**LAPORAN PRAKTIKUM KECERDASAN BUATAN**

**SISTEM PENDUKUNG KEPUTUSAN UNTUK MENENTUKAN KOMPONEN PC BERDASARKAN BUDGET DAN KEBUTUHAN**

****

**Disusun Oleh :**

Muhammad Kharis 1515015004

Hanif Aulia 1515015012

Hasyim Asyari 1515015030

**Asisten Praktikum :**

|  |  |  |
| --- | --- | --- |
| Niken Novirasari  1415015064 | Anisa Nur Afiyah  1415015068 | 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**

Dengan mengucapkan puji syukur kehadirat Tuhan yang Maha Esa, yang telah memberikan berkat, rahmat, serta karunia-Nya, sehingga kami dapat menyelesaikan laporan ini. Adapun maksud dan tujuan dari penulisan laporan ini untuk mengetahui seberapa besar kita mendalami materi praktikum yang telah diajarkan oleh asistan laboratorium masing – masing dan merupakan tugas akhir dari praktikum kami di semester 4 ini. Namun dalam penyusunannya, kami menyadari masih banyak kekurangan dan jauh dari taraf kesempurnaan. Selain itu banyak sekali kesulitan yang kami alami dalam membuat laporan ini terutama dalam membuat program. Oleh karena itu, dengan rendah hati kami membutuhkan saran dan kritik yang sifatnya membangun dari semua pembaca Dalam kesempatan ini perkenankanlah kami menyampaikan rasa terima kasih kepada Dosen, Aslab Praktikum dan Aslab Penanggung Jawab Mata Kuliah Kecerdasan Buatan yaitu :

1. Joan Angelina W, S.kom, M.kom
2. Masnawati, MT
3. Niken Novirasari
4. Anisa N. Afiyah
5. M. Hilmy Adi S

Satu harapan yang kami inginkan semoga laporan ini dapat berguna bagi pembaca.

Samarinda, 4 Mei 2017

Penulis

**DAFTAR ISI**

**Halaman Judul** i

**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 2

**BAB II Landasan Teori**  3

* 1. Fungsi-Fungsi Visual Prolog yang Digunakan 3

2.1.1 Unifikasi dan Lacakbalik 3

2.1.2 Data Object Sederhana dan Jamak 5

2.1.3 Perulangan dan Rekursi ................................................... 7

2.1.4 List ................................................................................... 8

2.2 Teori Keilmuan yang diimplementasikan ..................................... 9

**BAB III Metodologi** 10

3.1 Alur Pembuatan Sistem 10

**BAB IV Hasil dan Pembahasan** 14

4.1 Tabel Kebenaran 14

4.2 Analisis Aplikasi 14

**BAB V Penutup** 18

5.1 Kesimpulan 18

5.2 Saran 18

**Daftar Pustaka**  19

**Lampiran**  20

1 Source Code 20

2.Kartu Konsul 29

**BAB I**

**Pendahuluan**

* 1. **Latar Belakang**

Ilmu pengetahuan dan teknologi selalu berkembang dan mengalami kemajuan, sesuai dengan perkembangan zaman dan perkembangan cara berpikir manusia. Bangsa Indonesia sebagai salah satu negara berkembang tidak akan bisa maju selama belum memperbaiki kualitas sumber daya manusia bangsa kita. Kualitas hidup bangsa dapat meningkat jika ditunjang dengan sistem pendidikan yang mapan. Dengan sistem pendidikan yang mapan, memungkinkan kita berpikir kritis, kreatif, dan produktif. Jadi kami disini menyediakan suatu program untuk membantu menentukan bagaimana cara memilih komponen *PC* dengan sesuai *budget* yang anda punya. Misalkan anda akan membeli suatu komponen *PC* untuk *Game* dengan *budget* sekian. Dan *program* ini akan langsung menunjukkan beberapa komponen *PC* yang dibutuhkan yang sesuai dengan budget yang anda miliki.

* 1. **Rumusan Masalah**

Berdasarkan deskripsi masalah yang dipaparkan diatas, maka rumusan masalah dalam pembuatan program menentukan komponen *PC* berdasarkan *budget* dan kebutuhan adalah :

1. Bagaimana membuat program sistem pendukung menentukan komponen *PC* berdasarkan *budget* dan kebutuhan dengan menggunakan *Visual* *Prolog* ?
2. Bagaimana cara membuat suatu program dapat membantu dalam menentukan komponen *PC* berdasarkan budget dan kebutuhan ?
   1. **Batasan Masalah**
3. Program ini hanya bisa menentukan komponen *PC* dengan *budget* yang sudah di tentukan.
4. Program ini dibuat dengan menggunakan *Visual Prolog.*
   1. **Tujuan dan Manfaat**
      * **Tujuan**
5. Merancang suatu program yang dapat memudahkan pada saat pembelian komponen pc dengan budget yang ada.
6. Membuat program sistem pendukung keputusan untuk menentukan komponen *PC* berdasarkan *budget* dan kebutuhan dalam *Visual Prolog.*
   * + **Manfaat**
7. Untuk mempermudah dalam menentukan komponen *PC* berdasarkan budget dan kebutuhan yang diinginkan.

