

**LAPORAN PRAKTIKUM KECERDASAN BUATAN  
SISTEM PENENTU KEPUTUSAN PEMBELIAN HANDPHONE  
BERDASARKAN SPESIFIKASI**



**Disusun Oleh :**

Tasik Somba Buttuba Lekong	1515015145
Muhammad Meidy Fachreza	1515015155
Agung Layang Donga	1515015206

**Asisten Praktikum :**

Alvian Nur Wahyudhi	Annisa Nur Afiah
1415015066	1415015068

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

## **KATA PENGANTAR**

Puji Syukur kami panjatkan Kepada Tuhan Yang Maha Esa yang telah memberikan Rahmat dan karunia-Nya kepada kami sebagai penyusun laporan tugas akhir praktikum sehingga laporan ini dapat kami susun dan terselesaikan. Laporan ini kami buat sebagai tugas akhir praktikum dengan judul “Sistem Penentu Keputusan Pembelian Handphone Berdasarkan Spesifikasi” dan sebagai salah satu syarat untuk menyelesaikan mata kuliah praktikum Kecerdasan Buatan. Pada laporan tugas akhir ini berisi program yang umumnya di jumpai dalam proses logika dalam menyelesaikan atau melihat hasil akhir (goal) yang ada pada suatu program. Dimana dalam program tersebut diberikan sebuah fakta-fakta dan aturan-aturan yang ada juga diberikan sebuah variable yang nantinya pada goal tersebut menampilkan hasil akhir. Pada kesempatan ini, kami selaku penulis ingin mengucapkan terima kasih khususnya kepada Ayah dan Ibu yang telah memberikan dukungan doa yang tiada hentinya kepada kami.

Serta dalam kesempatan ini, kami selaku penulis juga mengucapkan terima kasih kepada yang terhormat:

1. Ibu Joan Angelina W,M.Kom selaku Dosen mata kuliah Kecerdasan Buatan yang telah mendidik dan membimbing kami sehingga ilmu pengetahuan kami akan mata kuliah Kecerdasan Buatan dapat bertambah.
2. Ibu Masnah Wati, MT selaku Dosen mata kuliah Kecerdasan Buatan yang telah mendidik dan membimbing kami sehingga ilmu pengetahuan kami akan mata kuliah Kecerdasan Buatan dapat bertambah
3. Kak Alvian Nur Wahyudhi dan Kak Anisa Nur Afiyah selaku Asisten Laboratorium Kelompok C2, yang telah memberikan ilmu, motivasi serta waktu dan kesempatan untuk mengajar kami melalui Pretest, Posttest dan praktikum selama kurang lebih 9 pertemuan.
4. Seluruh dukungan teman-teman atas nasehat dan saran-saran yang membangun.

Kami menyadari bahwa dalam pembuatan laporan tugas akhir ini masih banyak kekurangan, untuk itu saran dan kritik yang bersifat membangun sangat kami butuhkan

demi perbaikan laporan tugas akhir ini. Akhir kata, kami berharap semoga laporan tugas akhir ini dapat bermanfaat bagi kami khususnya dan para pembaca umumnya.

Samarinda, 06 Mei 2017

Kelompok 2

## DAFTAR ISI

<b>KATA PENGANTAR .....</b>	<b>i</b>
<b>DAFTAR ISI .....</b>	<b>iii</b>
<b>DAFTAR TABEL .....</b>	<b>v</b>
<b>DAFTAR GAMBAR .....</b>	<b>v</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar belakang.....	1
1.2. Rumusan masalah .....	1
1.3. Batasan masalah.....	1
1.4. Tujuan dan Manfaat .....	2
<b>BAB II LANDASAN TEORI.....</b>	<b>3</b>
2.1 Fungsi-Fungsi Visual Prolog yang Digunakan .....	3
2.1.1 Aritas Jamak .....	3
2.1.2 Konversi Tipe Otomatis.....	3
2.1.3 Unifikasi dan Lacak Balik .....	3
2.1.4 Perulangan dan Rekursi .....	5
2.1.5 List .....	6
2.1.6 Data Objek Sederhana .....	7
2.1.7 Predikat Fail .....	7
2.2 Teori Keilmuan yang Diimplementasikan .....	8

2.2.1 Cara Mengetahui Spesifikasi Handphone.....	8
<b>BAB III METODOLOGI.....</b>	<b>9</b>
3.1 Alur Pembuatan Sistem.....	9
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>10</b>
4.1 Tabel Keputusan.....	10
4.2 Analisis Aplikasi .....	10
<b>BAB V PENUTUP .....</b>	<b>18</b>
5.1 Kesimpulan.....	18
5.2 Saran.....	18
<b>DAFTAR PUSTAKA .....</b>	<b>19</b>
<b>LAMPIRAN .....</b>	<b>20</b>
1 Source Code .....	20
2 Kartu Konsul .....	22

