**LAPORAN PRAKTIKUM KECERDASAN BUATAN**

**PERMAINAN JAN-KEN-PON**



**Disusun Oleh :**

Dony Ramadhan 1515015066

Mahathir Muhammad 1515105070

Reza Hervandi 1515015076

Ardan 1515015124

**Asisten Praktikum :**

|  |  |  |
| --- | --- | --- |
| Niken Novirasari  1415015064 |  | M. Hilmy Ady S.  1415015058 |

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI DAN KOMUNIKASI**

**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**

**UNIVERSITAS MULAWARMAN**

**2017**

**KATA PENGANTAR**

Puji Syukur senantiasa kita panjatkan kehadirat Tuhan Yang Maha Esa karena berkat Rahmat dan Izin-Nya jualah maka Laporan Akhir Praktikum ini dapat diselesaikan tepat waktu. Adapun tujuan dalam penulisan Laporan ini adalah untuk memenuhi tugas akhir Praktikum Kecerdasan Buatan pada semester tiga.

Laporan ini terdiri dari beberapa bab yang bertujuan untuk memberikan pemaparan tentang projek akhir yang berjudul **Permainan Jan-Ken-Pon** 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 M.Kom dan Ibu Masnawati S.Kom, M.T selaku dosen mata kuliah Kecerdasan Buatan.
3. Asisten Laboratorium Praktikum Kecerdasan Buatan Kelompok B yang telah membimbing dalam pembuatan program dan penyelesaian Laporan.

Penulis telah berusaha untuk membuat Laporan Akhir ini dengan sebaik-baiknya, meskipun demikian penulis yakin jika Laporan ini masih jauh dari sempurna. Untuk itu penulis sangat terbuka untuk menerima kritik dan saran yang bersifat membangun. Terimakasih.

Samarinda, April 2017

Penulis

**DAFTAR ISI**

**KATA PENGANTAR i**

**DAFTAR ISI ii**

**DAFTAR GAMBAR iii**

**DAFTAR TABEL........................................................................................ v**

**BAB I PENDAHULUAN 1**

1.1.Latar belakang 1

1.2.Rumusan masalah 2

1.3.Batasan masalah 2

1.4.Tujuan dan Manfaat 2

**BAB II LANDASAN TEORI**  3

1.1 Fungsi-Fungsi Visual Prolog yang Digunakan 4

1.1.1 Dasar Dasar Prolog 4

1.1.2 Program Visual Prolog 5

1.1.3 Unifikasi Dan Lacakbalik 6

1.1.4 Data Objek Sederhana Dan Jamak 7

1.1.5 Perulangan Dan Rekursi 7

1.1.6 List 7

1.1.7 Section Facts 7

1.2 Menerapkan Kecerdasan Buatan dalam Game 8

1.2.1 Mengejar Dan Menghindar 8

1.2.2 Pola Pergerakan 8

1.2.3 Pathfinding 9

1.2.4 Jaringan Saraf Tiruan 9

1.2.5 Algoritma Genetis 9

1.3 Jan-Ken-Pon 9

**BAB III METODOLOGI** 11

3.1 Alur Pembuatan 11

**BAB IV Hasil dan Pembahasan** 12

4.1 Analisis 12

**BAB V Penutup** 23

5.1 Kesimpulan 23

5.2 Saran 23

**Daftar Pustaka**  24

**Lampiran**  25

1.Source Code 25

2.Kartu Konsul 25

**DAFTAR GAMBAR**

Gambar 3.1.11 Flowchart 11

Gambar 1.12 JepunChance.pro 12

Gambar 2.13 Tampilan Syntax Program 13

Gambar 3.14 Tampilan Syntax Program 14

Gambar 5.15 Tampilan Syntax Program 15

Gambar 6.16 Tampilan Syntax Program 16

Gambar 7.17 Tampilan Syntax Program 17

Gambar 8.18 Tampilan Awal 19

Gambar 9.18 Tampil Memulai Permainan 19

Gambar 10.18 Kembali Ke Menu Awal 20

Gambar 11.19 Tampilan Cara Bermain 20

Gambar 12.19 Tampilan Credit 21

Gambar 13.19 Tampilan Exit 21

**Daftar Table**

Table 1.12 Table Kemungkinan 12

**BAB I**

**PENDAHULUAN**

* 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) 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.

**Permainan** atau **Gim** merupakan sebuah aktivitas [rekreasi](https://id.wikipedia.org/wiki/Rekreasi) dengan tujuan bersenang-senang, mengisi waktu luang, atau berolahraga ringan. Hampir setiap anak menyukai *game*, apapun bentuk game itu sendiri. Mulai dari *game* yang sifatnya sederhana sampai game yang paling modern sekalipun. Dalam *game* berbasis kecerdasan buatan, ada banyak teknik yang diadaptasi dari bidang kecerdasan buatan untuk diterapkan pada *game*. Kali ini kmai akan membuat suatu game bernama Jan-Ken-Pon yang menggunakan sebuat peluang dan juga penganruh psikologis untuk memenangkan permainan ini.

