**LAPORAN PRAKTIKUM KECERDASAN BUATAN**  
**[SISTEM PAKAR PENDETEKSI SINDROM DEPRESI]**

****

**Disusun Oleh :**

Hadi Yoga 1515015133

Abdul Aziz 1515015150

Muhammad Jodi Saputra 1515015179

**Asisten Praktikum :**

Annisa Nur Afiyah Alvian Nur Wahyudhi

1415015064 1415015066

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI DAN KOMUNIKASI**  
**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**  
**UNIVERSITAS MULAWARMAN**  
**2017**

**KATA PENGANTAR**

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa yang telah melimpahkan rahmat, dan hidayahNya sehingga kami dapat menyelesaikan laporan akhir mata kuliah Pratikum Kecerdasan Buatan mengenai Sistem Penentu pakar pendeteksi sindrom depresi. Penyusunan laporan ini sebagai salah satu persyaratan tugas akhir mata kuliah Pratikum Kecerdasan Buatan.

Laporan ini kami susun dengan semaksimal mungkin dan tidak lepas dari bantuan berbagai pihak sehingga dapat memperlancar pembuatan laporan ini. Untuk itu kami menyampaikan banyak terimakasih kepada semua pihak yang telah berkontribusi dalam pembuatan laporan ini. Terlepas dari semua itu, kami menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasanya. Oleh karena itu dengan tangan terbuka kami menerima segala saran dan kritik dari pembaca agar kami lebih baik lagi ke depannya.

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

6 Mei 2017

Kelompok 7

**DAFTAR ISI**

**Kata Pengantar** ............................................................................................ ii  
**Daftar Isi** ...................................................................................................... iii  
**Daftar Tabel** ................................................................................................ iv  
**Daftar Gambar** ........................................................................................... v  
**BAB I Pendahuluan** ..................................................................................... 1

1.1. Latar belakang 1

1.2. Rumusan masalah 1

1.3. Batasan masalah 1

1.4. Tujuan dan Manfaat 1   
**BAB II Landasan Teori** ............................................................................................................. 2

2.1 Fungsi-Fungsi Visual Prolog yang Digunakan ............................................................ 2

2.1.1 Unifikasi dan Lacakbalik 3

2.1.2 Data Objek Sederhana........................................................................................ 3

2.1.3 Section dasar Visual Prolog 4

2.1.4 Section Facts 5

2.1.5 Perulangan dan Rekursi 6

2.1.6 List 6  
**BAB III Metodologi**  7

3.1 Alur Pembuatan Sistem 7  
**BAB IV Hasil dan Pembahasan** 8

4.1 Tabel Keputusan 8

4.2 Analisis Aplikasi 8  
**BAB V Penutup** 11

5.1 Kesimpulan 11

5.2 Saran 11  
**Daftar Pustaka** 12

**Lampiran** 13

1. Source code 13
2. Kartu konsul 19

**DAFTAR TABEL**

Tabel 4.1 Tabel Keputusan 7

Tabel 4.2 Tabel Keputusan 7

Tabel 4.3 Tabel Keputusan 8

**DAFTAR GAMBAR**

Gambar 4.2.1 Inputan Data 8

Gambar 4.2.2 Hasil Reward jika semua hasil pilhan Y terjawab 9

Gambar 4.2.3 Hasil Reward jika pilihan Y di jawab 3 kali 9

Gambar 4.2.4 Hasil Reward jika pilihan Y di jawab 2 kali 10

Gambar 4.2.5 Punishment pertama 10

Gambar 4.2.6 Punishment kedua 10

**BAB I**  
**PENDAHULUAN**

**1.1 Latar Belakang**

Kecerdasan Buatan (*Artificial Intelligence*) menyebabkan lahirnya berbagai teknologi yang dapat dikatakan bersifat cerda, misalnya permainan (*game*), sistem pakar (*expert system*), atau sistem penentu keputusan. Kecerdasan buatan ini dapat dimanfaatkan diberbagai bidang kebutuhan manusia, contohnya pada program yang kelompok kami buat. Kelompok kami membuat program tentang Sistem Pakar Pendeteksi sindrom depresi, yang pada program ini dibuat untuk mendeteksi sindrom depresi dimana program memudahkan orang untuk mendeteksi gejala pada sindrom depresi yang terbagi menjadi depresi ringan,depresi sedang,depresi akut