## DAFTAR TABEL

Tabel 1.1 Tabel Spesifikasi RAM .....	8
Tabel 4.1 Tabel Keputusan .....	10

## DAFTAR GAMBAR

Gambar 3.1 Alur Program .....	9
Gambar 4.2.1 Source Code Domains .....	11
Gambar 4.2.2 Source Code Predikat .....	12
Gambar 4.2.3a Source Code Clauses .....	13
Gambar 4.2.3b Source Code Clauses .....	14
Gambar 4.2.3c Source Code Clauses .....	15
Gambar 4.3.3d Source Code Clauses .....	16
Gambar 4.3.4 Source Code Goal .....	17

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Banyak sekali dari berbagai kalangan masyarakat yang ingin memenuhi kebutuhan hidup sehari-harinya, salah satu-nya yaitu Handphone atau Gadget yang semakin tahun semakin berkembang dengan berbagai fitur-fitur dan layanan-layanan konsumen dalam memenuhi kebutuhan dunianya. Sehingga banyak sekali konsumen-konsumen yang ingin memiliki dan membeli sesuai dengan selera masing-masing konsumen.

Dengan program yang kami rancang ini kami membantu masyarakat dalam era globalisasi untuk menentukan apa yang cocok sesuai dengan apa yang diminati dan apa yang menjadi kebutuhan masyarakat sehingga program kami membantu menentukan handphone seperti apa yang cocok dalam memenuhi kebutuhan masyarakat.

### **1.2. Rumusan Masalah**

Dalam era globalisasi baik kalangan ekonomi yang rendah hingga kalangan ekonomi yang tinggi memiliki selera yang cukup sama dalam memenuhi salah satu kebutuhan sehari-hari. Sehingga tiap-tiap konsumen memiliki selera atau ciri-ciri yang hampir sama dalam memilih sebuah gadget atau smartphone dengan keterbatasan dari setiap kalangan ekonomi yang di alami.

1. Fakta-fakta apa yang di minati pada era globalisasi di zaman ini?
2. Spesifikasi apa saja yang terbaik dalam memenuhi kebutuhan konsumen?
3. Apa hasil penentu goal yang sesuai dengan fakta dan spesifikasi yang di inputkan?

### **1.3. Batasan Masalah**

Dalam susunan program ini kami membatasi penentuan sebuah handphone yang telah di berikan fakta-fakta pada sebuah Predicates dengan membatasi kecocokan setiap spesifikasi yang di inputkan pada goal yang ada dengan menentukan spesifikasi yang cukup banyak di minati dikalangan masyarakat zaman sekarang.

### **1.4. Tujuan dan Manfaat**



Dengan dibuat-nya program “Sistem Penentu Keputusan Pembelian Handphone Berdasarkan Spesifikasi” ini dapat membantu setiap konsumen dalam memilih gadget atau smartphone yang diinginkan sesuai dengan selera tiap-tiap konsumen. Dengan memasukan ( menginput ) fakta-fakta atau selera yang kita inginkan dalam sebuah PREDICATES sehingga pada akhir GOAL-nya (output-nya) akan mengeluarkan hasil keputusan pembelian berdasarkan spesifikasi yang kita harapkan.

## BAB II

### LANDASAN TEORI

#### 2.1. Fungsi-Fungsi Visual Prolog

##### 2.1.1. Aritas Jamak

Aritas (*arity*) suatu predikat adalah jumlah argumen yang ada pada predikat tersebut. Visual Prolog memperbolehkan kita mempunyai 2 atau lebih predikat dengan nama yang sama namun dengan aritas yang berbeda. Aritas yang berbeda dari nama predikat yang sama harus dikelompokkan bersama baik pada section **predicates** maupun pada section **clauses**. Perbedaan aritas oleh Visual Prolog akan diperlakukan secara berbeda pula.

##### 2.1.2. Konversi Tipe Otomatis (Automatic Type Conversions)

Ketika visual prolog mencocokkan 2 variabel, keduanya tidak selalu berasal dari domain yang sama. Juga kadang variabel diikat (*bound*) menjadi konstan dari domain lain. Percampuran domain ini diperbolehkan karena Visual Prolog melakukan konversi tipe otomatis dengan syarat konversi bisa terjadi bila:

- Antara **strings** dan **symbols**.
- Antara semua domain **integer** dan juga **real**. Ketika suatu karakter (**char**) dikonversikan ke nilai numeris, angka nilai ASCII dari karakter tersebut yang digunakan.

##### 2.1.3. Unifikasi Dan Lacak Balik

###### 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 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.

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).

