

**LAPORAN PRAKTIKUM KECERDASAN BUATAN**  
**GAME TEBAK ANGKA YANG DIPIKIRKAN DAN**  
**TEBAK NILAI X DAN Y DALAM PERSAMAAN GARIS**



**Disusun Oleh :**

**KELOMPOK B/11/2015**

Syahrul Andi Subakti	1515015060
Arinda Mulawardani K.	1515015081
Chrisman Bonor Sinaga	1515015082

**Asisten Praktikum :**

M. Hilmy Ady Saputra	Niken Novirasari
1415015058	1415015064

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

**2017**

## **KATA PENGANTAR**

Puji Syukur kehadiran Allah Yang Maha Kuasa karena atas Rahmat dan Hidayah-Nyalah kami dapat menyelesaikan laporan praktikum mata kuliah kecerdasan Buatan. Tujuan penulisan Laporan ini adalah untuk memenuhi tugas akhir Praktikum Kecerdasan Buatan.

Laporan ini bertujuan untuk memberikan pemaparan tentang proyek akhir yang berjudul “game tebak angka yang dipikirkan dan tebak nilai x dan y dalam persamaan garis”. Dan dalam situasi Laporan ini pada awalnya penulis mengalami beberapa kesulitan dan terutama karena kurangnya pengetahuan, namun berkat bimbingan dan dukungan dari berbagai pihak sehingga Laporan ini dapat terselesaikan walaupun masih ada kekurangannya.

Dalam penulisan Laporan Akhir ini, penulis dengan kerendahan dan penuh pengharapan mengucapkan terima kasih kepada pihak-pihak yang membantu dalam menyelesaikan Laporan ini, khususnya kepada :

1. Tuhan Yang Maha Kuasa atas limpahan Rahmat-Nya.
2. Ibu Joan Angelina W, M.Kom dan Ibu Masna Wati, M.T selaku dosen mata kuliah Kecerdasan Buatan..
3. Asisten Laboratorium Praktikum Kecerdasan Buatan yang membimbing dan membantu pembuatan program maupun penyusunan laporan.

Masih banyak kekurangan dalam laporan ini . Selain itu dalam penyusunan tugas atau materi ini, tidak sedikit hambatan yang kami hadapi. Namun kami menyadari bahwa kelancaran dalam penyusunan laporan ini tidak lain berkat bantuan banyak pihak sehingga kendala - kendala yang kami hadapi dapat teratasi.

Samarinda, 3 Mei 2017

Penyusun

## DAFTAR ISI

<b>Halaman Judul .....</b>	<b>-</b>
<b>Kata Pengantar .....</b>	<b>i</b>
<b>Daftar Isi .....</b>	<b>ii</b>
<b>Daftar Tabel.....</b>	<b>iii</b>
<b>Daftar Gambar .....</b>	<b>iv</b>
<b>BAB I Pendahuluan .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan dan Manfaat .....	2
<b>BAB II Landasan Teori .....</b>	<b>3</b>
2.1 Fungsi-fungsi Visual Prolog yang Digunakan .....	3
2.2 Teori Keilmuan yang Diimplementasikan .....	10
<b>BAB III Metodologi .....</b>	<b>13</b>
3.1 Alur Pembuatan Sistem.....	13
<b>BAB IV Hasil dan Pembahasan .....</b>	<b>17</b>
4.1 Tabel Kebenaran .....	17
4.2 Analisis Aplikasi .....	18
<b>BAB 5 Penutup .....</b>	<b>24</b>
5.1 Kesimpulan.....	24
5.2 Saran.....	25
<b>Daftar Pustaka.....</b>	<b>26</b>
<b>Lampiran .....</b>	<b>27</b>

## **DAFTAR TABEL**

Tabel 4.1.1 List Deret Angka Pertama .....	17
Tabel 4.1.2 List Deret Angka Kedua .....	17
Tabel 4.1.3 List Deret Angka Ketiga .....	17
Tabel 4.1.4 List Deret Angka Keempat .....	18
Tabel 4.1.5 List Deret Angka Kelima .....	18

## DAFTAR GAMBAR

Gambar 3.1.1 .....	13
Gambar 4.2.1 .....	19
Gambar 4.2.2 .....	19
Gambar 4.2.3 .....	20
Gambar 4.2.4 .....	20
Gambar 4.2.5 .....	21
Gambar 4.2.6 .....	21
Gambar 4.2.7 .....	22
Gambar 4.2.8 .....	22
Gambar 4.2.9 .....	23
Gambar 4.2.10 .....	23

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Kecerdasan Buatan atau kecerdasan yang ditambahkan kepada suatu sistem yang bisa diatur dalam konteks ilmiah atau Intelegensi Artifisial (bahasa Inggris: Artificial Intelligence atau hanya disingkat AI) didefinisikan sebagai kecerdasan entitas ilmiah. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (games), logika fuzzy, jaringan syaraf tiruan dan robotika.

Di era modern yang pesat ini, pesatnya teknologi menuntut perusahaan-perusahaan untuk selalu menciptakan permainan atau games yang tidak ketinggalan zaman meskipun waktu terus berlalu. Dari semua games yang ada saat ini games bertema asah otak yang permainannya menggunakan metode angka masih diminati banyak orang dan sering dimainkan.

Salah satu bidang yang menggunakan kecerdasan buatan adalah permainan komputer (games). Sangat banyak games yang beredar sekarang, dari yang berguna untuk mengasah otak, logika, sampai hanya sebagai sarana penghilang jenuh semata atau bisa dikatakan untuk bersenang-senang, dan disini kami akan menjelaskan lebih lanjut tentang program kita yang mengangkat tema “Tebak Angka”.

