**LAPORAN PRAKTIKUM KECERDASAN BUATAN**

**SISTEM PENDUKUNG KEPUTUSAN POSISI PARKIR MOTOR MAHASISWA DI FKTI**

****

**Disusun Oleh :**

Ingan Zakaria 1515015173

Adli Dzil Ikram 1515015164

Melky Sinduwinata Yohanes 1515015157

**Asisten Praktikum :**

|  |  |  |
| --- | --- | --- |
| Niken Novirasari  1415015064 |  | Asdar Zulkiawan  1415015052 |

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI DAN KOMUNIKASI**

**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**

**UNIVERSITAS MULAWARMAN**

**2017**

**KATA PENGANTAR**

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada kami, sehingga kami dapat menyelesaikan Laporan Praktikum Kecerdasan Buatan di Universitas Mulawarman Fakultas Ilmu Komputer dan Teknologi Informasi Prodi Teknik Infomatika, mengenai Sistem Pendukung Keputusan Posisi Parkir Motor Mahasiswa di FKTI.

Penyusunan Laporan ini sebagai salah satu persyaratan project akhir praktikum matakuliah Kecerdasan Buatan. Laporan ini kami susun dengan semaksimal mungkin dan tidak lepas dari bantuan berbagai pihak sehingga dapat memperlancar pembuatan Laporan ini. Untuk itu kami menyampaikan banyak terima kasih kepada semua pihak yang telah berkontribusi dalam pembuatan Laporan ini.

Terlepas dari semua itu, kami menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasanya. Oleh karena itu, dengan rendah hati kami membutuhkan saran dan kritik yang sifatnya membangun dari semua pembaca agar lebih baik lagi ke depannya.

Dalam kesempatan ini, kami mengucapkan Terima Kasih sebesar-besarnya kepada Ibu Masna Wati,S.Si,MT, selaku dosen pembimbing mata kuliah Kecerdasan Buatan dan kepada Niken Novirasari dan Asdar Zulkiawan selaku asisten laboratorium, yang telah membimbing kami dalam Praktikum Matakuliah Kecerdasan Buatan.