**1.2 Rumusan Masalah**

Program kami memiliki perumusan masalah, yaitu bagaimana suatu program mampu membantu menyelesaikan masalah dalam menentukan jenis depresi.  
**1.3 Batasan Masalah**

1. Pada program yang kami buat tidak dapat benar-benar mendeteksi gejala depresi secara detil Karena Batasan dari program visual prolog berbentuk procedural dalam program berstruktur procedural, apabila masukkan dari program tidak ada sama sekali, maka program akan berjalan kembali tanpa solusi
2. Dalam penggunaan nya di kehidupan nyata aplikasi ini tidak dapat berjalan sesuai fungsi Karena pendeteksian depresi tidak seharusnya di golongkan berdasarkan pertanyaan yang diberikan.

**1.4 Tujuan dan Manfaat**

Tujuan nya memudahkan untuk mendeteksi sindrom depresi yang ada pada manusia serta solusi dari gejala yang sudah diketahui pada saat program ini menampilkan hasil

**BAB II**   
**LANDASAN TEORI**

**2.1 Fungsi Visual Prolog yang digunakan**  
**2.1.1 Kecerdasan Buatan**

Kecerdasan Buatan atau kecerdasan yang ditambahkan kepada suatu sistem yang bisa diatur dalam konteks ilmiah atau Intelegensi Artifisial didefinisikan sebagai kecerdasan entitas ilmiah. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin komputer agar dapat melakukan pekerjaan seperti yang dilakukan manusia

**2.1.2 Visual Prolog**

Prolog menggunakan teknik pencaharian yang disebut heuristik dengan menggunakan pohon logika. Prolog banyak dikembangkan *di University of Melbourne oleh Lee Naish* dan *John Loyd*. Prolog dalam perkembangannya telah dikombinasikan dengan berbagai bahasa pemrograman terutama functional programming. Bahasa prolog ini secara intensif digunakan pada proyek komputer generasi ke-5 di Jepang. Walaupun demikian sebetulnya penggunaannya tidak terbatas untuk AI saja. Prolog adalah bahasa pemrograman logika atau di sebut juga sebagai bahasa non-procedural. Namanya diambil dari bahasa Perancis programmation en logique (Pemrograman logika). Bahasa ini diciptakan oleh *Alain Colmerauer* dan *Robert Kowalski* sekitar tahun 1972 dalam upaya untuk menciptakan suatu bahasa pemrograman yang memungkinkan pernyataan logika alih-alih rangkaian perintah untuk dijalankan komputer. Berbeda dengan bahasa pemrograman yang lain, yang menggunakan algoritma konvensionl sebagai teknik pencariannya seperti Pada *Delphi,Pascal, BASIC, COBOL* dan bahasa pemrograman yang sejenisnya, maka prolog menggunakan teknik pencarian yang di sebut *heuristik (heutistic)* dengan menggunakan pohon logika.  
  
**2.1.3 *Section* dalam Program**

Konsep yang digunakan untuk menyusun program dan penerapan visual prolog ini adalah Dasar-dasar Prolog, Data *Object* Sederhana, Lacakbalik, dan *SectionFacts*.

**2.1.3.1 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.

**2.1.3.2 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 diidentifikasikan sebagai sebuah *object* bukan *subject* yang nilainya bisa bervariasi, sepertisebuah karakter(*char*), angka(*integer*atau*real*) atau sebuah atom(simbolataustring).

**2.1.3.3 *Section* Dasar Visual Prolog**

Secara umum, program Visual Prolog terdiri dariempat *section*dasar, yaitu *section clauses, section predicates,sectiondomains,*danterakhir*sectiongoal*. Berikut akan dijelaskan secara singkat masing masing *section* tersebut.

**a.*SectionClauses***  
*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.

**b. *Section Predicates***

Sebelum mendefinsikan 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.

