IMPLEMENTASI *ARTIFICIAL INTELLIGENCE* DALAM PENYELESAIAN MASALAH DENGAN METODE *UNIFICATION* DAN *BACK TRACKING VISUAL PROLOG*

(Studi Kasus: pemilihan mahasiswa terbaik)

Citra Kurniawan, ST, MM [1]
Program Studi Teknik Elektronika
Sekolah Tinggi Teknik Malang

ABSTRAK

Manusia menyelesaikan sebuah permasalahan dengan dasar pengetahuan dan pengalaman. Pengetahuan diperoleh dari proses belajar yang dilakukan manusia, semakin banyak pengetahuan yang dimiliki maka permasalahan yang ada dapat diselesaikan dengan mudah. Manusia juga memiliki kemampuan untuk menalar yang baik. Tanpa nalar yang baik manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah. Demikian juga dengan kemampuan menalar yang sangat baik,namun tanpa bekal pengetahuan dan pengalaman yang memadai,manusia juga tidak akan bisa menyelesaikan masalah dengan baik. Kemampuan untuk menalar yang ditunjang dengan pengetahuan dan pengalaman diperoleh dengan proses waktu yang sangat lama, sedangkan permasalahan kadang muncul secara tiba-tiba. Permasalahan seperti di atas dapat diatasi dengan menggunakan ilmu *Artificial Intelligence*.

Artificial Intelligence atau kecerdasan buatan diperlukan untuk memahami bahasa komputer. Artificial Intelligence merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Dengan adanya *Artificial Intelligence* ini maka memungkinkan manusia untuk menyelesaikan masalah dengan cepat tanpa harus membutuhkan proses pengalaman, pengetahuan dan nalar yang baik karena pengalaman, pengetahuan dan nalar tersebut sudah disimpan berupa data referensi yang tersimpan. *Artificial Intelligence* menggunakan bahasa pemograman *Visual prolog* untuk mendeskripsikan logika variabel. Implementasi *Artificial Intelligence* dengan menggunakan *Visual prolog* ini dapat mempermudah manusia untuk menyelesaikan masalah.

Kata kunci : *Artificial Intelligence*, bahasa komputer, kemampuan nalar, pengetahuan, Variabel, pengalaman, bahasa pemograman, *Visual Prolog*

1. PENDAHULUAN

1.1 LATAR BELAKANG

Proses penyelesaian masalah dilakukan dengan cara menganalisa terlebih dahulu solusi yang diharapkan dari permasalahan. Solusi permasalahan didapatkan dari proses penyelesaian berdasarkan pengalaman, pengetahuan dan nalar. Untuk mendapatkan penyelesaian masalah yang cepat maka dibutuhkan sebuah implementasi *Artificial Intelligence*. Dengan adanya *Artificial Intelligence* maka setiap permasalahan dapat dianalisis menjadi suatu struktur data, dimana data tersebut dapat digunakan sebagai referensi.

Proses penyelesaian masalah dapat dilakukan dengan menggunakan *Articial Intellegence*. Dimana bahasa pemograman yang dipakai adalah bahasa pemograman *Visual Prolog*. Bahasa pemrograman ini menggunakan bahasa deklaratif dimana pemrograman memberi fakta dan aturan untuk selanjutnya diselesaikan oleh *Visual Prolog* secara deduktif sehingga menghasilkan suatu kesimpulan. *Visual prolog* menyelesaikan permasalahan secara deduktif atau menurunkan kesimpulan sebagai jawaban berdasarkan fakta (*fact)* dan aturan (*rule*) dengan pencarian dari atas ke bawah.

Berdasarkan uraian di atas, penulisan penelitian ini akan membahas mengenai bagaimana *"IMPLEMENTASI ARTIFICIAL INTELLIGENCE DALAM PENYELESAIAN MASALAH DENGAN METODE UNIFICATION DAN BACK TRACKING VISUAL PROLOG.(Studi Kasus: pemilihan mahasiswa terbaik)*"

1.2 RUMUSAN MASALAH

Berdasarkan latar belakang di atas, maka dapat diambil rumusan masalah sebagai berikut:

1. Bagaimana proses penyelesaian masalah pada *Artificial Intelligence*.
2. Bagaimana proses kerja *Visual Prolog* pada *Artificial Intelligence.*
3. Bagaimana implementasi *Visual Prolog* dalam penyelesaian masalah.

1.3 BATASAN MASALAH

Berdasarkan latar belakang di atas, maka dapat diambil batasan masalah sebagai berikut:

1. Aplikasi A*rtificial Intelligence* menggunakan bahasa pemograman *Visual Prolog*.
2. Penyelesaian masalah diselesaikan berdasarkan logika pemikiran bahasa pemograman.