Dan berdasarkan latar belakang tersebut kami mengangkat judul “TEBAK ANGKA”. Permainan tersebut terdiri dari 2 jenis permainan yaitu, pertama tentang tebak angka yang dilakukan oleh komputer, dimana kita sebagai pengguna memikirkan angka dari 1 sampai 31 lalu komputer akan menebak angka apa yang kita pikirkan. Kedua yaitu permainan dimana pengguna diminta untuk menebak nilai X dan Y dimana terdapat *clue* yaitu selisih dari nilai X dan Y.

## **1.2. Rumusan Masalah**

Sebagaimana penjelasan deskripsi masalah diatas, terdapat rumusan masalah yaitu:

- 2.1. Bagaimana cara merancang sebuah games yang bertema mengasah otak seperti menebak angka yang dipikirkan seseorang ?
- 2.2. Bagaimana program menampilkan persis angka yang diinginkan ?
- 2.3. Bagaimana program menentukan nilai X dan Y pada Visual Prolog ?

## **1.3. Batasan Masalah**

Batasan-batasan masalah yang ada adalah sebagai berikut:

1. Ini adalah berupa games yang berfungsi untuk mengasah kemampuan berfikir pengguna.
2. Dalam tampilan menu tersedia 5 pilihan.
3. Permainan yang tersedia hanya 2 jenis permainan.
4. Dalam games Tebak Angka yang Anda Pikirkan angka yang tersedia hanya 1 sampai 31.
5. Dalam games Mengungkap Misteri X dan Y pengguna hanya diberi clue berupa selisih X dan Y.
6. Bahasa Pemrograman yang digunakan adalah visual prolog.

## **1.4. Tujuan dan Manfaat**

Sesuai dengan judul kami yakni **TEBAK ANGKA** diharapkan permainan ini tidak hanya menghibur tetapi juga bermanfaat untuk mengasah kemampuan berfikir pengguna.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Fungsi-fungsi Visual Prolog yang Digunakan**

##### **2.1.1. Unifikasi dan Lacakbalik**

Saat 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 variable.

Beberapa hal penting dalam proses pencocokan atau unifikasi, yaitu:

1. Pada waktu Prolog berusaha untuk memenuhi sebuah goal, Prolog memulainya dari bagian paling atas (top) dari program dalam rangka mencari pencocokan.
2. Ketika sebuah panggilan baru terjadi, pencarian pencocokan juga dimulai dari bagian paling atas dari program.
3. Ketika sebuah panggilan mengalami pencocokan yang sukses, pemanggil kembali (is said to return), dan giliran subgoal berikutnya diuji.
4. Ketika suatu variabel telah diikat (bound) pada sebuah klausa, cara-cara satu-satunya untuk membebaskan ikatan tersebut adalah melalui lacakbalik (backtracking).



Seringkali seseorang harus menelusuri suatu jalur untuk mendapatkan konklusi yang logis saat menyelesaikan masalah. Jika konklusi ini tidak memberikan jawaban yang dicari, orang tersebut harus memilih jalur yang lain. Contohnya pada permainan 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.

Visual Prolog menyediakan 2 alat yang memperbolehkan kita untuk mengendalikan mekanisme lacak balik yaitu predikat fail yang digunakan untuk memaksa lacakbalik dan predikat cut (tanda !) yang digunakan untuk mencegah lacakbalik, serta tambahan yaitu not.

1. Fail akan memulai lacakbalik jika ada panggilan yang gagal. Pada situasi tertentu, ada kebutuhan untuk memaksa lacakbalik dalam rangka mencari alternatif solusi. Visual Prolog menyediakan predikat khusus fail untuk memaksa kegagalan sehingga memicu terjadinya lacakbalik.

2. Predikat Cut (!) 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).
3. 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). 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 variabel bebas dipanggil melalui not, maka Visual Prolog akan mengeluarkan pesan kesalahan Free variables not allowed in 'not' or 'retractall'.

#### 2.1.2. Data Objek Sederhana dan Jamak

Data objek sederhana terdiri dari 2 yaitu variabel atau konstanta. Konstanta 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).

### 1. 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.

### 2. 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.

### 3. 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)

### 4. Angka

Angka merupakan salah satu dari integer atau real.

### 5. 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.

Sedangkan Data objek 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. Deklarasi suatu domain yang mengandung data object jamak tanggal :

DOMAINS

tanggal\_jamak = tanggal(unsigned, string, unsigned)

CLAUSES

T = tanggal(1,"desember",1999).

Penulisan ini mirip dengan penulisan suatu fakta tetapi ini bukan fakta. Ini adalah sebuah data objek. Data objek 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 objek jamak yang didalamnya terdapat argumen. Argumen dari sebuah data objek jamak bisa dalam bentuk data objek jamak pula.

### 2.1.3. Perulangan dan Rekursi

Struktur pengulangan diwujudkan dengan suatu aturan(aturan induk) yang terdiri dari aksi inisialisasi dan aturan lain(aturan cabang) yang terdiri dari sejumlah aksi lain baik yang hendak diulang dalam struktur, pengulangan tersebut maupun aksi menjadi penentu kondisi berhentinya pengulangan.

Nilai indeks awal dan nilai indeks akhir pengulangan pada aturan cabang akan menjadi nilai yang digunakan untuk menentukan awal dan akhir dari pengulangan, yaitu pada aksi perbandingan di aturan cabang. Aksi hendak yang hendak diulang dapat berupa aksi aritmatika maupun aksi input-output.

contoh :

```
get_answer(Ans):-
write('Enter answer to question'),nl,
repeat,write('answer yes or no'),read(Ans),
valid(Ans),write('Answer is '),write(Ans),nl.
valid(yes). valid(no).
```

Repeat until digunakan untuk melakukan aksi pengulangan dimana kita tidak tahu dengan pasti berapa kali aksi pengulangan akan dilakukan, yang kita tahu apabila kondisi tertentu tak dipenuhi maka aksi pengulangan tidak dilakukan lagi. Pada struktur pengulangan ini minimal aksi pengulangan dilakukan satu kali karena aksi perbandingan untuk menentukan kondisi terpenuhi atau tidak diletakkan pada bagian akhir.

contoh :

#### PREDICATES

```
repeat(integer,integer,integer)
repeataksi(integer,integer,integer)
until(integer,integer,integer)
```

#### CLAUSES

```
repeat(Awal,Next,Akhir):-
I=Awal,
repeataksi(I,Next,Akhir).
repeataksi(I,Next,Akhir):- I1 & I1=I,
repeataksi(I1,Next,Akhir).
until(I1,_,Akhir):- I1 >= Akhir.
```

#### GOAL

```
repeat(1,1,10).
```

Suatu prosedur dikatakan rekursif jika memanggil dirinya sendiri. dengan rekursif tidak sulit untuk menyimpan nilai hasil program karena perhitungan loop, jumlah dan hasil sementara dapat dikirim ke tiap pengulangan dan rekursif.