Di Indonesia terkenal dengan yang namanya "Hompimpah". Suatu permainan kalah dan menang dengan mengadu jari atau telapak tangan yang beranggotakan dua orang atau lebih. Di Jepang, permainan ini sama saja, tetap menggunakan jari dan tangan, namun namanya dan bentuk permainannya berbeda. Namanya, "Jankenpon" -disaat bersamaan dua orang anak akan menurunkan tangannya, bila kata jankenpon itu disebutkan.

* 1. **Rumusan Masalah**

Berdasarkan deskripsi masalah diatas, maka rumusan masalah yang dapat diambil adalah:

1. Bagaimana cara merancang dan membuat program Permainan Jan-Ken-Pon berbasis kecerdasan buatan?
2. Bagaiman mengimplementasikan rancangan tersebut kedalam aplikasi Visual Prolog?
3. Bagaimana membuat analisis program untuk memudahkan pembaca dalam memahami program yang telah dibuat?
4. Bagaimana menjelaskan konsep materi praktikum yang digunakan ?
   1. **Batasan Masalah**

Batasan-batasan masalah yang dapat diambil dari program Permainan Jan-Ken-Pon ini adalah:

1. Program ini adalah sebuah permainan sederhana tentang suit berbasis Kecerdasan Buatan.
2. Pemain dapat memilih pilihan berupa batu, kertas dan gunting dan tidak boleh lebih.
3. Pemain akan melawan musuh berupa program tersebut.
4. Pemain dapat memilih berapa kali dia ingin bermain.
5. Software yang digunakan dalam pembuatan program ini adalah Visual Prolog v.5.2
   1. **Tujuan dan manfaat**

Tujuan yang akan dicapai dalam pembuatan program permainan Jan-Ken-Pon ini adalah untuk memenuhi Projek Akhir praktikum Kecerdasan Buatan, dan manfaat dari program permainan Jan-Ken-Pon ini adalah agar para peminat *game* serta masyarakat luas dapat mencoba memainkan permainan berbasis kecerdasan yang kami buat ini meskipun hanya permainan yang sederhana.

**BAB II**

**LANDASAN TEORI**

* 1. **Fungsi-Fungsi Visual Prolog yang Digunakan**
     1. **Dasar-dasar prolog**
* **Kalimat: Fakta dan Aturan**

Fakta adalah apa yang diketahui. Suatu fakta memiliki nama relasi diikuti *object* atau *object-object* di dalam tanda kurung. Sebagaimana kalimat, fakta tersebut diakhiri dengan tanda titik (.)

Aturan adalah apa yang dapat disimpulkan dari fakta yang ada.aturan adalah suatu konklusi diketahui benar jika satu atau lebih konklusi atau fakta lain ditemukan benar. an sifat (*property*) dari suatu *object* sebagaimana suatu relasi.

* **Query**

Kalau kita sudah memberikan Prolog sekumpulan fakta, selanjutkan dapat diajukan pertanyaan sehubungan dengan fakta-fakta. Ini dikenal dengan nama memberikan query ke sistem Prolog (*querying the Prolog system*). Pertanyaan yang diajukan ke Prolog sama tipenya seperti dalam bahasa natural.

* **Variabel**

Untuk memberikan nama variabel dalam Visual Prolog harus diawali dengan huruf besar (*capital letter*) atau garis bawah (*underscore*), berikutnya dapat berupa huruf (besar atau kecil), angka (“0-9”) dan garis bawah (“\_”).

* **Komentar**

Komentar dengan baris jamak harus dimulai dengan karakter /\* (*slash, asterik*) dan diakhiri dengan karakter \*/ (*asterik, slash*). Untuk memberikan komentar dengan baris tunggal dapat menggunakan karakter yang sama atau dapat dimulai dengan tanda persen (%).

* + 1. **Program Visual Prolog**
* **Section Clauses**

Section clauses merupakan section yang paling penting pada program Visual Prolog. Pada section inilah kita meletakkan fakta dan aturan.

* **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.

* **Secton Domain**

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.

* **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

* + 1. **Unifikasi dan Lacakbalik**
* **Unifikasi (Unification)**

Pada waktu Visual Prolog mencoba untuk mencocokkan suatu panggilan (dari sebuah subgoal) ke klausa (pada section clauses), maka proses tersebut melibatkan suatu prosedur yang dikenal dengan unifikasi (*unification*), yang mana berusaha untuk mencocokkan antara struktur data yang ada di panggilan (subgoal) dengan klausa yang diberikan.

* **Lacakbalik**

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. Visual Prolog menggunakan metode lacakalik untuk menemukan suatu solusi dari permasalahan yang diberikan.

* + 1. **Data Objek Sederhana dan Jamak**
* **Data ObjekctSederhana**

Terdiri dari 2 yaitu variabel atau konstanta. Yang dimaksud dengan konstanta di sini adalah apapun yang diidentifikasikan sebagai sebuah *object* bukan *subject* yang nilainya bisa bervariasi, seperti sebuah karakter (**char**), angka (integer atau real) atau sebuah **atom** (symbol atau string).