1.4 TUJUAN PENELITIAN

Tujuan yang ingin dicapai pada penelitian ini adalah sebagai berikut :

1. Menyelesaikan masalah dengan metode *Artificial Intelligence*.
2. Menjelaskan proses kerja *Visual Prolog* pada *Artificial Intelligence.*
3. Implementasi *Visual Prolog* dalam penyelesaian masalah.

1.5 MANFAAT PENELITIAN

Manfaat dari penelitian ini adalah menentukan metode penyelesaian masalah melalui bahasa pemograman *Visual Prolog*.

2. TINJAUAN PUSTAKA

2.1 Bahasa Pemograman sebagai bahasa mesin

Bahasa pemrograman adalah teknik komunikasi standar untuk mengekspresikan instruksi kepada komputer. Layaknya bahasa manusia, setiap bahasa memiliki tata tulis dan aturan tertentu. Bahasa pemrograman memfasilitasi seorang programmer secara tepat menetapkan data apa yang sedang dilakukan oleh komputer selanjutnya, bagaimana data tersebut disimpan dan dikirim, dan apa yang akan dilakukan apabila terjadi kondisi yang variatif.

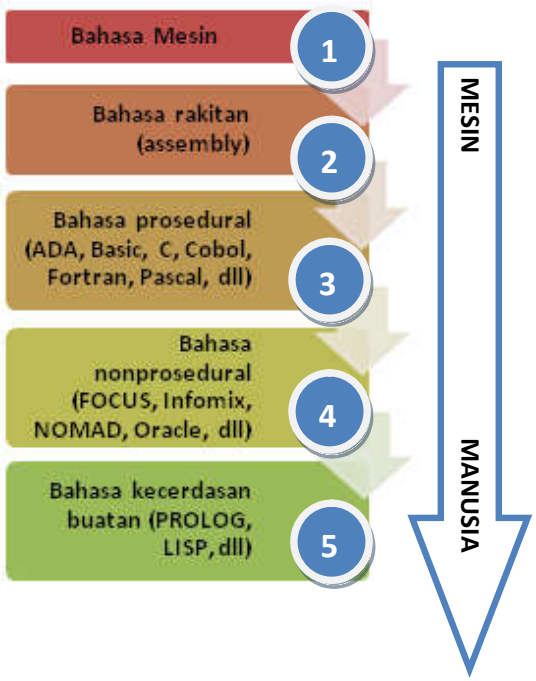
Bahasa pemrogaman merupakan sebuah instruksi standar untuk memerintah komputer agar mempunyai fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

Bahasa pemograman membantu manusia untuk saling berhubungan dengan mesin (computer). Bahasa pemrograman dapat diklasifikasikan menjadi tingkat rendah, menengah, dan tingkat tinggi. Pergeseran tingkat dari rendah menuju tinggi menunjukkan kedekatan terhadap "bahasa manusia". Berdasarkan tingkat kedekatan dengan mesin komputer, bahasa pemrogaraman dapat dibedakan sebagai berikut :

1. Bahasa mesin, bahasa yang memberikan perintah kepada computer dengan memakai bahasa binner.
2. Bahasa tingkat rendah, bahasa pemograman ini biasanya disebut

bahasa assembly, memberikan perintah kepada computer dengan memakai kode – kode singkat (kode mnemonic),), contohnya MOV, SUB, CMP, JMP, JGE, JL, LOOP.

3. Bahasa tingkat menengah, bahasa pemrogaman yang mempergunakan intruksi bahasa manusia yang bersifat simbolik, , contohnya {, }, ?, <<, >>, &&, ||.
4. Bahasa tingkat tinggi, bahasa pemograman yang mempergunakan instruksi yang berasal dari unsure kata-kata manusia, contohnya begin, end, if, for, while, and, or.



Gambar 1. Perkembangan bahasa pemograman

Dari gambar 1 dapat dilihat bahwa bahasa pemograman berkembang pada lima generasi. Pada generasi pertama, bahasa yang dipergunakan berorientasi pada mesin. Program disusun menggunakan bahasa mesin/kode mesin. Bahasa mesin ini sangat sulit dipahami oleh orang awam sehingga programmer harus menguasai operasi komputer secara teknis. Kode binner "0" dan "1" dipergunakan dengan cara mengkonbinasikan kode tersebut untuku berkomunikasi kepada mesin.

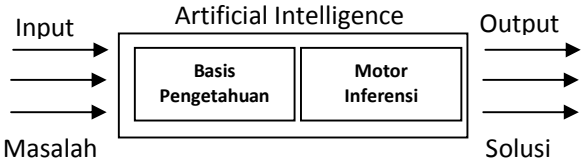
Pada bahasa pemograman generasi kedua, bahasa pemrograman yang digunakan adalah bahasa rakitan / *Assembly.* Bahasa *Assembly* adalah bahasa pemrograman yang menggunakan instruksi yang sama seperti pada bahasa mesin berupa instruksi dan variable yang mempunyai nama sehingga mempermudah proses pemrograman. Pada bahasa pemograman generasi ketiga menggunakan pendekatan procedural. Instruksi program ditulis menggunakan kata-kata yang biasa digunakan oleh manusia. Contoh : *WRITE* (untuk menampilkan kelayar), *READ* (untuk membaca data masukan dari keyboard).