**c.*SectionDomains***  
*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.

**d. *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 *sectionfacts*.

**e. *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.

**2.1.3.4*SectionFacts***

*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 kalua di deklarasikan pada *section predicates*.

**2.1.3.5 Perulangan dan 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 melewatkan (*passing*) pencacah, total, dan hasil sementara sebagai argumen dari satu iterasi ke iterasi berikutnya. Berikut ini merupakan contoh program untuk mencari factorial dari suatu angka.

**2.1.3.6 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 \*/

**BAB III**  
**METODOLOGI**

**3.1 Alur Pembuatan Sistem**

Penelitian ini akan dilaksanakan dengan mengikuti beberapa langkah yang kami gunakan, yaitu : Tahap inputan : yaitu langkah pertama di dalam program kami dengan memasukkan inputan nama, jenis kelamin dan jika sudah dilengkapi akan lanjut ke tahap selanjutnya. Tahap kedua : di tahap ini program kami akan menampilkan sebuah pertanyaan-pertanyaan yang akan dijalankan dan dijawab user dan menampilkan pilihan Y/T. Tahap ketiga : jika telah melalui tahap kedua, jika user banyak menginputkan pilihan Y program akan mengeluarkan outputan atau hasil yang akan menampilkan jenis depresi, jika dari semua pertanyaan tersebut di jawab T, rogram langsung mengeluarkan hasil atau solusi.

**BAB IV**  
**HASIL DAN PEMBAHASAN**

**4.1 Tabel keputusan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pertanyaan** | **Pilihan jawaban** | | **Hasil** |
| **Y** | **T** | **Depresi akut** |
| 1. **Saya merasa tidak berharga** | **√** |  |
| 1. **Saya merasa lelah bahkan setelah istirahat yang cukup** | **√** |  |
| 1. **Saya merasa hampa secara terus-menerus** | **√** |  |
| 1. **Saya selalu terpikir tentang mengakhiri hidup saya** | **√** |  |
| 1. **Masalah-masalah dalam hidup saya cenderung semakin bertambah, mengarah ke kondisi depresi karena saya merasa saya tidak dapat mengendalikan apakah suatu masalah telah terselesaikan dengan baik** | **√** |  |

**4.2 Tabel keputusan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pertanyaan** | **Pilihan jawaban** | | **Hasil** |
| **Y** | **T** | **Depresi sedang** |
| 1. **Saya merasa bahwa saya hanya bisa menjadi beban bagi orang-orang di sekitar saya** | **√** |  |
| 1. **Saya merasa ingin menangis tanpa ada alasan yang jelas** | **√** |  |
| 1. **Saya terus berpikir tentang bagaimana hal-hal yang berlangsung dalam kehidupan saya tiap malam** | **√** |  |
| 1. **Saya merasa tidak dapat memulai tugas atau proyek penting yang harus saya kerjakan** | **√** |  |

**4.3 Tabel keputusan**

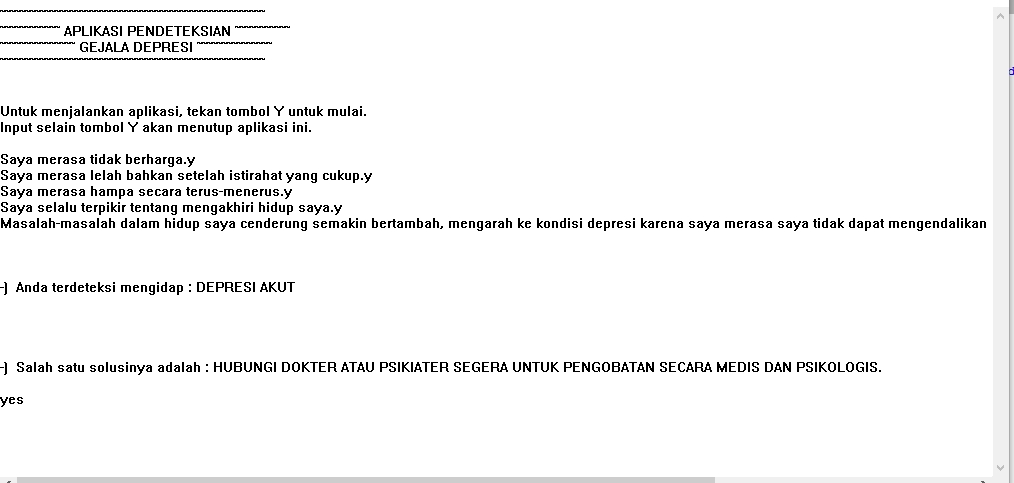
|  |  |  |  |
| --- | --- | --- | --- |
| **Pertanyaan** | **Pilihan jawaban** | | **Hasil** |
| **Y** | **T** | **Depresi ringan** |
| 1. **Saya mengacaukan apa saja yang saya sentuh** | **√** |  |
| 1. **Mengambil keputusan terasa seperti siksaan bagi saya** | **√** |  |
| 1. **Saya merasa melambat, baik secara mental maupun secara fisik.** | **√** |  |