#### 2.1.4. List

Dalam Prolog, list dituliskan dengan menggunakan tanda kurung [] dan setiap elemen dipisahkan oleh tanda koma(.). Sebagai contoh:

[] → list kosong

[f, t, i] → list dengan tiga elemen

Setiap list dalam Prolog dapat dituliskan sebagai [H | T] dimana H adalah kepala (head) yang menunjukkan elemen pertama dari suatu list dan T adalah ekor (tail), yaitu list tanpa elemen pertama (sama seperti Haskell! semoga kawan2 masih ingat).

Nilai H dan T ini dapat dibandingkan dengan operasi car dan cdr pada pemrograman fungsional (Haskell atau LISP). Oleh karena itu, list [f, t, i] dapat dituliskan sebagai:

[f,t,i|[]]

[f,t|[i]]

[f|[t,i]]

Sebagai contoh;

GOAL = [H | T ] = [f,t,i].

YES. H = f, T = [t,i]

#### 2.1.5. Section Facts

Facts pada prolog berupa fakta – fakta yang mana fakta – fakta tersebut dapat ditambah dan dihapus secara langsung dari sebuah program pada saat program sedang berjalan. Kita dapat mendeklarasikan sebuah predikat pada section facts seperti pada section predicates. Yang perlu dicatat bahwa perbedaan pendeklarasian fakta – fakta yang ada pada section facts dengan yang ada pada section predicates yaitu :

1. Bahwa penambahan predikat pada section facts hanya berlaku sebagai fakta saja, tidak bisa sebagai rule.
2. Fakta – fakta yang ada di section facts tidak boleh mempunyai variabel bebas.
3. Beberapa predikat built in dalam penggunaan section facts antara lain :
4. assert , assertdn assertz : digunakan untuk menambahkan fakta baru pada section facts
5. retractdan retractall : digunakan untuk menghapus fakta yang ada
6. consult : digunakan untuk membaca fakta dari sebuah file dan menyertakan fakta tersebut ke dalam fakta internal
7. save : digunakan isi fakta internal ke dalam sebuah file

## **2.2. Teori Keilmuan yang Diimplementasikan**

### **2.2.1. Tabel Angka**

Matematika merupakan suatu pelajaran yang tersusun secara beraturan, logis, berjenjang dari yang paling mudah hingga yang paling rumit. Hampir semua cabang ilmu pasti memerlukan perhitungan, maka dari itu matematika juga berarti ilmu pengetahuan yang didapat dengan berfikir.

Program kami menerapkan perhitungan matematika didalamnya, mengelompokkan bilangan-bilangan yang sudah dihitung dan memasukkannya ke dalam tabel-tabel. Karena program ini adalah permainan tebak angka dimana program akan menebak apa nomor yang dipikirkan seseorang (antara 1 sampai 31), maka langkah pertama adalah dengan meng-konversi bilangan decimal tersebut menjadi biner. Setelah itu adalah dengan membuat 5 tabel yang isinya sudah disesuaikan berdasarkan kode binernya.

Permainan ini sangatlah sederhana, pengguna akan dilihatkan satu persatu dengan tabel-tabel yang telah tersusun tadi dan disuruh untuk menjawab apakah nomor yang dipikirkan itu ada di dalam tabel tersebut. Secara bertahap, prosesnya yaitu :

1. Pengguna dihibau untuk memikirkan sebuah angka antara 1 sampai 31
2. Program akan menampilkan tabel pertama yang isinya berupa angka-angka.
3. Pengguna menjawab apakah nomor yang dipikirkan ada di dalam tabel pertama.
4. Setelah itu pengguna akan dihadapkan dengan tabel kedua dan disuruh menjawab, begitu seterusnya sampai tabel kelima.
5. Setelah selesai, program akan menampilkan secara akurat nomor yang dipikirkan pengguna.

#### 2.2.2. Persamaan Garis Lurus

Secara sederhana persamaan garis lurus dapat didefinisikan sebagai sebuah garis lurus dimana posisinya ditentukan oleh sebuah persamaan dan apabila persamaan tersebut digambarkan pada bidang cartesius maka akan menghasilkan sebuah garis yang lurus. Salah satu contoh persamaan yang menghasilkan garis lurus adalah  $x + y = 3$ .

Salah satu cara yang bisa kita lakukan untuk membuktikan persamaan garis lurus adalah dengan menggambarkan garis lurus ke dalam bidang cartesius dengan menggunakan koordinat yang dihasilkan dari persamaan tersebut, contohnya:

Kita misalkan  $x = 0$ , maka:

$$x + y = 3$$

$$0 + y = 3$$

$$y = 3$$

titik pertama yang kita peroleh adalah koordinat (0, 3)

Dengan teori persamaan garis lurus, kami membuat program untuk menebak nilai X dan Y pada sebuah persamaan. Kami juga menambahkan perhitungan skor yang akan muncul ketika pengguna salah dalam menjawab nilai tersebut. Untuk lebih jelasnya dapat dilihat dari langkah-langkah di bawah ini :