Samarinda 04 Mei 2107

Penulis

**DAFTAR ISI**

**Kata Pengantar**  **i**

**Daftar Isi** **ii**

**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**

2.1 Fungsi-Fungsi Visual Prolog yang Digunakan **3**

2.1.1 Unifikasi dan Lacakbalik **3**

2.1.1.1 Unifikasi **3**

2.1.1.2 Lacakbalik **3**

2.1.2 Data Object Jamak **4**

2.1.3 Perulangan dan Rekursi **4**

2.1.3.1 Lacak Balik **5**

2.1.3.2 Rekursi **5**

2.1.4 List **5**

2.1.5 Section Facts **5**

2.1.5.1 Deklarasi Section Facts **6**

2.1.5.2 Menambah fakta pada saat run time **6**

2.1.5.3 Menghapus fakta pada saat run time **6**

2.1.5.4 Menyimpan database fakta-fakta pada saat run time **6**

2.2 Teori Keilmuan.... **6**

2.2.1 Pilih tujuan ruangan **6**

2.2.2 Keadaan cuaca **6**

**BAB III Metodologi** **7**

3.1 Alur Pembuatan Sistem **7**

**BAB IV Hasil dan Pembahasan** **9**

4.1 Tabel Kebenaran **9**

4.2 Analisis Aplikasi **9**

**BAB V Penutup** **13**

5.1 Kesimpulan **13**

5.2 Saran **13**

**Daftar Pustaka**  **14**

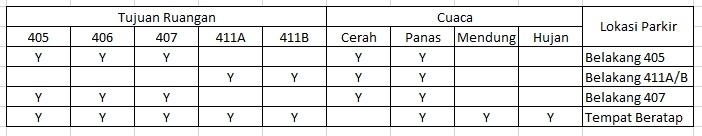
**Lampiran**  **15**

1 Source Code **15**

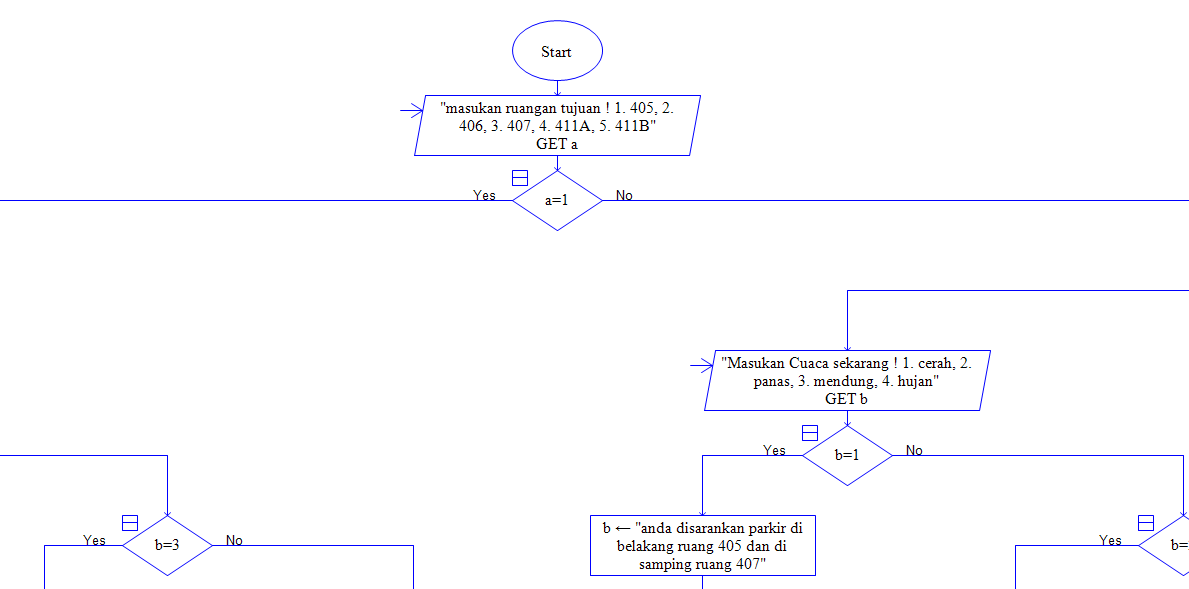
2 Kartu Konsul 23

**Daftar Tabel**

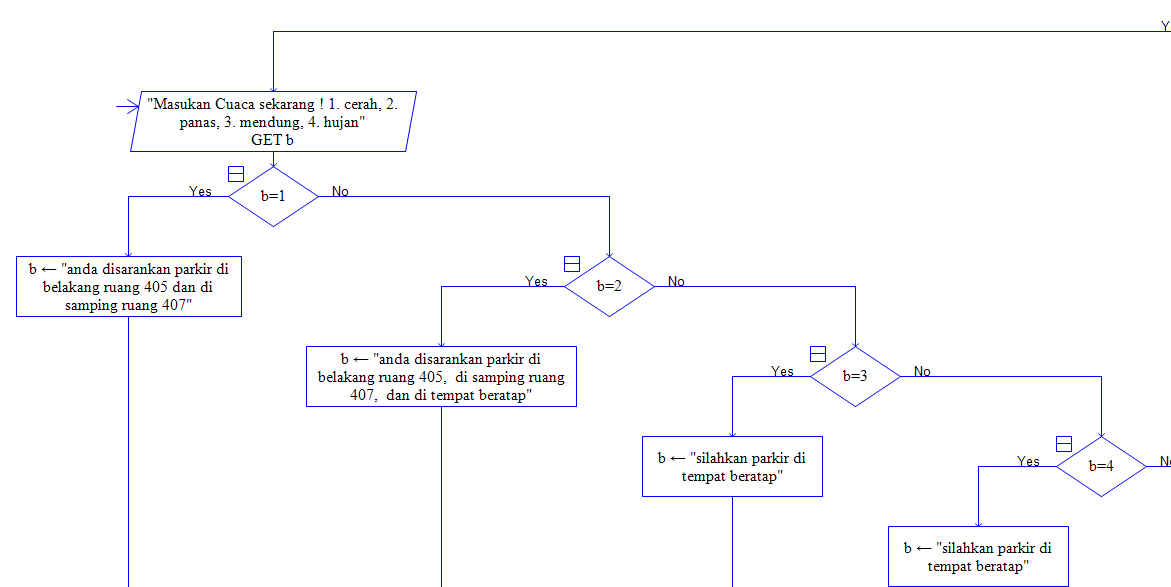
**-Tabel Keputusan**

****

**Daftar Gambar**

**-Gambar Flowchart**

**Gambar.1 Flowchart**

****

**Gambar.2 Flowchart**

**BAB I**

**Pendahuluan**

* 1. **Latar belakang**

Parkiran di FKTI adalah suatu hal yang sangat penting bagi mahasiswa, karena ketika mahasiswa sampai di FKTI terkadang mahasiswa jadi bingung menentukan dimana dia harus parkir yang tepat, oleh karena itu kami mebuat Program SPK Posisi Parkir Motor Mahasiswa di FKTI agar dapat memberikan saran kepada mahasiswa dimana seharusnya mahasiswa harus parkir yang tepat dan cocok dengan keadaan, contohnya mahasiswa datang kekampus FKTI dan memilih tujuan ruangannya adalah 405 dengan keadaan cuaca yang cerah maka program SPK ini akan memberikan saran kepada mahasiswa dimana mahasiswa harus parkir, misalkan harus parkir diarea belakang ruangan 405 dan diarea samping ruangan 407.

Dan setiap ruangan yang dipilih oleh mahasiswa dan keadaan cuaca pada hari itu program SPK ini akan memberikan saran kepada mahasiswa dimana seharusnya mahasiswa itu harus parkir yang tepat agar tidak jauh dari ruangan yang dipakai dan parkir nya juga sesuai dengan keadaan cuacanya.

* 1. **Rumusan masalah**

1. Program ini membantu mahasiswa untuk memutuskan area parkir
2. Program ini digunakan sesuai dengan keadaan dan tempat yang dipilih oleh mahasiswa.
   1. **Batasan masalah**
3. Program SPK ini hanya ditujukan kepada mahasiswa di FKTI
4. Program SPK ini hanya menyarankan tempat parkir kepada mahasiswa di FKTI
5. Program SPK ini hanya memberikan saran tempat parkir yang tepat dengan keadaan cuaca yang ada.
6. Program ini hanya memilih tujuan ruangan dan keadaan cuaca
   1. **Tujuan dan Manfaat**
7. **Tujuan :**
8. Agar mahasiswa dapat menerapkan konsep dan materi kecerdasan buatan yang diberikan pada saat praktikum dilaksanakan.
9. Agar mahasiswa dapat membuat program yang seefektif mungkin dan dapat digunakan oleh mahasiswa sendiri seperti program SPK ini.
10. Program dan laporan ini adalah salah satu persyaratn untuk menyelesaikan tugas akhir praktikum matakuliah keserdasan buatan.
11. **Mamfaat :**
12. Dapat memperluas kemampuan mahasiswa untuk membuat program seperti SPK parkir motor untuk mahasiswa di FKTI.
13. Dapat menghasilkan solusi serta hasilnya dapat digunakan.
14. Dapat meningkatkan efektifitas terhadap keputusan yang diambil mahasiswa

**BAB II**

**Landasan Teori**

**2.1 Fungsi-Fungsi Visual Prolog yang Digunakan**

**2.1.1 Unifikasi**

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 variabel.

**2.1.1 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. 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.

* **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.

* **Predikat cut**

Visual Prolog memiliki *cut* yang 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*).

**2.1.2 Data Object 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.

**2.1.3 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).

**2.1.3.1 Lacak balik**

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.