**Catatan:**

1. jika sesi pernyataan 1 di jawab “Y” secara keselurahan maka hasil nya adalah depresi akut dan jika salah satu atau lebih jawaban “T” maka hasil nya belum tentu akut dan akan di lanjutkan ke sesi pernyataan 2
2. jika sesi pernyataan 2 di jawab “Y” secara keselurahan maka hasil nya adalah depresi sedang dan jika salah satu atau lebih jawaban “T” maka hasil nya belum tentu sedang dan akan di lanjutkan ke sesi pernyataan 3
3. jika sesi pernyataan 3 di jawab “Y” atau “T” maka hasil nya secara otomatis Normal

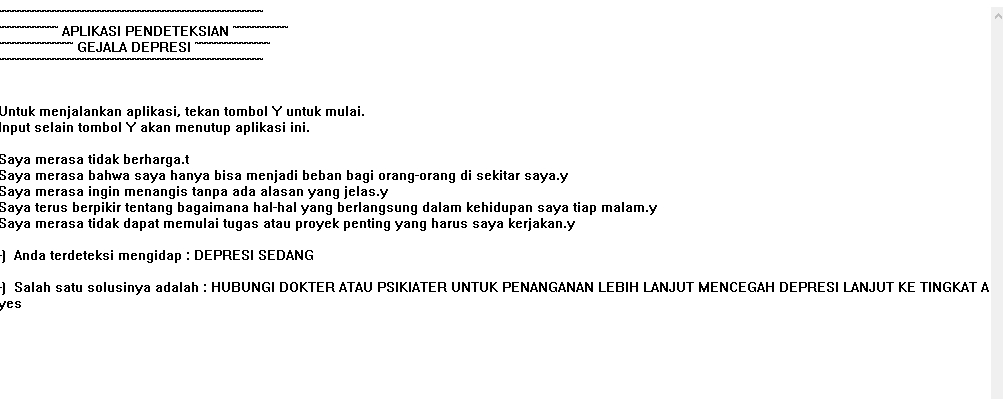
**4.2 Analisis Aplikasi**

a. Tahap awal jika kita menginputkan “Y” semua maka hasil akan DEPRESI AKUT



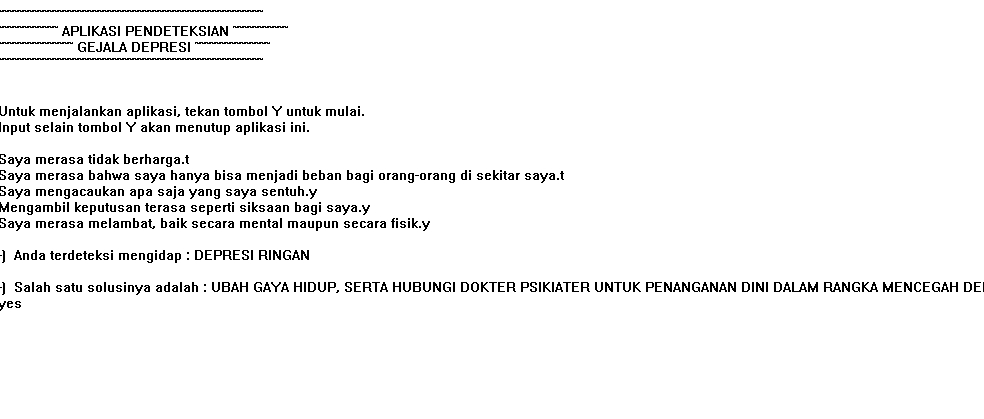
Gambar 4.2.1 Hasil Pendeteksi

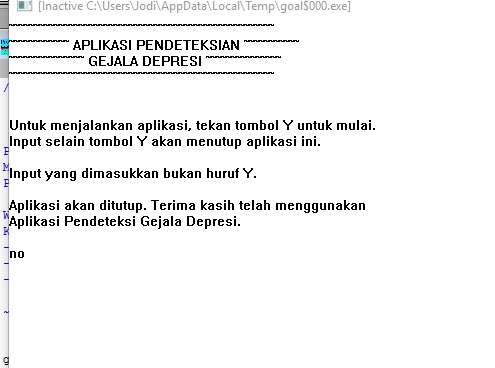
b. jika salah satu jawaban “T” maka akan melanjutkan ke pertanyaan ke 2



Gambar 4.2.2 Hasil Pendeteksi 2

c. Jika dari pertanyaan 1 dan 2 di jawab “T” maka akan melanjutkan ke pernyataan ke 3 yang hasilnya DEPRESI RINGAN

  
Gambar 4.2.3 HAsil pendeteksi 3

d.Jika jawaban yang kita input selain huruf “Y” dan “T” maka program akan tertutup  
****

Gambar 4.2.4 HAsil jika inputan selain “Y” atau“T”

**BAB V**  
**PENUTUP**

**5.1 Kesimpulan**

jadi dalam program yang dibuat ini dapat digunakan untuk setiap orang yang ingin mengetahui kadar depresi yang sudah di golongkan serta memberikan solusi sesuai dengan kadar depresi nya, solusi tersebut dimaksudkan untuk pencegahan dari kadar depresi