**BAB II**

**Pembahasan**

* 1. **Fungsi – Fungsi Visual Prolog yang Digunakan**
     1. **Unifikasi Dan Lacakbalik**

1. Unifikasi (*Unification*)

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

Ada beberapa hal penting dalam proses pencocokan atau unifikasi, yaitu:

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

1. Lacakbalik (*Backtracking*)

Pada waktu menyelesaikan masalah, seringkai 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.

* + 1. **Data Object Sederhana dan Jamak**

1. Data Object sederhana

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

* *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.

* *Konstanta*

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

* Karakter

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

Jika kita menginginkan sebuah *backslash* atau petik tunggal menjadi karakter menuliskannya harus didahului dengan sebuah tanda *backslash*. Contoh : ‘\\’ (*backslash*), ‘\’’ (tanda petik tunggal). Beberapa karakter mempunyai fungsi khusus ketika didahului oleh karakter *escape* ( \ ). Contoh : ‘\n’ (ganti baris), ‘\t’ (tabulasi).

Konstanta karakter dapat juga ditulis berdasarkan kode *ASCII*-nya, dengan didahului *backslash*. Contoh : ‘\64’ (@), ‘\90’ (Z).

* Angka

Angka merupakan salah satu dari integer atau real.

* Atom

Sebuah atom terdiri dari sebuah simbol atau sebuah *string.* Perbedaan antara simbol dan string berkaitan dengan representasi dan implementasi mesin. *Visual Prolog* melakukan *konversi* tipe otomatis diantara *domain string* dan *domain* simbol. Jadi simbol dapat digunakan untuk *domain string* dan *string* dapat digunakan untuk *domain* simbol. Namun *string* dinyatakan sebagai sesuatu yang berada diantara tanda petik ganda sedang simbol tidak membutuhkan tanda petik ganda.

Simbol dimulai dengan sebuah huruf kecil dan hanya boleh berisikan huruf, angka, dan *underscore*. *String* adalah sesuatu yang diapit tanda petik ganda dan berisikan kombinasi dari karakter, kecuali *ASCII NULL* (0), yang dipakai untuk menandai akhir dari *string*.

1. Data objek Jamak

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

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

Deklarasi domain-campuran jamak bermaksud:

* memiliki sebuah argumen dengan kemungkinan lebih dari satu tipe argumen;
* memiliki beberapa macam argumen, masing-masing dengan tipe yang berbeda;
* memiliki beberapa macam argumen, beberapa diantaranya dengan kemungkinan lebih dari satu tipe argumen.
* Agar suatu *predikat* *Visual Prolog* dapat menerima suatu argumen yang memberikan informasi dengan tipe yang berbeda maka *functor* tersebut harus dideklarasikan. Contoh berikut memperlihatkan klausa umur yang dapat menerima suatu argumen usia dengan tipe yang berbeda yaitu *string*, *real* atau *integer*.
  + 1. **Perulangan dan Rekursi**

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

1. Proses Perulangan

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

* Lacakbalik

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

* Implementasi Lacakbalik dengan Loop

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

* *Rekursi*

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

* Rekursi Ekor (*Tail Recursion*)

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

1. Struktur Data Rekursif

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

* + 1. **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.

***Head* dan *Tail* (Kepala dan Ekor)**

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

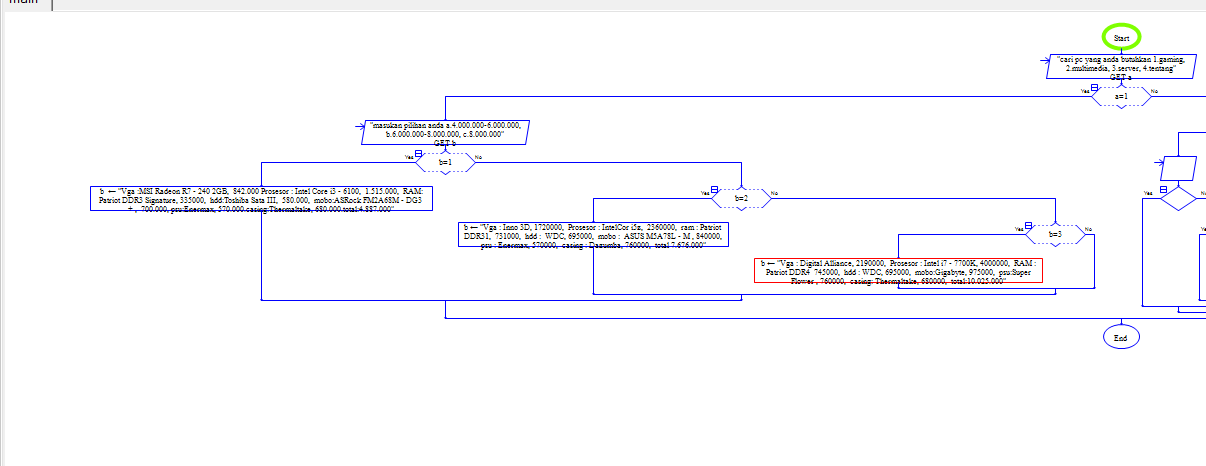
* 1. **Teori Keilmuan yang di Implementasikan**