**2.1.3.2 Rekursi**

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

**2.1.4 List**

Pada Prolog, yang dimaksud dengan list adalah sebuah object yang didalamnya mengandung sejumlah object yang lain (jumlahnya dapat berubah-ubah). List dalam bahasa pemrograman lain bisa disamakan dengan tipe data pointer (C dan Pascal). Berikut ini cara penulisan list pada Prolog.

[ 1, 2, 3 ] /\* list yang mengandung integer 1, 2 dan 3 \*/  
[ kucing, anjing, tikus ] /\* list yang terdiri dari 3 buah symbol \*/  
[ “Syarif Musadek”, “Yusida Andriani”, “Diana Putri” ]  
/\* list yang terdiri dari 3 buah string \*/

**2.1.5 Section Facts**

*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 runtime*).

Kita dapat mendeklarasikan sebuah predikat pada *section facts* dan predikat tersebut dapat digunakan sama halnya seperti kalau dideklarasikan pada *section predicates*.  
Visual Prolog menyediakan beberapa predikat *built-in* untuk menangani hal yang  
berkaitan dengan penggunaan *section facts*, antara lain:

• assert, asserta dan assertz untuk menambah fakta baru pada *section facts*.

• retract dan retractall untuk menghapus fakta yang ada.

• consult untuk membaca fakta dari sebuah file dan menyertakan fakta tersebut ke dalamfaktainternal.

• save menyimpan isi fakta internal ke dalam sebuah file.

**2.1.5.1 Deklarasi Section Facts**

Kata kunci facts atau bisa juga database menandai permulaan sederetan deklarasi dari predikat yang ada pada section facts.

Kita dapat menambahkan fakta-fakta (bukan rule) pada suatu section facts dari keyboard pada saat run time dengan menggunakan asserta dan  
assertz atau memanggil predikat consult untuk mengambil fakta tambahan dari sebuah file.

**2.1.5.2 Menambah fakta pada saat run time**

Pada saat run time, fakta-fakta dapat ditambah ke section facts dengan menggunakan predikat assert, asserta dan assertz atau me-load sebuah file yang berisikan fakta menggunakan predikat consult.

**2.1.5.3 Menghapus fakta pada saat run time**

Predikat retract mengunifikasi suatu fakta dan menghapus fakta tersebut dari sectionfacts.

**2.1.5.4 Menyimpan database fakta-fakta pada saat *run time***

Predikat save berfungsi untuk menyimpan fakta-fakta yang ada pada section facts ke dalam sebuah file.

**2.2 Teori Keilmuan**

Teori yang di implementasikan SPK ini memiliki kemudahan bagi penggunanya. Kemampuan dari SPK ini sangat bagus untuk memudahkan kondisi parkir dari teori-teori yang dikumpulkan.

**2.2.1 Pilih tujuan ruangan**

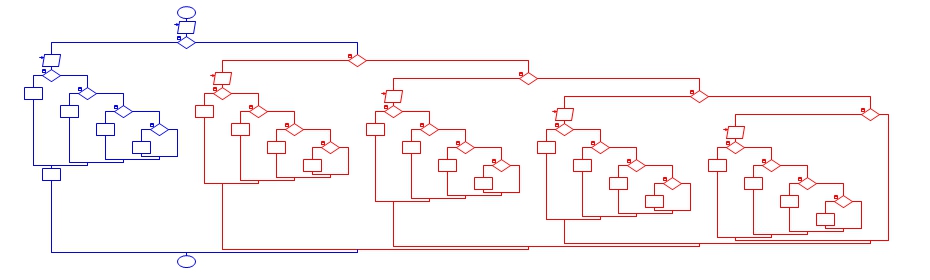
Jadi mahasiswa akan memilih ruangan yang akan dimasuki dan program SPK menyediakan ruangan 405,406,407,411A,411B.

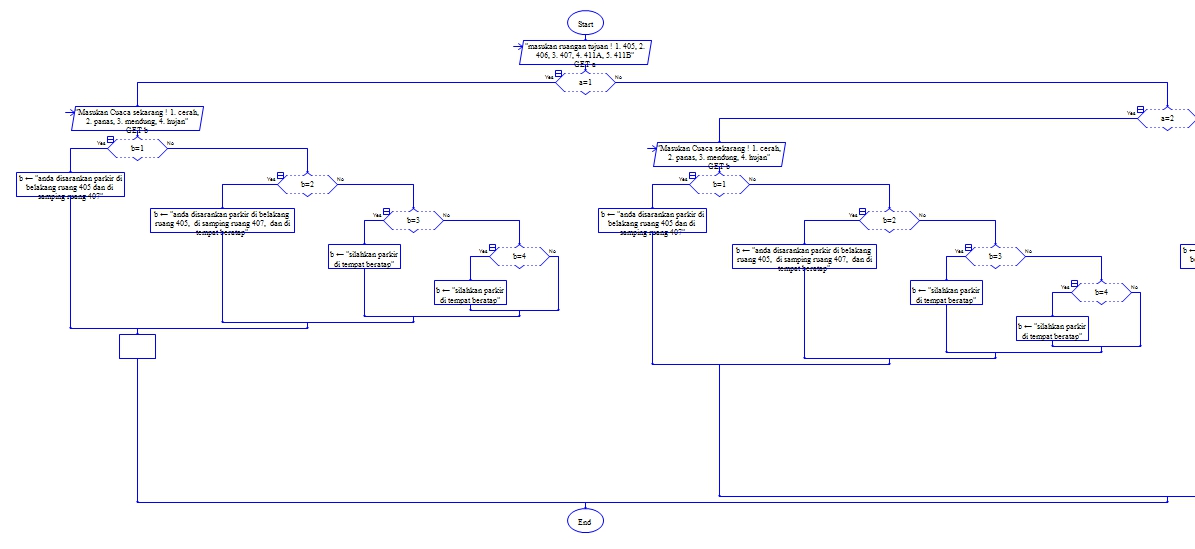
**2.2.2 Keadaan cuaca**

Disini akan menentukan cuaca agar program SPK dapat memberikan saran tempat parkir yang tepat dengan keadaan cuaca. Keadaan cuaca nya ada cerah,terang,mendung,hujan.