* **Data Object Jamak**

Data object jamak memperlakukan beberapa informasi sebagai sebuah item tunggal.

* + 1. **Perulangan dan Rekursi**

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

* **Lacakbalik**

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

* **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 faktorial dari suatu angka.

* + 1. **List**

*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)

* + 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*). Kita dapat mendeklarasikan sebuah predikat pada *section facts* dan predikat tersebut dapat digunakan sama halnya seperti kalau dideklarasikan pada *section predicates*.

* 1. **Menerapkan Kecerdasan Buatan dalam Game**

Di dunia akademis, bidang kecerdasan buatan dipelajari secara serius untuk meningkatkan kualitas hidup manusia. Para peneliti dan mahasiswa (ilmu komputer atau teknik informatika) terus menerus mengembangkan teknik-teknik pada bidang ini untuk menghasilkan mesin yang semakin mengerti, dan memahami kebutuhan manusia. Dalam game berbasis kecerdasan buatan, ada banyak teknik yang diadaptasi dari bidang kecerdasan buatan untuk diterapkan pada game. beberapa diantaranya, yaitu:

* + 1. **Mengejar dan Menghindar**

Mengejar dan menghindar merupakan teknik dasar yang diterapkan pada banyak game berbasis kecerdasan buatan dari yang sederhana sampai yang kompleks. apakah itu space shooters, RPG, atau game strategi.

* + 1. **Pola Pergerakan**

Pola pergerakan merupakan cara yang sederhana untuk memberikan ilusi kecerdasan pada sebuah game. Metode standar untuk menerapkan pola pergerakan adalah dengan cara menyimpan pola tersebut dalam suatu array. Array tersebut terdiri dari serangkaian koordinat atau perintah pergerakan dengan pola tertentu untuk mengontrol koordinat dari objek. Dengan metode ini, bisa didapatkan pola-pola pergerakan seperti melingkar, garis lurus, zig-zag atau bahkan kurva tak beraturan.

* + 1. **Pathfinding**

Metode pathfinding paling mudahditemui pada game-game bertipe strategi dimana kita menunjuk satu tokoh untuk digerakkan ke lokasi tertentu dengan mengklik lokasi yang hendak dituju.

* + 1. **Jaringan saraf tiruan (neural network)**

Neural network cukup baik ketika diterapkan pada kasus-kasus yang sifatnya non-linier atau mengambil keputusan yang tidak dapat dilakukan dengan metode tradisional. Penerapannya seringkali pada game-game yang memerlukan kemampuan adaptif atau belajar dari pengalaman.

* + 1. **Algoritma Genetis (genetic algorithm)**

Algoritma genetis sedikit banyak dipengaruhi oleh teori evolusi yang dicetuskan Darwin, yaitu bahwa spesies akan terus menerus beradaptasi dengan lingkungannya dan ciri khasnya yang terletak pada kromosom, akan diturunkan pada generasi berikutnya.

Beberapa teknik lain yang umum digunakan antara lain: *finite state machine, fuzzy logic, ruled-based AI, basic probability,* dan keputusan tak pasti dan lain-lain.

* 1. **Teori Jan-Ken-Pon**

Jan-Ken-Pon adalah sebuah [permainan](https://id.wikipedia.org/wiki/Permainan) menggunakan tangan dua orang atau lebih dengan membentuk suatu *gesture*. Permainan ini sering digunakan untuk [pemilihan](https://id.wikipedia.org/w/index.php?title=Pemilihan&action=edit&redlink=1) [acak](https://id.wikipedia.org/w/index.php?title=Acak&action=edit&redlink=1), seperti halnya pelemparan [koin](https://id.wikipedia.org/wiki/Koin), [dadu](https://id.wikipedia.org/wiki/Dadu), dan lain-lain. Beberapa permainan dan [olahraga](https://id.wikipedia.org/wiki/Olahraga) menggunakannya untuk menentukan peserta mana yang bermain terlebih dahulu. Kadang ia juga dipakai untuk menentukan [peran](https://id.wikipedia.org/w/index.php?title=Peran&action=edit&redlink=1) dalam permainan peran, maupun dipakai sebagai sarana [perjudian](https://id.wikipedia.org/wiki/Judi). Permainan ini dimainkan di berbagai belahan dunia.

Terdapat tiga [isyarat](https://id.wikipedia.org/wiki/Isyarat) tangan dalam permainan ini. Batu digambarkan oleh tangan mengepal, gunting digambarkan oleh jari telunjuk dan tengah, kertas digambarkan oleh tangan terbuka. Tujuan dari permainan adalah mengalahkan lawan bermain. Aturan standar adalah batu mengalahkan gunting, gunting mengalahkan kertas, dan kertas mengalahkan batu.