Teori yang di implementasikan SPK ini memiliki kemudahan bagi penggunanya. SPK ini sangat bagus karena untuk memudahkan orang untuk mencari komponen PC dengan *budget* yang sudah disediakan. SPK ini hanya membantu seorang pembeli dalam mencari komponen *PC* yang sesuai dengan *budget* dan kebutuhan.

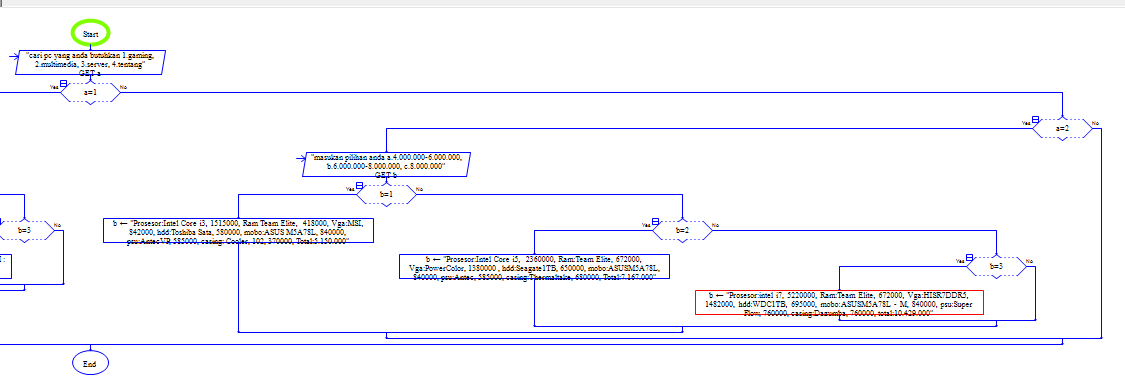
**BAB III**

**Metedologi**

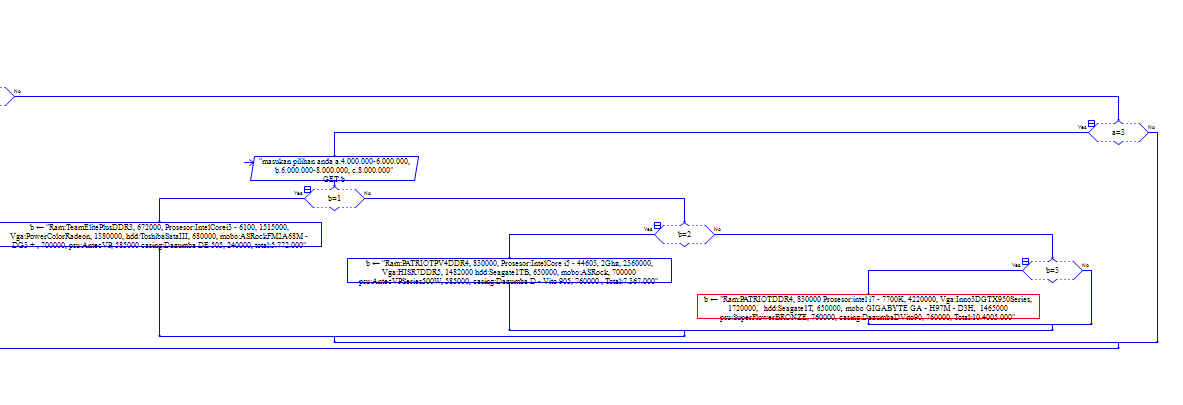
* 1. **Alur Pembuatan Sistem**
* **Flowchart**