Pada bahasa pemograman generasi ke empat, bahasa pemrograman dirancang untuk mengurangi waktu pemrogram untuk membuat program sehingga pembuatan program dibuat dengan waktu lebih cepat. Program ini dapat digunakan oleh pemakai yang kurang mengenal hal-hal teknis pemrograman tanpa perlu bantuan seorang programmer professional. Sebagai contoh adalah *Microsoft Access*. Sedangkan pada bahasa pemograman ke lima. bahasa pemrograman yang ditujukan untuk menangani kecerdasan buatan (*Artificial Antelligence*) (AI). AI adalah disiplin dari ilmu komputer yang mempelajari cara komputer meniru kecerdasan manusia. Contoh bahasa pemrograman : PROLOG dan LISP.

2.2 Konsep Artificial Intelligence

A*rtificial Intelligence* adalah sub bidang pengetahuan yang mempelajari bagaimana implementasi bahasa pemograman dapat memdeskripsikan bahasa manusia berdasarkan bahasa mesin (computer) . Berdasarkan John McCarthy (1956), A*rtificial Intelligence* adalah Metode untuk mengetahui dan memodelkan proses – proses pikiran manusia dan mendesain mesin agar dapat meniru perilaku manusia.

Pengertian A*rtificial Intelligence* dapat ditinjau dari dua pendekatan antara lain pendekatan ilmiah *(A Scientific Approach)* dan pendekatan teknik *(An Engineering Approach)*. Pendekatan Ilmiah *(A Scientific Approach)* merupakan pendekatan dasar ilmiah yang berkembang sebelum invansi ke computer. Pendekatan ilmiah melihat batas sementara dari komputer, dan dapat diatasi dengan perkembangan teknologi lanjutan. Sedangkan pendekatan teknik *(An Engineering Approach)*, mengacu pada metode-metode yang digunakan dan menggunakan sistematika logika. Penggunaan logika sangat penting untuk pembentukan nalar pada saat penyelesaian masalah. Penggunaan nalar pada metode ini didukung oleh pengetahuan dan pengalaman yang dimiliki.



Gambar 2. Proses pemecahan masalah Artificial Intelligence

Artificial Intelligence dipakai diberbagai bidang. Penggunaan *Artificial Intelligence* memberikan manfaat sebagai berikut:

1. Aplikasi komputer yang sangat mudah bagi pemakai.

2. Peningkatan pemecahan masalah secara cepat dan efisien.
3. Penyelesaian masalah yang memiliki keterbatasan data dan fakta.
4. Peningkatan produktifitas
5. Efektifitas pencarian data atau pola yang mempunyai data yang besar.

2.3 *Visual Prolog*

Visual Prolog menggunakan teknik pencarian yang disebut heuristik (*heuristic*) dengan menggunakan metode pohon logika. Bahasa pemograman *Visual Prolog* banyak dikembangkan di University of Melbourne oleh Lee Naish dan John Loyd. Prolog dalam perkembangannya telah dikombinasikan dengan berbagai bahasa pemrograman terutama *functional programming*.

Bahasa *Visual Prolog* ini secara intensif digunakan pada proyek komputer generasi ke-5 di Jepang. Walaupun demikian sebetulnya penggunaannya tidak terbatas untuk Artificial Intelligence saja. *Visual Prolog* adalah bahasa pemrograman logika atau di sebut juga sebagai bahasa non-procedural. Namanya diambil dari bahasa Perancis programmation en logique (Pemrograman logika).

Bahasa ini diciptakan oleh Alain Colmerauer dan Robert Kowalski sekitar tahun 1972 dalam upaya untuk menciptakan suatu bahasa pemrograman yang memungkinkan pernyataan logika alih-alih rangkaian perintah untuk dijalankan komputer. Berbeda dengan bahasa pemrograman yang lain, yang menggunakan algoritma konvensional sebagai teknik pencariannya seperti pada Delphi, Pascal, BASIC, COBOL dan bahasa pemrograman yang sejenisnya, maka prolog menggunakan teknik pencarian yang di sebut heuristik (heutistic) dengan menggunakan pohon logika. Berikut adalah kegunaan dari visual prolog :

1. Visual prolog menyelesaikan permasalahan secara deduktif atau menurunkan kesimpulan sebagai jawaban berdasarkan fakta (*fact*) dan aturan (*rule*) dengan pencarian dari atas ke bawah.
2. Perbedaan antara bahasa deklaratif dengan prosedural sebagai berikut:
3. Bahasa deklaratif hanya membutuhkan deklarasi atau uraian masalah sedangkan pada bahasa prosedural memerlukan perintah.
4. Visual prolog adalah *goal oriented* yaitu apa yang harus dipecahkan, sedangkan bahasa prosedural

menjawab bagaimana harus memecahkan masalah.