**5.2 Saran**

Tak lupa juga penulis mencantumkan saran karena di dalam laporan ini masih terdapat banyak kekurangan oleh sebab itu penulis masih sangat membutuhkan saran bagi para pembaca. Diharapkan juga agar program ini dapat lebih dikembangkan untuk ke depannya dan bisa digunakan sebagaimana semestinya.

**DAFTAR PUSTAKA**

<https://www.psychologytoday.com/tests/health/depression-tes> (3 Mei 2017)

<https://psychcentral.com/quizzes/depression-quiz.htm> (3 Mei 2017)

**LAMPIRAN**  
**Source Code**

domains

gejala = symbol

symptom = symbol

pernyataan = string

respons = char

facts

ya\_ya(symptom)

tidak\_tidak(symptom)

predicates

nondeterm mulai

nondeterm prompt

nondeterm pernyataan(symptom)

nondeterm ya(pernyataan,symptom)

nondeterm tidak(pernyataan,symptom)

nondeterm deteksi(gejala)

terdeteksi(gejala)

nondeterm penanganan(gejala)

nondeterm statement(pernyataan,symptom,respons)

simpan(symptom,respons)

jawaban(char,char)

nondeterm pilihan(char)

clauses

/\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ START ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\*/

jawaban(Y,Y):-!.

jawaban(\_,\_):-fail.

mulai:-

write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"),nl,

write("~~~~~~~~~~~~ APLIKASI PENDETEKSIAN ~~~~~~~~~~~"),nl,

write("~~~~~~~~~~~~~~~ GEJALA DEPRESI ~~~~~~~~~~~~~~~"),nl,

write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"),nl,nl,nl,

write("Untuk menjalankan aplikasi, tekan tombol Y untuk mulai."),nl,

write("Input selain tombol Y akan menutup aplikasi ini."),nl,nl,

readchar(A), pilihan(A).