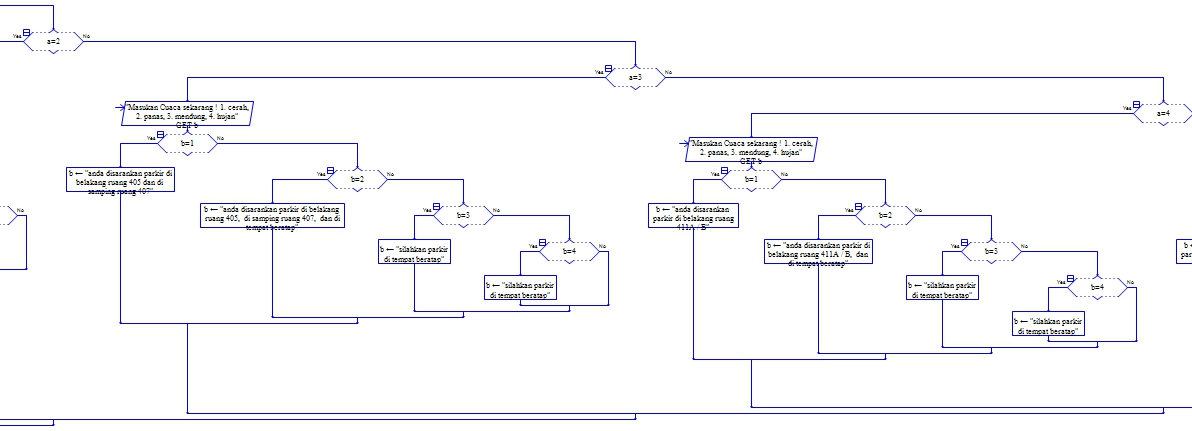
**Bab III**

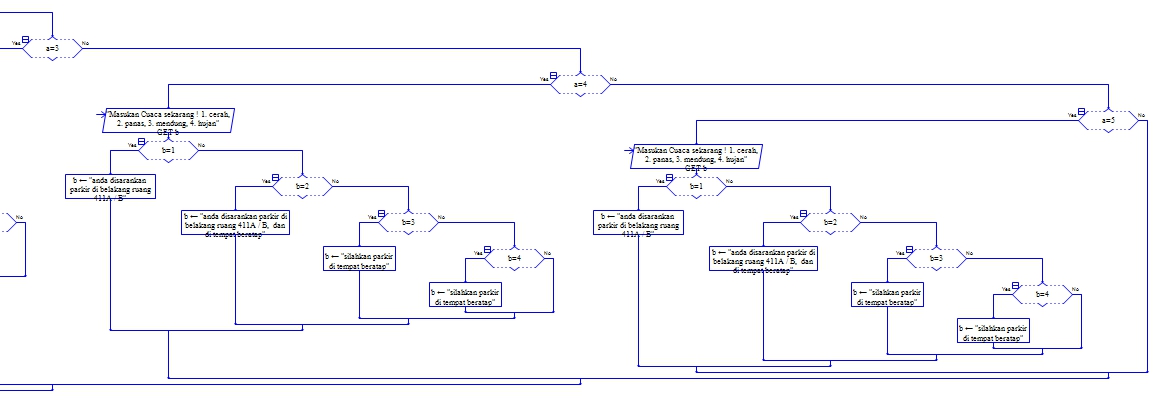
**Metodologi**

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

**Gambar.1**

**Gambar.2**

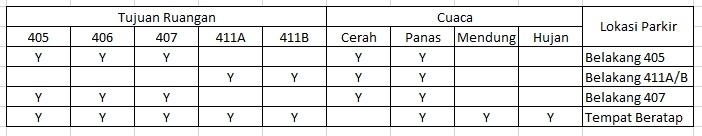
****

**Gambar.3**

**Gambar.4**

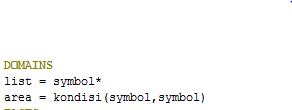
**Bab IV**

**Hasil dan Pembahasan**

* 1. **Tabel Keputusan**

**Gambar.5**

* 1. **Analisis Aplikasi**
* **Domains**

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

**Gambar.6 Domains**

* **Facts**

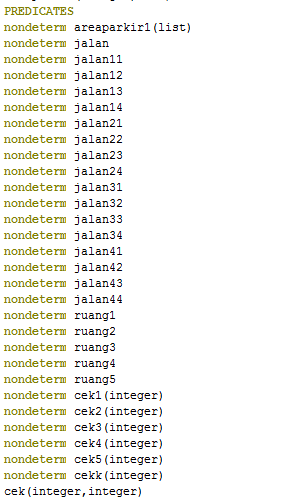
Terdiri dari fakta-fakta yang dapat ditambah dan dihapus secara langsung saat program sedang berjalan.



**Gambar.7**

* **Predicates**

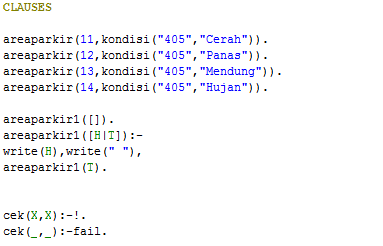
Pada predikat digunakan nondeterm yang fungsinya adalah untuk mengatasi apabila pada saat run terdapat lebih dari satu data yang sama, sehingga tetap dapat muncul hasilnya. Dan cek merupakan variabel baru.



