi agar tidak terjadi penyakit kulit lagi.

5.2. Saran

Program ini masih memiliki kekurangan oleh karena itu kritik beserta saran sangat dibutuhkan untuk perbaikan dan menjadikan lebih baik lagi.

Program ini hanya berkonsentrasi pada 3 jenis penyakit kulit yaitu penyakit panu, kurap dan jerawat, serta pada program ini telah disediakan pilihan-pilihan yang telah memiliki jawaban yang pasti dan tidak memiliki jawaban lain.

DAFTAR PUSTAKA

Sistemphp.com. *Manfaat Sistem Pendukung Keputusan.*

<https://www.sistemphp.com/manfaat-sistem-pendukung-keputusan/>.

Diakses pada tanggal 02 Mei 2017.

Sistemphp.com. *Sistem Pendukung Keputusan Pemilihan Obat.*

<https://www.sistemphp.com/program-sistem-pendukung-keputusan-pemilihan-obat-php/> . Diakses pada tanggal 02 Mei 2017

LAMPIRAN

1. Source Code

DOMAINS

daftar = string*

input = integer

nama,kelas=string

nim=integer

**kelompok=anggota1(nama,nim,kelas);anggota2(nama,nim,kelas);anggota3(nam
a,nim,kelas)**

FACTS

nondeterm cetak_daftar(daftar)

nondeterm data_kelompok(kelompok)

PREDICATES

nondeterm detail_data_kelompok

nondeterm credit

nondeterm pilih(input)

nondeterm pilih1

nondeterm pilih2

nondeterm pilih3

nondeterm panu(input)

nondeterm panuobat1

nondeterm panuobat2

nondeterm obatpanuorganik1

nondeterm obatpanuorganik2

nondeterm obatpanuorganik3
nondeterm obatpanunonorganik1
nondeterm obatpanunonorganik2
nondeterm obatpanunonorganik3
nondeterm solusipanu1(input)
nondeterm solusipanu2(input)
nondeterm kurap(input)
nondeterm kurapobat1
nondeterm kurapobat2
nondeterm obatkuraporganik1
nondeterm obatkuraporganik2
nondeterm obatkuraporganik3
nondeterm obatkurapnonorganik1
nondeterm obatkurapnonorganik2
nondeterm obatkurapnonorganik3
nondeterm solusikurap1(input)
nondeterm solusikurap2(input)
nondeterm jerawat(input)
nondeterm jerawatobat1
nondeterm jerawatobat2
nondeterm obatjerawatorganik1
nondeterm obatjerawatorganik2
nondeterm obatjerawatorganik3
nondeterm obatjerawatnonorganik1
nondeterm obatjerawatnonorganik2
nondeterm obatjerawatnonorganik3
nondeterm solusijerawat1(input)
nondeterm solusijerawat2(input)

nondeterm perulangan(input)
nondeterm perulangan1(input)
nondeterm perulangan2(input)
nondeterm perulangan3(input)
nondeterm perulangan4(input)
nondeterm perulangan5(input)
nondeterm perulangan6(input)
nondeterm perulangan7(input)
nondeterm perulangan8(input)
nondeterm perulangan9(input)
nondeterm perulangan10(input)
nondeterm perulangan11(input)
nondeterm perulangan12(input)
nondeterm perulangan13(input)
nondeterm perulangan14(input)
nondeterm perulangan15(input)
nondeterm perulangan16(input)
nondeterm perulangan17(input)
nondeterm lekassembuh
nondeterm tampil
rx(integer,integer)

CLAUSES

rx(Y,Y):- !.

rx(_,_):- fail.

person('Evi Lolita ',1515015128).

person('Kelvin Bramastovic',1515015000).

```
person("Nadya Octavia Islami ",1515015147).
```

credit :-

```
write("\n"),
```

```
write('++++++  
++++++  
+++++\n\n'),
```

```
write("-----").
```

```
write("SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN OBAT PADA
PENYAKIT KULIT"),
```

```
write("-----\n\n").
```

```
write("\n").
```

```
write("\n"),
```

```
write("\t\t1. Penyakit Panu\n"),
```

```
write("\t\t2. Penyakit Kurap\n"),
```

```
write("\t\t3. Jerawat\n"),
```

```
write("\n"),
```

```
write("\tSilahkan Masukan Pilihan Keluhan Anda: "),
```

readint(P),pilih(P).

pilih(P):- rx(P,1),pilih1.

pilih(P):- rx(P,2),pilih2.

pilih(P):- rx(P,3),pilih3.

pilih(_):- credit.