5. Prolog tidak memiliki struktur seperti deklarasi, dan lain-lainnya, yang ada hanyalah *clause*

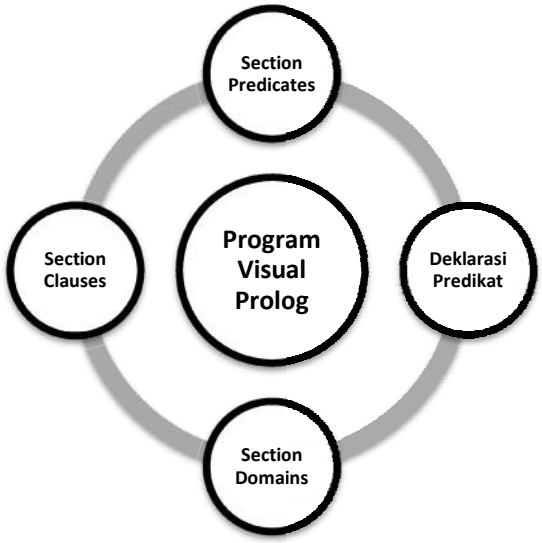
Berikut perbedaan bahasa pemogramam umum dan bahasa pemograman prolog.

Tabel 1. Perbedaan bahasa pemograman prolog dan bahasa pemograman umum.

Bahasa Pemograman Umum	**Bahasa Permograman Prolog**
Algoritma / prosedur untuk memecahkan masalah (procedural languange)	Object oriented languange atau declarative language
Program menjalankan prosedur yang sama berulang-ulang dengan data masukan yang berbeda-beda	Tidak terdapat prosedur, tetapi hanya kumpulan data-data objek (fakta) yang akan diolah, dan relasi antar objek tersebut membentuk aturan yang diperlukan untuk mencari suatu jawaban
.Prosedur dan pengendalian program ditentukan oleh programmer dan perhitungan dilakukan sesuai dengan prosedur yang telah dibuat	Programmer menentukan tujuan (goal), dan komputer menentukan bagaimana cara mencapai tujuan tersebut serta mencari jawabannya.

Dari table 1 di atas dapat dilihat bahwa bahasa pemograman visual prolog dapat digunakan untuk memecahkan masalah yang tidak terstruktur, dan prosedur pemecahannya tidak diketahui, khususnya untuk memecahkan masalah non numeric. Selain itu, visual . Prolog bekerja seperti pikiran manusia, proses pemecahan masalah bergerak di dalam ruang masalah menuju suatu tujuan (jawaban tertentu).

Visual prolog bekerja menggunakan empat dasar. Bagian dasar dari prolog dapat dilihat pada gambar 2 berikut :



Gambar 3. Bagian dasar program visual prolog
Sumber : Boer, Thomas (2008)

Pada gambar 2 di atas, dapat dilihat bahwa program prolog terdiri dari empat bagian dasar yaitu, Section Domains, Section Predicates, Section Clauses, dan Deklarasi predikat. Section domains merupakan bagian dasar dari visual dasar yang dapat memdeklarasikan variabel terhadap data yang sejenis, sebagai contoh :

DOMAINS
nama, jenis_kelamin = symbol
umur = integer

PREDICATES
orang(nama, , jenis_kelamin, umur)

Pada DOMAINS di atas, variabel nama dan jenis_kelamin dikelompok menjadi satu jenis tipe variabel “symbol”, sedangkan umur dikelompokkan menjadi satu jenis tipe variabel “integer”. Pengelompokan variabel berdasarkan tipe variabel yang sama untuk mencegah kesalahan logika pada program dan mempermudah pengelompokan variabel pada fungsi utama program.

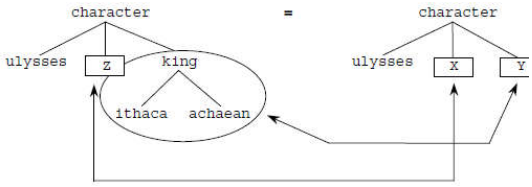
Bagian dasar yang kedua adalah Section Predicates. Sebelum mendefinsikan predikat di section clauses, maka predikat tersebut harus dideklarasikan terlebih dahulu di section predicates karena Visual Prolog tidak akan mengenal predikat yang kita tuliskan tersebut. Proses selanjutnya adalah Section Clauses. Pada Section Clauses, aturan dan fakta dipergunakan untuk untuk mendeklarasikan struktur dari fungsi program. Bagian dasar yang keempat adalah Deklarasi Predikat. Program utama dideklarasikan pada deklarasi predikat. Fungsi-fungsi yang merupakan program utama dimasukkan pada bagian Deklarasi Predikat.