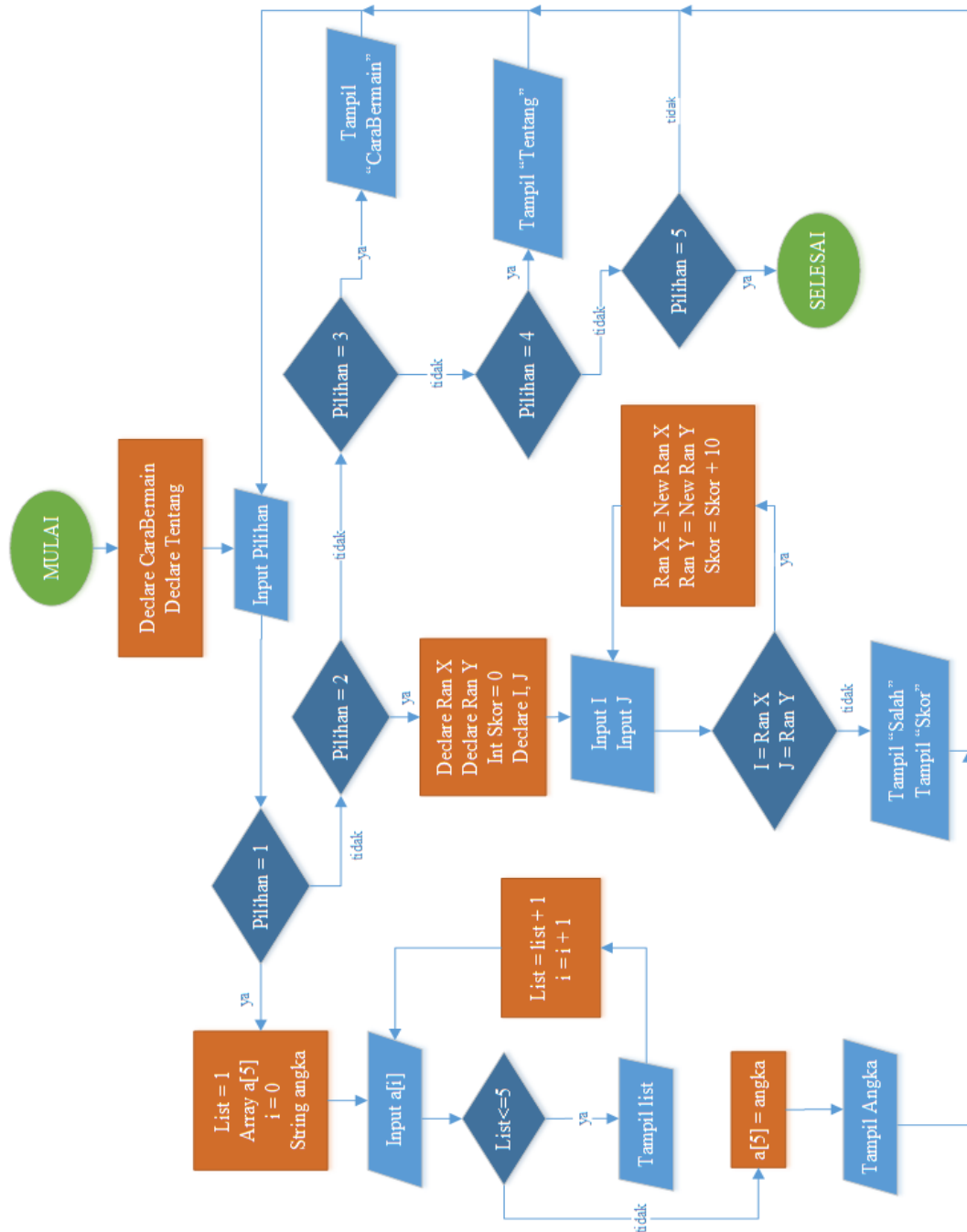


1. Pengguna diperlihatkan sebuah persamaan, contohnya  $3x+2y=12$
2. Berdasarkan persamaan di atas, akan diperlihatkan juga sebuah *clue* yaitu **nilai X dikurangi Y sama dengan 1**.
3. Dari 2 tampilan tersebut, pengguna dihibau untuk mengetikkan jawaban dari nilai X.
4. Setelah mengisi nilai X, lanjut untuk mengisi nilai Y.
5. Program akan menginisiasi apakah jawaban tersebut benar atau tidak.
6. Apabila salah, program akan berhenti dan menampilkan keterangan salah dan menampilkan hasil skor yang pengguna peroleh.
7. Apabila benar, program akan lanjut untuk menampilkan persamaan lain beserta *clue* nya, begitu seterusnya sampai pengguna memasukkan nilai yang salah, setelah itu menampilkan skor akhir yang diperoleh

## BAB III









### METODOLOGI

#### 3.1. Alur Pembuatan Sistem



**Gambar 3.1.1** Flowchart Tebak Gambar

Gambar 3.1.1 merupakan diagram alur kerja program berbentuk *flowchart*. Flowchart sering disebut dengan Diagram Alir yang digunakan untuk menggambarkan proses-proses operasionalnya sehingga mudah dipahami dan mudah dilihat berdasarkan urutan langkah dari suatu proses ke proses lainnya.

Simbol	Keterangan
	<i>Terminator</i> Menggambarkan kegiatan awal atau akhir dari suatu proses.
	<i>Proses</i> Menggambarkan suatu proses.
	<i>Data</i> Menggambarkan kegiatan masukan atau keluaran yang dihasilkan.
	<i>Decision</i> Menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu.
	<i>Predefine Proses</i> Menggambarkan proses-proses yang masih bisa dijabarkan dalam algoritma.
	<i>Line Connector</i> Menghubungkan suatu simbol dengan simbol lain pada modul yang sama.
	<i>On-page Reference</i> Menghubungkan suatu simbol dengan simbol yang lainnya pada halaman yang sama.
	<i>Off-page Reference</i> Menghubungkan suatu simbol dengan simbol yang lainnya pada halaman yang berbeda.