### **Lacakbalik (Backtracking)**

Pada waktu menyelesaikan masalah, seringkali seseorang harus menelusuri suatu jalur untuk mendapatkan konklusi yang logis. Jika konklusi ini tidak memberikan jawaban yang dicari, orang tersebut harus memilih jalur yang lain. Perhatikan permainan maze berikut. Untuk mencari jalan keluar dari maze, seseorang harus selalu mencoba jalur sebelah kiri terlebih dahulu pada setiap percabangan hingga menemukan jalan buntu. Ketika menemukan jalan buntu maka orang tersebut harus kembali ke percabangan terakhir (back-up) untuk mencoba lagi (try again) ke jalur kanan dan jika menemukan percabangan lagi maka tetap harus mencoba jalur kiri terlebih dahulu. Jalur kanan hanya akan sekali-sekali dipilih. Dengan begitu orang tersebut akan bisa keluar dari maze, dan memenangkan permainan.

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.

### **Pengendalian Proses Lacakbalik**

Mekanisme lacak balik dapat menghasilkan pencarian yang tidak perlu, akibatnya program menjadi tidak efisien. Misalnya adanya beberapa jawaban yang muncul ketika kita hanya membutuhkan solusi tunggal dari masalah yang diberikan. Pada kasus lain, suatu kebutuhan untuk memaksa Visual Prolog untuk melanjutkan mencari jawaban tambahan walaupun goal tersebut sudah terpenuhi. Untuk kasus-kasus tersebut, kita harus mengontrol proses lacakbalik. Visual Prolog menyediakan 2 alat yang memperbolehkan kita untuk mengendalikan mekanisme lacak balik yaitu predikat fail yang digunakan untuk memaksa lacakbalik dan predikat cut (ditandai dengan !) yang digunakan untuk mencegah lacakbalik.

#### **2.1.4. Perulangan dan Rekursi**

Pengulangan → melakukan aksi yang sama sejumlah kali dengan jumlah yang telah ditentukan atau sampai kondisi berhenti pengulangan tercapai. Pada prolog 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 yang menjadi penentu kondisi berhentinya pengulangan. Aksi inisialisasi tersebut merupakan pemberian nilai awal dari salah satu parameter aturan induk yang merupakan nilai indeks awal pengulangan ke salah satu parameter aturan cabang. Nilai indeks akhir pengulangan diperoleh aturan cabang dengan menggunakan parameter aturan induk yang berisi nilai indeks akhir pengulangan sebagai parameternya. 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, sedangkan aksi yang menjadi penentu kondisi berhentinya pengulangan adalah aksi perbandingan. Operator untuk aksi aritmatika terdiri dari +, -,

\*, /, mod, div, aksi input-ouput terdiri dari fungsi write dan read, sedangkan operator untuk aksi perbandingan terdiri dari operator <, >, <=, >=, =, <>.

## **Rekursi**

Rekursi dalam bahasa apa pun adalah fungsi yang dapat memanggil dirinya sendiri sampai tujuan telah berhasil. Dalam Prolog, rekursif muncul ketika sebuah predikat mengandung tujuan yang mengacu pada dirinya sendiri. Sebagaimana telah kita lihat dalam bab-bab sebelumnya, ketika aturan disebut Prolog membuat query baru dengan variabel baru. Jadi tidak membuat perbedaan cuaca aturan panggilan peraturan lain atau telepon itu sendiri. Dalam Prolog dan dalam bahasa apapun, definisi rekursif selalu memiliki paling sedikit dua bagian. Fakta pertama yang bertindak seperti kondisi berhenti dan aturan yang panggilan itsefl disederhanakan. Pada setiap tingkat fakta pertama diperiksa. Jika kenyataannya adalah benar maka rekursi berakhir. Jika tidak rekursif terus. Aturan rekursif harus tidak pernah menyebut dirinya dengan argumen sames! Jika itu terjadi maka program tidak akan pernah berakhir.

### **2.1.5. List**

Salah satu bentuk data yang populer dalam pemrograman adalah struktur data list dimana dalam Prolog dituliskan dengan menggunakan tanda kurung [] dan setiap elemen dipisahkan oleh tanda koma(.). Sebagai contoh:

[]

list kosong

[a, b, c]

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. Nilai H dan T in i dapat dibandingkan dengan operasi car dan cdr pada pemrograman fungsional. Oleh karena itu, list [a, b, c] dapat dituliskan sebagai:

[a,b,c|[]]

[a,b|[c]]

[a|[b,c]]

### 2.1.6. Data objek 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 diidentifikasi sebagai sebuah *object* bukan *subject* yang nilainya bisa bervariasi, seperti sebuah karakter (*char*), angka (integer atau real) atau sebuah atom (symbol atau string).