****

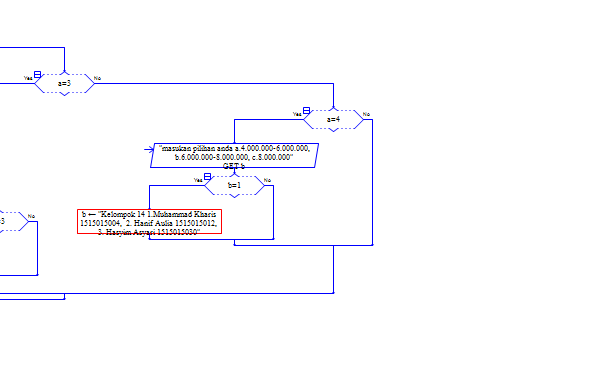
**Gambar 1**

****

**Gambar 2**

****

**Gambar 3**

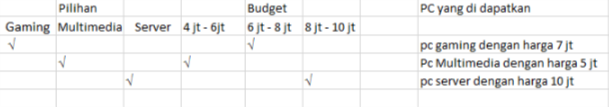
****

**Gambar 4**

**BAB IV**

**Hasil dan Pembahasan**

**4.1. Tabel Kebenaran**

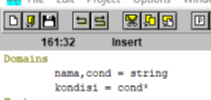


**Gambar 5. Tabel Kebenaran**

**4.2. Analisis Aplikasi**

* **Domains**

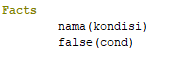
Domains digunakan untuk mendeklarasikan predikat, dan juga bisa digunakan untuk deklarasi tipe data.



**Gambar 6 Domains**

* **Facts**

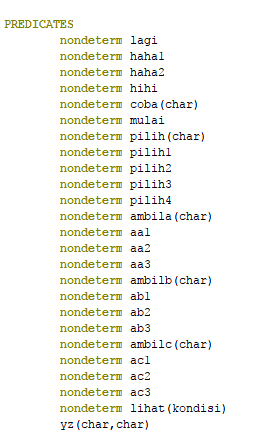
Terdiri dari fakta – fakta yang dapat ditambah dan dihapus secara langsung saat program sedang dijalankan.



**Gambar 7. Facts**

* **Predikat**

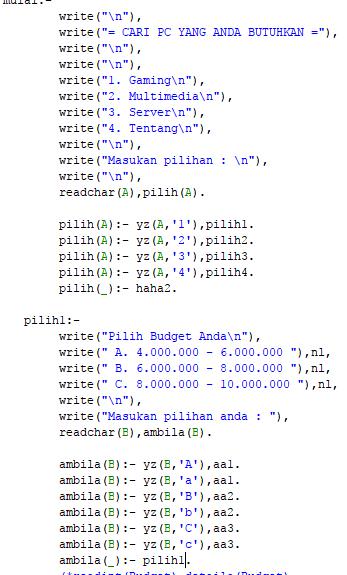
Pada predikat digunakan nondeterm yang fungsinya adalah untuk mengatasi apabila pada saat *RUN* terdapatlebih dari satu data yang sama, sehingga tetap dapat muncul hasilnya. Dan yz sendiri adalah sebuah variable baru.



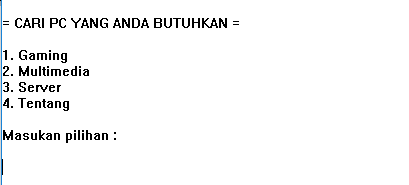
**Gambar 8. Predikat pada program**

* **Unifikasi**

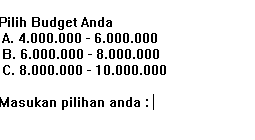
Unifikasi adalah suatu pencocokan suatu klausa



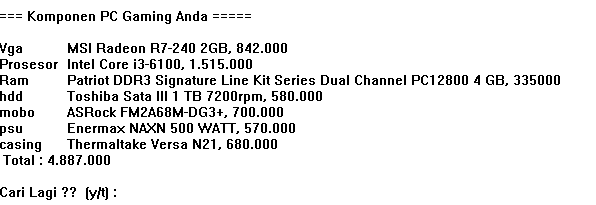
**Gambar 9. Unifikasi pada program**



**Gambar 10. Tampilan awal program**

****

**Gambar 11. Tampilan kedua program**

****

**Gambar 12. Tampilan hasil keputusan**

**BAB V**

**Penutup**

**5.1. Kesimpulan**

Aplikasi ini hanya untuk menampilkan pencarian semua komponen PC yang dibutuhkan dengan data yang sudah ada. Dan aplikasi ini juga membantu pembeli yang ingin membeli suatu komponen PC dengan *budget* yang ada.

**5.2. Saran**

Semoga aplikasi ini bisa dikembangkan oleh ingin membuatnya lebih baikm lagi dari pada program yang kami buat dan efektif dan lebih bermanfaat lagi.

**DAFTAR PUSTAKA**

<https://jalantikus.com/tips/trik/rakit-pc-gaming-5-jutaan/>

(Di akses 26 – 04 – 2017).

[**http://vinnick86.blogspot.co.id/2007/11/unifikasi-dan-lacakbalik.html**](http://vinnick86.blogspot.co.id/2007/11/unifikasi-dan-lacakbalik.html).