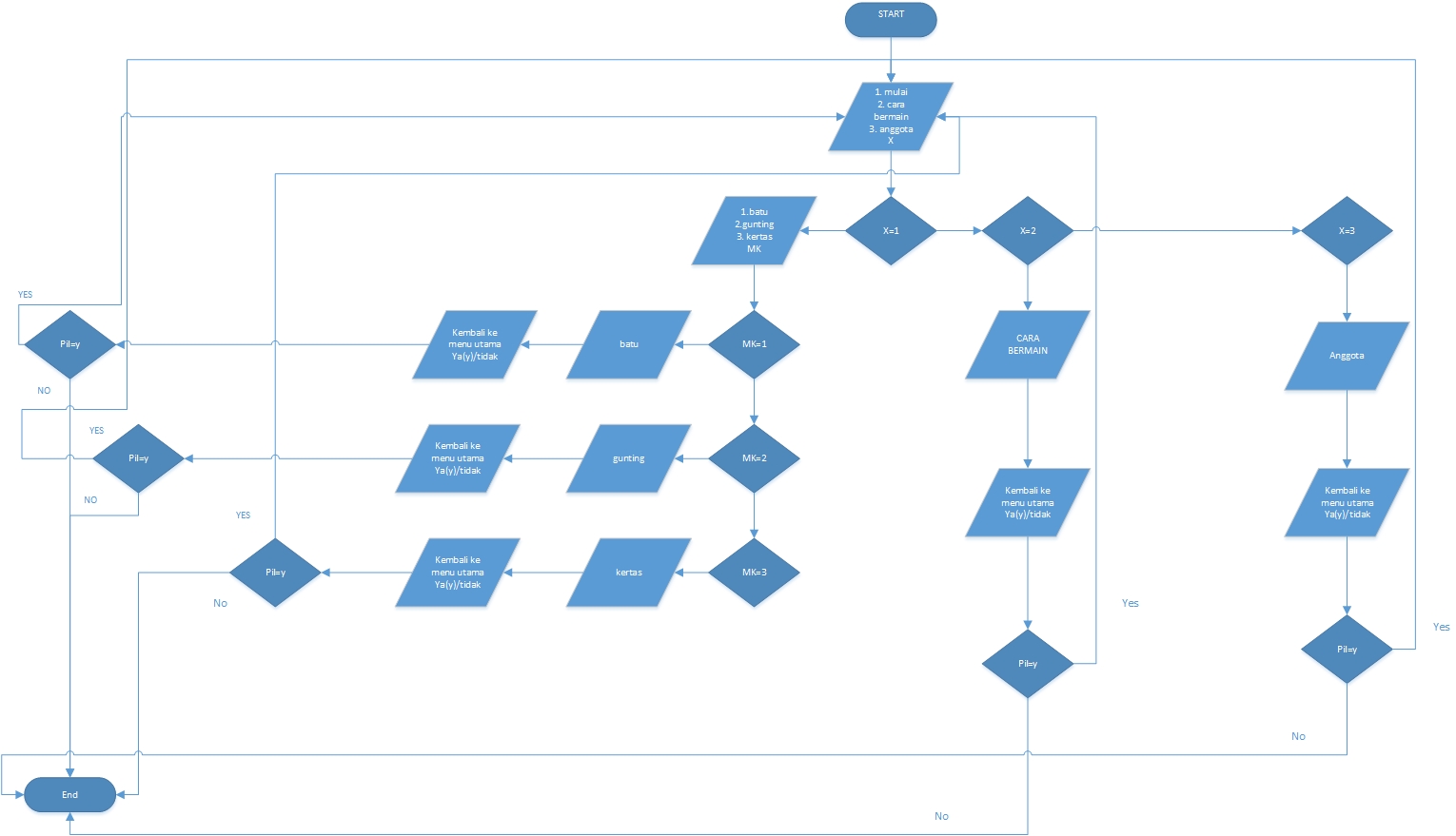
Jika kedua pemain mengeluarkan isyarat yang sama, maka permainan diulang. Kadangkala pemain menggunakan sistem berulang-ulang artinya sekali kemenangan tidak cukup untuk menghentikan permainan. Misalnya pemain yang menang 5 kali terlebih dahulu menjadi pemenang.

Ada "guu" (batu), bila jari tangan dikepalkan membentuk tinju. Lalu ada "Paa" (kertas), telapak tangan dibentangkan memperlihatkan ke lima jari. Dan "choki" (gunting), cukup dua jari membentuk huruf "V", batu akan kalah lawan kertas, karena kertas itu bisa menutupi batu. kertas akan kalah dengan gunting, karena kertas bisa dipotong oleh gunting. gunting akan kalah lawan batu, karena batu sangat keras dan membuat gunting akan tumpul. permainan ini sering dimainkan oleh orang jepang apabila mereka adu kalah atau menang pada sebuah permainan.

**BAB III**

**METODOLOGI**

* 1. **Alur Pembuatan Sistem**

****

Gambar 3.1.11 Flowchart

Pada gambar terlihat bahwasan nya flowchart yang kami buat mengikuti alur pembuatan program. Mulai dari star ke pilihan menu, terdapat 3 pilihan menu utama. Lalu pada alur main kita disuruh pilih antar gunting, batu atau kertas. Jika sudah memilih maka ada pilihan mau main lagi atau tidak. Jika iya maka akan kembali ke menu awal sedangkan jika pilih tidak maka program akan berakhir.