Dan untuk permainan tebak gambar ini dapat diketahui alur proses pengguna berdasarkan flowchart sebagai berikut :

1. Pengguna memulai dengan “MULAI” yang diwakili oleh simbol *terminator* yang menggambarkan kegiatan awal atau akhir dari suatu proses. Pada langkah simbol ini, program menampilkan menu-menu yang ada sebagai pilihan. Ada 5 menu pilihan yang sudah kami sediakan mulai dari urutan pertama yaitu Game 1, Game 2, Cara Bermain, Tentang, dan keluar. Pada tahap ini juga program sudah memuat database sebagai referensi pengetahuan yang nantinya digunakan saat memainkan game.

2. Apabila pengguna memilih “1” maka akan menuju tampilan daftar tabel angka yang sudah disediakan, sebelum itu program akan mulai mendeklarasikan apa-apa saja nilai yang perlu seperti array dan inputan. Setelah itu pengguna dihadapkan dengan list tabel pertama yang memuat nomor-nomor, lalu user menginputkan nilai apakah benar angka yang ada dipikirkannya tampil pada list tabel pertama atau tidak. Setelah itu program akan mengecek kondisi yang diwakili oleh simbol decision yang berfungsi menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu.
3. Di sini program akan mengecek apakah list masih dalam kondisi kurang dari sama dengan ( $\leq$ ) 5 atau belum. Karena tadi masih merupakan list pertama jadi program akan mengulang meng input lagi. Begitu seterusnya sampai list sudah mencapai lebih dari 5. Apabila sudah lebih program otomatis akan memproses nilai-nilai inputan yang sudah dimasukkan sebelumnya dicocokkan dengan database angka dan menjadi suatu nilai yaitu angka yang telah dipikirkan pengguna.
4. Setelah itu program akan kembali lagi ke menu pilihan. Apabila memilih “2”, program otomatis menginisiasi nilai X dan Y secara random, juga mendeklarasikan variable inputan, serta nilai skor = 0 untuk awal.
5. Apabila memilih “2”. Userpun akan diberi tampilan berupa persamaan garis namun dengan nilai X dan Y belum diketahui, dan tampilan clue terkait nilai dari X dan Y tersebut. setelah itu pengguna mengetikkan inputan yaitu nilai dari X, setelah itu mengisi Y.
6. Setelah selesai program akan mengecek lagi apakah nilai inputan sama dengan random nilai X dan Y, maka akan ada proses lagi (ditandai dengan simbol persegi panjang) yaitu inisiasi nilai Random baru untuk X dan Y. program akan looping terus-menerus selama inputan yang pengguna ketik bernilai benar, dan setiap satu kali benar akan ada penambahan nilai skor yaitu 10.

7. Apabila salah satu atau semua nilai inputan pengguna tidak sesuai, maka proses looping akan berhenti dan langsung dilanjutkan dengan tampilan “salah” dan hasil jumlah dari skor yang telah dikumpulkan.
8. Sebuah opsi menu yang sangat berguna apabila pengguna belum mengerti cara bermain tebak angka ini, maka kami sediakan pilihan menu 3 yaitu tampilan cara bermain untuk memudahkan.
9. Opsi menu yang lain yaitu apabila kita memilih 4 yaitu menu tentang, di dalamnya terdapat informasi mengenai siapa saja yang berkontribusi dalam pembuatan *project* ini.
10. Opsi terakhir adalah apabila ingin keluar dari program, pengguna cukup menekan angka 5 saja, maka proses akan terhenti dan tidak dapat dijalankan lagi kecuali mengulang membukanya lagi.
11. Program hanya mengenal inputan mulai dari 1 sampai angka 5 saja, selain dari 5 pilihan tersebut program akan mengulang kembali ke pemilihan menu dan menghimbau pengguna agar memberika inoutan yang sesuai.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Tabel

Dalam game pertama yaitu tebak angka yang dipikirkan pengguna, kami mempunyai 5 tabel yang sudah disediakan untuk mengarahkan pengguna kepada jawaban ya atau tidak.

1	3	5	7
9	11	13	15
17	19	21	23
25	27	29	31

**Tabel 4.1.1** Daftar deret angka pertama

Tabel ini diperoleh berdasarkan perhitungan konversi dari decimal ke biner. Karena cakupannya hanya sampai 31 maka akan didapat susunan 5 deret angka yang mewakili nilai biner.

$$2^0 + 2^1 + 2^2 + 2^3 + 2^4 = 31$$

Contohnya 1 nilai binernya adalah 10000, 3 nilai binernya 11000, jadi semua yang mempunyai nilai 1 di urutan depan akan dimasukkan ke dalam list pertama.

2	3	6	7
10	11	14	15
18	19	22	23
26	27	30	31

**Tabel 4.1.2** Daftar deret angka kedua

Tabel kedua diperuntukkan untuk bilangan-bilangan yang mempunyai kode biner 1 di urutan ke dua, seperti contohnya 6 kode binernya 01100, terlihat bahwa 6 mempunyai nilai kode biner 1 di urutan kedua.

4	5	6	7
12	13	14	15
20	21	22	23
28	29	30	31

**Tabel 4.1.3** Daftar deret angka ketiga

Tabel ketiga diperuntukkan untuk bilangan-bilangan yang mempunyai kode biner 1 di urutan ke tiga, seperti contohnya 5 kode binernya 10100, terlihat bahwa 5 mempunyai nilai kode biner 1 di urutan ketiga.

8	9	10	11
12	13	14	15
24	25	26	27
28	29	30	31

**Tabel 4.1.4** Daftar deret angka keempat

Sama seperti sebelumnya, tabel keempat diperuntukkan untuk bilangan-bilangan yang mempunyai kode biner 1 di urutan ke empat, seperti contohnya 9 kode binernya 10010, terlihat bahwa 9 mempunyai nilai kode biner 1 di urutan keempat.