Dasar pemrograman Visual Prolog dalam bahasa pemrograman dikenal beberapa istilah. Istilah dalam Visual Prolog antara lain adalah fakta, aturan, Klausa, Relasi, dan Variabel. Fakta adalah kenyataan atau kebenaran yang diketahui, dan menyatakan relasi antara dua object atau lebih. Fakta dapat juga dideklarasikan berupa kata sifat. Aturan merupakan logika yang dirumuskan dalam bentuk relasi sebab akibat dan menunjukkan bagaimana fakta – fakta berinteraksi satu sama lain untuk membentuk suatu kesimpulan. Aturan yang dideklarasikan pada Visual prolog merupakan suatu kalimat yang bersyarat.

Klausa adalah suatu ungkapan atau susunan kata yang didalam Prolog dapat berupa fakta atau aturan. Dalam suatu klausa dapat terdiri dari beberapa sub-klausa, yang dihubungkan satu dengan yang lain dengan tanda (,) atau (;). Sedangkan relasi adalah tabel dengan jumlah n kolom dan terdiri dari beberapa baris fakta maupun aturan dan variabel adalah predikat yang mempunyai nilai.

2.3 Unifikasi dan Lacak Balik (Unification and Backtracking)

Unifikasi adalah proses pencocokan atau perbandingan untuk mencari jawaban pada nilai suatu variabel. Pada proses unifikasi terdapat hal-hal yang perlu diperhatikan terkait dengan konsistensi logika. Proses unifikasi terjadi pada sesama argument yang sama.



Gambar 4. Bentuk Unifikasi
Sumber: Bratko (2001)

Pada unifikasi, hasil yang didapatkan tidak boleh mengakibatkan variabel mempunyai nilai ganda atau bertentangan. Dengan adanya teknik Unifikasi ini, variabel yang digunakan dapat dimasukkan pada kelompok variabel yang sama jika memiliki kesamaan tipe. Unifikasi pada Visual Prolog mengimplementasikan beberapa prosedur yang juga dilakukan oleh beberapa bahasa dengan melewatkan parameter, menyeleksi tipe data, membangun struktur, mengakses struktur dan pemberian nilai (assignment). Adanya unifikasi variabel menyebabkan proses eksekusi program berjalan cepat karena deklarasi data tidak banyak.

Selain unifikasi, dalam penyelesaian masalah didasarkan pada metode backtracking atau runut balik. Runut balik merupakan strategi pencarian yang arahnya kebalikan dari runut maju. Proses pencarian dimulai dari tujuan, yaitu kesimpulan yang menjadi solusi permasalahan yang dihadapi. 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. *Visual Prolog* menggunakan metode runut balik 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. *Visual Prolog* 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 runut balik terakhir dan mencoba alternatif subgoal yang lain.

Mekanisme runut balik dapat menghasilkan pencarian yang tidak perlu, akibatnya program menjadi tidak efisien. Pada kasus lain, suatu kebutuhan untuk memaksa Visual Prolog untuk melanjutkan mencari jawaban tambahan walaupun goal tersebut sudah terpenuhi. *Visual Prolog* menyediakan dua metode 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.

Pada metode 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.. Pada 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. Ketika proses melewati cut, pemanggil ke cut dinyatakan sukses dan subgoal berikutnya (jika ada) dipanggil.

3. METODOLOGI PENELITIAN

3.1. Studi Literatur

Studi literatur yang dibutuhkan adalah proses penyelesaian masalah dan *Visual Prolog.*