(Di akses 01 – 05 – 2017)

Modul Praktikum Kecerdasan Buatan

(Di akses 01 – 05 – 2017)

**LAMPIRAN**

1. **Source Code**

Domains

nama,cond = string

kondisi = cond\*

Facts

nama(kondisi)

false(cond)

PREDICATES

nondeterm lagi

nondeterm haha1

nondeterm haha2

nondeterm hihi

nondeterm coba(char)

nondeterm mulai

nondeterm pilih(char)

nondeterm pilih1

nondeterm pilih2

nondeterm pilih3

nondeterm pilih4

nondeterm ambila(char)

nondeterm aa1

nondeterm aa2

nondeterm aa3

nondeterm ambilb(char)

nondeterm ab1

nondeterm ab2

nondeterm ab3

nondeterm ambilc(char)

nondeterm ac1

nondeterm ac2

nondeterm ac3

nondeterm lihat(kondisi)

yz(char,char)

CLAUSES

yz(Y,Y):- !.

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

lagi:-

write("Cari Lagi ?? (y/t) : \n"),

readchar(A),coba(A).

coba(A):- yz(A,'y'),mulai.

coba(A):- yz(A,'t'),hihi.

coba(\_):- haha1.

haha1:-

write("Mohon Masukan Yang Benar !!\n"),

lagi.

haha2:-

write("Mohon Masukan Yang Benar !!\n"),

mulai.

hihi:-

write("=== Terima Kasih Telah Mencoba Semoga Membantu Anda ===\n"),

exit.

mulai:-

write("\n"),

write("= CARI PC YANG ANDA BUTUHKAN ="),

write("\n"),

write("\n"),

write("1. Gaming\n"),

write("2. Multimedia\n"),

write("3. Server\n"),

write("4. Tentang\n"),

write("\n"),

write("Masukan pilihan : \n"),

write("\n"),

readchar(A),pilih(A).

pilih(A):- yz(A,'1'),pilih1.

pilih(A):- yz(A,'2'),pilih2.

pilih(A):- yz(A,'3'),pilih3.

pilih(A):- yz(A,'4'),pilih4.

pilih(\_):- haha2.

pilih1:-

write("Pilih Budget Anda\n"),

write(" A. 4.000.000 - 6.000.000 "),nl,

write(" B. 6.000.000 - 8.000.000 "),nl,

write(" C. 8.000.000 - 10.000.000 "),nl,

write("\n"),

write("Masukan pilihan anda : "),

readchar(B),ambila(B).

ambila(B):- yz(B,'A'),aa1.

ambila(B):- yz(B,'a'),aa1.

ambila(B):- yz(B,'B'),aa2.

ambila(B):- yz(B,'b'),aa2.

ambila(B):- yz(B,'C'),aa3.

ambila(B):- yz(B,'c'),aa3.

ambila(\_):- pilih1.

/\*readint(Budget),detaila(Budget).

detaila(Budget):-

Budget>4000000,aa1.

detaila(Budget):-

Budget>6000000,aa2.\*/

aa1:-

write("\n"),

write("\n"),

write("=== Komponen PC Gaming Anda ====="),nl,

write("\n"),

write("Vga\t"),

write(" MSI Radeon R7-240 2GB, 842.000 \n"),

write("Prosesor\t"),

write(" Intel Core i3-6100, 1.515.000 \n"),

write("Ram\t"),

write(" Patriot DDR3 Signature Line Kit Series Dual Channel PC12800 4 GB, 335000 \n"),

write("hdd\t"),

write(" Toshiba Sata III 1 TB 7200rpm, 580.000 \n"),

write("mobo\t"),

write(" ASRock FM2A68M-DG3+, 700.000 \n"),

write("psu\t"),

write(" Enermax NAXN 500 WATT, 570.000 \n"),

write("casing\t"),

write(" Thermaltake Versa N21, 680.000 \n"),

write(" Total : 4.887.000\n"),

write("\n"),

lagi.

aa2:-

write("\n"),

write("\n"),

write("=== Komponen PC Gaming Anda====="),nl,

write("\n"),

write("Vga\t"),

write(" Inno 3D GTX 950 2 GB OC Series,1720000 \n"),

write("Prosesor\t"),

write(" Intel Core i5-4460 3,2Ghz, 2360000 \n"),

write("Ram\t"),

write(" Patriot DDR3 Viper 3 Series Dual Channel PC19000 8 GB CL11, 731000 \n"),

write("hdd\t"),

write(" WDC Caviar Blue 1TB,695000 \n"),

write("mobo\t"),

write(" ASUS M5A78L-M LE/USB,840000 \n"),

write("psu\t"),

write(" Enermax NAXN 500 WATT,570000 \n"),

write("casing\t"),

write(" Dazumba D-Vito 905,760000 \n"),

write(" Total : 7.676.000\n"),

write("\n").