16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31

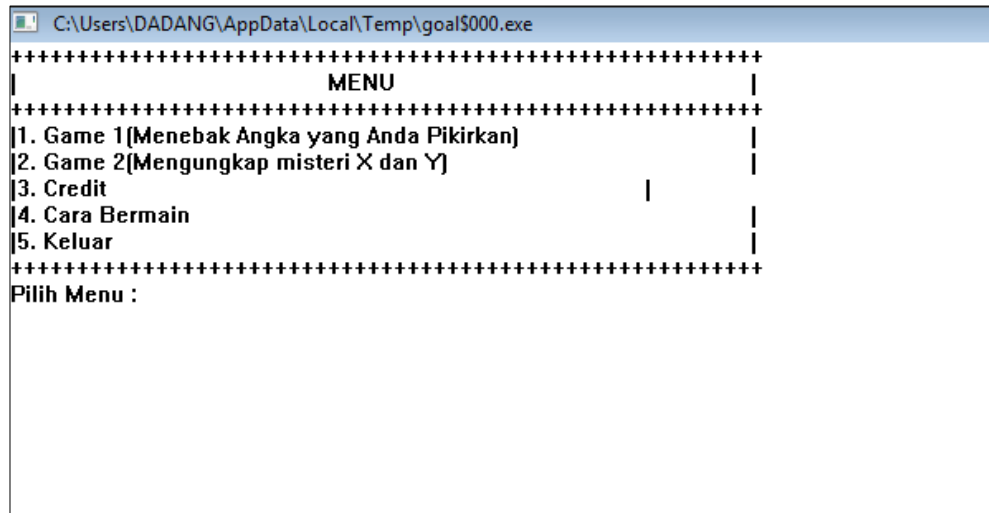
**Table 4.1.5** Daftar deret angka kelima

Selanjutnya tabel kelima yang memuat bilangan-bilangan yang mempunyai kode biner 1 di urutan ke kelima, seperti contohnya 20 kode binernya 00101, terlihat bahwa 20 mempunyai nilai kode biner 1 di urutan kelima.

## 4.2. Analisis Aplikasi

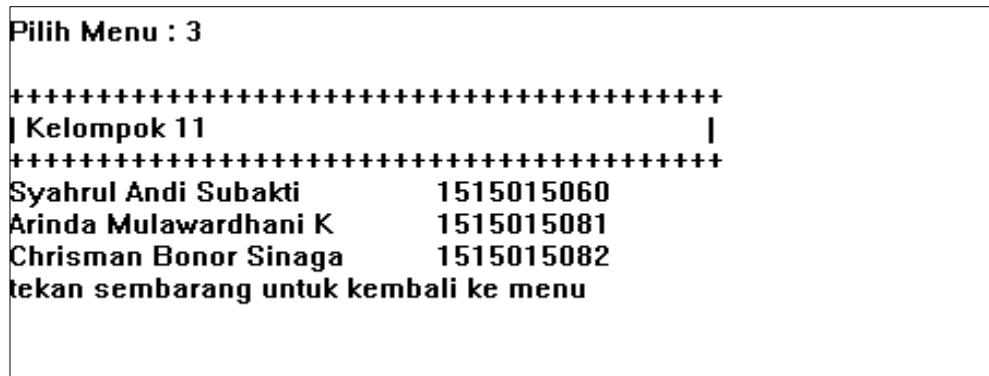
Pada gambar dibawah merupakan tampilan awal dari permainan tebak angka kami, yang terdiri dari 5 pilihan yaitu:

1. Pilihan nomor 1 merupakan pilihan untuk memainkan game “Menebak Angka Yang Anda Pikirkan”.
2. Pilihan nomor 2 merupakan pilihan untuk memainkan game “Mengungkapkan Misteri X dan Y”.
3. Pilihan nomor 3 merupakan pilihan credit.
4. Pilihan nomor 4 merupakan cara memainkan game.
5. Pilihan nomor 5 merupakan pilihan untuk mengakhiri game.



**Gambar 4.2.1** Tampilan Awal Game

Selanjutnya kita memasukkan pilihan nomor 3 yaitu credit, pada pilihan ini terdapat nama-nama anggota kelompok 11. Diantaranya adalah Syahrul Andi Subakti, Arinda Mulawardani Kustiawan, dan Chrisman Bonor Sinaga



**Gambar 4.2.2** Tampilan Credit

Selanjutnya jika kita memasukkan pilihan nomor 4 maka akan ditampilkan cara untuk memainkan game “TEBAK ANGKA”.



**Pilih Menu : 4**

**Cara Bermain Game 1 :**

1. Pada game pertama, pemain diminta untuk memikirkan angka 1 hingga 31
2. Akan tampil deretean angka-angka, bila pemain menemukan angka yang dipilih maka pemain memasukkan angka '1', bila angka yang dipilih tidak ada pemain memasukkan angka '0'

**Cara Bermain Game 2 :**

1. Pada game kedua, pemain diminta untuk menebak nilai dari X dan Y
2. Terdapat clue yang merupakan selisih nilai dari X dan Y
2. Nilai X dan Y adalah bilangan bulat positif3. Raihlah score tertinggi

**Gambar 4.2.3** Tampilan Cara Bermain

Selanjutnya kita akan memainkan permainan yang ada di game “TEBAK ANGKA”. Pertama kita akan memainkan permainan pertama yaitu Menebak Angka yang Anda Pikirkan, pertama pilih pilihan nomor 1 lalu akan ditampilkan perintah untuk memikirkan angka antara 1 sampai 31. Setelah itu kita klik enter atau yang lainnya juga bisa, setelah itu akan ditampilkan beberapa angka, dan kita harus memperhatikan apakah angka yang kita cari ada didalam daftar atau tidak. Angka-angka yang akan ditampilkan akan muncul sebanyak 5 kali, setelah itu dapat ditentukan apakah angka yang sedang anda pikirkan tadi.