### 2.1.7 Predikat fail

Visual Prolog 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. Efek dari fail sama dengan efek memberikan perbandingan  $2=3$  atau subgoal yang tidak mungkin (impossible) lainnya.

## 2.2 Teori yang diImplementasikan

### 2.2.1. Cara Mengetahui Spesifikasi Handphone

Spesifikasi handphone merupakan salah satu ciri-ciri layanan handphone atau fakta yang ada dalam sebuah layanan handphone dimana. Dimana banyak yang membeli sebuah handphone dengan melihat spesifikasi RAM tiap-tiap handphone yang ada. Dimana RAM yang lebih besar kapasitasnya akan semakin cepat prosesnya dan tidak lama loadingnya(lelet).

Spesifikasi RAM Handphone

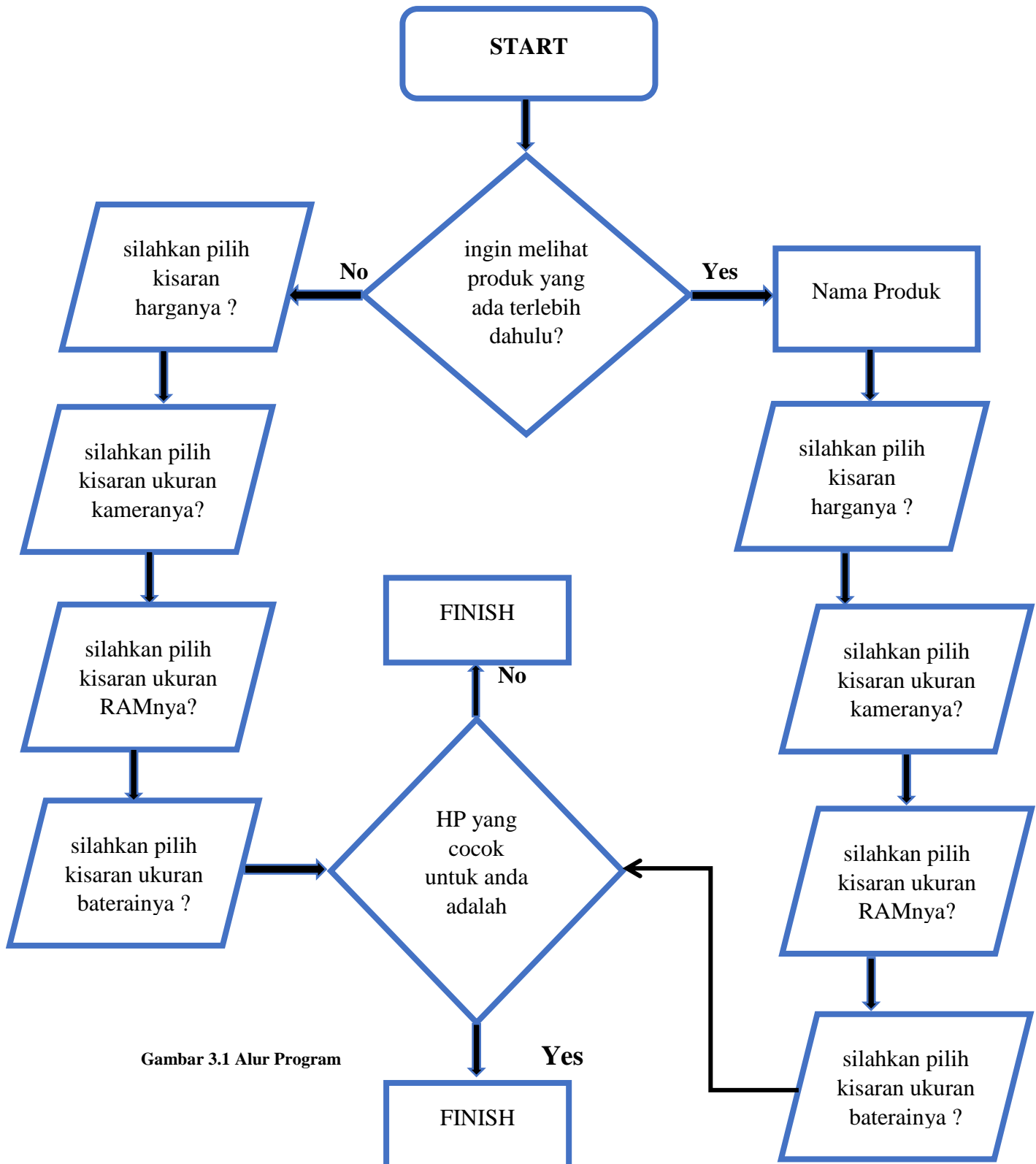
Nama Handphone	Spesifikasi RAM
iphone 5s	1GB - 2GB
	2GB - 3GB
iphone 6 – Nokia N6	3GB - 4GB