pilihan(A):-jawaban(A,'y'), prompt.

pilihan(A):-jawaban(A,'Y'), prompt.

pilihan(\_):-

write("Input yang dimasukkan bukan huruf Y."),nl,nl,

write("Aplikasi akan ditutup. Terima kasih telah menggunakan"),nl,

write("Aplikasi Pendeteksi Gejala Depresi."),nl,nl,fail.

prompt:-

deteksi(\_),!,

save("deteksi.txt").

prompt:-

write("Selamat, anda tidak memiliki gejala mengidap sindrom depresi."),nl.

ya(\_,Symptom):-

ya\_ya(Symptom),!.

ya(Pernyataan,Symptom):-

not(tidak\_tidak(Symptom)),

statement(Pernyataan,Symptom,Respons),

Respons='y'.

tidak(\_,Symptom):-

tidak\_tidak(Symptom),!.

tidak(Pernyataan,Symptom):-

not(ya\_ya(Symptom)),

statement(Pernyataan,Symptom,Respons),

Respons='t'.

statement(Pernyataan,Symptom,Respons):-

write(Pernyataan),

readchar(Respons),

write(Respons),nl,

simpan(Symptom,Respons).

simpan(Symptom,'y'):-

asserta(ya\_ya(Symptom)).

simpan(Symptom,'t'):-

asserta(tidak\_tidak(Symptom)).

/\* ~~~~~ SET LIST OF QUESTIONS ASKED ~~~~ \*/

pernyataan(Symptom):-

ya\_ya(Symptom),!.

pernyataan(Symptom):-

tidak\_tidak(Symptom),!,fail.

pernyataan(set1\_1):-

ya("Saya merasa tidak berharga.",set1\_1).

pernyataan(set1\_2):-

ya("Saya merasa lelah bahkan setelah istirahat yang cukup.",set1\_2).

pernyataan(set1\_3):-

ya("Saya merasa hampa secara terus-menerus.",set1\_3).

pernyataan(set1\_4):-

ya("Saya selalu terpikir tentang mengakhiri hidup saya.",set1\_4).

pernyataan(set1\_5):-

ya("Masalah-masalah dalam hidup saya cenderung semakin bertambah, mengarah ke kondisi depresi karena saya merasa saya tidak dapat mengendalikan apakah suatu masalah telah terselesaikan dengan baik.",set1\_5).

pernyataan(set2\_1):-

ya("Saya merasa bahwa saya hanya bisa menjadi beban bagi orang-orang di sekitar saya.",set2\_1).

pernyataan(set2\_2):-

ya("Saya merasa ingin menangis tanpa ada alasan yang jelas.",set2\_2).

pernyataan(set2\_3):-

ya("Saya terus berpikir tentang bagaimana hal-hal yang berlangsung dalam kehidupan saya tiap malam.",set2\_3).

pernyataan(set2\_4):-

ya("Saya merasa tidak dapat memulai tugas atau proyek penting yang harus saya kerjakan.",set2\_4).

pernyataan(set3\_1):-

ya("Saya mengacaukan apa saja yang saya sentuh.",set3\_1).

pernyataan(set3\_2):-

ya("Mengambil keputusan terasa seperti siksaan bagi saya.",set3\_2).

pernyataan(set3\_3):-

ya("Saya merasa melambat, baik secara mental maupun secara fisik.",set3\_3).

/\*~~~~~~~ DETEKSI ~~~~~~~~\*/

deteksi("Severe Depression"):-

pernyataan(set1\_1),

pernyataan(set1\_2),

pernyataan(set1\_3),

pernyataan(set1\_4),

pernyataan(set1\_5),

terdeteksi("Depresi Akut"),

penanganan("Hubungi dokter atau psikiater segera untuk pengobatan secara medis dan psikologis.").

deteksi("Moderate Depression"):-

pernyataan(set2\_1),

pernyataan(set2\_2),

pernyataan(set2\_3),

pernyataan(set2\_4),

terdeteksi("Depresi Sedang"),

penanganan("Hubungi dokter atau psikiater untuk penanganan lebih lanjut mencegah depresi lanjut ke tingkat akut.").

deteksi("Mild Depression"):-

pernyataan(set3\_1),

pernyataan(set3\_2),

pernyataan(set3\_3),

terdeteksi("Depresi Ringan"),

penanganan("Ubah gaya hidup, serta hubungi dokter psikiater untuk penanganan dini dalam rangka mencegah depresi berakumulasi ke tingkat lanjut.").