**Gambar.8 Predicates pada program**

* **Clauses**

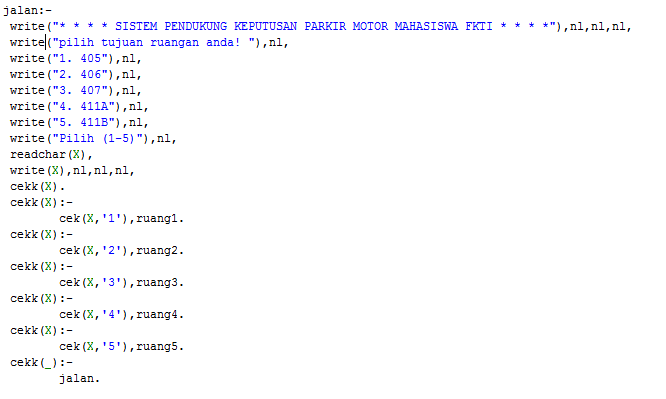
Clauses berisi fakta-fakta dan aturan suatu program.



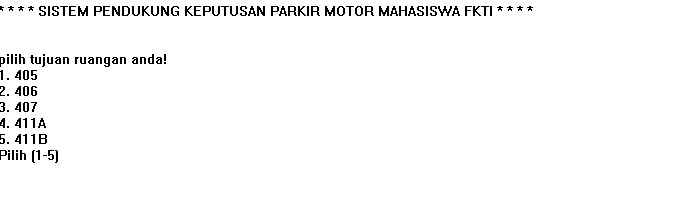
**Gambar.9 Clauses pada program**

* **Unifikasi**

Unifikasi adalah suatu pencocokan suatu klausa



**Gambar.10 Unifikasi pada program**

****

**Gambar.11 Tampilan awal program**

****

**Gambar.12 Tampilan kedua program**

**Gambar.13 Tampilan hasil keputusan**

**BAB V**

**Penutup**

* 1. **Kesimpulan**
* Adanya program ini yang sudah kami buat akan membantu mahasiswa memilih tempat parkir yang tepat dengan keadaan cuaca yang ada dengan menggunakan visual prolog.
* Solusi yang diberikan disesuaikan dengan kondisi dan situasi lingkungan sekitar, karena solusi yang diberikan merupakan solusi secara umum.
* Program ini cukup membantu dan sangat efisien, karena tidak perlu membuang waktu yang banyak untuk memberikan keputusan parkir terbaik untuk mahasiswa.
  1. **Saran**
* Program ini Memberikan Keputusan Kepada Mahasiswa FKTI untuk memarkir di tempat yang tepat.
* Dengan memasukan kondisi cuaca sekitar dan tujuan ruangannya, maka program ini akan memberikan keputusannya.

**Daftar Pustaka**

-Modul Paraktikum Kecerdasan Buatan

-http://id.wikipedia.org/wiki/Visual\_Prolog (diakses 27 mei 2015)

-http://documents.tips/documents/laporan-praktikum-kecerdasan-buatan.html

-https://www.scribd.com/doc/147033210/Laporan-Praktikum-Kecerdasan-Buatan

-http://sutri95.blogspot.co.id/2017/02/laporan-akhir-lab-ti-pengantar\_52.html

**Lampiran**

1. **Source Code**

DOMAINS

list = symbol\*

area = kondisi(symbol,symbol)

FACTS

areaparkir(integer,area)

PREDICATES

nondeterm areaparkir1(list)

%nondeterm areaparkir(integer,area)

nondeterm jalan

nondeterm jalan11

nondeterm jalan12

nondeterm jalan13

nondeterm jalan14

nondeterm jalan21

nondeterm jalan22

nondeterm jalan23

nondeterm jalan24

nondeterm jalan31

nondeterm jalan32

nondeterm jalan33

nondeterm jalan34

nondeterm jalan41

nondeterm jalan42

nondeterm jalan43

nondeterm jalan44

nondeterm ruang1

nondeterm ruang2

nondeterm ruang3

nondeterm ruang4

nondeterm ruang5

nondeterm cek1(integer)

nondeterm cek2(integer)

nondeterm cek3(integer)

nondeterm cek4(integer)

nondeterm cek5(integer)

nondeterm cekk(integer)

cek(integer,integer)

CLAUSES

areaparkir(11,kondisi("405","Cerah")).

areaparkir(12,kondisi("405","Panas")).

areaparkir(13,kondisi("405","Mendung")).

areaparkir(14,kondisi("405","Hujan")).

areaparkir1([]).

areaparkir1([H|T]):-

write(H),write(" "),

areaparkir1(T).

cek(X,X):-!.

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

jalan:-

write("\* \* \* \* SISTEM PENDUKUNG KEPUTUSAN PARKIR MOTOR MAHASISWA FKTI \* \* \* \*"),nl,nl,nl,

write("pilih tujuan ruangan anda! "),nl,

write("1. 405"),nl,

write("2. 406"),nl,

write("3. 407"),nl,

write("4. 411A"),nl,

write("5. 411B"),nl,

write("Pilih (1-5)"),nl,

readchar(X),

write(X),nl,nl,nl,

cekk(X).

cekk(X):-

cek(X,'1'),ruang1.

cekk(X):-

cek(X,'2'),ruang2.

cekk(X):-

cek(X,'3'),ruang3.

cekk(X):-

cek(X,'4'),ruang4.

cekk(X):-

cek(X,'5'),ruang5.

cekk(\_):-

jalan.

ruang1:-

write("Bagaimana Cuaca saat ini? "),nl,

write("1. Cerah"),nl,

write("2. Panas"),nl,

write("3. Mendung"),nl,

write("4. Hujan"),nl,

write("Pilih (1-4)"),nl,

readchar(X),

write(X),nl,nl,nl,

cek1(X).

cek1(X):-

cek(X,'1'),jalan11.

cek1(X):-

cek(X,'2'),jalan12.

cek1(X):-

cek(X,'3'),jalan13.

cek1(X):-

cek(X,'4'),jalan14.

cek1(\_):-

ruang1.