Table 1.1. Spesifikasi RAM

## BAB III

### METODOLOGI

#### 3.1. Alur Pembuatan Sistem



Gambar 3.1 Alur Program

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Tabel Keputusan

No	Kisaran Harga								Ukuran Kamera								RAM			Baterai			Nama HP
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	1	2	3	
1.	y								y								y			y			iphone 5s
2.	y									y									y			y	iphone 6
3.	y									y									y			y	iphone 7
4.	y									y									y			y	samsung S5
5.	y									y									y			y	samsung S6
6.	y									y									y			y	samsung S7
7.	y									y									y			y	lenovo A600
8.	y									y									y			y	lenovo K2
9.	y									y									y			y	lenovo P2
10.	y									y									y	y			xiaomi redmi note
11.	y									y									y			y	xiaomi mi4
12.	y									y									y			y	xiaomi mi5
13.	y									y									y			y	Vivo V
14.	y									y									y			y	Vivo Y2
15.	y									y									y			y	Vivo Y5
16.	y									y									y			y	Nokia 5500
17.	y									y									y			y	Nokia N
18.	y									y									y			y	Nokia N6

Tabel 4.1.Tabel Keputusan

#### 4.2 Analisis Program



#### 4.2.1. Domains

Dalam program kami juga memakai fungsi domains yang dalam hal ini fungsi visual prolog ini dapat mempunyai 2 manfaat utama, yaitu pertama, kita dapat memberikan nama yang berarti untuk domain, walaupun secara internal domain tersebut sama tipenya dengan domain yang telah ada; yang kedua, kita dapat mendeklarasi domain khusus yang digunakan untuk mendeklarasikan struktur data yang tidak didefinisikan oleh standar domain. Dengan mendeklarasikan domain juga dapat mencegah kesalahan logika pada program. Dalam program kami memakai Domains untuk memberikan tipe yang berbeda pada setiap simbol yang ada dalam program kami. Agar suatu predikat Visual Prolog dapat menerima suatu argumen yang memberikan informasi dengan tipe yang berbeda maka functor tersebut harus dideklarasikan dalam program kami. Contohnya dalam program yang kami buat ini tipe yang kami deklarasikan dalam fungsi Domains adalah :

```
domains
kriteria = kriteria(harga,kamera,ram,baterai)
angka = integer
nama= string
harga= integer
kamera= integer
baterai= integer
ram= integer
list = string*
```

Gambar 4.2.1 Source Code Domains

Dimana nama simbol angka,harga,kamera,baterai dan ram di deklarasikan sebagai tipe integer dan nama serta list di deklarasikan sebagai tipe String, juga nama simbol Kriteria di deklarasikan dengan ke-4 nama simbol yang telah diberikan tipe yaitu harga,kamera,ram, dan baterai. Pada simbol list Tanda asterik (\*) berarti domain tersebut merupakan sebuah *list*.

#### 4.2.2. Predicates

Pada program ini kami juga memakai salah satu fungsi visual yaitu Predicates yang dapat mendeklarasikan suatu nama simbol-simbol yang dideklarasikan dengan memberikan tipe atau variable pada simbol-simbol yang nantinya akan di pakai dalam fungsi Predicates pada sebuah program kami. Dimana pendeklarasian predicate dimulai dengan nama predikat diikuti tanda kurung buka, kemudian diikuti satu atau lebih argumen pada predicates (setiap argumen dipisah dengan tanda koma) dimana dalam setiap argumen yang ada di deklarasikan dengan tipe yang sudah di tentukan pada fungsi Domains, kemudian ditutup dengan tanda kurung tutup. Contohnya dalam program kami hanya terdapat 2 tipe argumen yang berbeda tetapi dapat dipakai berulang-ulang kali dalam satu simbol yang dideklarasikan pada predicate, misalnya dalam program kami :

```
predicates
nondeterm start
nondeterm run(angka)
nondeterm kriteria_produk(nama,kriteria)
nondeterm pilih(string)
cetak_list(list)
nama_produk(list)
pilihan(harga,kamera,ram,baterai)
```

**Gambar 4.2.2 Source Code Predicates**

Dimana dalam predicate program kami yaitu nama predicate start,run(angka),Kriteria\_produk(nama,kriteria),pilih(string) diberikan pendeklarasian nondeterm, dimana nondeterm berfungsi untuk mengatasi apabila pada saat run terdapat lebih dari satu data yang sama, sehingga tetap dapat muncul hasilnya.