/\* ~~~~~~~~ PENANGANAN LEBIH LANJUT ~~~~~~~ \*/

terdeteksi(Gejala):-

upper\_lower(AGejala, Gejala),nl,

write("-) Anda terdeteksi mengidap : ",AGejala),nl.

penanganan(Gejala):-

upper\_lower(AGejala, Gejala),nl,

write("-) Salah satu solusinya adalah : ",AGejala),nl.

goal

mulai.

nondeterm penanganan(gejala)

nondeterm statement(pernyataan,symptom,respons)

simpan(symptom,respons)

jawaban(char,char)

nondeterm pilihan(char)

clauses

/\*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ START ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\*/

jawaban(Y,Y):-!.

jawaban(\_,\_):-fail.

mulai:-

write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"),nl,

write("~~~~~~~~~~~~ APLIKASI PENDETEKSIAN ~~~~~~~~~~~"),nl,

write("~~~~~~~~~~~~~~~ GEJALA DEPRESI ~~~~~~~~~~~~~~~"),nl,

write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"),nl,nl,nl,

write("Untuk menjalankan aplikasi, tekan tombol Y untuk mulai."),nl,

write("Input selain tombol Y akan menutup aplikasi ini."),nl,nl,

readchar(A), pilihan(A).

pilihan(A):-jawaban(A,'y'), prompt.

pilihan(A):-jawaban(A,'Y'), prompt.

pilihan(\_):-

write("Input yang dimasukkan bukan huruf Y."),nl,nl,

write("Aplikasi akan ditutup. Terima kasih telah menggunakan"),nl,

write("Aplikasi Pendeteksi Gejala Depresi."),nl,nl,fail.

prompt:-

deteksi(\_),!,

save("deteksi.txt").

prompt:-

write("Selamat, anda tidak memiliki gejala mengidap sindrom depresi."),nl.

ya(\_,Symptom):-

ya\_ya(Symptom),!.

ya(Pernyataan,Symptom):-

not(tidak\_tidak(Symptom)),

statement(Pernyataan,Symptom,Respons),

Respons='y'.

tidak(\_,Symptom):-

tidak\_tidak(Symptom),!.

tidak(Pernyataan,Symptom):-

not(ya\_ya(Symptom)),

statement(Pernyataan,Symptom,Respons),

Respons='t'.

statement(Pernyataan,Symptom,Respons):-

write(Pernyataan),

readchar(Respons),

write(Respons),nl,

simpan(Symptom,Respons).

prompt:-

deteksi(\_),!,

save("deteksi.txt").

prompt:-

write("Selamat, anda tidak memiliki gejala mengidap sindrom depresi."),nl.

ya(\_,Symptom):-

ya\_ya(Symptom),!.

ya(Pernyataan,Symptom):-

not(tidak\_tidak(Symptom)),

statement(Pernyataan,Symptom,Respons),

Respons='y'.

tidak(\_,Symptom):-

tidak\_tidak(Symptom),!.

tidak(Pernyataan,Symptom):-

not(ya\_ya(Symptom)),

statement(Pernyataan,Symptom,Respons),

Respons='t'.

statement(Pernyataan,Symptom,Respons):-

write(Pernyataan),

readchar(Respons),

write(Respons),nl,

simpan(Symptom,Respons).

simpan(Symptom,'y'):-

asserta(ya\_ya(Symptom)).

simpan(Symptom,'t'):-

asserta(tidak\_tidak(Symptom)).