jalan11:-

areaparkir(11,Kondisi),write(Kondisi),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl.

jalan12:-

areaparkir(12,Kondisi),write(Kondisi),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan13:-

areaparkir(13,Kondisi),nl,write(Kondisi),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan14:-

areaparkir(14,Kondisi),nl,write(Kondisi),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

ruang2:-

write("Bagaimana Cuaca saat ini? "),nl,

write("1. Cerah"),nl,

write("2. Panas"),nl,

write("3. Mendung"),nl,

write("4. Hujan"),nl,

write("Pilih (1-4)"),nl,

readchar(X),

write(X),nl,nl,nl,

cek2(X).

cek2(X):-

cek(X,'1'),jalan21.

cek2(X):-

cek(X,'2'),jalan22.

cek2(X):-

cek(X,'3'),jalan23.

cek2(X):-

cek(X,'4'),jalan24.

cek2(\_):-

ruang2.

jalan21:-

areaparkir1(["406","Cerah"]),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl.

jalan22:-

areaparkir1(["406","Panas"]),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan23:-

areaparkir1(["406","Mendung"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan24:-

areaparkir1(["406","Hujan"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

ruang3:-

write("Cuaca saat ini "),nl,

write("1. Cerah"),nl,

write("2. Panas"),nl,

write("3. Mendung"),nl,

write("4. Hujan"),nl,

write("Pilih (1-4)"),nl,

readchar(X),

write(X),nl,nl,nl,

cek3(X).

cek3(X):-

cek(X,'1'),jalan31.

cek3(X):-

cek(X,'2'),jalan32.

cek3(X):-

cek(X,'3'),jalan33.

cek3(X):-

cek(X,'4'),jalan34.

cek3(\_):-

ruang3.

jalan31:-

areaparkir1(["407","Cerah"]),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl.

jalan32:-

areaparkir1(["407","Panas"]),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan33:-

areaparkir1(["407","Mendung"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan34:-

areaparkir1(["407","Hujan"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

ruang4:-

write("Cuaca saat ini "),nl,

write("1. Cerah"),nl,

write("2. Panas"),nl,

write("3. Mendung"),nl,

write("4. Hujan"),nl,

write("Pilih (1-4)"),nl,

readchar(X),

write(X),nl,nl,nl,

cek4(X).

cek4(X):-

cek(X,'1'),jalan41.

cek4(X):-

cek(X,'2'),jalan42.

cek4(X):-

cek(X,'3'),jalan43.

cek4(X):-

cek(X,'4'),jalan44.

cek4(\_):-

ruang4.

jalan41:-

areaparkir1(["411 A/B","Cerah"]),nl,

write("Silahkan parkir di belakang 411"),nl.

jalan42:-

areaparkir1(["411 A/B","Panas"]),nl,

write("Silahkan parkir di belakang 411"),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan43:-

areaparkir1(["411 A/B","Mendung"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan44:-

areaparkir1(["411 A/B","Hujan"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

ruang5:-

write("Cuaca saat ini "),nl,

write("1. Cerah"),nl,

write("2. Panas"),nl,

write("3. Mendung"),nl,

write("4. Hujan"),nl,

write("Pilih (1-4)"),nl,

readchar(X),

write(X),nl,nl,nl,

cek5(X).

cek5(X):-

cek(X,'1'),jalan41.

cek5(X):-

cek(X,'2'),jalan42.

cek5(X):-

cek(X,'3'),jalan43.

cek5(X):-

cek(X,'4'),jalan44.

cek5(\_):-

ruang5.

GOAL

save("projek\_akhir.txt"),

jalan.

nondeterm ruang1

nondeterm ruang2

nondeterm ruang3

nondeterm ruang4

nondeterm ruang5

nondeterm cek1(integer)

nondeterm cek2(integer)

nondeterm cek3(integer)

nondeterm cek4(integer)

nondeterm cek5(integer)

nondeterm cekk(integer)

cek(integer,integer)

CLAUSES

areaparkir(11,kondisi("405","Cerah")).

areaparkir(12,kondisi("405","Panas")).

areaparkir(13,kondisi("405","Mendung")).

areaparkir(14,kondisi("405","Hujan")).

areaparkir1([]).

areaparkir1([H|T]):-

write(H),write(" "),

areaparkir1(T).

cek(X,X):-!.

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

jalan:-

write("\* \* \* \* SISTEM PENDUKUNG KEPUTUSAN PARKIR MOTOR MAHASISWA FKTI \* \* \* \*"),nl,nl,nl,

write("pilih tujuan ruangan anda! "),nl,

write("1. 405"),nl,

write("2. 406"),nl,

write("3. 407"),nl,

write("4. 411A"),nl,

write("5. 411B"),nl,

write("Pilih (1-5)"),nl,

readchar(X),

write(X),nl,nl,nl,

cekk(X).

cekk(X):-

cek(X,'1'),ruang1.

cekk(X):-

cek(X,'2'),ruang2.

cekk(X):-

cek(X,'3'),ruang3.

cekk(X):-

cek(X,'4'),ruang4.

cekk(X):-

cek(X,'5'),ruang5.

cekk(\_):-

jalan.

cek1(X):-

cek(X,'1'),jalan11.

cek1(X):-

cek(X,'2'),jalan12.

cek1(X):-

cek(X,'3'),jalan13.

cek1(X):-

cek(X,'4'),jalan14.

cek1(\_):-

ruang1.

jalan11:-

areaparkir(11,Kondisi),write(Kondisi),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl.

jalan12:-

areaparkir(12,Kondisi),write(Kondisi),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan13:-

areaparkir(13,Kondisi),nl,write(Kondisi),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan14:-

areaparkir(14,Kondisi),nl,write(Kondisi),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

ruang2:-

write("Bagaimana Cuaca saat ini? "),nl,

write("1. Cerah"),nl,

write("2. Panas"),nl,

write("3. Mendung"),nl,

write("4. Hujan"),nl,

write("Pilih (1-4)"),nl,

readchar(X),

write(X),nl,nl,nl,

cek2(X).