**BAB IV**

**HASIL DAN PEMBAHASAN**

* 1. **Tabel Kemungkinan**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **B** | **G** | **K** |
| **B** | **D** | **W** | **L** |
| **G** | **L** | **D** | **W** |
| **K** | **W** | **L** | **D** |

Table 1.12 Table Kemungkinan

Ini adalah table kemunginan untuk menentukan kemenangan dalam permainan ini terdapat perhitungan 1/3, apabila B vs B = D, B vs G =W, B vs K =L. Untuk keterangannya ada di bawah ini

**B : Batu**

**G : Gunting**

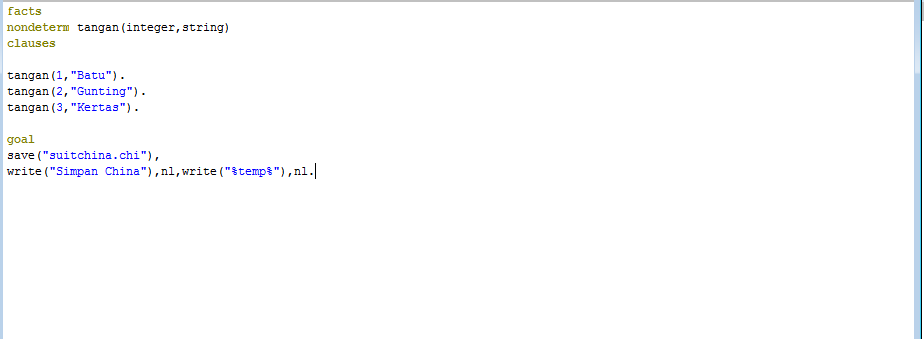
**K : Kertas**

**D : Draw**

**Win : Menang**

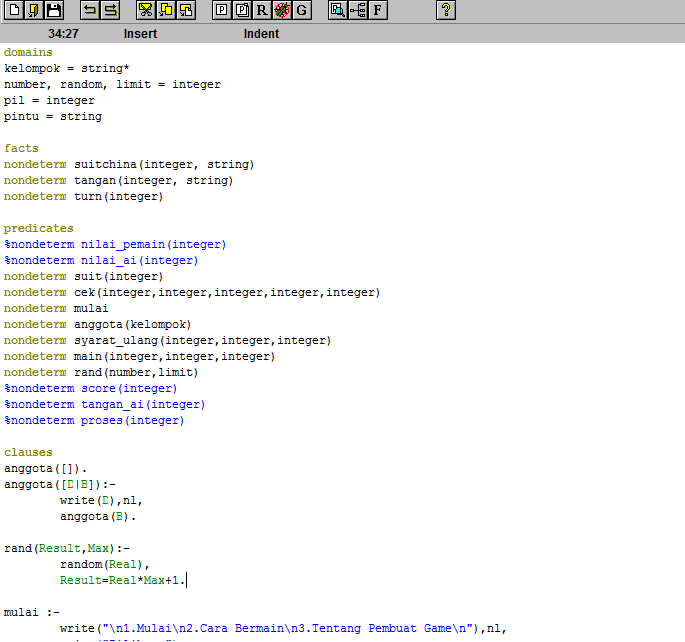
**Lose : Kalah**

* 1. **Analisis Aplikasi**



Gambar 1.12 JepunChance.pro

di file chinaChance. pro ini berisi fakta-fakta yang terdiri dari batu, gunting dan kertas yang berfungsi untuk random di file FhinalChina.pro

****

Gambar 2.13 Tampilan Syntax Program

**Domain** untuk menyatakan jenis argument atau jenis variable.  
**facts** merupakan isi dari sekumpulan fakta.

**Predicate** untuk memberi tau nama symbolic untuk relasi.

**determ** menentukan bahwa hanya memiliki fakta tunggal.

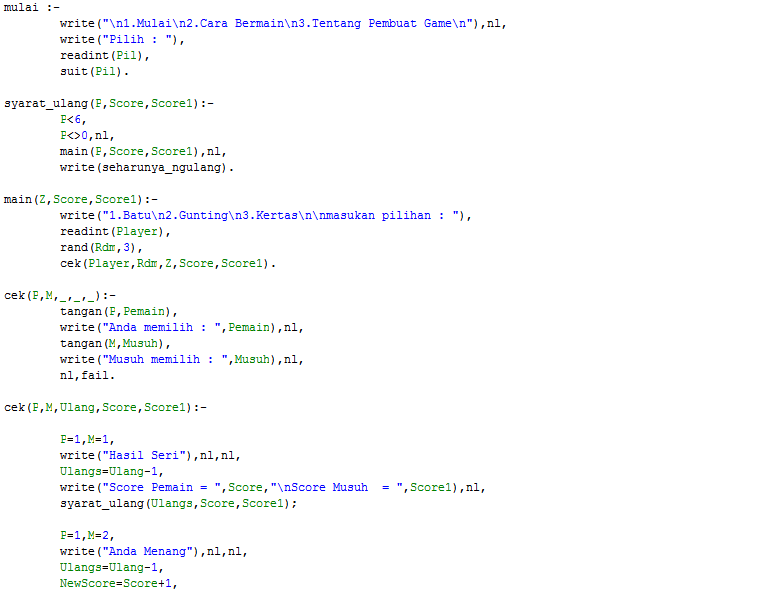
**Nondeterm** menentukan bahawa kemungkinan ada sejumlah fakta dari suatu. predikat sepanjang program berjalan.