#### **4.2.3. Clauses**

Dalam program ini kami memakai fungsi clauses yang merupakan section yang paling penting pada program Visual Prolog. Pada section inilah kita meletakkan fakta dan aturan. Ketika mencari jawaban, Visual Prolog akan mencari dari bagian paling atas dari section clauses, melihat setiap fakta dan aturan untuk mendapat jawaban benar,

hingga ke bagian paling bawah dari section ini. Contohnya pada program kami dibawah ini digunakan untuk membuat tampilan awal yaitu start, read dan write dibawah berfungsi untuk input data. Serta terdapat predikat fail yang berfungsi untuk mencegah lacakbalik. Dan fail digunakan untuk memaksa prolog ketika terdapat kegagalan sehingga memicu terjadinya lacakbalik.

```

clauses
start:-write("ingin melihat produk yang ada terlebih dahulu? <y/n>"),
      readln(ANS),
      pilih(ANS).

pilih("y"):-run(1).
pilih("Y"):-run(1).
pilih("n"):-run(2).
pilih("N"):-run(2).

run(1):-nl,nama_produk(List),
      cetak_list(List),nl,run(2).

run(2):-nl,
      pilihan(Harga,Kamera,Ram,Baterai),
      kriteria_produk(A,kriteria(Harga,Kamera,Ram,Baterai)),
      write("HP yang cocok untuk anda adalah ",A),nl,
      nl.

cetak_list([]).

cetak_list([H|T]):-
      write(H),nl,
      cetak_list(T).

```

**Gambar 4.2.3a Source Code Clauses**

pada gambar diatas bagian Run 1 : mengarahkan pada bagian nama\_produk yang akan di list-kan atau ketika program dirunning lalu diketikkan pilihan y maka nama produk akan ditampilkan. Lalu pada Run 2 : adalah fakta yang di tetapkan ketika selesai kita memilih spesifikasi yang cocok dimana pada proses itulah Unifikasi lacakbalik akan di proses dimana dia mencocokkan dari pilihan spesifikasi yang telah diinputkan Handphone apa yang cocok sesuai spesifikasi lalu dia menginputkan nama\_produk

ketika ada maka keluar hasil yes tetapi jika tidak ada nama\_produk yang sesuai dengan spesifikasi yang kita masukan maka hasil akhirnya tidak ditemukan atau = No.

```
pilihan(Harga,Kamera,Ram,Baterai):-write("Kisaran Harga"),nl,
    write("1. Rp.1.000.000 - Rp.2.000.000 "),nl,
    write("2. Rp.2.000.000 - Rp.3.000.000 "),nl,
    write("3. Rp.3.000.000 - Rp.4.000.000 "),nl,
    write("4. Rp.4.000.000 - Rp.5.000.000 "),nl,
    write("5. Rp.5.000.000 - Rp.6.000.000 "),nl,
    write("6. Rp.6.000.000 - Rp.7.000.000 "),nl,
    write("7. Rp.7.000.000 - Rp.9.000.000 "),nl,
    write("8. Rp.9.000.000 - Rp.11.000.000 "),nl,
    write("*silahkan pilih kisaran harganya : "),nl,
    readint(Harga),
    write("ukuran kamera"),nl,
    write("1. 1MP - 3MP"),nl,
    write("2. 3MP - 4MP"),nl,
    write("3. 5MP - 7MP"),nl,
    write("4. 7MP - 9MP"),nl,
    write("5. 9MP - 11MP"),nl,
    write("6. 11MP - 13MP"),nl,
    write("7. 13MP - 14MP"),nl,
    write("8. 14MP - 16MP"),nl,
    write("*silahkan pilih kisaran ukuran kameranya : "),nl,
    readint(Kamera),
    write("ukuran RAM"),nl,
    write("1. 1GB - 2GB"),nl,
    write("2. 2GB - 3GB"),nl,
    write("3. 3GB - 4GB"),nl,
    write("*silahkan pilih kisaran ukuran RAMnya : "),nl,
    readint(Ram),!,
    write("ukuran Baterai"),nl,
    write("1. 1000 - 2000"),nl,
    write("2. 2000 - 3000"),nl,
    write("3. 3000 - 4200"),nl,
    write("*silahkan pilih kisaran ukuran baterainya : "),nl,
    readint(Baterai).
```

**Gambar 4.2.3b Source Code Clauses**

Pada gambar di atas adalah source code tampilan Run dalam memilih spesifikasi yang di inginkan para konsumen dengan berbagai pilihan yang ada di tampilkan di akan disuruh menginputkan pilihan spesifikasi yang diinginkan pembeli.

```

nama_produk([
    "iphone 5s",
    "iphone 6",
    "iphone 7",
    "samsung S5",
    "samsung S6",
    "samsung S7",
    "lenovo A600",
    "lenovo K2",
    "lenovo P2",
    "xiaomi redmi note",
    "xiaomi mi4",
    "xiaomi mi5",
    "vivo V",
    "vivo Y2",
    "vivo Y5",
    "nokia 5500",
    "nokia N",
    "nokia N6"] ).