**Pilih Menu : 1**

**Pikirkan satu angka dari angka 1 hingga angka 31**

**Gambar 4.2.4** Pikirkan angka dari 1-31

Lalu setelah itu akan tampil beberapa angka yang ditampilkan secara acak, dan kita harus memperhatikan apakah angka yang kita pikirkan sudah ada atau tidak.

1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31  Apakah angka anda terdapat pada angka diatas ?(1=ya, 0=tidak)1  2 3 6 7 10 11 14 15 18 19 22 23 26 27 230 31  Apakah angka anda terdapat pada angka diatas ?(1=ya, 0=tidak)1  
--

**Gambar 4.2.5** Game menebak angka pemain

4 5 6 7 12 13 14 15 20 21 22 23 28 29 30 31  Apakah angka anda terdapat pada angka diatas ?(1=ya, 0=tidak)0  8 9 10 11 12 13 14 15 24 25 26 27 28 29 30 31  Apakah angka anda terdapat pada angka diatas ?(1=ya, 0=tidak)0  
--

**Gambar 4.2.6** Game menebak angka pemain

Setelah Game menebak angka yang dimaksud pemain sebanyak lima kali maka akhirnya Game akan mengetahui angka yang dimaksud pemain.

16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
<hr/> Apakah angka anda terdapat pada angka diatas ?(1=ya, 0=tidak)1 <hr/>
Angka yang anda pikirkan adalah 19 tekan sembarang untuk kembali ke menu

**Gambar 4.2.7** Game mengetahui angka pemain

Setelah permainan pertama telah dimainkan selanjutnya kita akan memainkan permainan yang kedua yaitu permainan “Mengungkap Misteri X dan Y” pada game ini kita diberikan sebuah pertanyaan nilai X dan Y dari sebuah persamaan, dan kita hanya diberi sebuah petunjuk yaitu selisih dari nilai X dan Y.

Pada gambar dibawah terdapat 3 pertanyaan tentang persamaan X dan Y. jika kita telah mengetahui jawabannya maka selanjutnya kita tulis jawaban untuk masing-masing X dan Y. jika jawaban benar maka akan diberi score sebanyak 10 poin.

<b>Pilih Menu : 2</b>  <b>Berapa kah nilai X dan Y pada persamaan ini <math>3X + 1Y = 18</math></b> <b>CLUE = Nilai dari X-Y adalah 2</b> <b>X = 5</b>  <b>Y = 3</b> <b>Anda benar</b> <b>Berapa kah nilai X dan Y pada persamaan ini <math>4X + 3Y = 24</math></b> <b>CLUE = Nilai dari X-Y adalah -1</b> <b>X = 3</b>  <b>Y = 4</b> <b>Anda benar</b> <b>Berapa kah nilai X dan Y pada persamaan ini <math>4X + 3Y = 28</math></b> <b>CLUE = Nilai dari X-Y adalah 0</b> <b>X = 4</b>  <b>Y = 4</b> <b>Anda benar</b>
--

**Gambar 4.2.8** Pemain menjawab pertanyaan

Dalam permainan “ Mengungkap Misteri X dan Y” tidak ada batasan soal, jika kita bisa menjawab permainan akan terus berlanjut. Tetapi jika kita salah menjawab pertanyaan maka permainan dianggap selesai, dan kita diberitahu berapa score yang telah kita dapatkann selama bermain.

**Berapa kah nilai X dan Y pada persamaan ini  $1X + 2Y = 8$**   
**CLUE = Nilai dari X-Y adalah 2**  
**X = 1**  
**Y = 2**  
**Anda salah, Score anda 30**  
**tekan sembarang untuk kembali ke menu**

**Gambar 4.2.9** Permainan selesai

Jika kita ingin mengakhiri permainan dan ingin keluar dari Game “TEBAK ANGKA” maka kita masukkan pilihan nomor 5 untuk keluar dari Game.

**Pilih Menu : 5**  
**bye**  
**yes**

**Gambar 4.2.10** Keluar dari Game

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Dari Program dan referensi yang ada terkait Visual Prolog diatas, kami dapat mengambil kesimpulan bahwa:

1. 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).
2. Kita dapat mendeklarasikan sebuah predikat pada section facts dan predikat tersebut dapat digunakan sama halnya seperti kalau dideklarasikan pada section predicates.
3. Visual Prolog menyediakan beberapa predikat built-in untuk menangani hal yang berkaitan dengan penggunaan section facts, antara lain:
  - a. assert, asserta dan assertz untuk menambah fakta baru pada section facts.
  - b. retract dan retractall untuk menghapus fakta yang ada.
  - c. consult untuk membaca fakta dari sebuah file dan menyertakan fakta tersebut ke dalam fakta internal.
  - d. save menyimpan isi fakta internal ke dalam sebuah file.

Ada banyak sekali yang dapat dilakukan dalam pemrograman visual prolog, karena Bahasa yang digunakan berbeda daripada Bahasa pemrograman lain yaitu sebagai Bahasa deskriptif. Programmer Prolog hanya membutuhkan deskripsi/gambaran permasalahan, lalu menerjemahkannya ke bahasa Prolog. Selanjutnya tinggal sistem Prolog yang menentukan bagaimana mencari solusinya. Prolog mempunyai mesin inferensi (infrence 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.

Penerapan ilmu matematika juga kami terapkan di dalam pemrograman ini. Dengan memberikan pola-pola tertentu pada deret bilangan, kami dapat mengimplementasikan suatu aturan dan fakta tersendiri yang nantinya Prolog sendirilah yang akan menemukan masalahnya.

## **5.2. Saran**

Dalam pembuatan program tebak angka ini, dibutuhkan pemahaman lebih dalam Prolog terutama bagaimana meletakkan fakta-fakta dan aturan menjadi satu kesatuan, apalagi menerapkan konsep matematika di dalamnya yang membuat pengerjaan menjadi lebih sulit. Demikian laporan akhir yang dapat kami paparkan dan tentunya masih banyak kekurangan dan kelemahannya, karena keterbatasan pengetahuan dan referensi, maka diharapkan pembaca dapat memberikan kritik dan saran yang bersifat membangun laporan akhir ini di kesempatan selanjutnya.