**Clause** untuk meletakan fakta dan aturan .

**Anggota ([])** bahwa list kosong.

**Anggota([D][B])**  adalah kondisi nilai yang akan di keluarkan terlebih dahulu adalah variable D lalu di susul dengan variable B.

**Rand** hasil dari random akan di dapat nilainya.

****

Gambar 3.14 Tampilan Syntax Program

**syarat\_ulang** berfungsi untuk menerima nilai batasan dan nilai hasil pertandingan antara pemain dan komputer. Pada kondisi tersebut ada p<6 maksudnya bahwa kita hanya dapat memainkan 5 kali permainan dan p<> itu maksudnya bahwa nilai p tidak boleh = 0 .

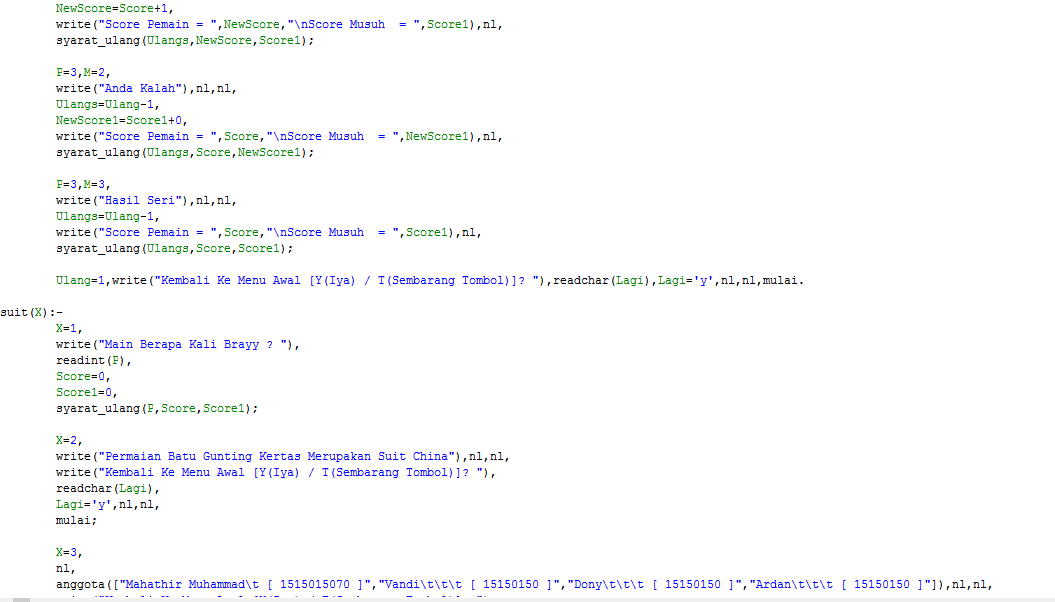
**main** untuk menentukan inputan batasan yang di terima dari syarat ulang serta menerima inputan dari pemilihan batu, gunting dan kertas, dan juga terdapat parameter dari cek.

**cek** berfungsi untuk menerima hasil dari inputan yang telah di kondisikan.



Gambar 5.15 Tampilan Syntax Program

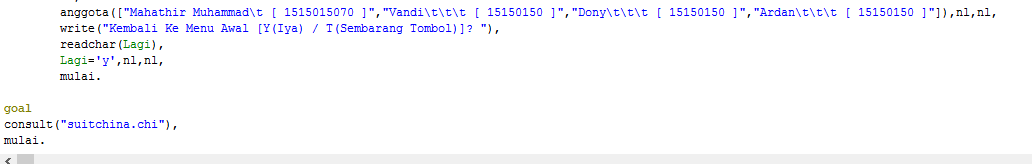
**write** kondisi untuk menentukan bahwa yang di pilih pemain dan yang di pilih program akan mengeluarkan hasil pertandingan tersebut dan Ulangs=Ulang-1 ini berfungsi untuk menentukan niai berapa kali kita ingin melakukan permainan akan berkurang 1sehingga looping dapat berhenti.