aa3:-

write("\n"),

write("\n"),

write("=== Komponen PC Gaming Anda====="),nl,

write("\n"),

write("Vga\t"),

write(" Digital Alliance GeForce GTX 1050 Ti 4 GB DDR5 StormX Series,2190000 \n"),

write("Prosesor\t"),

write(" Intel i7-7700K (Box) (4.2Ghz Up To 4.5Ghz, 4000000 \n"),

write("Ram\t"),

write(" Patriot DDR4 Viper 4 Xtreme Dual Channel PC19200 8 GB CL10 (2X4 GB), 745000 \n"),

write("hdd\t"),

write(" WDC Caviar Blue 1TB,695000 \n"),

write("mobo\t"),

write(" Gigabyte GA-H110M-DS2,975000 \n"),

write("psu\t"),

write(" Super Flower BRONZE FX 550 W 80+ Bronze,760000 \n"),

write("casing\t"),

write(" Thermaltake Versa N21,680000 \n"),

write(" Total : 10.025.000\n"),

write("\n").

pilih2:-

write("Pilih Budget Anda\n"),

write(" A. 4.000.000 - 6.000.000 "),nl,

write(" B. 6.000.000 - 8.000.000 "),nl,

write(" C. 8.000.000 - 10.000.000 "),nl,

write("\n"),

write("Masukan Pilihan Anda : "),

readchar(C),ambilb(C).

ambilb(C):- yz(C,'A'),ab1.

ambilb(C):- yz(C,'a'),ab1.

ambilb(C):- yz(C,'B'),ab2.

ambilb(C):- yz(C,'b'),ab2.

ambilb(C):- yz(C,'C'),ab3.

ambilb(C):- yz(C,'c'),ab3.

ambilb(\_):- pilih3.

ab1:-

write("\n"),

write("\n"),

write("=== Komponen PC Multimedia Anda====="),nl,

write("\n"),

write("Prosesor\t"),

write(" Intel Core i3-6100, 1515000 \n"),

write("Ram\t"),

write(" Team Elite Plus DDR3 PC12800 1600Mhz Dual Channel 4 GB (2X2) GB, 418000 \n"),

write("Vga\t"),

write(" MSI Radeon R7-240 2GB, 842000 \n"),

write("hdd\t"),

write(" Toshiba Sata III 1 TB 7200rpm 3.5 Inch, 580000 \n"),

write("mobo\t"),

write(" ASUS M5A78L-M LE/USB3,840000 \n"),

write("psu\t"),

write(" Antec VP Series 500 W,585000 \n"),

write("casing\t"),

write(" Cooler Master CMP 102,370000 \n"),

write(" Total : 5.150.000\n"),

write("\n"),

lagi.

ab2:-

write("\n"),

write("\n"),

write("=== Komponen PC Multimedia Anda====="),nl,

write("\n"),

write("Prosesor\t"),

write(" Intel Core i5-4460 3,2Ghz, 2360000 \n"),

write("Ram\t"),

write(" Team Elite Plus DDR 3 PC 12800 Dual Channel 8 GB, 672000 \n"),

write("Vga\t"),

write(" PowerColor Radeon RX 460 2 GB Red Dragon, 1380000 \n"),

write("hdd\t"),

write(" Seagate 1TB,650000 \n"),

write("mobo\t"),

write(" ASUS M5A78L-M LE/USB3, 840000 \n"),

write("psu\t"),

write(" Antec VP Series 500 W, 585000 \n"),

write("casing\t"),

write(" Thermaltake Versa N21, 680000 \n"),

write(" Total : 7.167.000\n"),

write("\n"),

lagi.

ab3:-

write("\n"),

write("\n"),

write("=== Komponen PC Multimedia Anda====="),nl,

write("\n"),

write("Prosesor\t"),

write(" intel i7-7700K (Box) (4.2Ghz Up To 4.5Ghz, 5220000 \n"),

write("Ram\t"),

write(" Team Elite Plus DDR 3 PC 12800 Dual Channel 8 GB, 672000 \n"),

write("Vga\t"),

write(" HIS R7 360 Icooler OC 2GB DDR5, 1482000 \n"),

write("hdd\t"),

write(" WDC Caviar Blue 1TB, 695000 \n"),

write("mobo\t"),

write(" ASUS M5A78L-M LE/USB3,840000 \n"),

write("psu\t"),

write(" Super Flower BRONZE FX 550 W 80+ Bronze, 760000 \n"),

write("casing\t"),

write(" Dazumba D-Vito 905, 760000 \n"),

write(" Total : 10.429.000\n"),

write("\n"),

lagi.

pilih3:-

write("Pilih Budget Anda\n"),

write(" A. 4.000.000 - 6.000.000 "),nl,

write(" B. 6.000.000 - 8.000.000 "),nl,

write(" C. 8.000.000 - 10.000.000 "),nl,

write("\n"),

write("Masukan pilihan : "),

readchar(D),ambilc(D).

ambilc(D):- yz(D,'A'),ac1.

ambilc(D):- yz(D,'a'),ac1.

ambilc(D):- yz(D,'B'),ac2.

ambilc(D):- yz(D,'b'),ac2.