## DAFTAR PUSTAKA

[http://www.academia.edu/3290599/Modul\\_Praktek\\_AI](http://www.academia.edu/3290599/Modul_Praktek_AI). Diakses 03/04/2017.

<http://rumusrumus.com/persamaan-garis-lurus/>. Diakses 03/04/2017.

[https://www.academia.edu/9946906/MODUL\\_I\\_LINGKUNGAN\\_VISUAL\\_PROLOG\\_VISUAL\\_PROLOG\\_ENVIRONMENT\\_VPE](https://www.academia.edu/9946906/MODUL_I_LINGKUNGAN_VISUAL_PROLOG_VISUAL_PROLOG_ENVIRONMENT_VPE). Diakses 03/04/2017.

<https://katahugo.wordpress.com/2017/04/10/program-tebak-angka-dalam-bahasa-c/>. Diakses 03/04/2017.

## LAMPIRAN

% Kamus Jawaban Tebak Angka

clauses	list([1,1,0,1,1],27).
list([0,0,0,0,0],0).	list([0,0,1,1,1],28).
list([1,0,0,0,0],1).	list([1,0,1,1,1],29).
list([0,1,0,0,0],2).	list([0,1,1,1,1],30).
list([1,1,0,0,0],3).	list([1,1,1,1,1],31).
list([0,0,1,0,0],4).	
list([1,0,1,0,0],5).	list_angka(1,1,"1 3 5 7").
list([0,1,1,0,0],6).	list_angka(1,2,"9 11 13 15").
list([1,1,1,0,0],7).	list_angka(1,3,"17 19 21 23").
list([0,0,0,1,0],8).	list_angka(1,4,"25 27 29 31").
list([1,0,0,1,0],9).	
list([0,1,0,1,0],10).	list_angka(2,1,"2 3 6 7").
list([1,1,0,1,0],11).	list_angka(2,2,"10 11 14 15").
list([0,0,1,1,0],12).	list_angka(2,3,"18 19 22 23").
list([1,0,1,1,0],13).	list_angka(2,4,"26 27 230 31").
list([0,1,1,1,0],14).	
list([1,1,1,1,0],15).	list_angka(3,1,"4 5 6 7").
list([0,0,0,0,1],16).	list_angka(3,2,"12 13 14 15").
list([1,0,0,0,1],17).	list_angka(3,3,"20 21 22 23").
list([0,1,0,0,1],18).	list_angka(3,4,"28 29 30 31").
list([1,1,0,0,1],19).	
list([0,0,1,0,1],20).	list_angka(4,1,"8 9 10 11").
list([1,0,1,0,1],21).	list_angka(4,2,"12 13 14 15").
list([0,1,1,0,1],22).	list_angka(4,3,"24 25 26 27").
list([1,1,1,0,1],23).	list_angka(4,4,"28 29 30 31").
list([0,0,0,1,1],24).	
list([1,0,0,1,1],25).	list_angka(5,1,"16 17 18 19").
list([0,1,0,1,1],26).	list_angka(5,2,"20 21 22 23").
	list_angka(5,3,"24 25 26 27").
	list_angka(5,4,"28 29 30 31").



% pemrosesan logika game 1, menebak angka yang dipikirkan seseorang

```
%GAME1=====
nondeterm binary(lis,lis,integer)
nondeterm predict(lis,integer)
nondeterm cek_input(integer)
nondeterm opsi(string)
nondeterm kelompok(string,string,integer)
nondeterm draft(integer)
%=====
%GAME 1
binary([],[],5).
binary([_ | Tail],[NewHead | NewTail],Count):-
    NewCount=Count+1,
    list_angka(NewCount,1,List1),
    list_angka(NewCount,2,List2),
    list_angka(NewCount,3,List3),
    list_angka(NewCount,4,List4),
    nl,
    write(List1),nl,
    write(List2),nl,
    write(List3),nl,
    write(List4),nl,
    nl,
    write("-----"),nl,
    write("Apakah angka anda terdapat pada angka diatas ?(1=ya, 0=tidak)",
    readint(In),cek_input(In),nl,
    write("-----"),nl,
    NewHead=In,
    binary(Tail,NewTail,NewCount).
predict(NewNumb,Numb):-
    list(NewNumb,Numb).
```

% pemrosesan logika game 2, menebak nilai X dan Y

```
%game2=====
nondeterm tes_ran(number,limit)
nondeterm var_x(integer)
nondeterm var_y(integer)
nondeterm kons_x(integer)
nondeterm kons_y(integer)
nondeterm cek(integer,integer,integer,integer,integer)
nondeterm clue(integer,integer)
nondeterm game2(integer)
%=====
%GAME 2
tes_ran(Result,Max):-
    random(Real),Result=Real*Max+1.

var_x(Result1):-
    tes_ran(Result1,4).
var_y(Result2):-
    tes_ran(Result2,4).
kons_x(Result3):-
    tes_ran(Result3,3).
kons_y(Result4):-
    tes_ran(Result4,3).
cek(X,Y,A,B,Score):-
    X=A,Y=B,write("Anda benar"),NewScore=Score+10,nl,
    game2(NewScore);
    write("Anda salah, Score anda ",Score).
clue(A,B):-
    C=A-B,
    write("CLUE = Nilai dari X-Y adalah ",C).
game2(Score):-
    var_x(Result1),
    var_y(Result2),
    kons_x(Result3),
    kons_y(Result4),
    Hasil=Result3*Result1+Result4*Result2,
    write("Berapa kah nilai X dan Y pada persamaan ini ",Result3,"X +
",Result4,"Y = ",Hasil),nl,clue(Result1,Result2),
    nl,
    write("X = "),readint(X),nl,
    write("Y = "),readint(Y),cek(X,Y,Result1,Result2,Score).
```