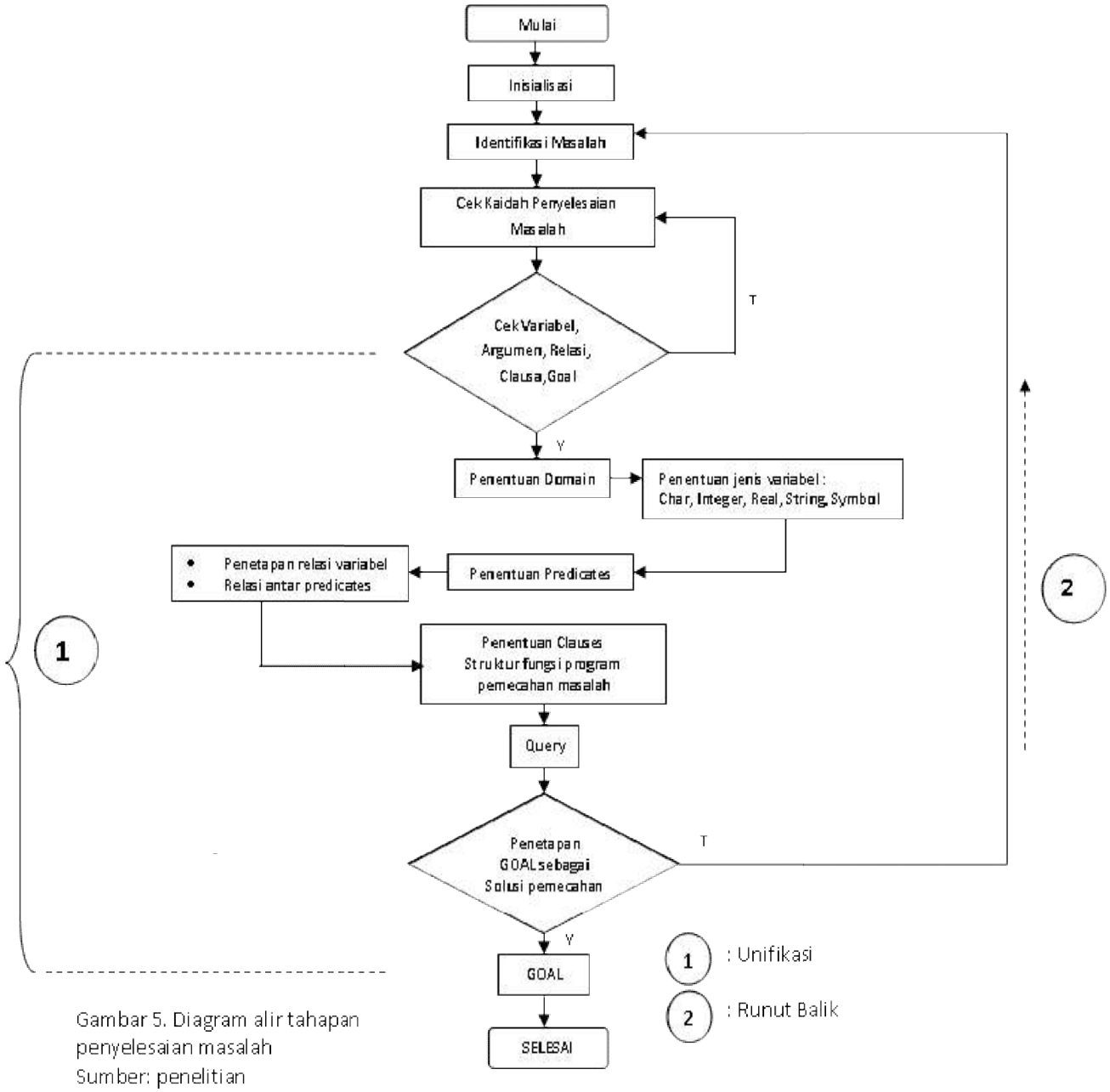
3.2. Analisis Kebutuhan

Kebutuhan tools sebagai alat untuk simulasi adalah bahasa pemograman *Visual Prolog 5.2*

3.3. Desain Penelitian

Perencanaan dengan *Visual Prolog* dilakukan dengan cara mengidentifikasi terlebih dahulu masalah yang terjadi, kemudian permasalahan tersebut dianalisis berdasarkan metode Unifikasi dan *Backtracking*.

3.4. Metode Pelaksanaan



Gambar 5. Diagram alir tahapan penyelesaian masalah
Sumber: penelitian

3.5. Pengujian

Pengujian pada penelitian ini didasarkan pada efektifitas penyelesaian masalah.

3.5 Pengujian

Pengujian pada penelitian ini didasarkan pada efektifitas penyelesaian masalah

3.6. Teknik Analisis Data

Untuk mengetahui apakah bahasa *Visual Prolog* dapat menyelesaikan masalah.

3.7. Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan proses unifikasi dan runut balik dari visual prolog dilakukan.

4. PEMBAHASAN DAN HASIL PENELITIAN

Bab ini memuat pembahasan proses penelitian, hasil penelitian dan analisis terhadap hasil penelitian yang telah dibuat. Proses pelaksanaan penelitian untuk penyelesaian masalah dibagi menjadi dua tahap, tahap unifikasi (*Unification*) dan tahap runut balik (*Backtracking*).