Gambar 6.16 Tampilan Syntax Program

**suit** berfungsi untuk menerima inputan berapa kali main

**readint** untuk mengetahui bahwa nilai yang kita inputkan bernilai data integer

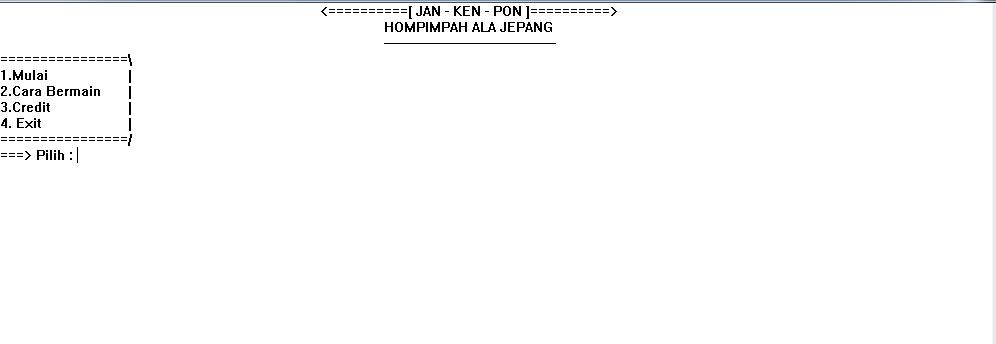


Gambar 7.17 Tampilan Syntax Program

**anggota**  untuk menampilkan anggota

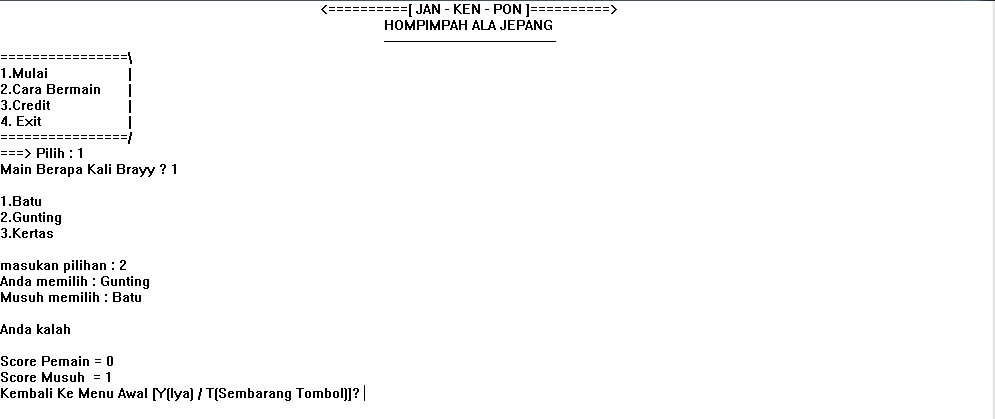
**readchar** untuk menginput data

**goal** untuk mengeksekusi program



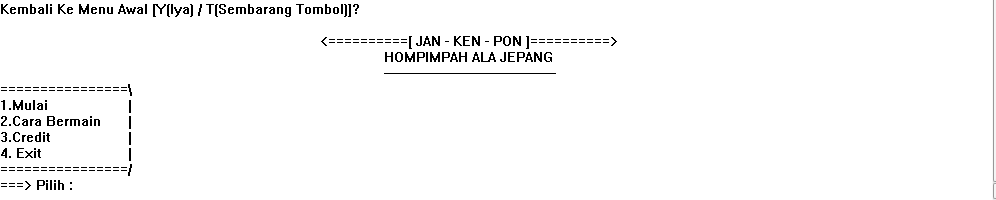
Gambar 8.18 Tampilan Awal

Tampilan awal ketika kita menjalankan program, di tampilan menu utama ini terdapat 4 pilihan yaitu mulai untuk memulai permainan, cara bermain untuk mengetahui cara bermain, credit untuk melihat anggota kelompok dan exit untuk keluar dari program



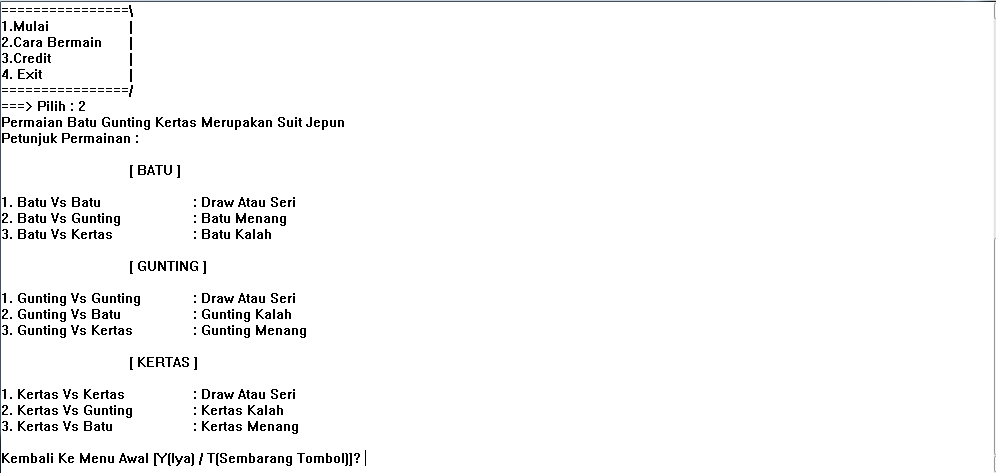
Gambar 9.18 Tampilan Memulai Permainan

jika kita pilih angka 1 maka kita akan memulai permainan dan kita akan di minta untuk bermain berapa kali di program ini kami memberikan batasan sebanyak 5 kali setelah itu maka kita akan di suruh memilih pilihan yang sudah di tentukan jika kita sudah memilih maka akan menampilkan hasilnya.