```

**Gambar 4.2.3c Source Code Clauses**

Pada gambar di atas adalah predicate nama\_produk dimana akan ditampilkan ketika pada awal Start dalam pilihan awal kita memilih Y untuk menampilkan nama produk, sehingga source code diatas akan di proses dalam sebuah program.

```

kriteria_produk("iphone 5s", kriteria(1,1,1,1) ).
kriteria_produk("iphone 6", kriteria(1,2,3,3) ).
kriteria_produk("iphone 7", kriteria(1,2,3,3) ).
kriteria_produk("samsung S5", kriteria(1,2,3,3) ).
kriteria_produk("samsung S6", kriteria(1,2,3,3) ).
kriteria_produk("samsung S7", kriteria(1,2,3,3) ).
kriteria_produk("lenovo A600", kriteria(1,2,3,3) ).
kriteria_produk("lenovo K2", kriteria(1,2,3,3) ).
kriteria_produk("lenovo P2", kriteria(1,2,3,3) ).
kriteria_produk("xiaomi redmi note 3", kriteria(1,2,3,1) ).
kriteria_produk("xiaomi mi4", kriteria(1,2,3,3) ).
kriteria_produk("xiaomi mi5", kriteria(1,2,3,3) ).
kriteria_produk("vivo V3", kriteria(1,2,3,3) ).
kriteria_produk("vivo Y2", kriteria(1,2,3,3) ).
kriteria_produk("vivo Y5", kriteria(1,2,3,3) ).
kriteria_produk("nokia 5500", kriteria(1,2,3,3) ).
kriteria_produk("nokia N3", kriteria(1,2,3,3) ).
kriteria_produk("nokia N6", kriteria(1,2,3,3) ).

```

**Gambar 4.2.3d Source Code Clauses**

pada Gambar di atas ada fakta dan aturan yang dalam sebuah program kami dimana setelah proses penginputan 4 pilihan yang diberikan akan ada hasil atau nilai akhir proses pilihannya, nah ketika selesai menginput pilihan pada source di atas yang akan di lacakbalik apakah pilihan yang dimasukan ada atau tidak nama\_produk-nya yang sesuai dengan Kriteria-produk yang telah kami tetapkan dalam program kami.

#### **4.2.4. Goal**

Fungsi goal ini adalah fungsi ketetapan yang kami pakai dalam program kami dimana pada visual prolog fungsi Goal adalah nilai akhir atau hasil akhir dari pertanyaan atau fakta-fakta apa aja yang ada pada sebuah program. Dimana sebelum mendeklarasikan Goal yang ada pada visual prolog harus di deklarasikan terlebih dahulu pada fungsi Predicates dan fungsi Clauses. Sehingga dalam program yang kami buat ini hanya memberikan simbol yaitu :

GOAL

start.

dimana simbol run telah di deklarasikan pada Predicates dan Clauses :

```
goal
start.
```

#### **4.2.4. Source Code Goal**

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Sistem Visual Prolog yang kami buat adalah mengenai Sistem Penentu Keputusan Pembelian Handphone berdasarkan Spesifikasi. Dimana pembeli atau konsumen dapat melakukan banyak pilihan sesuai dengan selera atau kebutuhan-nya dalam memenuhi standar spesifikasi yang baik menurut standar kebutuhan tiap-tiap konsumen dalam membantu kebutuhan tiap konsumen.

#### **5.2. Saran**

Kami berharap setiap dalam program yang kami buat bisa kalian semua pahami termasuk dalam Sistem Penentu Keputusan Pembelian Handphone berdasarkan Spesifikasi. Dimana jelas sekali kami memberikan sebuah fakta atau aturan yang ada dalam spesifikasi handphone yang ada dimasa kini.

Bagi Kakak Asisten Laboratorium agar dapat memahami setiap kemampuan mahasiswa dalam memahami materi yang disampaikan, karena daya tangkap setiap mahasiswa tidaklah sama.

Bagi Mahasiswa untuk lebih sering berlatih mengerjakan soal, baik latihan soal yang telah disediakan dimodul maupun mencari latihan soal yang diperoleh dari internet. Mahasiswa pun diharapkan agar lebih sering bertanya kepada teman maupun kakak tingkat mengenai hal-hal atau soal yang belum dipahami.