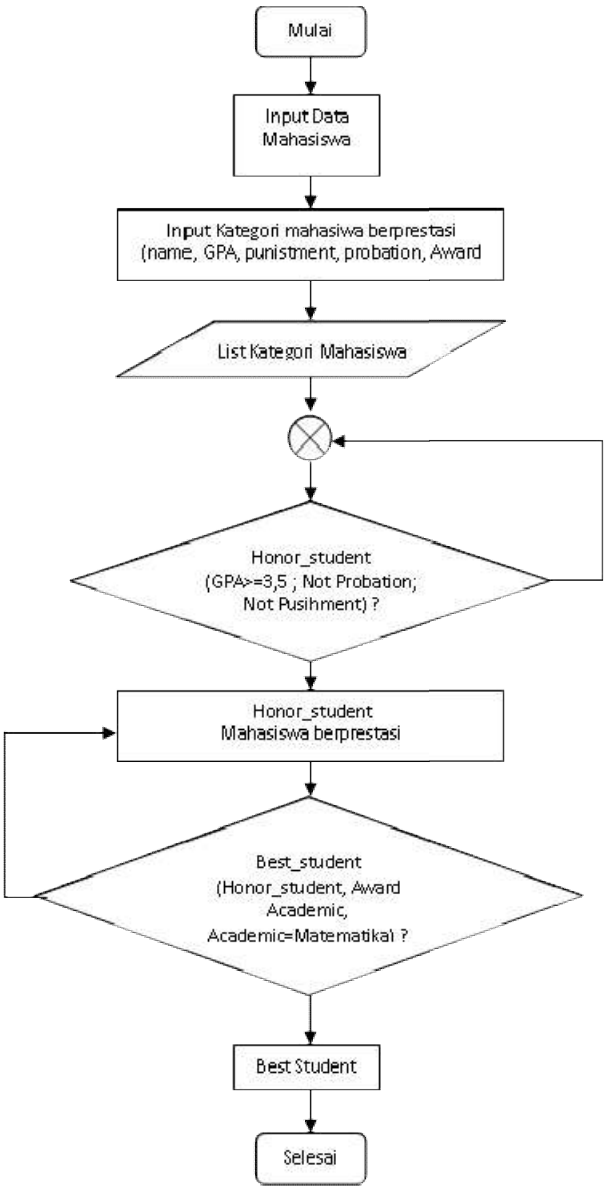
- Unifikasi

Berikut adalah proses unifikasi pada penyelesain masalah. Sebagai contoh dari masalah adalah penentuan mahasiswa terbaik. Mahasiswa kategori berprestasi memiliki syarat sebagai berikut : 1) Mempunyai IPK sama dengan atau di atas 3,5 (skala 4), (variabel GPA). 2) Mahasiswa tidak dalam masa cuti kuliah (variabel Probation). 3) Mahasiswa tidak dalam masa hukuman (variabel Punishment). 4) Mahasiswa kandidat berprestasti adalah mahasiswa yang pernah mendapatkan reward (predikat honor_student). Berikut adalah data mahasiswa yang akan dipilih:

Tabel 2. Daftar mahasiswa

Mhs	GPA	Probation	Punishment	Award
Billy Kurniawan	3,5	**no**	**yes**	**no**
Joni Santoso	2,0	**no**	**no**	Non Akademik
Raymond Pribadi	3,7	**no**	**no**	Akademik
David Hedler	3,4	**no**	**no**	Akademik
Sandra lee	2,75	**no**	**yes**	**Non Akademik**
Mandy Few	3,8	**yes**	**no**	Akademik
David Moch	3,1	**yes**	**no**	no

Dari tujuh mahasiswa pada tabel 2 di atas, kemudian diproses sesuai dengan syarat untuk menjadi mahasiswa berprestasi. Mahasiswa terbaik yang dipilih harus memiliki kriteria yang sesuai dengan syarat yang ditetapkan.



Gambar 6. Proses pemilihan mahasiswa terbaik

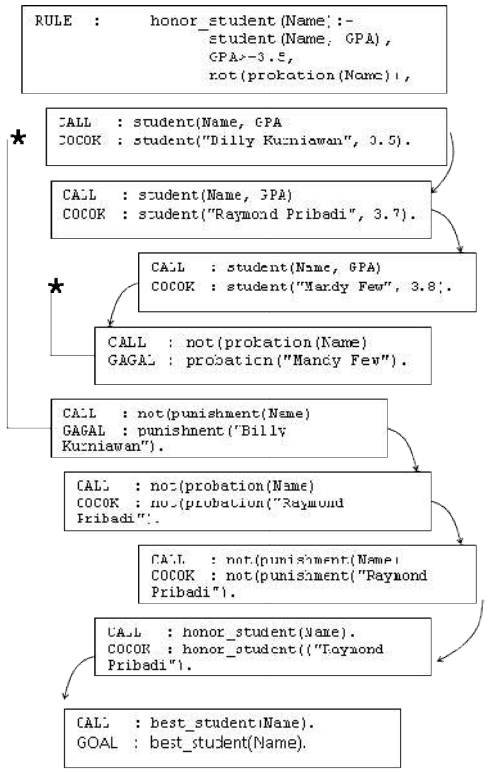
Pada gambar 6, dapat dilihat bahwa proses pemilihan mahasiswa terbaik dimulai dengan input data mahasiswa sesuai kategori pemilihan. Ada dua tahap dalam pemilihan, tahap pertama adalah seleksi untuk memilih mahasiswa berprestasi. Hasil dari mahasiswa berprestasi tersebut kemudian diseleksi kembali menjadi mahasiswa terbaik. Berikut listing program *Visual Prolog* :