pilih1:-

```
write("\n"),
```

```

write("\n"),
write("-----
"),
write("PEMILIHAN OBAT PADA PENYAKIT PANU"),
write("-----
\n\n"),
write("\n"),
write("\n"),
write("\t\t1. Organik\n"),
write("\t\t2. Non Organik\n"),
write("\n"),
write("\tSilahkan Masukan Pilihan Anda: "),
readint(M),panu(M).

panu(M):- rx(M,1),panuobat1.
panu(M):- rx(M,2),panuobat2.
panu(_):- panuobat1.

panuobat1 :-
write("\n"),
write("\n"),
write("-----"),
write("PEMILIHAN OBAT ORGANIK PADA PENYAKIT PANU"),
write("-----
\n\n"),
write("\n"),
write("\n"),
write("\t\t1. Ramuan Obat Panu Daun Ketepeng China \n"),

```

```

write("\t\t2. Ramuan Obat Panu Bawang Putih.\n"),
write("\t\t3. Ramuan Obat Panu Bunga Melati\n"),
write("\n"),
write("\tSilahkan Masukan Ramuan Obat Pilihan Anda: "),
readint(P),solusipanu1(P).

```

```

solusipanu1(M):- rx(M,1),obatpanuorganik1.
solusipanu1(M):- rx(M,2),obatpanuorganik2.
solusipanu1(M):- rx(M,3),obatpanuorganik3.
solusipanu1(_):- obatpanuorganik1.

```

obatpanuorganik1:-

```

write("\n"),
write("\n"),
write("\t\t^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^"),
write("RAMUAN OBAT PANU DAUN KETEPENG CHINA"),
write("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n"),
write("\t\t&&&&&& Harga : Rp 250.000,- &&&&&&\n\n"),
write("\t\t1. Minum 3x sehari(Pagi, Siang, Malam)),nl,
write("\t\t2. Sebelum diminum, dihangatkan kembali ramuan tersebut"),nl,
write("\t\t3. Dikonsumsi setelah makan"),nl,
write("\n"),
write("\t\t*****"),
write("SOLUSI OBAT PANU"),
write("*****\n\n"),
write("\n"),
write("\t\t1. Selalu Menjaga Kebersihan kulit dengan cara mandi minimal 2 kali
sehari"),nl,

```

```
write("\t\t2. Gunakan pakaian Bersih yang menyerap keringat"),nl,  
write("\t\t3. Dan Minum Ramuan Obat Panu Daun Ketepeng China"),nl,  
write("\n"),  
write("\n"),  
write("\t\t\t\t\t\t-----.\n"),  
write("\t\t\t\t\t\t| Apakah anda ingin mengulang? |"),  
write("\n"),  
write("\t\t\t\t\t\t| 1. Yes                               |"),  
write("\n"),  
write("\t\t\t\t\t\t| 2. No                                   |\n"),  
write("\t\t\t\t\t\t'-----'"),  
write("\n"),  
readint(A),perulangan(A).  
  
perulangan(A):- rx(A,1),credit.  
perulangan(A):- rx(A,2),lekassembuh.  
perulangan(_):- obatpanuorganik1.  
  
obatpanuorganik2:-  
write("\n"),  
write("\n"),  
write("\t\t^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  
^^"),  
write("RAMUAN OBAT PANU BAWANG PUTIH"),  
write("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n"),  
write ("\t\t&&&&& Harga : Rp 200.000,- &&&&&\n\n"),  
write("\t\t1. Minum 3x sehari(Pagi, Siang, Malam)'),nl,  
write("\t\t2. Dikonsumsi sebelum makan'),nl,
```

```
write("\n"),
write("\t\t*****"),
write("SOLUSI OBAT PANU"),
write("*****\n\n"),
write("\n"),
write("\t\t1. Selalu Menjaga Kebersihan kulit dengan cara mandi minimal 2 kali
sehari"),nl,
write("\t\t2. Gunakan pakaian Bersih yang menyerap keringat"),nl,
write("\t\t3. Dan Minum Ramuan Obat Panu Bawang Putih "),nl,
write("\n"),
write("\n"),
write("\n"),
write("\t\t\t\t\t-----.\n"),
write("\t\t\t\t\t Apakah anda ingin mengulang? |"),
write("\n"),
write("\t\t\t\t\t 1. Yes |"),
write("\n"),
write("\t\t\t\t\t 2. No |n"),
write("\t\t\t\t\t'-----'),
write("\n"),
readint(A),perulangan1(A).

perulangan1(A):- rx(A,1),credit.
perulangan1(A):- rx(A,2),lekassembuh.
perulangan1(_):- obatpanuorganik2.

obatpanuorganik3:-
write("\n"),
```