write("Bagaimana Cuaca saat ini? "),nl,

write("1. Cerah"),nl,

write("2. Panas"),nl,

write("3. Mendung"),nl,

write("4. Hujan"),nl,

write("Pilih (1-4)"),nl,

readchar(X),

write(X),nl,nl,nl,

cek2(X).

cek2(X):-

cek(X,'1'),jalan21.

cek2(X):-

cek(X,'2'),jalan22.

cek2(X):-

cek(X,'3'),jalan23.

cek2(X):-

cek(X,'4'),jalan24.

cek2(\_):-

ruang2.

jalan21:-

areaparkir1(["406","Cerah"]),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl.

jalan22:-

areaparkir1(["406","Panas"]),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan23:-

areaparkir1(["406","Mendung"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan24:-

areaparkir1(["406","Hujan"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

write("Silahkan parkir ditempat yang beratap"),nl.

jalan24:-

areaparkir1(["406","Hujan"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

ruang3:-

write("Cuaca saat ini "),nl,

write("1. Cerah"),nl,

write("2. Panas"),nl,

write("3. Mendung"),nl,

write("4. Hujan"),nl,

write("Pilih (1-4)"),nl,

readchar(X),

write(X),nl,nl,nl,

cek3(X).

cek3(X):-

cek(X,'1'),jalan31.

cek3(X):-

cek(X,'2'),jalan32.

cek3(X):-

cek(X,'3'),jalan33.

cek3(X):-

cek(X,'4'),jalan34.

cek3(\_):-

ruang3.

jalan31:-

areaparkir1(["407","Cerah"]),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl.

jalan32:-

areaparkir1(["407","Panas"]),nl,

write("Silahkan parkir di belakang 405"),nl,

write("Silahkan parkir di samping 407"),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan33:-

areaparkir1(["407","Mendung"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan34:-

areaparkir1(["407","Hujan"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

ruang4:-

write("Cuaca saat ini "),nl,

write("1. Cerah"),nl,

write("2. Panas"),nl,

write("3. Mendung"),nl,

write("4. Hujan"),nl,

write("Pilih (1-4)"),nl,

readchar(X),

write(X),nl,nl,nl,

cek4(X).

cek4(X):-

cek(X,'1'),jalan41.

cek4(X):-

cek(X,'2'),jalan42.

cek4(X):-

cek(X,'3'),jalan43.

cek4(X):-

cek(X,'4'),jalan44.

cek4(\_):-

ruang4.

jalan41:-

areaparkir1(["411 A/B","Cerah"]),nl,

write("Silahkan parkir di belakang 411"),nl.

jalan42:-

areaparkir1(["411 A/B","Panas"]),nl,

write("Silahkan parkir di belakang 411"),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan43:-

areaparkir1(["411 A/B","Mendung"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan44:-

areaparkir1(["411 A/B","Hujan"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

write("Silahkan parkir di belakang 411"),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan43:-

areaparkir1(["411 A/B","Mendung"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

jalan44:-

areaparkir1(["411 A/B","Hujan"]),nl,

write("Silahkan parkir ditempat yang beratap"),nl.

ruang5:-

write("Cuaca saat ini "),nl,

write("1. Cerah"),nl,

write("2. Panas"),nl,

write("3. Mendung"),nl,

write("4. Hujan"),nl,

write("Pilih (1-4)"),nl,

readchar(X),

write(X),nl,nl,nl,

cek5(X).

cek5(X):-

cek(X,'1'),jalan41.

cek5(X):-

cek(X,'2'),jalan42.

cek5(X):-

cek(X,'3'),jalan43.

cek5(X):-

cek(X,'4'),jalan44.

cek5(\_):-

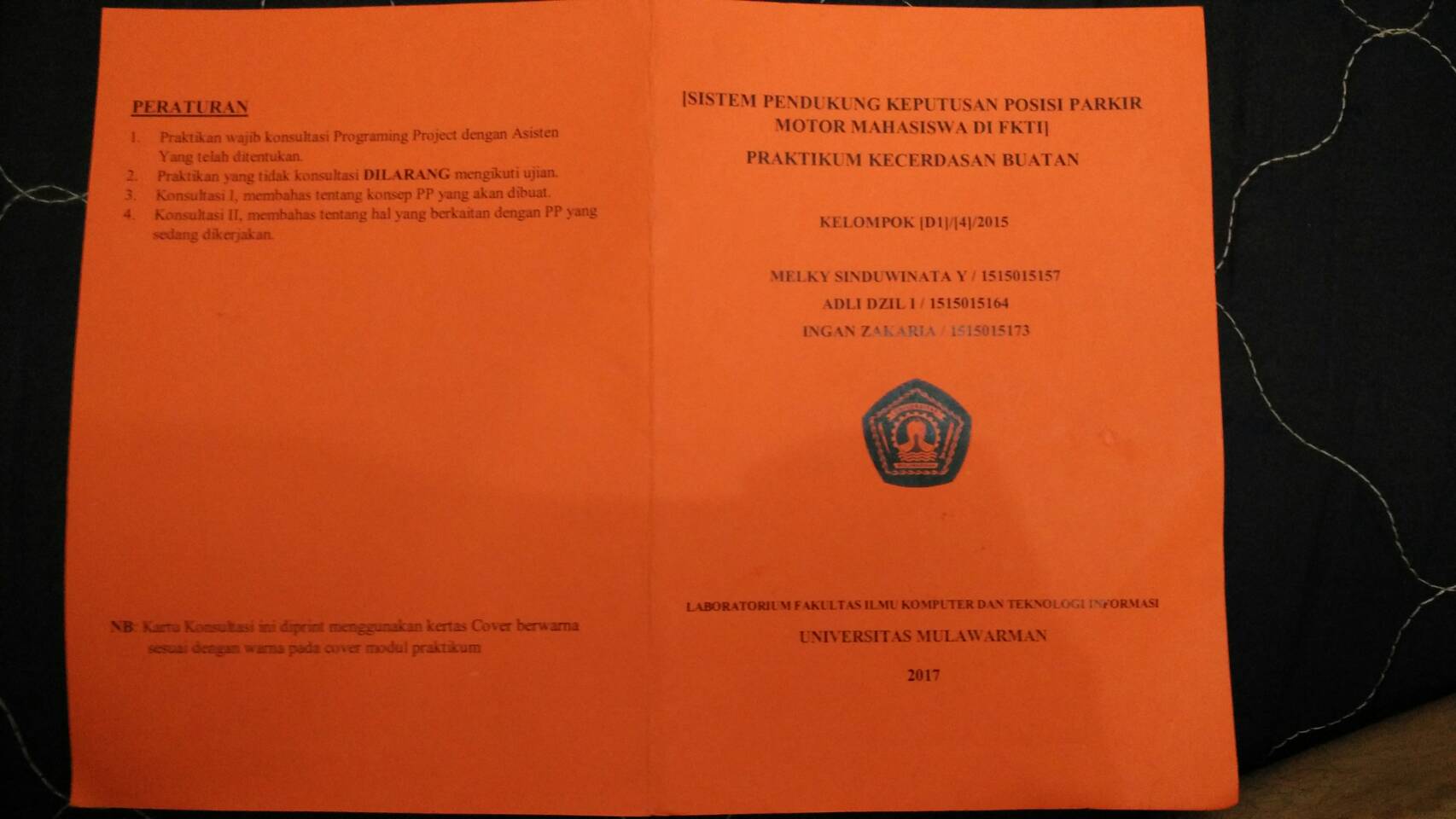
ruang5.