Gambar 10.18 Kembali Ke Menu Awal

Jika kita menekan tombol Y maka kita akan kembali ke menu awal, apabila menekan tombol T maka program akan berhenti.



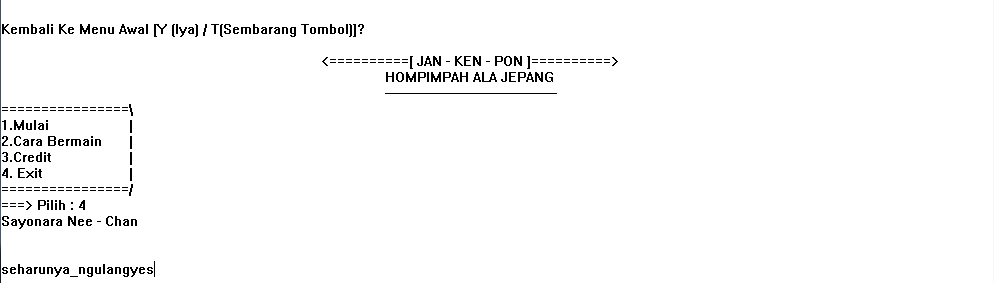
Gambar 11.19 Tampilan Cara Bermain

Jika kita pilih angka 2 maka kita akan di berikan aturan dalam permainan seperti pada gambar



Gambar 12.19 Tampilan Credit

Pada tampilan ini akan muncul anggota kelompok 6



Gambar 13.19 Tampilan Exit

Apabila kita menekan tombol 4 maka akan muncul pesan seperti gambar di atas dan program akan keluar secara otomatis

**BAB V**

**PENUTUP**

* 1. **Kesimpulan**

Dari project akhir Kecerdasan Buatan yang berjudul ”Permainan Jan-Ken-Pon” yang telah kami buat dapat kami simpulkan bahwa :

1. Permaianan Jan-Ken-Pon ini merupakan solusi terbaik dalam mengatasi Kejenuhan bila kita merasa bosan terhadapat permainan lainnya
2. Permainan ini pula dapat dikembangkan kembali jauh lebih maju dengan tampilan visual.
   1. **Saran**

Kekurangan yang ada pada program ini yang kami buat ini adalah ketidak adaan nya pemain melawan pemain, oleh karena itu program ini hanya dapat kita lakukan bermain melawan program itu sendiri. Maka dari itu Seiring berjalannya waktu, kami akan terus berusaha untuk menyempurnakan program yang kami buat ini agar program kami dapat digunakan dan bermanfaat lagi.

**DAFTAR PUSTAKA**

* <http://rico-ariyanto.blogspot.co.id/2010/12/permainan-suit-jepang.html>
* <http://www.otakjepang.com/2014/05/inilah-3-langkah-cara-memenangkan-suit-jepang.html>
* <http://www.anibee.tv/news/id/games-hobby/5782/apakah-kalian-tahu-sejarah-tentang-suit-jepang-batu-gunting-dan-kertas>
* <https://softwareengineering.stackexchange.com/questions/84407/why-is-prolog-good-for-ai-programming>
* <https://en.wikipedia.org/wiki/Prolog>

**LAMPIRAN**

domains

kelompok = string\*

number, random, limit = integer

pil = integer

pintu = string

facts

nondeterm suitchina(integer, string)

nondeterm tangan(integer, string)

nondeterm turn(integer)

predicates

%nondeterm nilai\_pemain(integer)

%nondeterm nilai\_ai(integer)

nondeterm suit(integer)

nondeterm cek(integer,integer,integer,integer,integer)

nondeterm mulai

nondeterm anggota(kelompok)

nondeterm syarat\_ulang(integer,integer,integer)

nondeterm main(integer,integer,integer)

nondeterm rand(number,limit)

%nondeterm score(integer)

%nondeterm tangan\_ai(integer)

%nondeterm proses(integer)

clauses

anggota([]).

anggota([D|B]):-

write(D),nl,

anggota(B).

rand(Result,Max):-

random(Real),

Result=Real\*Max+1.

mulai :-

write("\t\t\t\t\t<==========[ JAN - KEN - PON ]==========>"),nl,

write("\t\t\t\t\t\tHOMPIMPAH ALA JEPANG"),nl,

write("\t\t\t\t\t\t-------------------------------------------"),nl,

write("================\\\n1.Mulai\t\t|\n2.Cara Bermain\t|\n3.Credit\t\t|\n4. Exit\t\t|\n"),

write("================/\n===> Pilih : "),

readint(Pil),

suit(Pil).