Listing Program :

```
domains
        name = symbol
        gpa  = real

non_academic,competition,academic,punishment,t
ype_award,type_academic = string

award = award(name, type_award, type_academic)

predicates
        nondeterm honor_student(name)
        nondeterm student(name, gpa)
        nondeterm probation(name)
        nondeterm best_student(name)
        nondeterm punishment(name)
        nondeterm academic(name)
        nondeterm non_academic(name)
        academic(competition)
        awarded(name,competition)

clauses
/*-----------------------------------*/
        honor_student(Name):-
                student(Name, GPA),
                GPA>=3.5,
                not(probation(Name)),
                not(punishment(Name)).

        best_student(Name):-
                honor_student(Name),

        not(non_academic(competition)),
                academic(matematics).

  student("Billy Kurniawan", 3.5).
  student("Joni Santoso", 2.0).
  student("Raymond Pribadi", 3.7).
  student("David Hedler", 3.4).
  student("Sandra lee", 2.75).
  student("Mandy Few", 3.8).
  student("David Moch",3.1).
  academic(matematics).
  academic(phisics).
  academic(biology).
  non_academic(chess).
  non_academic(karate).
  awarded("Raymond Pribadi",matematics).
  awarded("Sandra lee",chess).
  awarded("Mandy Few",biology).
  awarded("David Hedler",matematics).

  probation("Mandy Few").
  probation("David Moch").
  punishment("Billy Kurniawan").
  punishment("Sandra lee").


/*---------------------------------------*/

goal
/*---------------------------*/
best_student(Name).
```

- **Backtracking**

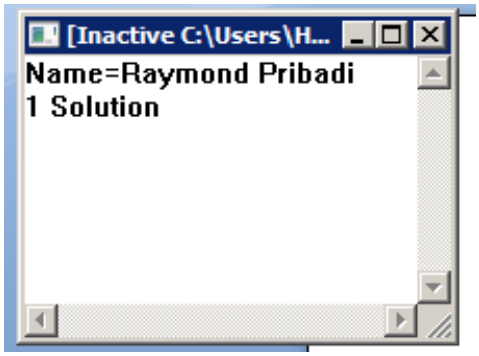
Untuk mengetahui terjadinya runut balik, akan diamati langkah demi langkah bagaimana *Visual Prolog* mencari solusi dari goal yang diberikan. *Visual prolog* mencoba *untuk memanggil variabel "gpa", "punishment","probation",* kemudian mencari variabel tersebut pada proses variabel unifikasi. Jika variabel yang dipanggil mengalamai kecocokan maka proses selanjutnya adalah memasukkan variabel tersebut pada "clauses" sebagai fungsi program. Pemanggilan kembali tersebut diperlukan sebagai proses runut balik variabel jika variabel yang digunakan tidak memenuhi syarat yang ditentukan fungsi program.



Gambar 7. Proses runut balik pemecahan masalah

- **Hasil Penelitian**

Hasil penelitian dari bahasa visual prolog adalah mahasiswa dengan nama "Raymond Pribadi". Raymond terpilih sebagai mahasiswa terbaik karena memenuhi dua kriteria yaitu merupakan "honor_student" dan pernah mendapatkan award "academic".



Gambar 8. Hasil dari pemilihan mahasiswa terbaik.

Dari gambar 8 dapat dilihat bahwa terdapat dua solusi yang saling menguatkan, "Raymond Pribadi" sebagai mahasiswa *"honor_student"* dan mahasiswa *"best_student"*.

5. KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah:

- Penelitian dilakukan secara sistematis dengan memetakan goal / tujuan terlebih dahulu.
- *Visual Prolog* untuk menyelesaikan masalah dengan deskripsi logika dan variabel.

- Penggunaan syarat dan relasi pada *Visual Prolog* mempermudah penyelesaian masalah dan hasil *Visual Prolog* tidak ambigu.

5.2 SARAN

Saran pada penelitian ini adalah sebagai berikut :

- Pengembangan *Visual Prolog* dengan metode yang lain.
- Penggunaan relasi dan unifikasi yang lebih kompleks.

DAFTAR PUSTAKA

[1] Boer, Thomas, 2008, A Beginners Guide to Visual Prolog

[2] Bratko, 2001,An Introduction to Prolog

[3] Endriss, Ulle, 2014, An Intruduction to prolog programming, Universiteit van Amsterdam

[4] Lin, Jian, Vipin Kumar, Clement Leung, 1980, An Intelligent Backtracking Algorithm for Parallel Execution of Logic Program. Depatment Of Computer Science The University of Texas

[5] Merritt, Dennis, 2000, Building Expert Systems in Prolog, Amzi Inc

[6] Praktikum Artificial Intelligence. AKAKOM Yogyakarta

[7] Rowe, Neil C, 1988, Artificial : Intelligence through Prolog, Prentice-Hall