/\* ~~~~~ SET LIST OF QUESTIONS ASKED ~~~~ \*/

pernyataan(Symptom):-

ya\_ya(Symptom),!.

pernyataan(Symptom):-

tidak\_tidak(Symptom),!,fail.

pernyataan(set1\_1):-

ya("Saya merasa tidak berharga.",set1\_1).

pernyataan(set1\_2):-

ya("Saya merasa lelah bahkan setelah istirahat yang cukup.",set1\_2).

pernyataan(set1\_3):-

ya("Saya merasa hampa secara terus-menerus.",set1\_3).

pernyataan(set1\_4):-

ya("Saya selalu terpikir tentang mengakhiri hidup saya.",set1\_4).

pernyataan(set1\_5):-

ya("Masalah-masalah dalam hidup saya cenderung semakin bertambah, mengarah ke kondisi depresi karena saya merasa saya tidak dapat mengendalikan apakah suatu masalah telah terselesaikan dengan baik.",set1\_5).

pernyataan(set2\_1):-

ya("Saya merasa bahwa saya hanya bisa menjadi beban bagi orang-orang di sekitar saya.",set2\_1).

pernyataan(set2\_2):-

ya("Saya merasa ingin menangis tanpa ada alasan yang jelas.",set2\_2).

pernyataan(set2\_3):-

ya("Saya terus berpikir tentang bagaimana hal-hal yang berlangsung dalam kehidupan saya tiap malam.",set2\_3).

pernyataan(set2\_4):-

ya("Saya merasa tidak dapat memulai tugas atau proyek penting yang harus saya kerjakan.",set2\_4).

pernyataan(set3\_1):-

ya("Saya mengacaukan apa saja yang saya sentuh.",set3\_1).

pernyataan(set3\_2):-

ya("Mengambil keputusan terasa seperti siksaan bagi saya.",set3\_2).

pernyataan(set3\_3):-

ya("Saya merasa melambat, baik secara mental maupun secara fisik.",set3\_3).

pernyataan(set3\_1):-

ya("Saya mengacaukan apa saja yang saya sentuh.",set3\_1).

pernyataan(set3\_2):-

ya("Mengambil keputusan terasa seperti siksaan bagi saya.",set3\_2).

pernyataan(set3\_3):-

ya("Saya merasa melambat, baik secara mental maupun secara fisik.",set3\_3).

/\*~~~~~~~ DETEKSI ~~~~~~~~\*/

deteksi("Severe Depression"):-

pernyataan(set1\_1),

pernyataan(set1\_2),

pernyataan(set1\_3),

pernyataan(set1\_4),

pernyataan(set1\_5),

terdeteksi("Depresi Akut"),

penanganan("Hubungi dokter atau psikiater segera untuk pengobatan secara medis dan psikologis.").

deteksi("Moderate Depression"):-

pernyataan(set2\_1),

pernyataan(set2\_2),

pernyataan(set2\_3),

pernyataan(set2\_4),

terdeteksi("Depresi Sedang"),

penanganan("Hubungi dokter atau psikiater untuk penanganan lebih lanjut mencegah depresi lanjut ke tingkat akut.").

deteksi("Mild Depression"):-

pernyataan(set3\_1),

pernyataan(set3\_2),

pernyataan(set3\_3),

terdeteksi("Depresi Ringan"),

penanganan("Ubah gaya hidup, serta hubungi dokter psikiater untuk penanganan dini dalam rangka mencegah depresi berakumulasi ke tingkat lanjut.").

pernyataan(set3\_2),

pernyataan(set3\_3),

terdeteksi("Depresi Ringan"),

penanganan("Ubah gaya hidup, serta hubungi dokter psikiater untuk penanganan dini dalam rangka mencegah depresi berakumulasi ke tingkat lanjut.").

/\* ~~~~~~~~ PENANGANAN LEBIH LANJUT ~~~~~~~ \*/

terdeteksi(Gejala):-

upper\_lower(AGejala, Gejala),nl,

write("-) Anda terdeteksi mengidap : ",AGejala),nl.

penanganan(Gejala):-

upper\_lower(AGejala, Gejala),nl,

write("-) Salah satu solusinya adalah : ",AGejala),nl.

goal

mulai.

**Kartu Konsul**