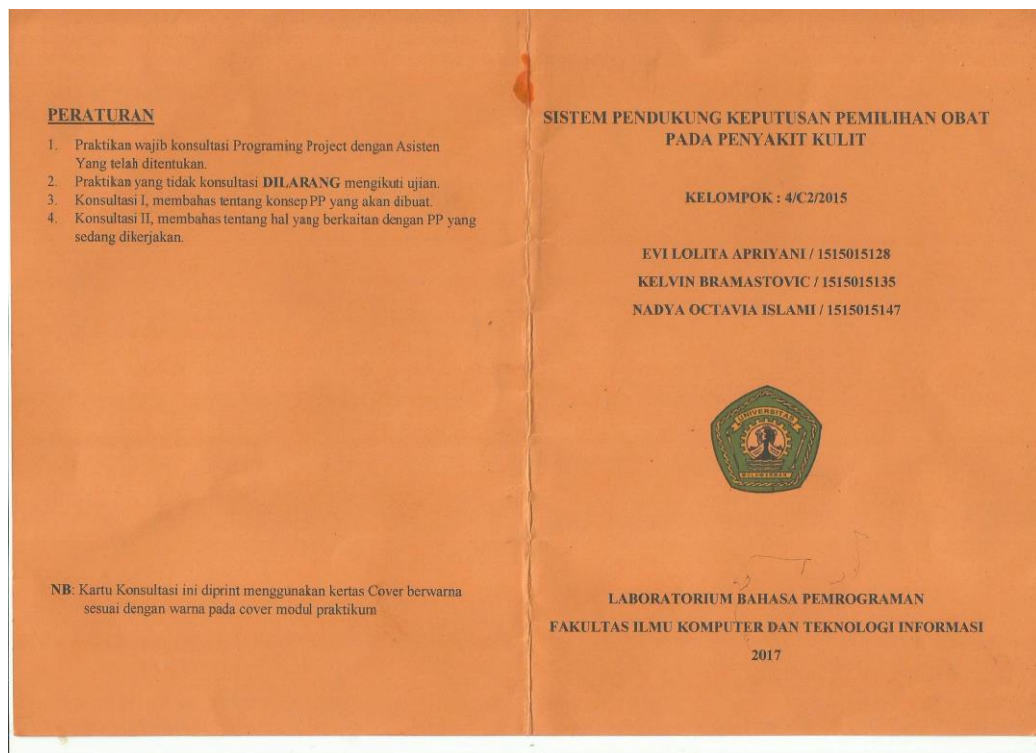
```
write("\n"),
write('\t\t^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^'),
write("RAMUAN OBAT PANU BUNGA MELATI"),
write('^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n'),
write ('\t&&&&& Harga : Rp 150.000,- &&&&&\n\n'),
write('\t1. Minum 3x sehari(Pagi, Siang, Malam)'),nl,
write('\t2. Sebelum diminum, dihangatkan kembali ramuan tersebut'),nl,
write('\t3. Dikonsumsi setelah makan'),nl,
write('\n'),
write('\t*****'),
write("SOLUSI OBAT PANU"),
write('*****\n\n'),
write('\n'),
write('\t1. Selalu Menjaga Kebersihan kulit dengan cara mandi minimal 2 kali
sehari'),nl,
write('\t2. Gunakan pakaian Bersih yang menyerap keringat'),nl,
write('\t3. Dan Minum Ramuan Obat Panu Bunga Melati '),nl,
write('\n'),
write('\n'),
write('\n'),
write('\t\t\t\t\t-----.\n'),
write('\t\t\t\t\t Apakah anda ingin mengulang? |'),
write('\n'),
write('\t\t\t\t\t 1. Yes                                     |'),
write('\n'),
write('\t\t\t\t\t 2. No                                       |\n'),
write('\t\t\t\t\t'-----'''),
```

```
write("\n"),
readint(A),perulangan2(A).


perulangan2(A):- rx(A,1),credit.
perulangan2(A):- rx(A,2),lekassembuh.
perulangan2(_):- obatpanuorganik3.
```

2. Kartu Konsul

a. Halaman Depan



b. Halaman Belakang

Tanggal Konsultasi : 01-05-2017	Tanggal Konsultasi : 03-05-2017
Uraian : - Perbaikan program - Revisi Alur Program	Uraian : - Perbaikan program - Revisi Laporan
Tanda Tangan Asisten : 	Tanda Tangan Asisten : 