ambilc(D):- yz(D,'C'),ac3.

ambilc(D):- yz(D,'c'),ac3.

ambilc(\_):- pilih3.

ac1:-

write("\n"),

write("\n"),

write("=== Komponen PC Server Anda====="),nl,

write("\n"),

write("Ram\t"),

write(" Team Elite Plus DDR 3 PC 12800 Dual Channel 8 GB, 672000 \n"),

write("Prosesor\t"),

write(" Intel Core i3-6100, 1515000 \n"),

write("Vga\t"),

write(" PowerColor Radeon RX 460 2 GB Red Dragon, 1380000 \n"),

write("hdd\t"),

write(" Toshiba Sata III 1 TB 7200rpm 3.5 Inch, 680000 \n"),

write("mobo\t"),

write(" ASRock FM2A68M-DG3+, 700000 \n"),

write("psu\t"),

write(" Antec VP Series 500 W, 585000 \n"),

write("casing\t"),

write(" Dazumba DE 505,240000 \n"),

write(" Total : 5.772.000\n"),

write("\n"),

lagi.

ac2:-

write("\n"),

write("\n"),

write("=== Komponen PC Server Anda====="),nl,

write("\n"),

write("Ram\t"),

write(" PATRIOT PV4 8G 240 C5K ( 2 x 4Gb ) DDR4, 830000 \n"),

write("Prosesor\t"),

write(" Intel Core i5-4460 3,2Ghz, 2360000 \n"),

write("Vga\t"),

write(" HIS R7 360 Icooler OC 2GB DDR5, 1482000 \n"),

write("hdd\t"),

write(" Seagate 1TB, 650000 \n"),

write("mobo\t"),

write(" ASRock FM2A68M-DG3+, 700000 \n"),

write("psu\t"),

write(" Antec VP Series 500 W, 585000 \n"),

write("casing\t"),

write(" Dazumba D-Vito 905,760000 \n"),

write(" Total : 7.367.000\n"),

write("\n"),

lagi.

ac3:-

write("\n"),

write("\n"),

write("=== Komponen PC Server Anda====="),nl,

write("\n"),

write("Ram\t"),

write(" PATRIOT PV4 8G 240 C5K ( 2 x 4Gb ) DDR4, 830000 \n"),

write("Prosesor\t"),

write(" intel i7-7700K (Box) (4.2Ghz Up To 4.5Ghz, 4220000 \n"),

write("Vga\t"),

write(" Inno 3D GTX 950 2 GB OC Series, 1720000 \n"),

write("hdd\t"),

write(" Seagate 1T,650000 \n"),

write("mobo\t"),

write(" GIGABYTE GA-H97M-D3H, 1465000 \n"),

write("psu\t"),

write(" Super Flower BRONZE FX 550 W 80+ Bronze,760000 \n"),

write("casing\t"),

write(" Dazumba D-Vito 90,760000 \n"),

write(" Total : 10.4005.000\n"),

write("\n"),

lagi.

nama(["1.Muhammad Kharis 1515015004",

"2.Hanif Aulia 1515015012",

"3.Hasyim Asyari 1515015030"]).

pilih4:-

write("\tKelompok 14\n"),

nama(Kami),

lihat(Kami).

lihat ([H|T]):-

not(false(H)),

write(H),nl,

lihat(T).