GOAL

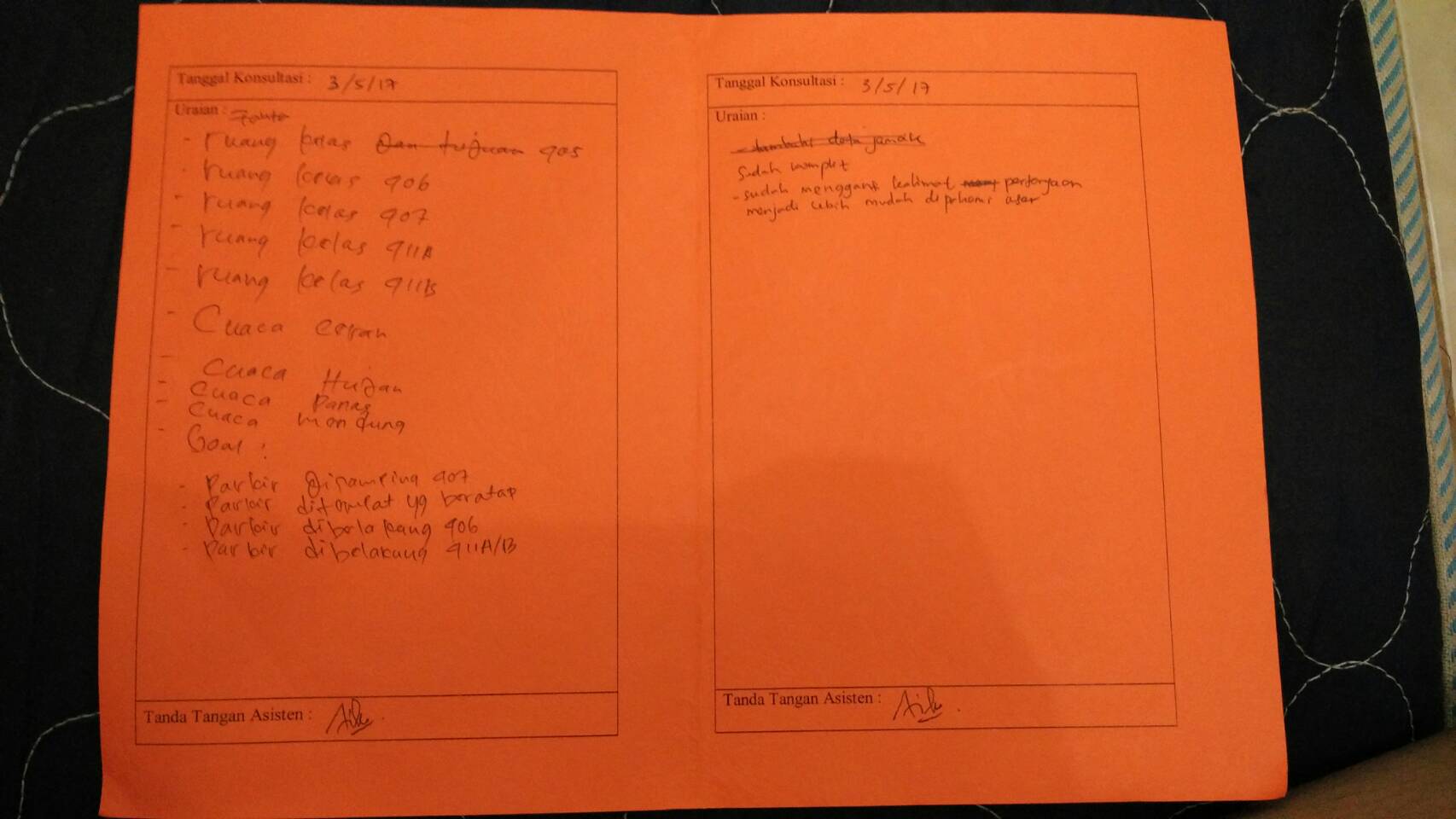
save("projek\_akhir.txt"),

jalan.

**2.Kartu Konsul**

****

**Gambar.14 Kartu konsul bagian depan**

****

**Gambar.15 Kartu konsul dalam**