## **DAFTAR PUSAKA**

<http://web.ipb.ac.id/~julio/webaku/isi/kom204/notes/4.pdf> : 04 Mei 2017

<http://wiwidrainshinbi.blogspot.co.id/2010/10/rekursif-perulangan-dalam-prolog.html> :  
04 Mei 2017

<http://vinnick86.blogspot.co.id/2007/11/unifikasi-dan-lacakbalik.html> : 05 Mei 2017

<http://pikikuka.blogspot.com/2015/12/section-section-yang-ada-di-visual.html> : 05 Mei  
2017



## LAMPIRAN

### 1. Source Code

```
domains
angka = integer
nama= string
harga= integer
kamera= integer
baterai= integer
ram= integer
list = string*
predicates
nondeterm start
nondeterm nama_produk(list)
nondeterm run(angka)
nondeterm pilihan(harga,kamera,ram,baterai)
nondeterm kriteria_produk(nama,harga,kamera,ram,baterai)
cetak_list(list)
nondeterm pilih(string)
clauses
start:-write("ingin melihat produk yang ada terlebih dahulu? <y/n>"),
        readln(ANS),
        pilih(ANS).
pilih("y"):-run(1).
pilih("Y"):-run(1).
pilih("n"):-run(2).
pilih("N"):-run(2).
```

```

run(1):-nl,nama_produk(List),
        cetak_list(List),nl,run(2).

run(2):-nl,
pilihan(Harga,Kamera,Ram,Baterai),
kriteria_produk(A,Harga,Kamera,Ram,Baterai),
write("HP yang cocok untuk anda adalah ",A),nl,
nl.


cetak_list([]).

cetak_list([H|T]):-
write(H),nl,
cetak_list(T).

pilihan(Harga,Kamera,Ram,Baterai):-write("Kisaran Harga"),nl,
                                     write("1. Rp.1.000.000 - Rp.2.000.000 "),nl,
                                     write("2. Rp.2.000.000 - Rp.3.000.000 "),nl,
                                     write("3. Rp.3.000.000 - Rp.4.000.000 "),nl,
                                     write("4. Rp.4.000.000 - Rp.5.000.000 "),nl,
                                     write("5. Rp.5.000.000 - Rp.6.000.000 "),nl,
                                     write("6. Rp.6.000.000 - Rp.7.000.000 "),nl,
                                     write("7. Rp.7.000.000 - Rp.9.000.000 "),nl,
                                     write("8. Rp.9.000.000 - Rp.11.000.000 "),nl,
                                     write("*silahkan pilih kisaran harganya : "),nl,
                                     readint(Harga),

```

## 2. Kartu Konsul

<p><b>PERATURAN</b></p> <ol style="list-style-type: none"><li>1. Praktikan wajib konsultasi Programing Project dengan Asisten Yang telah ditentukan.</li><li>2. Praktikan yang tidak konsultasi <b>DILARANG</b> mengikuti ujian.</li><li>3. Konsultasi I, membahas tentang konsep PP yang akan dibuat.</li><li>4. Konsultasi II, membahas tentang hal yang berkaitan dengan PP yang sedang dikerjakan.</li></ol> <p><b>NB:</b> Kartu Konsultasi ini di print menggunakan kertas Cover berwarna sesuai dengan warna pada cover modul praktikum</p>	<p><b>SISTEM PENENTU KEPUTUSAN PEMBELIAN HANDPHONE BERDASARKAN SPESIFIKASI KECERDASAN BUATAN</b></p> <p><b>KELOMPOK : C2/02/2015</b></p> <table border="0"><tr><td><b>TASIK SOMBA BUTTUBA LEKONG</b></td><td><b>1515015145</b></td></tr><tr><td><b>MUHAMMAD MEIDY FACHREZA</b></td><td><b>1515015155</b></td></tr><tr><td><b>AGUNG LAYANG DONGA</b></td><td><b>1515015206</b></td></tr></table> <div style="text-align: center;"></div> <p><b>LABORATORIUM BAHASA PEMROGRAMAN FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI</b></p> <p><b>2017</b></p>	<b>TASIK SOMBA BUTTUBA LEKONG</b>	<b>1515015145</b>	<b>MUHAMMAD MEIDY FACHREZA</b>	<b>1515015155</b>	<b>AGUNG LAYANG DONGA</b>	<b>1515015206</b>
<b>TASIK SOMBA BUTTUBA LEKONG</b>	<b>1515015145</b>						
<b>MUHAMMAD MEIDY FACHREZA</b>	<b>1515015155</b>						
<b>AGUNG LAYANG DONGA</b>	<b>1515015206</b>						

Tanggal Konsultasi : 29 APRIL 2017

Uraian :

- REVISI PROGRAM
- FIX KONSEP
- MODUL 4 & 5 SUDAH
- MODUL 6 & 7 BELUM

Tanda Tangan Asisten :

Tanggal Konsultasi : 03 MEI 2017

Uraian :

- REVISI PROGRAM
- MODUL 6 & 7 SUDAH.

Tanda Tangan Asisten :