lihat([H|\_]):-

assertz(false(H)),

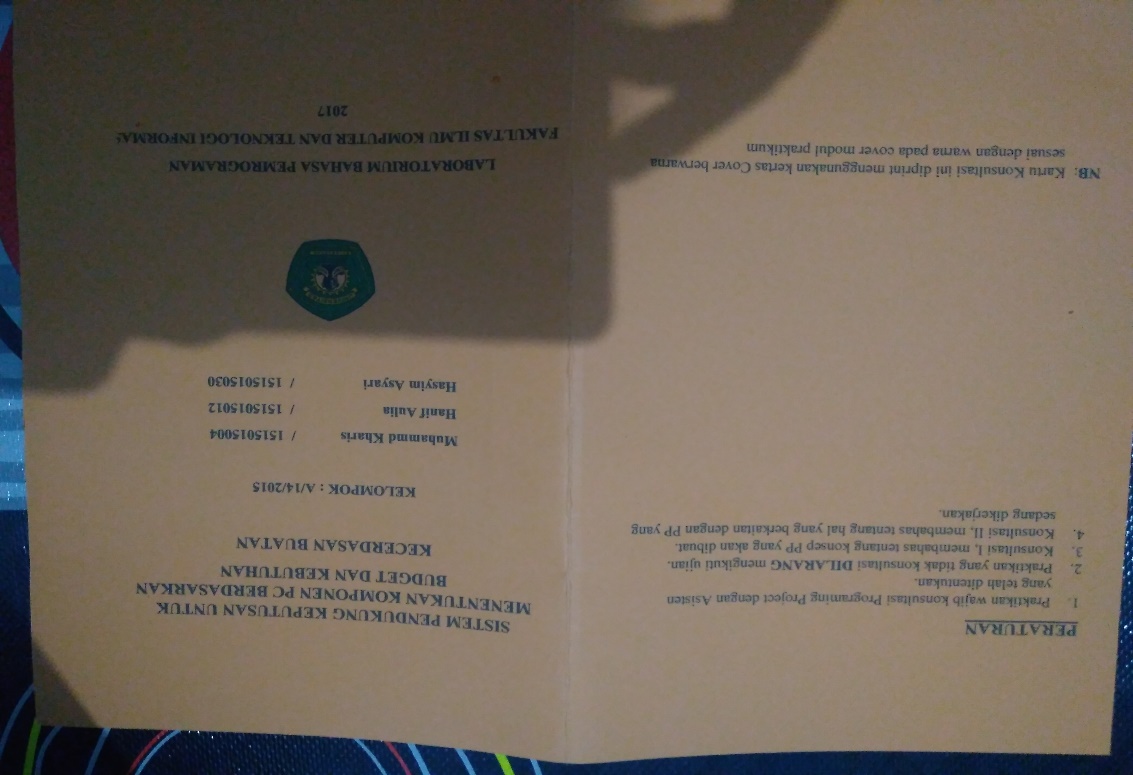
write("\n"),

lagi.

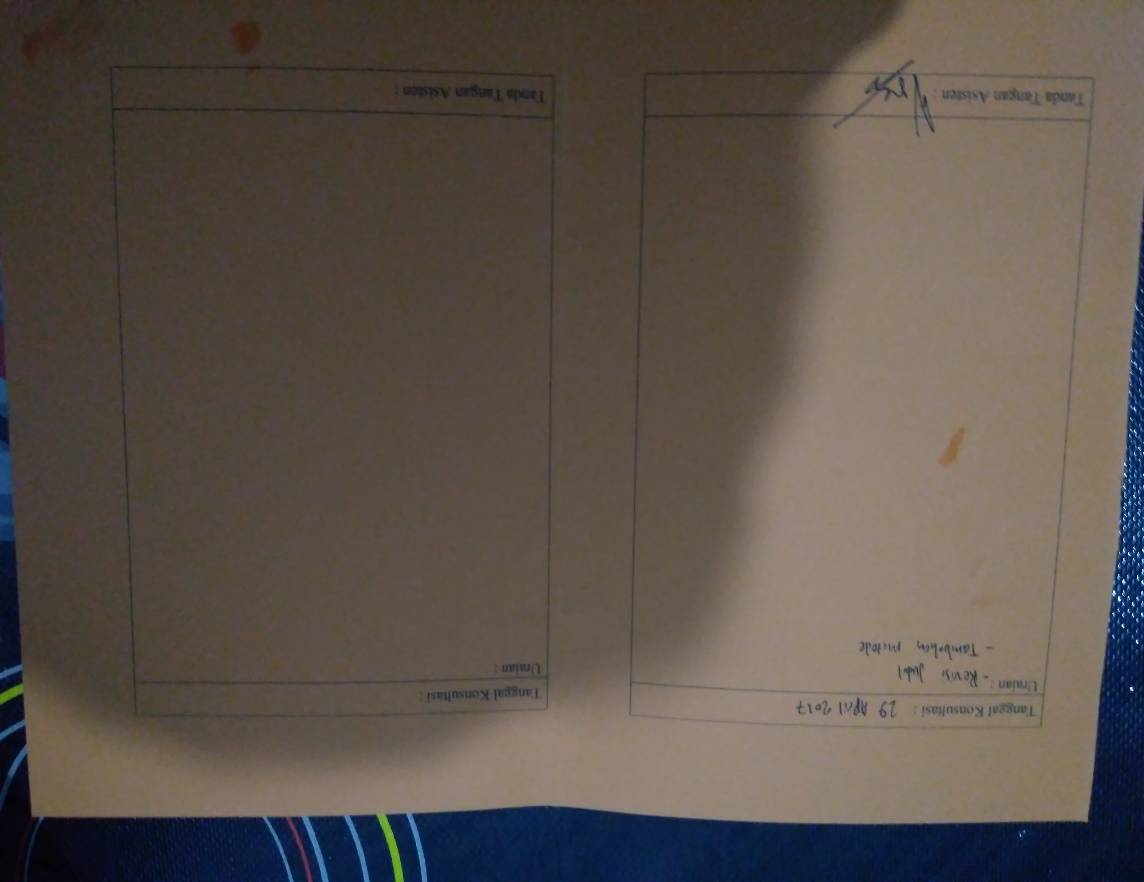
GOAL

mulai.

1. **Kartu Konsul**

****

**Gambar 13. Tampilan depan kartu konsul**

****

**Gambar 14. Tampilan dalam kartu konsul**